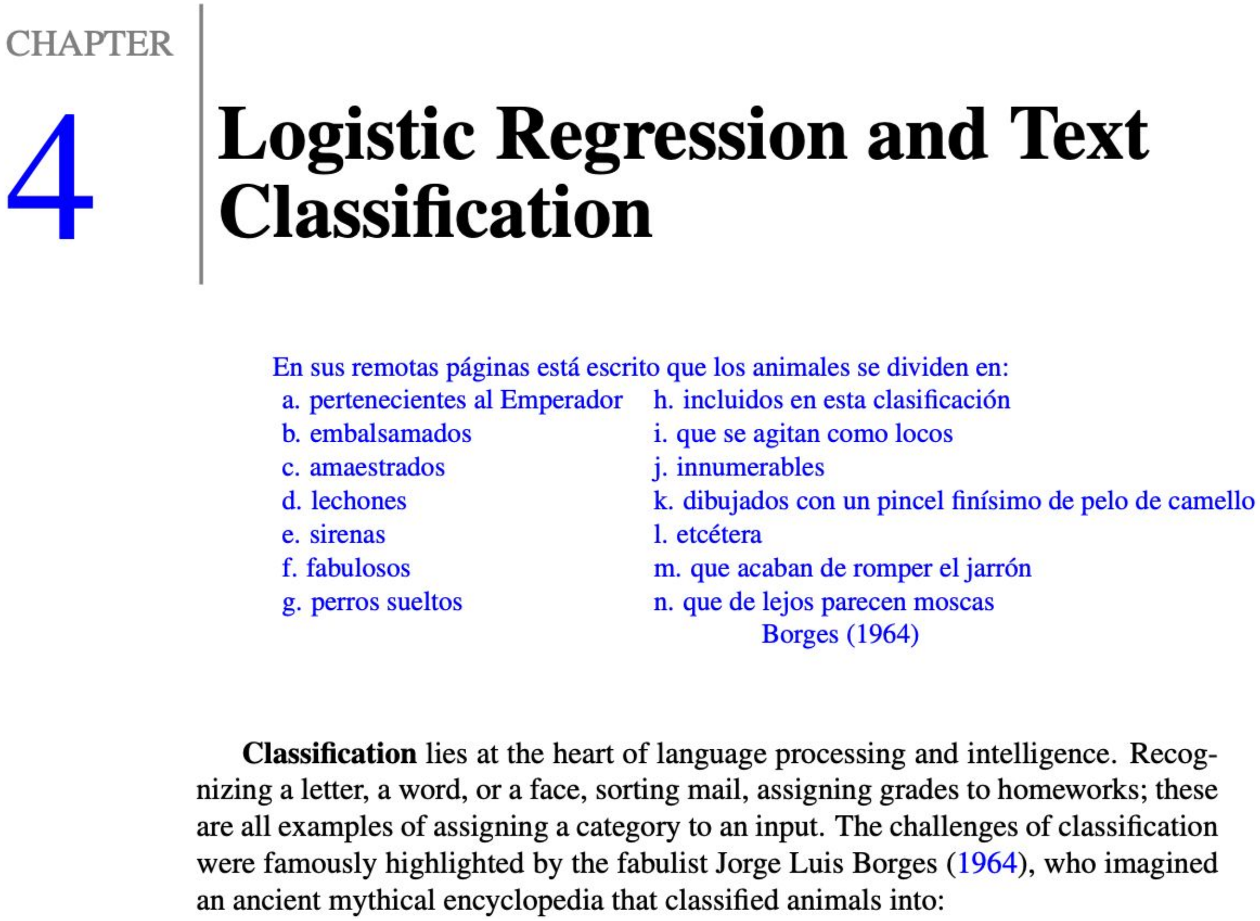
IMPERIAL

Text Classification

Shamsuddeen Muhammad
Google DeepMind Academic Fellow,
Imperial College London
https://shmuhammadd.github.io/

Reference



CHAPTER

4

Logistic Regression and Text Classification

En sus remotas páginas está escrito que los animales se dividen en:

a. pertenecientes al Emperador
b. embalsamados
c. amaestrados
d. lechones
e. sirenas
f. fabulosos
g. perros sueltos
h. incluidos en esta clasificación
i. que se agitan como locos
j. innumerables
k. dibujados con un pincel finísimo de pelo de camello
l. etcétera
m. que acaban de romper el jarrón
n. que de lejos parecen moscas

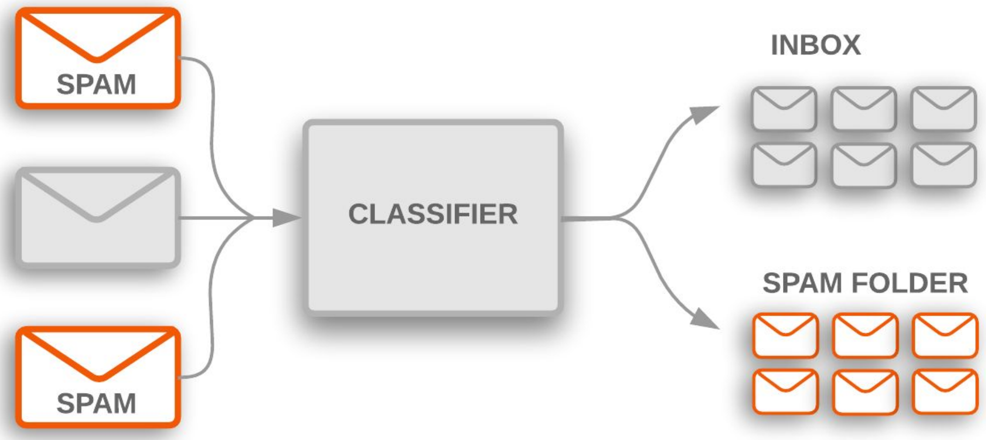
Borges (1964)

Classification lies at the heart of language processing and intelligence. Recognizing a letter, a word, or a face, sorting mail, assigning grades to homeworks; these are all examples of assigning a category to an input. The challenges of classification were famously highlighted by the fabulist Jorge Luis Borges (1964), who imagined an ancient mythical encyclopedia that classified animals into:

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

Text is everywhere

- When you open your email, a model decides: **“spam or not?”**
- When you post on social media, a model checks: **“harmful or safe?”**
- When working with product reviews, the system must determine: “Is this review **positive, or negative?**”



Text Classification

- Text classification is one of the most widely deployed NLP applications

- Recent industry reports estimate that over *70% of AI-driven language technologies* used in production environments rely on some form of classification.

Examples of Text Classification Tasks

AfriSenti: A Twitter Sentiment Analysis Benchmark for African Languages

Shamsuddeen Hassan Muhammad[1,2*+], Idris Abdulmumin[3+], Abinew Ali Ayele[4], Nedjma Ousidhoum[5,6], David Ifeoluwa Adelani[7*], Seid Muhie Yimam[8], Ibrahim Sa'id Ahmad[2+], Meriem Beloucif [9], Saif M. Mohammad[10], Sebastian Ruder[11], Oumaima Hourrane[12], Pavel Brazdil[1]3, Alípio Jorge [1,13], Felermino Dário Mário António Ali[1], Davis David[14], Salomey Osei[15], Bello Shehu Bello[2], Falalu Ibrahim[16], Tajuddeen Gwadabe[*+], Samuel Rutunda[17], Tadesse Belay[18], Wendimu Baye Messelle[4], Hailu Beshada Balcha[19], Sisay Adugna Chala[20], Hagos Tesfahun Gebremichael[4], Bernard Opoku[21], Steven Arthur[21]

[1]University of Porto, Portugal [2]Bayero University Kano, [3]Ahmadu Bello University, Zaria, [4]Bahir Dar University, [5]University of Cambridge, [6]Cardiff University, [7]University College London, [8]Universität Hamburg, [9]Uppsala University, [10]National Research Council Canada, [11]Google Research, [12]Hassan II University of Casablanca, [13]LIAAD - INESC TEC, [14]dLab, [15]University of Deusto, [16]Kaduna State University, [17]Digital Umuganda, [18]Wollo University, [19]Jimma University, [20]Fraunhofer FIT, [21]Accra Institute of Technology, [*]Masakhane NLP, [+]HausaNLP

shmuhammad.csc@buk.edu.ng

Abstract

Africa is home to over 2,000 languages from more than six language families and has the highest linguistic diversity among all continents. These include 75 languages with at least one million speakers each. Yet, there is little NLP research conducted on African languages. Crucial to enabling such research is the availability of high-quality annotated datasets. In this paper, we introduce AfriSenti, a sentiment analysis benchmark that contains a total of >110,000 tweets in 14 African languages (Amharic, Algerian Arabic, Hausa, Igbo, Kinyarwanda, Moroccan Arabic, Mozambican Portuguese, Nigerian Pidgin, Oromo, Swahili, Tigrinya, Twi, Xitsonga, and Yorùbá) from four language families. The tweets were annotated by native speakers and used in the AfriSenti-SemEval shared task [1].

We describe the data collection methodology, annotation process, and the challenges we dealt with when curating each dataset. We further report baseline experiments conducted on the different datasets and discuss their usefulness.

Figure 1: Countries and languages represented in AfriSenti: Amharic, Algerian Arabic, Hausa, Igbo, Kinyarwanda, Moroccan Arabic, Mozambican Portuguese, Nigerian Pidgin, Oromo, Swahili, Tigrinya, Twi, Xitsonga, and Yorùbá.

