Major Course Output 2 **Release Date:** October 2, 2025
PinoyBot **Due Date:** November 14, 2025 (Friday), 11:59 PM

1 Project Description

In this project, students will apply the concepts and theories they have learned in the course to develop an intelligent system that can perform language identification. Through this project, students should be able to demonstrate the following learning outcomes:

- **LO3.** Collaboratively design and implement machine learning models for supervised and unsupervised tasks
- **LO5.** Articulate ideas and present results in correct technical written and oral English.

2 Project Specifications

This section contains the specifications for this major course output.

2.1 Overview

In this project, you will train and evaluate a supervised machine learning classifier, specifically a word-level **language identifier** tailored for Filipino code-switched text. Filipino is a morphologically rich language, featuring complex word formations through affixes, infixes, and reduplication, which can make automated processing challenging. Additionally, Filipino exhibits code-switching, a common linguistic phenomenon where speakers alternate between languages within a single utterance or text. This can occur at the inter-word level (e.g., mixing full words from different languages, such as "bawal corrupt") or intra-word level (e.g., applying Filipino morphological rules to English stems, such as "pinull" or "naglunch").

The goal of this project is to build an intelligent system that can accurately identify the language of each word in a code-switched text. Language identification is an important role in the development natural language processing (NLP) applications such as machine translation, sentiment analysis, and speech recognition. This is especially true in multilingual contexts like the Philippines where English and Filipino are frequently intermixed. By addressing training intelligent systems that are sensitive to code-switching, your system can contribute to the better handling of real-world. Filipino data.

2.2 Data Annotation

The foundation of your language identifier will be a high-quality dataset. As a class, you will collectively annotate a real-world dataset. The dataset will be provided by the Center for Language Technologies (CeLT) of the College of Computer Studies, sourced from the Corpus of Historical Filipino English (or CoHFiE). This corpus includes historical texts that exhibit code-switching between Filipino and English, reflecting real-world usage in diverse contexts.

To speed up the annotation process, a BSMS-CS student has already tagged the dataset with a preliminary annotation by a large language model (e.g., GPT-4, Grok, or similar). However, **The LLM is expected to commit mistakes, so the first phase of your project is to validate the LLM annotations manually**, carefully taking into consideration the annotation criteria described below. To make this manageable, we will use a crowdsourced approach, with each group contributing to the effort. Each group will receive a set of samples to validate. The group must check the annotations and correct any errors, resolve ambiguities (e.g., in intra-word code-switching cases), and add notes on edge cases. If discrepancies arise, discuss them in class or group sessions to reach a consensus.

While it may sound tedious, it is essential to go through this manual validation process to (1) appreciate the importance of the role of data collection and annotation in the machine learning process, (2) help you understand the nuances of Filipino morphology and code-switching, (3) foster a culture of collaboration in further advancing the state of NLP in the Philippines, and (4) personally engage with the output of generative AI tools to get a better understanding of their strengths and flaws.

Each group must **independently** submit their annotations to the specified repository on or before **an agreed-upon deadline by the class**. Penalties will be incurred by the group if (a) they fail to submit the validated dataset on time, or (b) other groups consistently flag a particular group's contributions as inaccurate. Please understand that the validation is a **collaborative effort** and must be taken seriously, as poor contributions will affect the quality of the entire class's models, and future research on Filipino NLP. The validated datasets from all groups will serve as the final dataset for this course's major course output.

2.2.1 Criteria for Annotation

In validating the annotated dataset, please **read the following guidelines carefully and ensure compliance**.

Each word must be tagged with the following labels, whenever appropriate.

Tag	Definition
Fil	The word is purely Filipino in origin and form, including those with native affixes (e.g., "bawal"). This also includes loanwords that have already been assimilated into Filipino, and already have Filipino spellings.
Eng	The word is an English word and uses an English spelling, even if used in a Filipino context (e.g., "corrupt")
CS	The word is an intra-code switched word, combining both Filipino and English elements in the same word (e.g., "naglunch", "pina-explain").
NE	The word is a named entity. That is, proper names, such as people, places, organizations, or brands (e.g., "Manila", "Elon Musk"). An NE can also be tagged as English (e.g., the "park" in Rizal Park) or Filipino (e.g., "Komisyon ng Wikang Pilipino"). Proper names, such as names of people and places, do **NOT** have a language regardless of their perceived origin.
Num	Numerical values (e.g., 2023, 2.45)
Sym	Punctuation marks, emojis, and other non-alphanumeric synbols.
Abb	The word is an abbreviation, or shortened forms of words or phrases (e.g., "Dr", "PhD", "Brgy", "DPWH"). Abbreviations must also be tagged as either English, Filipino when it is clear ("Brgy" is Filipino, "DPWH" is English)
Expr	Onomatopoeic words or expressions representing sounds (e.g., "haha", "grr", "ah")
Unk	Any word that doesn't fit the above categories or is ambiguous or unrecognizable.

2.2.2 Example Annotations

23	na	flood	control	projects	ang	na-award	sa	St	.	Gerrard	Construction	.
Num	Fil	Eng	Eng	Eng	Fil	CS	Fil	Eng-Abb-NE	Sym-NE	NE	Eng-NE	Sym

Ang	Chinito	Walkers	ay	mga	sikat	na	content	creators	mula	sa	DLSU	.
Fil	Fil-NE	Eng-NE	Fil	Fil	Fil	Fil	Eng	Eng	Fil	Fil	Eng-Abb-NE	Sym

Ginanap	ang	13	trillion	peso	march	sa	EDSA	monument	at	Luneta	Park	.
Fil	Fil	Num	Eng	Eng	Eng	Fil	NE-Abb	Eng-NE	Fil	NE	Eng-NE	Sym

"	You	think	hindi	ako	napipikon	?	Pikon	naman	din	ako	ah	"	,	ani	Heart	.
Sym	Eng	Eng	Fil	Fil	Fil	Sym	Fil	Fil	Fil	Fil	Expr	Sym	Sym	Fil	NE	Sym

2.3 PinoyBot Implementation

You are provided a skeleton Python file named `pinoybot.py`. This contains a function called `tag_language`, which you must complete for this project. The function accepts a single Filipino passage, and is expected to automatically tag each token in the language with the appropriate tag. For simplicity, we will only have three classes for this project: `ENG` for English, `FIL` for Filipino, and `OTH` for Other. The following rules should be observed:

- Symbols, numbers, onomatopoeic words / expressions, unknown, and abbreviations (regardless of their language), will all be treated as **Others** (`OTH`) class.
- Named entities will be treated based on their language, if it exists. For example, in $\underset{\text{NE}}{\text{Rizal}}$ $\underset{\text{Eng-NE}}{\text{Park}}$, "Rizal" will be treated as **Others** (`OTH`), but "Park" will treated as **English** (`ENG`).
- Intra-word code switched words, such as "nag-lunch" and "pinush", will be treated as **Filipino** (`FIL`).

The parameter for the function is a list of strings, which represent the tokens. For example, the sentence $\underset{\text{Eng}}{\text{Love}}$ $\underset{\text{Fil}}{\text{Kita}}$ $\underset{\text{Sym}}{\text{.}}$ will be passed to the function as `['Love', 'kita', '.']`, containing 3 tokens. The function is expected to return another list, containing the predicted tags for each token. In this case, if the predictions are correct, the function is expected to return `['ENG', 'FIL', 'OTH']`.

2.3.1 Model Training and Evaluation

To predict the language of each word, you are expected to train a supervised machine learning model (e.g., decision tree, Naïve Bayes) that takes a list a features extracted from a word, and then outputs a language tag for that word (either `FIL`, `ENG`, or `OTH`). To train the model, you can use the `scikit-learn` library, a popular machine learning library in Python, which can handle the training and evaluation process for you.

It is recommended that you go through a few tutorials showing how to use the sklearn library to gain a good understanding of how to format the input data, instantiate and train the model, make predictions, and interpret the results:

- Decision Tree reference for sklearn
- Naïve Bayes reference for sklearn

You must identify a set of relevant features (i.e., feature engineering) for identifying the language of a given word. Some ideas you may want to consider are:

- presence of letters (number of a's, ratio of vowels, etc.)
- capitalization (is first letter capital, ratio of capital letters, etc.)
- arrangement of letters (character-level n-grams, etc.)
- properties of words surrounding it in the sentence (previous word is predicted as English, previous word ends in a vowel, etc.)