2007) and in recent years, it has seen a prolifera

AfriHate: A Multilingual Collection of Hate Speech and Abusive Language Datasets for African Languages

Shamsuddeen Hassan Muhammad[1,2*], Idris Abdulmumin[3*], Abinew Ali Ayele[4,5], David Ifeoluwa Adelani[6], Ibrahim Said Ahmad[2,7], Saminu Mohammad Aliyu[2], Nelson Odhiambo Onyango[8], Lilian D. A. Wanzare[8], Samuel Rutunda[9], Lukman Jibril Aliyu[10], Esubalew Alemneh[4], Oumaima Hourrane[12], Hagos Tesfahun Gebremichael[4], Elyas Abdi Ismail[20], Meriem Beloucif[13], Ebrahim Chekol Jibril[14], Andiswa Bukula[15], Rooweither Mabuya[15], Salomey Osei[16], Abigail Oppong[17], Tadesse Destaw Belay[18,19], Tadesse Kebede Guge[11], Tesfa Tegegne Asfaw[4], Chiamaka Ijeoma Chukwuneke[21], Paul Röttger[22], Seid Muhie Yimam[5], Nedjma Ousidhoum[23]

[1]Imperial College London, [2]Bayero University Kano, [3]DSFSI, University of Pretoria, [4]Bahir Dar University, [5]University of Hamburg, [6]Mila, McGill University & Canada CIFAR AI Chair, [7]Northeastern University, [8]Maseno University, [9]Digital Umuganda, [10]HausaNLP, [11]Haramaya University, [12]Al Akhawayn University, [13]Uppsala University, [14]Istanbul Technical University, [15]SADiLaR, [16]University of Deusto, [17]Independent Researcher, [18]Instituto Politécnico Nacional, [19]Wollo University, [20]Jigjiga University, [21]Lancaster University, [22]Bocconi University, [23]Cardiff University

Contact: s.muhammad@imperial.ac.uk, seid.muhie.yimam@uni-hamburg.de

Abstract

Hate speech and abusive language are global phenomena that need socio-cultural background knowledge to be understood, identified, and moderated. However, in many regions of the Global South, there have been several documented occurrences of (1) absence of moderation and (2) censorship due to the reliance on keyword spotting out of context. Further, high-profile individuals have frequently been at the center of the moderation process, while large and targeted hate speech campaigns against minorities have been overlooked. These limitations are mainly due to the lack of high-quality data in the local languages and the failure to include local communities in the collection, annotation, and moderation processes. To address this issue, we present **AfriHate**: a multilingual collection of hate speech and abusive language datasets in 15 African languages, annotated by native speakers. We report the challenges related to the construction of the datasets and present various classification baseline results with and without using LLMs. We find that model performance highly depends on the language and that multilingual models can help boost the performance in low-resource

1 Introduction

> *No one is born hating another person because of the color of his skin, or his background, or his religion. People must learn to hate, and if they can learn to hate, they can be taught to love, for love comes more naturally to the human heart than its opposite.* – (Mandela, 1994)

Hate speech and abusive language are global phenomena that highly depend on specific socio-cultural contexts. Although they deviate from the norm on social media, hate speech and abusive language quickly attract significant attention, spread among online communities (Mathew et al., 2019), and incite harm or violence on individuals in real life (Saha et al., 2019). Tangible efforts to address these problems must take social and cultural contexts into account (Shahid and Vashistha, 2023). However, in the absence of high-quality data or when excluding local voices from the collection and annotation processes, one may fail to build assistive tools that help address the problem and moderate such content.

Collecting hate speech and offensive language

BRIGHTER: BRIdging the Gap in Human-Annotated Textual Emotion Recognition Datasets for 28 Languages

arXiv:2502.11926v4 [cs.CL] 29 May 2025

Shamsuddeen Hassan Muhammad[1,2*], Nedjma Ousidhoum[3*], Idris Abdulmumin[4], Jan Philip Wahle[5], Terry Ruas[5], Meriem Beloucif[6], Christine de Kock[7], Nirmal Surange[8], Daniela Teodorescu[9], Ibrahim Said Ahmad[10], David Ifeoluwa Adelani[11,12,13], Alham Fikri Aji[14], Felermino D. M. A. Ali[15], Ilseyar Alimova[31], Vladimir Araujo[16], Nikolay Babakov[17], Naomi Baes[7], Ana-Maria Bucur[18,19], Andiswa Bukula[20], Guanqun Cao[21], Rodrigo Tufiño[22], Rendi Chevi[14], Chiamaka Ijeoma Chukwuneke[23], Alexandra Ciobotaru[18], Daryna Dementieva[24], Murja Sani Gadanya[2], Robert Geislinger[25], Bela Gipp[5], Oumaima Hourrane[26], Oana Ignat[27], Falalu Ibrahim Lawan[28], Rooweither Mabuya[20], Rahmad Mahendra[29], Vukosi Marivate[4,30], Alexander Panchenko[31,32], Andrew Piper[12], Charles Henrique Porto Ferreira[33], Vitaly Protasov[32], Samuel Rutunda[34], Manish Shrivastava[8], Aura Cristina Udrea[35], Lilian Diana Awuor Wanzare[36], Sophie Wu[12], Florian Valentin Wunderlich[5], Hanif Muhammad Zhafran[37], Tianhui Zhang[38], Yi Zhou[3], Saif M. Mohammad[39]

[1]Imperial College London, [2]Bayero University Kano, [3]Cardiff University, [4]Data Science for Social Impact, University of Pretoria, [5]University of Göttingen, [6]Uppsala University, [7]University of Melbourne, [8]IIIT Hyderabad, [9]University of Alberta, [10]Northeastern University, [11]MILA, [12]McGill University, [13]Canada CIFAR AI Chair, [14]MBZUAI, [15]LIACC, FEUP, University of Porto, [16]Sailplane AI, [17]University of Santiago de Compostela, [18]University of Bucharest, [19]Universitat Politècnica de València, [20]SADiLaR, [21]University of York, [22]Universidad Politécnica Salesiana, [23]Lancaster University, [24]Technical University of Munich, [25]Hamburg University, [26]Al Akhawayn University, [27]Santa Clara University, [28]Kaduna State University, [29]Universitas Indonesia, [30]Lelapa AI, [31]Skoltech, [32]AIRI, [33]Centro Universitário FEI, [34]Digital Umuganda, [35]National University of Science and Technology Politehnica Bucharest, [36]Maseno University, [37]Institut Teknologi Bandung, [38]University of Liverpool, [39]National Research Council Canada

Contact: s.muhammad@imperial.ac.uk, OusidhoumN@cardiff.ac.uk

Abstract

People worldwide use language in subtle and complex ways to express emotions. Although emotion recognition–an umbrella term for several NLP tasks–impacts various applications within NLP and beyond, most work in this area has focused on high-resource languages. This has led to significant disparities in research efforts and proposed solutions, particularly for under-resourced languages, which often lack high-quality annotated datasets. In this paper, we present BRIGHTER–a collection of multilabeled, emotion-annotated datasets in 28 different languages and across several domains. BRIGHTER primarily covers low-resource languages from Africa, Asia, Eastern Europe, and Latin America, with instances labeled by fluent speakers. We highlight the challenges related to the data collection and annotation processes, and then report experimental results for monolingual and crosslingual multi-label emotion identification, as well as emotion intensity recognition. We analyse the variability in performance across languages and text domains, both with and without the use of LLMs, and show that the BRIGHTER datasets represent a meaningful step towards addressing the gap in text-based emotion recognition.

Figure 1: **Languages included in BRIGHTER** and their language families.

[*]Equal contribution

Examples of Text Classification Tasks

MasakhaNEWS: News Topic Classification for African languages

David Ifeoluwa Adelani[1*], Marek Masiak[1*], Israel Abebe Azime[2], Jesujoba Oluwadara Alabi[2],
Atnafu Lambebo Tonja[3,6], Christine Mwase[4], Odunayo Ogundepo[5], Bonaventure F. P. Dossou[6,7,8,9],
Akintunde Oladipo[5], Doreen Nixdorf, Chris Chinenye Emezue[9,10], Sana Sabah al-azzawi[11],
Blessing K. Sibanda, Davis David[12], Lolwethu Ndolela, Jonathan Mukiibi[13], Tunde O. Ajayi[14],
Tatiana Moteu Ngoli[15], Brian Odhiambo, Abraham Toluwase Owodunni, Nnaemeka C. Obiefuna,
Muhidin Mohamed[16], Shamsuddeen Hassan Muhammad[17], Teshome Mulugeta Ababu[18],
Saheed S. Abdullahi[19], Mesay Gemeda Yigezu[3], Tajuddeen Gwadabe, Idris Abdulmumin[20],
Mahlet Taye Bame, Oluwabusayo O. Awoyomi[21], Iyanuoluwa Shode[22], Tolulope Anu Adelani,
Habiba Abdulganiy Kailani, Abdul-Hakeem Omotayo[23], Adetola Adeeko, Afolabi Abeeb,
Anuoluwapo Aremu, Olanrewaju Samuel[24], Clemencia Siro[25], Wangari Kimotho[26],
Onyekachi Raphael Ogbu, Chinedu E. Mbonu[27], Chiamaka I. Chukwuneke[27,28], Samuel Fanijo[29],
Jessica Ojo, Oyinkansola F. Awosan, Tadesse Kebede Guge[30], Sakayo Toadoum Sari[26,31],
Pamela Nyatsine, Freedmore Sidume[32], Oreen Yousuf, Mardiyyah Oduwole[33], Kanda P. Tshinu,
Ussen Kimanuka[34], Thina Diko, Siyanda Nxakama, Sinodos G. Nugussie[18], Abdulmejid Tuni Johar,
Shafie Abdi Mohamed[34], Fuad Mire Hassan[35], Moges Ahmed Mehamed[36], Evrard Ngabire[37],
Jules Twagirayezu, Ivan Ssenkungu, and Pontus Stenetorp[1]

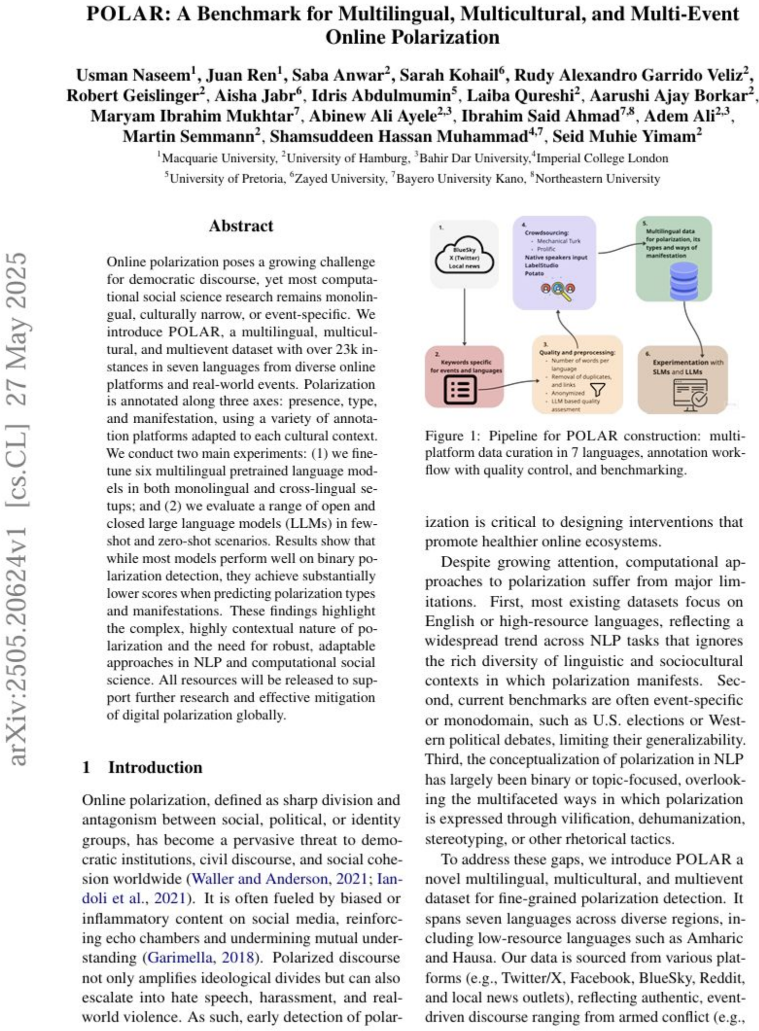
[♡] Masakhane NLP, Africa, [1]University College London, United Kingdom, [2] Saarland University, Germany,
[3] Instituto Politécnico Nacional, Mexico, [4]Fudan University, China, [5] University of Waterloo, Canada, [6] Lelapa AI,
[7]McGill University, Canada, [8] Mila Quebec AI Institute, Canada, [9] Lanfrica, [10] Technical University of Munich, Germany
[11] Luleå University of Technology, Sweden, [12]Tanzania Data Lab, Tanzania [13] Makerere University, Uganda,
[14] Insight Centre for Data Analytics, Ireland, [15]Paderborn University, Germany, [16]Aston University, UK,
[17]University of Porto, Portugal, [18] Dire Dawa University, Ethiopia [19]Kaduna State University, Nigeria,
[20]Ahmadu Bello University, Nigeria, [21]The College of Saint Rose, USA [22]Montclair State University, USA,
[23] University of California, Davis, [24]University of Rwanda, Rwanda [25]University of Amsterdam, The Netherlands,
[26]AIMS, Cameroon, [27]Nnamdi Azikiwe University, Nigeria , [28]Lancaster University, United Kingdom,
[29]Iowa State University, USA, [30]Haramaya University, Ethiopia, [31]AIMS, Senegal, [32]BIUST, Botswana,
[33]NOUN, Nigeria, [34] PAUSTI, Kenya, [34] Jamhuriya University, Somalia, [35]Somali National University,
[36]Wuhan University of Technology, China, [37]Deutschzentrum an der Universität Burundi, Burundi

Correspondence: d.adelani@ucl.ac.uk

Abstract

Despite representing roughly a fifth of the world population, African languages are underrepresented in NLP research, in part due to a lack of datasets. While there are individual language-specific datasets for several tasks, only a handful of tasks (e.g. named entity recognition and machine translation) have datasets covering geographical and typologically-diverse African languages. In this paper, we develop MasakhaNEWS—the largest dataset for news topic classification covering 16 languages widely spoken in Africa. We provide and evaluate a set of baseline models by training classical machine learning models and fine-tuning several language models. Furthermore, we explore several alternatives to full fine-tuning of language models that are better suited for zero-shot and few-shot learning, such as: cross-lingual parameter-efficient fine-tuning (MAD-X), pattern exploiting training (PET), prompting language models (ChatGPT), and prompt-free sentence transformer fine-tuning (SetFit and the co:here embedding API). Our evaluation in a few-shot setting, shows that with as little as 10 examples per label, we achieve more than 90% (i.e. 86.0 F1 points) of the performance of fully supervised training (92.6 F1 points) leveraging the PET approach. Our work shows that existing supervised approaches work well for all African languages and that language models with only a few supervised samples can reach competitive performance, both findings which demonstrate the applicability of existing NLP techniques for African languages.

[*] Equal contribution



POLAR: A Benchmark for Multilingual, Multicultural, and Multi-Event Online Polarization

Usman Naseem[1], Juan Ren[1], Saba Anwar[2], Sarah Kohail[6], Rudy Alexandro Garrido Veliz[2],
Robert Geislinger[2], Aisha Jabr[6], Idris Abdulmumin[5], Laiba Qureshi[2], Aarushi Ajay Borkar[2],
Maryam Ibrahim Mukhtar[7], Abinew Ali Ayele[2,3], Ibrahim Said Ahmad[7,8], Adem Ali[2,3],
Martin Semmann[2], Shamsuddeen Hassan Muhammad[4,7], Seid Muhie Yimam[2]
[1]Macquarie University, [2]University of Hamburg, [3]Bahir Dar University,[4]Imperial College London
[5]University of Pretoria, [6]Zayed University, [7]Bayero University Kano, [8]Northeastern University

Abstract

Online polarization poses a growing challenge for democratic discourse, yet most computational social science research remains monolingual, culturally narrow, or event-specific. We introduce POLAR, a multilingual, multicultural, and multievent dataset with over 23k instances in seven languages from diverse online platforms and real-world events. Polarization is annotated along three axes: presence, type, and manifestation, using a variety of annotation platforms adapted to each cultural context. We conduct two main experiments: (1) we fine-tune six multilingual pretrained language models in both monolingual and cross-lingual setups; and (2) we evaluate a range of open and closed large language models (LLMs) in few-shot and zero-shot scenarios. Results show that while most models perform well on binary polarization detection, they achieve substantially lower scores when predicting polarization types and manifestations. These findings highlight the complex, highly contextual nature of polarization and the need for robust, adaptable approaches in NLP and computational social science. All resources will be released to support further research and effective mitigation of digital polarization globally.

1.
BlueSky
X (Twitter)
Local news
2.
Keywords specific for events and languages
3.
Quality and preprocessing:
- Number of words per language
- Removal of duplicates, and links
- Anonymized
- LLM based quality assesment
4.
Crowdsourcing:
- Mechanical Turk
- Prolific
Native speakers input
LabelStudio
Potato
5.
Multilingual data for polarization, its types and ways of manifestation
6.
Experimentation with SLMs and LLMs

Figure 1: Pipeline for POLAR construction: multi-platform data curation in 7 languages, annotation workflow with quality control, and benchmarking.

1 Introduction

Online polarization, defined as sharp division and antagonism between social, political, or identity groups, has become a pervasive threat to democratic institutions, civil discourse, and social cohesion worldwide (Waller and Anderson, 2021; Iandoli et al., 2021). It is often fueled by biased or inflammatory content on social media, reinforcing echo chambers and undermining mutual understanding (Garimella, 2018). Polarized discourse not only amplifies ideological divides but can also escalate into hate speech, harassment, and real-world violence. As such, early detection of polarization is critical to designing interventions that promote healthier online ecosystems.