To train the model, you must convert the annotated text samples into a feature matrix and label list, such as the one shown below:

Word (not included as feature)	Feature 1	Feature 2	Feature 3	...	**Label**
Word 1	value	value	value	...	ENG
Word 2	value	value	value	...	FIL
Word 3	value	value	value	...	OTH
...	...	...	...	...	...

This will serve as the training data for your classifier model/s.

You must observe a 70-15-15 train-validation-test split in training and evaluating the model. It is up to you implement this split appropriately. You are expected to test your classifier model and report its performance on the set.

2.3.2 General Flow for PinoyBot

You must train the test the classifier model separately from the tag_language function. Once trained, you can save the model so that it can be called in the tag_language model to make predictions. The general flow for the tag_language function is as follows:

1. Input: list of strings, representing the tokens in the passage
2. Convert the list of tokens into the feature matrix by extracting the relevant features
3. Call the trained model to predict the tag for each token, assign the results to a new list
4. Return the list of tags

Please make sure that your function is returned in the correct format. The length of the list returned should be equal to the number of tokens in the input, and each tag can only be either ENG, FIL, or OTH. Failure to comply with this may cause problems with the automated checking process.

Please observe the following restrictions:

- You are not allowed to use English or Tagalog dictionaries to directly infer the language of a word (e.g., searching a list of words in a Filipino dictionary and if a specific word appears there, then it must be a Filipino word). While this approach can be helpful in many cases (but not all), we want you to focus your efforts on training the model based on the features of the word itself, rather than using existing repositories.
- You are not allowed to use any other existing language identification models, as this would defeat the purpose of training your own.

3 Report

In addition to the PinoyBot codes, you are required to write a report documenting the implementation of the project. The report should contain the following:

1. **Classifier Model**: A section describing what classifier model was used, the hyperparameters, and the list of features used. Explain briefly what influenced the final list of features and the final configuration of the model.
2. **Evaluation and Performance**: A section containing an empirical evaluation of the classifier model on the test set, using relevant metrics for classification such as accuracy, precision, recall, and F1-score. It should also mention how the test set was split from the training data. Finally, a concise analysis of the results, including the strengths and weakness of the classifier should be included.
3. **Challenges**: A discussion on the difficulties encountered in developing the bot. What are the challenges that make it difficult for computers to automatically perform language identification? What are some future directions we can explore to address these challenges?
4. **Table of Contributions:** A table showing the contributions of each member to the project.

The report should **not** exceed four pages with A-4 sized paper. If there is a strong need to have more pages, the group must ask permission from the instructor and justify it. There is no restriction on the formatting, as long as it is readable. Please be concise in writing the report and avoid repeating already-known definitinons. The bulk of the report should contain the group's personal ideas, insights, and implementations. Minimize the reiteration of already-known definitions. **You are not allowed to use generative AI for the report, even for writing improvements**. The report should be an accurate reflection of the group's personal understanding and engagement with the project.

4 Deliverables

There are two deliverables for this project: a zip file containing the source files for the project, and a pdf file containing the report. The zip file should contain the following: (1) all the source files needed to run the bot (including `pinoybot.py` which contains the `tag_language` function, as well as the trained model file), (2) the codes used for training the model (this can be placed in a separate folder). There is no need to include the dataset that was used to train the model, but **please be ready to present it upon the request of the instructor**.

5 Academic Honesty

Honesty policy applies. Please take note that you are **NOT** allowed to borrow and/or copy-and-paste in full or in part any existing related program code from the internet or other sources (such as printed materials like books, or source codes by other people that are not online). **Violating this policy is a serious offense in De La Salle University and will result in a grade of 0.0 for the whole course**.

Please remember that the point of this project is for all members to learn something and increase their appreciation of the concepts covered in class. Each member is expected to be able to explain the different aspects of their submitted work, whether part of their contributions or not. **Failure to do this will be interpreted as a failure of the learning goals, and will result in a grade of 0 for that member**.

6 Data Usage

Please note that your contributions in validating the annotations of the datasets can be used in future research to advance the field of NLP in the Philippines.