Despite growing attention, computational approaches to polarization suffer from major limitations. First, most existing datasets focus on English or high-resource languages, reflecting a widespread trend across NLP tasks that ignores the rich diversity of linguistic and sociocultural contexts in which polarization manifests. Second, current benchmarks are often event-specific or monodomain, such as U.S. elections or Western political debates, limiting their generalizability. Third, the conceptualization of polarization in NLP has largely been binary or topic-focused, overlooking the multifaceted ways in which polarization is expressed through vilification, dehumanization, stereotyping, or other rhetorical tactics.

To address these gaps, we introduce POLAR a novel multilingual, multicultural, and multievent dataset for fine-grained polarization detection. It spans seven languages across diverse regions, including low-resource languages such as Amharic and Hausa. Our data is sourced from various platforms (e.g., Twitter/X, Facebook, BlueSky, Reddit, and local news outlets), reflecting authentic, event-driven discourse ranging from armed conflict (e.g.,

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

Babangida Sani[1,3], Aakansha Soy[1], Sukairaj Hafiz Imam[2,3], Ahmad Mustapha[1,3],
Lukman Jibril Aliyu[3], Idris Abdulmumin[3,4], Ibrahim Said Ahmad[2,3,5],
Shamsuddeen Hassan Muhammad[2,3,6]
[1]Kalinga University, [2]Bayero University, Kano, [3]HausaNLP
[4]DSFSI, University of Pretoria, [5]Northeastern University, [6]Imperial College London
correspondence: bsani480@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 scraped 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 Afri-centric models (AfriTeVa, AfriBERTa, AfroXLMR, and AfroXLMR-76L) on the resulting dataset and assessed their performance using accuracy and F1-score metrics. AfroXLMR achieved the highest performance with an accuracy of 99.23% and an F1 score of 99.21%, demonstrating its effectiveness for Hausa text detection. Our dataset is made publicly available[1] to enable further research.

Keywords: *Large Language Model (LLM), Natural Language Processing (NLP), Hausa, Transformer, Gemini, Fine-tune*

1 Introduction

Hausa is among the most spoken Chadic languages, belonging to the Afroasiatic phylum. Over 100 million people are estimated to speak the language, with the majority of speakers living in Northern Nigeria and the Republic of Niger, respectively (Inuwa-Dutse, 2021). However, from computational linguistics, it is regarded as a low-resource language, having insufficient resources to support tasks involving Natural Language Processing (NLP; Adam et al. 2023; Muhammad et al. 2023).

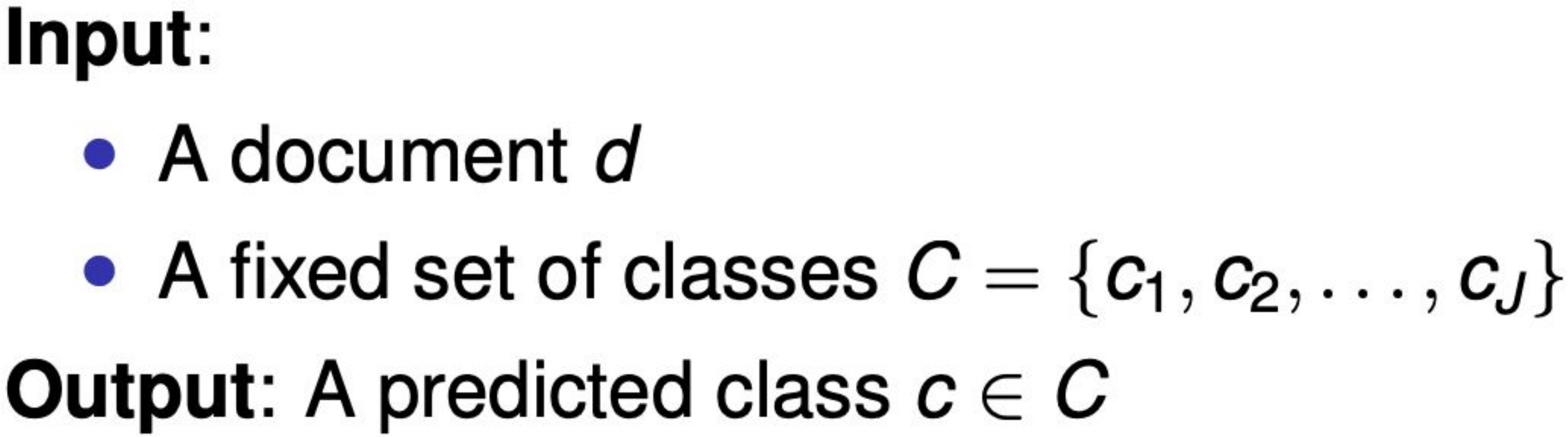
Hausa language is written in either the Latin (or ***Boko***) and Arabic (or ***Ajami***) script (Jaggar, 2006). The *Boko* script, existing since the 1930s, was introduced by the British colonial administration, and is used in education, government, and digital communication. The *Ajami* script, an 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 propagate harmful or misleading content, such as fake news or hate speech (Xie et al., 2024), or even fake or artificial scholarship. To ensure the authenticity, accuracy, and trustworthiness of content, there is a need for machine-generated text detectors. Extensive research has been undertaken to differentiate between machine-generated texts (MGTs) and human-generated texts (HGTs), primarily employing model-based approaches (Wang et al., 2023; Alshammari, 2024; Ji et al., 2024).

In existing studies, (i) focus has mainly been on high-resource languages like English; (ii) there are no reliable detectors for detecting human vs. AI-generated text in the Hausa language; (iii) ensuring content authenticity is difficult, especially for low resource languages like Hausa (Ji et al., 2024). We

[1] https://github.com/TheBangis/hausa_corpus

Text Classification: Definition



Input:

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

Output: A predicted class $c \in C$

Types of text Classification

- Text classification tasks differ based on the number of labels a model can assign.

- Understanding these types helps determine the right model, loss function, and evaluation metrics.

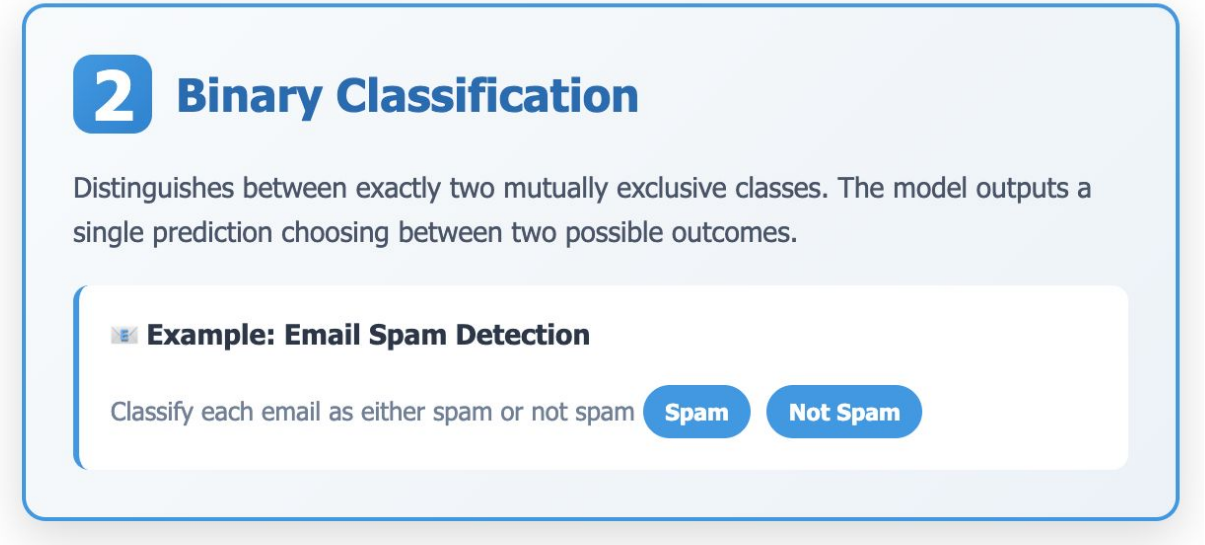
- Three types:
  - Binary Classification
  - Multiclass Classification
  - Multilabel Classification

Binary Classification

- The model chooses one of two possible classes.

- Labels: *{0,1}*, *{positive, negative}*, *{spam, not spam}*, *{hate, non-hate}*.

- Used when decisions are yes/no, true/false, or presence/absence.



2 Binary Classification

Distinguishes between exactly two mutually exclusive classes. The model outputs a single prediction choosing between two possible outcomes.

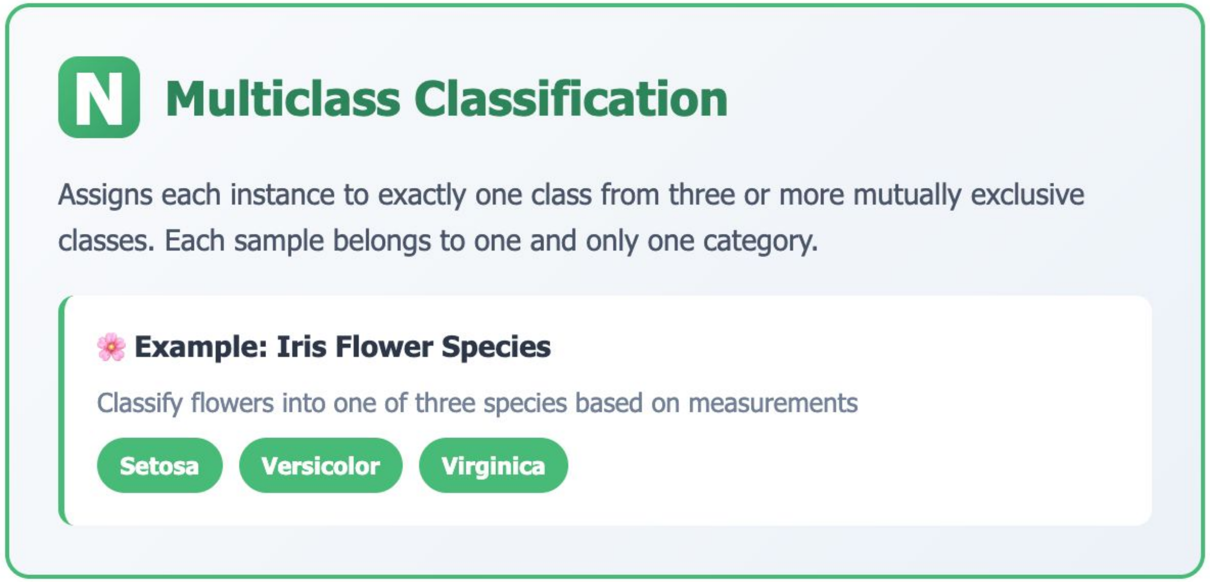
Example: Email Spam Detection

Classify each email as either spam or not spam

Muhammad et al. (2022). NaijaSenti: A Nigerian Twitter Sentiment Corpus for Multilingual Sentiment Analysis

Multiclass Classification

- The model selects one class from more than two categories.
- Only one label can be assigned per instance.
- Examples:
  - Flower classification: Setosa, Versicolor, Virginica
  - Emotion classification: anger, joy, sadness, fear, surprise.



N Multiclass Classification

Assigns each instance to exactly one class from three or more mutually exclusive classes. Each sample belongs to one and only one category.

🌸 Example: Iris Flower Species

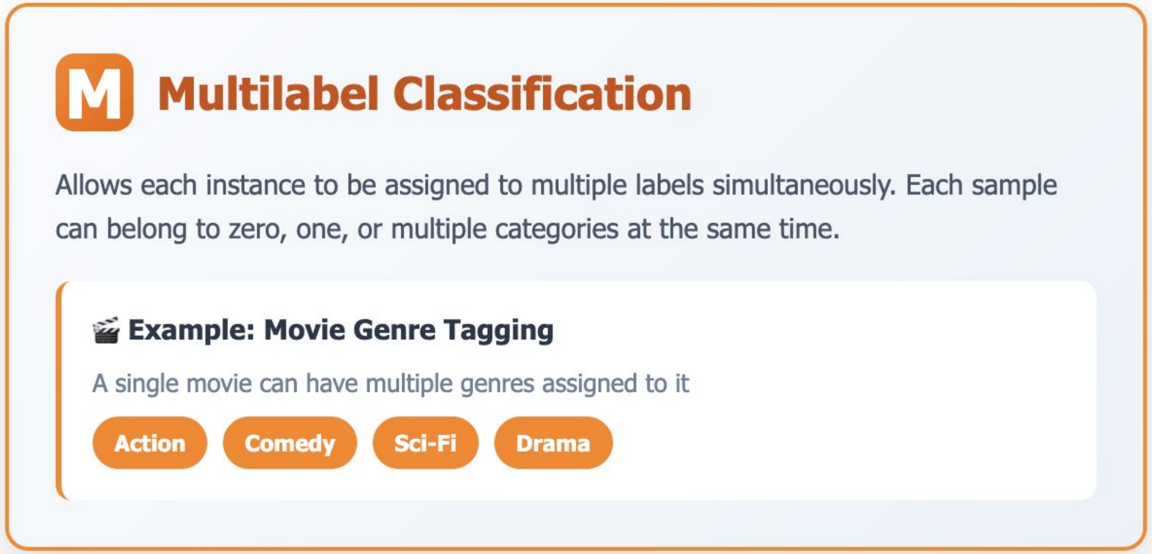
Classify flowers into one of three species based on measurements

Muhammad et al. (2025). BRIGHTER: Bridging Human-Annotated Emotion Datasets for 28 Languages. ACL 2025 (Best Resource Paper ACL2025).

Multilabel Classification

- The model assigns multiple labels simultaneously.

- Output is a set of labels, e.g., {politics, religion}.
  - Toxicity classification with multiple attributes: *insult, threat, hate, harassment*.



M Multilabel Classification

Allows each instance to be assigned to multiple labels simultaneously. Each sample can belong to zero, one, or multiple categories at the same time.

🎬 **Example: Movie Genre Tagging**

A single movie can have multiple genres assigned to it

Muhammad et al. (2025) AfriHate: A Benchmark for Hate and Abusive Language Detection in 15 African Languages.

Classification Methods

- **Rule-based Methods:** Use handcrafted linguistic or domain rules (e.g., keyword patterns, lexicons, regular expressions)

- **Probabilistic Models:** Use statistical estimates of likelihood (e.g., Naïve Bayes, Logistic Regression)

- **Supervised Machine Learning:** Learn patterns from labeled data (e.g., SVM, Decision Trees, Neural Networks)

- **Deep Learning Models***:* Automatically learn representations from text (e.g., CNNs, RNNs, Transformers, BERT-style models)

Classification Methods: Rule-based rules

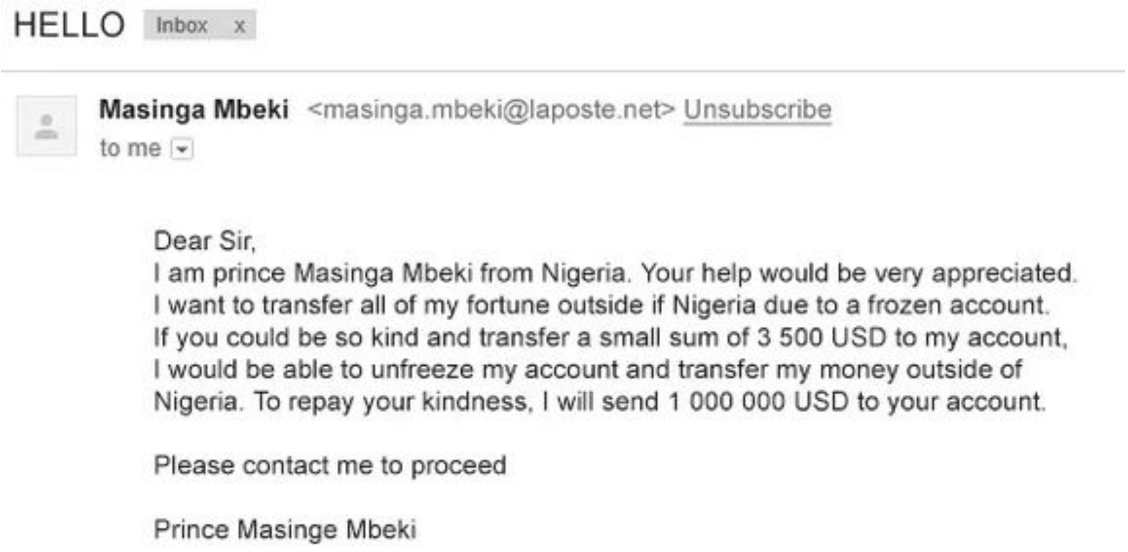
Models classify text using manually crafted rules built from keywords, patterns, or linguistic features.

If the message contains any of:

`"dollars"` AND `"you have been selected"`

`"black-listed address"` → **Classify as: SPAM**

`"Nigeria"` AND `"Prince"` → **Classify as: SPAM**



HELLO Inbox x

Masinga Mbeki <masinga.mbeki@laposte.net> Unsubscribe
to me

Dear Sir,
I am prince Masinga Mbeki from Nigeria. Your help would be very appreciated.
I want to transfer all of my fortune outside if Nigeria due to a frozen account.
If you could be so kind and transfer a small sum of 3 500 USD to my account,
I would be able to unfreeze my account and transfer my money outside of
Nigeria. To repay your kindness, I will send 1 000 000 USD to your account.

Please contact me to proceed

Prince Masinge Mbeki

Classification Methods: Rule-based rules

Strengths

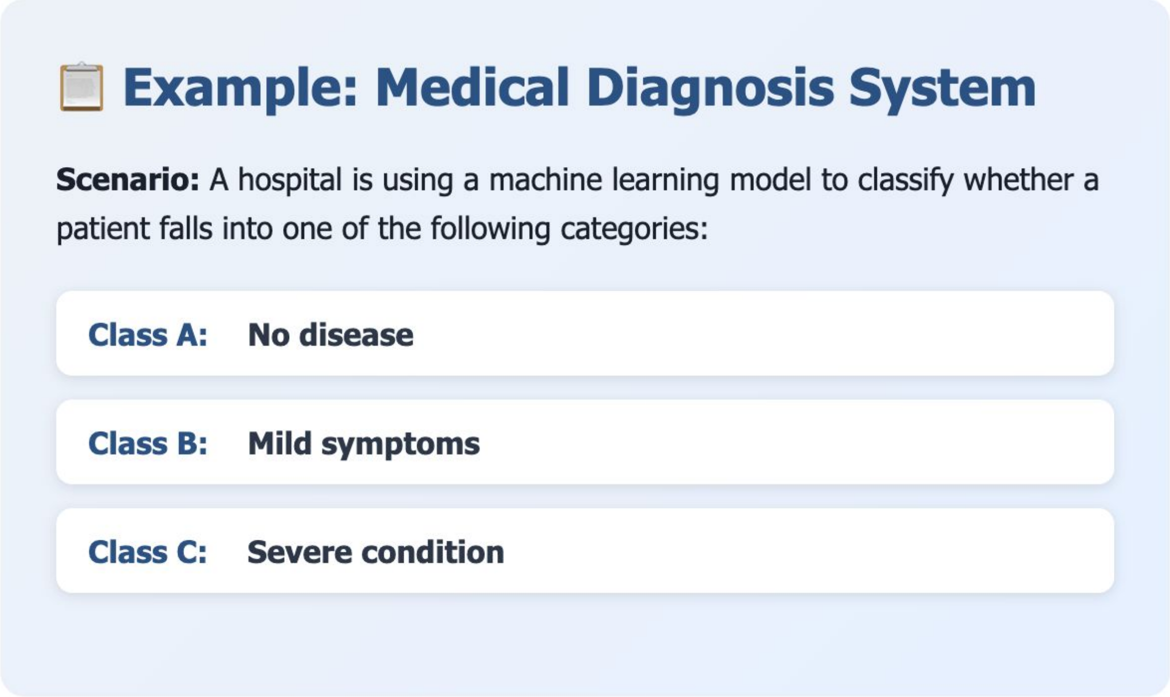
- High accuracy when domain experts write good rules
- Transparent and explainable (easy to understand why a prediction was made)

Limitations

- Time-consuming to build and maintain
- Rules break easily when language changes
- Not scalable for large or diverse datasets

Probabilistic classifier

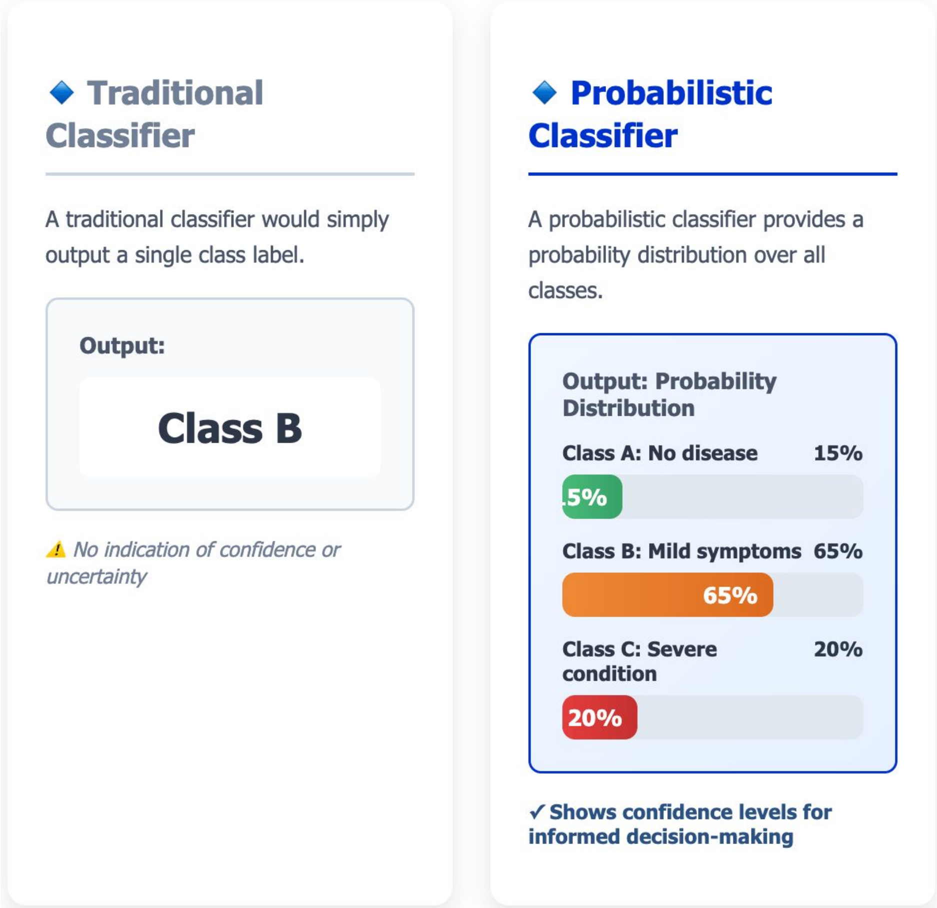
Instead of forcing a **hard label**, probabilistic classifiers show *how confident* the model is.



📋 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**



◆ Traditional Classifier

A traditional classifier would simply output a single class label.

Output:

Class B

⚠️ *No indication of confidence or uncertainty*

◆ Probabilistic Classifier

A probabilistic classifier provides a probability distribution over all classes.

Output: Probability Distribution

Class A: No disease **15%**

5%

Class B: Mild symptoms **65%**

65%

Class C: Severe condition **20%**

20%

✓ Shows confidence levels for informed decision-making

Probabilistic classifier

Strengths

- Provide confidence scores, enabling better *decision-making*
- Capture uncertainty instead of forcing a single hard label
- Useful in risk-sensitive applications (e.g., *medical, financial*)

Probabilistic classifier

Limitations

- Less interpretable when probabilities are poorly calibrated
- Can be computationally more expensive than simple rule-based systems
- Require large, representative training data for reliable probability estimates
- Sensitive to data imbalance (may output skewed probabilities)

Risk-Sensitive Example 1: Fraud Detection

A transaction is being checked for fraud.

- **P(fraud) = 0.10**

- Even 10% may be enough to trigger an extra verification step (e.g., sending an SMS or blocking temporarily).

- **Risk sensitive:** Prevents large financial losses.

Risk-Sensitive Example 1: Medical Diagnosis

A model predicts whether a patient has a severe disease.

P(severe disease) = 0.30

Even though the model's top class might be "***mild***", a 30% chance of a severe condition is too risky to ignore. The doctor may order more tests before taking action.

Risk-sensitive: The *cost of being wrong* is very high (a missed diagnosis).

Probabilistic classifier

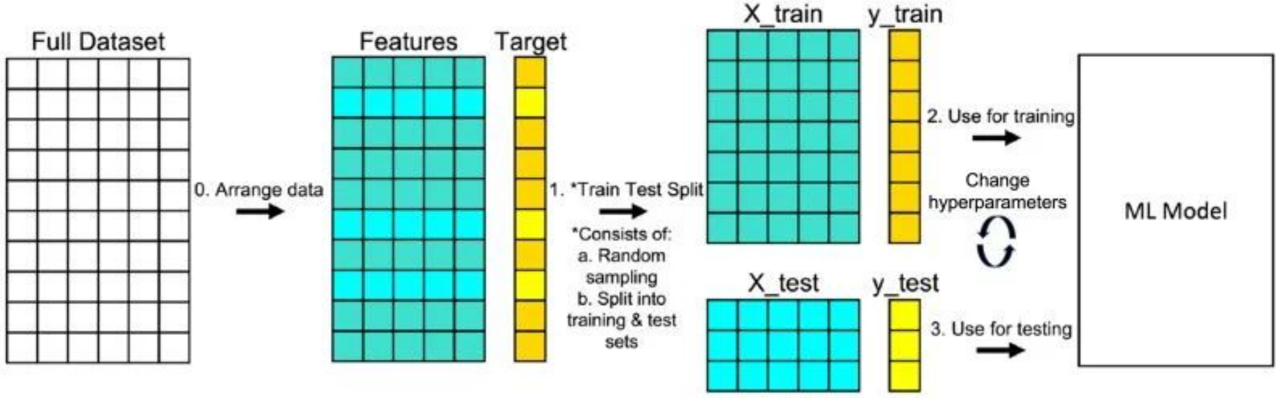
Probabilistic classifiers enhance *decision-making, reduce errors, and allow for more flexible system integration.* This makes them especially useful in high-stakes domains such as healthcare, finance, and automated systems.

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*



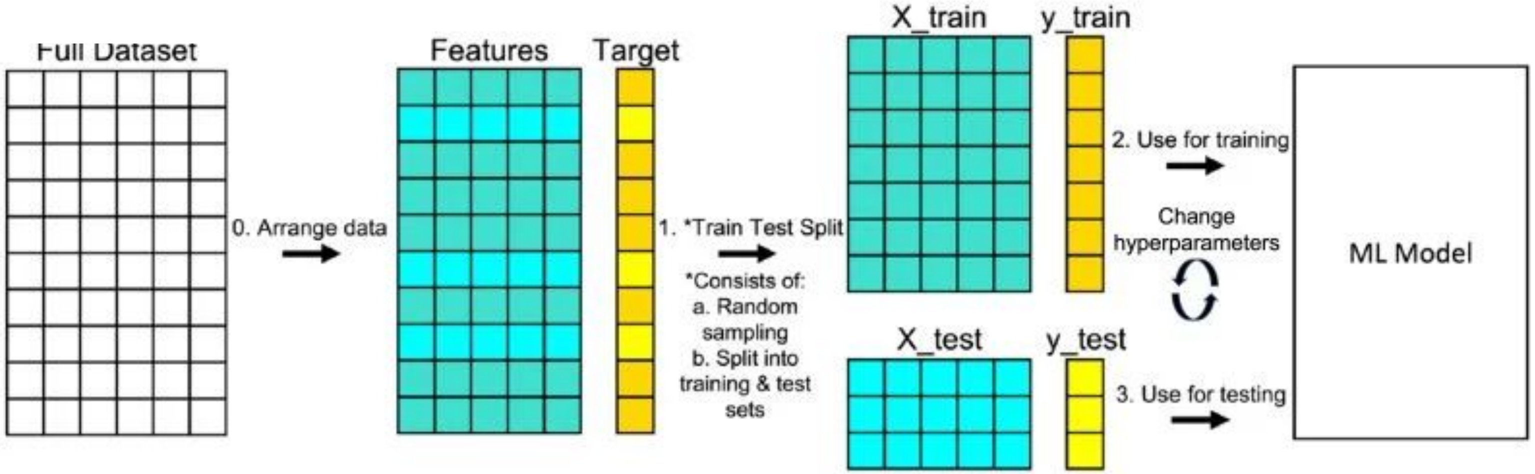
Supervised Machine Learning

Input:

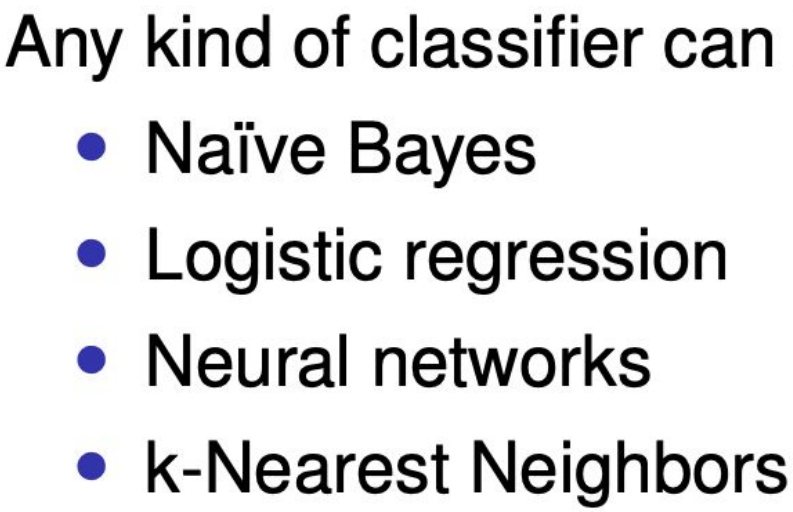
- A document d
- A fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
- A training set of m hand-labeled documents: $(d_1, c_1), (d_2, c_2), \ldots, (d_m, c_m)$

Output:

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



Supervised Machine Learning



Any kind of classifier can be used:

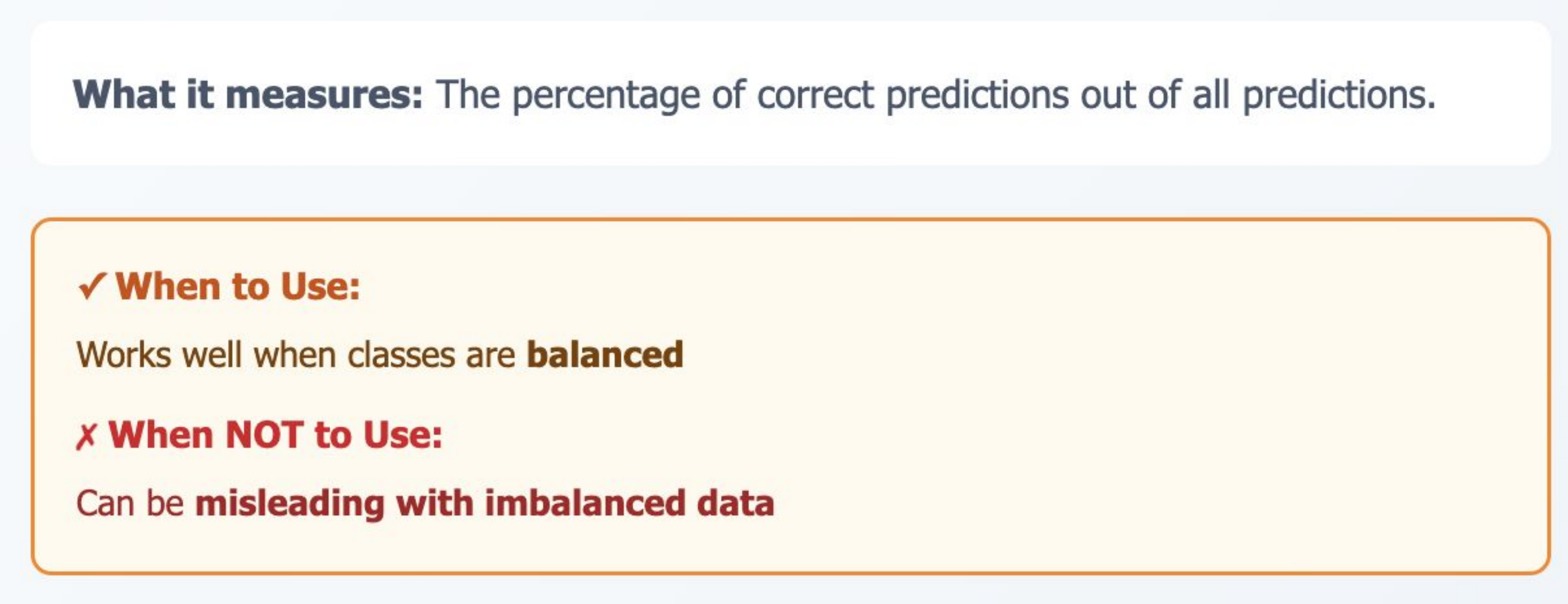
- Naïve Bayes
- Logistic regression
- Neural networks
- k-Nearest Neighbors

More details in your machine learning class

Evaluation of Text Classification

- Accuracy: Works when classes are balanced, but can be misleading with imbalance.
- Macro Precision / Macro Recall: Treats all classes equally; good when minority classes matter.
- Macro F1-Score: Balances precision and recall across all classes; preferred with class imbalance.
- Micro F1-Score: Aggregates over all instances; useful when overall performance is more important than per-class parity.

Accuracy



What it measures: The percentage of correct predictions out of all predictions.

✓ **When to Use:**

Works well when classes are **balanced**

✗ **When NOT to Use:**

Can be **misleading with imbalanced data**

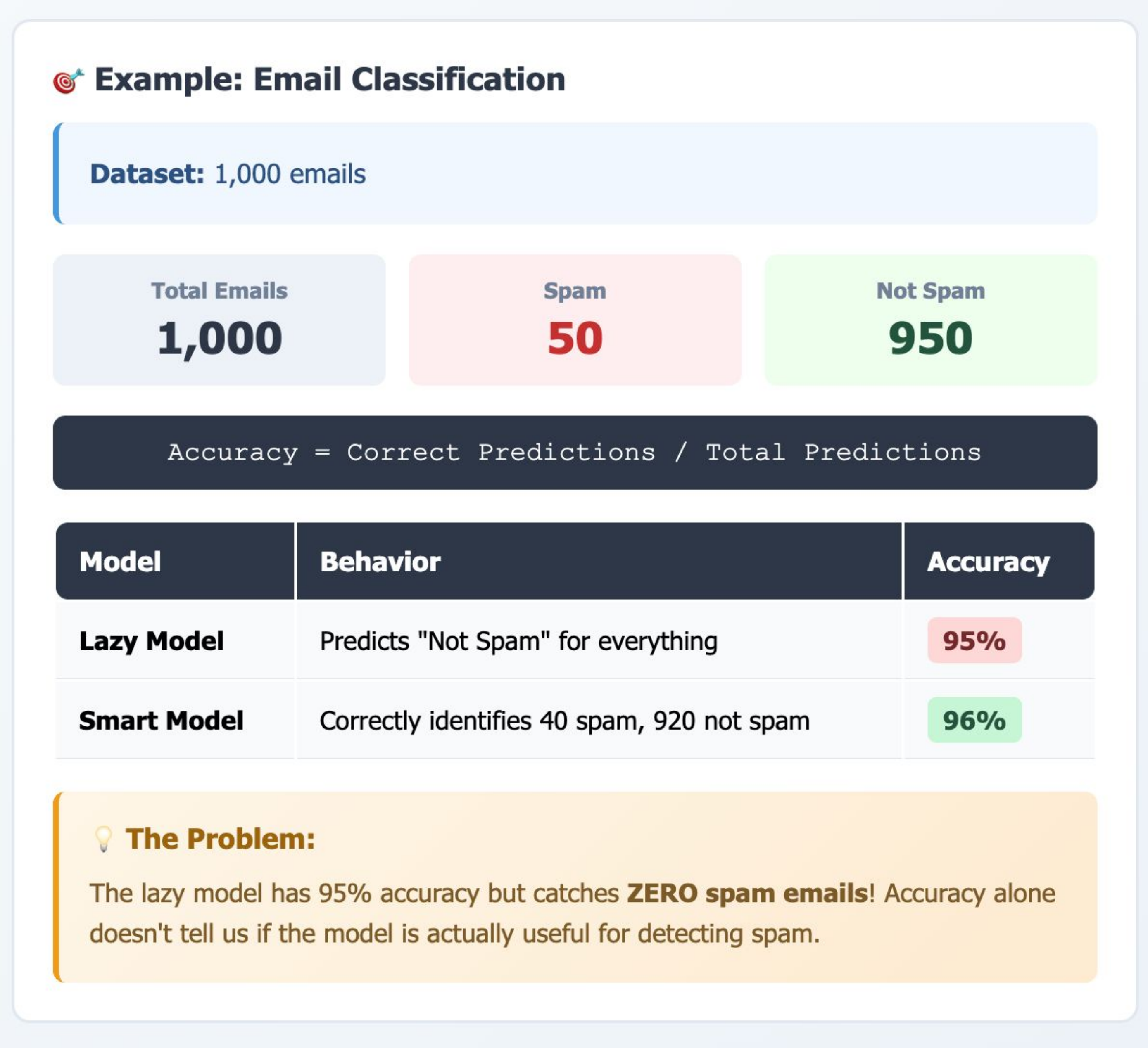
Accuracy

Lazy Model: Just predicts "Not Spam" for EVERYTHING

- It gets 950 emails correct (all the legitimate ones)
- It misses all 50 spam emails
- **Accuracy = 950/1000 = 95%**

Smart Model: Actually tries to detect spam

- Correctly identifies 40 spam emails
- Correctly identifies 920 legitimate emails
- Misses 10 spam + 30 false alarms
- **Accuracy = 960/1000 = 96%**



🎯 Example: Email Classification

Dataset: 1,000 emails

Total Emails	Spam	Not Spam
1,000	50	950

```
Accuracy = Correct Predictions / Total Predictions
```

Model	Behavior	Accuracy
Lazy Model	Predicts "Not Spam" for everything	95%
Smart Model	Correctly identifies 40 spam, 920 not spam	96%

💡 **The Problem:**

The lazy model has 95% accuracy but catches **ZERO spam emails**! Accuracy alone doesn't tell us if the model is actually useful for detecting spam.

Common Metrics for Classification

- **Accuracy**: Good for balanced datasets, Not reliable for imbalanced

- **Precision**: Out of predicted positives, how many are correct?

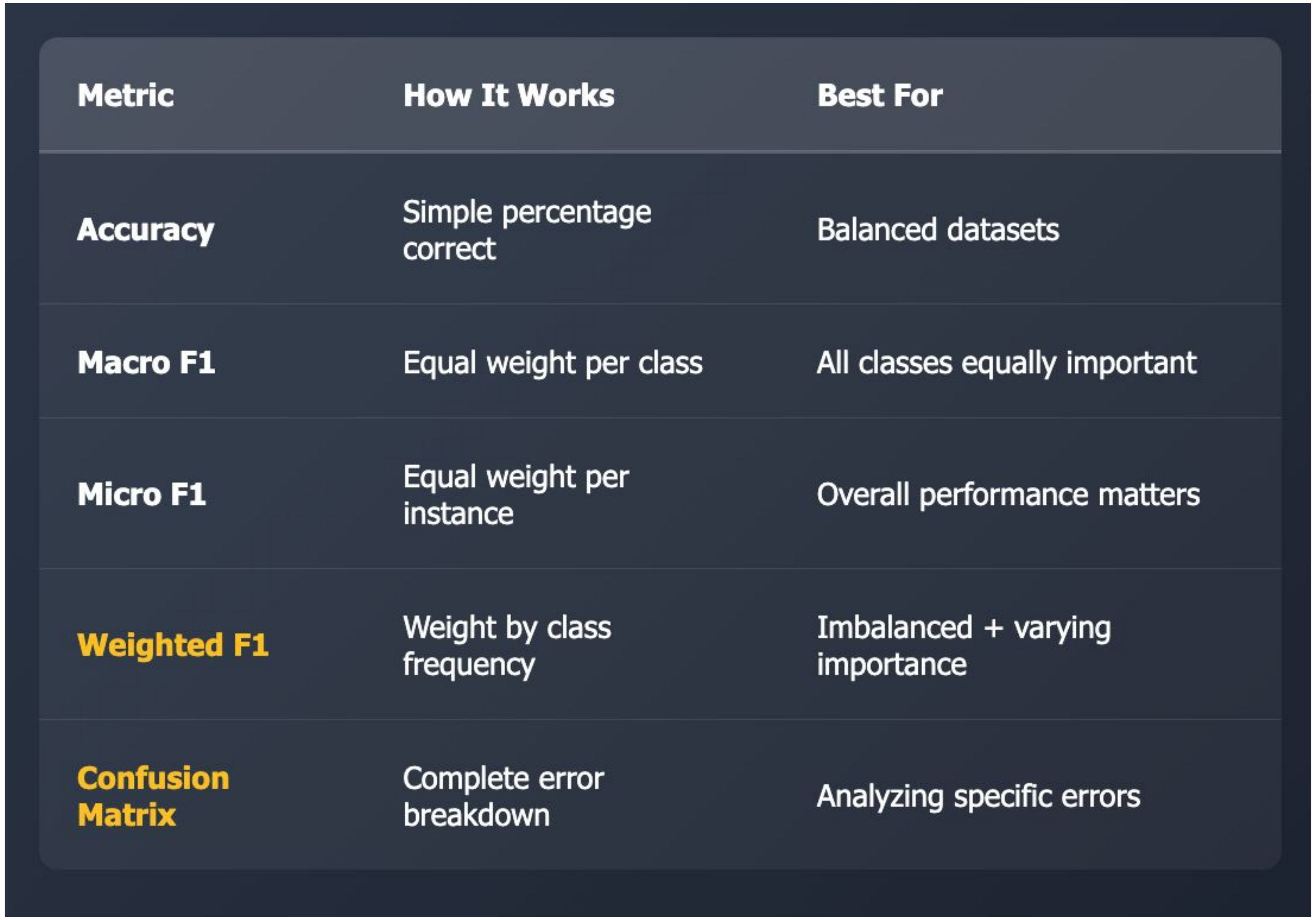
- **Recall**: Out of actual positives, how many were found?

- **F1-Score:** Harmonic mean of Precision & Recall

Evaluation of Text Classification

- Weighted F1-Score: Accounts for class frequency; good when dataset is imbalanced but class importance varies.

- Confusion Matrix: Essential for analyzing per-class errors.



Metric	How It Works	Best For
Accuracy	Simple percentage correct	Balanced datasets
Macro F1	Equal weight per class	All classes equally important
Micro F1	Equal weight per instance	Overall performance matters
Weighted F1	Weight by class frequency	Imbalanced + varying importance
Confusion Matrix	Complete error breakdown	Analyzing specific errors

Evaluation of Text Classification

- Assess how well the model generalizes to **unseen data**
- Choose appropriate evaluation metrics depending on the task and data characteristics (e.g., balanced and unbalanced dataset)
- Compare models and identify the best-performing approach

Train / Dev / Test Splits in NLP?

- NLP models must generalize to new, unseen text—not just memorize examples.

- To achieve this, we divide the dataset into three parts, each serving a specific purpose:
    1. **Training Set:** Learns the model parameters from labeled text.
    2. **Dev (Validation) Set:** Helps tune hyperparameters and choose the best model without touching the test set.
    3. **Test Set:** Used once at the end to measure true performance on unseen data.

The three-way split prevents overfitting and ensures that evaluation reflects real-world generalization.

Training Set: Learn the Model

Training set is used to **learn patterns from text**, such as word distributions, syntactic structures, or associations between input and labels.

- Used to fit model parameters (e.g., weights in logistic regression, transformer attention parameters).
- The model sees this data many times during training.

Training Set teach the model the statistical regularities of the language/task.

Development (Dev) Set: Tune and Select

Dev set is used for **hyperparameter tuning**, such as learning rate, model size, regularization strength, or max sequence length.

- Also used for **early stopping** to prevent overfitting.

- Prevents "peeking" at the test set, which would artificially inflate performance.

To choose the best version of the model without contaminating the final evaluation.

Test Set: Final Unbiased Evaluation

Test set is used **only once** after all training and tuning decisions are finished.

- It measures how well the NLP model performs on **unseen data**.
- Provides an unbiased estimate of generalization: e.g., does the classifier handle new tweets, new reviews, new Hausa or Swahili dialectal variations?

“The test set must be used only once, for final evaluation.”

Why Testing Directly on the Test Set Is Wrong?

- You leak information from the test set

- Models become overfitted to the test set

- You lose an unbiased measure of generalization

You leak information from the test set

If you tune hyperparameters or make design decisions based on the test results, you are indirectly letting the model **“see”** the test set.

- The model starts adapting to the specific examples in the test data.
- The test set is no longer unseen.
- Your evaluation is no longer valid.

This is called **test leakage**.

Models become overfitted to the test set

When you evaluate repeatedly on the same test set:

- You choose the model that performs best **on that exact test set**
- Not necessarily the model that generalizes best to new data
- Test accuracy becomes artificially inflated

This makes test performance **meaningless**, because it no longer reflects real-world performance.

You lose an unbiased measure of generalization

In NLP, true generalization means:

- handling new tweets, new dialects, new vocabulary
- working on unseen examples from different domains

If the test set has been used in training/tuning decisions, *you have no trustworthy way to measure that anymore.*

IMPERIAL

Q and A