# Energy Domain Classification Task Using FastText

## Objective

This project aims to construct a high-quality energy-related question dataset (benchmark queries) by filtering the large-scale ClueWeb22 corpus. As ClueWeb22 contains diverse and unlabeled content, we need to train a binary text classifier that can distinguish between energy-related and non-energy documents.

You will use FastText — a lightweight, efficient text classification tool — to build this classifier. The goal is to label documents as either:

- `\_\_label\_\_energy` (positive)

- `\_\_label\_\_nonenergy` (negative)

Once trained, the classifier will be used to extract energy-related documents from ClueWeb22.

## Workflow Overview

1. Collect labeled training data (positive = energy, negative = non-energy)

2. Preprocess and clean all documents

3. Format data into FastText-compatible input

4. Train and validate a FastText classifier

5. Apply the classifier to ClueWeb22 samples

6. Save filtered energy-related documents for later use

## Task Breakdown

Step-by-step tasks to be followed by the student:

## Step 1: Collect Training Data

\*\*Positive Samples\*\*

- Goal: Collect ~1000 documents in the energy domain.

- Source suggestions:

- arXiv.org (use keywords like "renewable energy", "electric grid", "natural gas", etc.)

- OpenAlex

- University library databases (e.g., IEEE, Elsevier, Springer)

- Acceptable formats: PDF, HTML, or plain text.

\*\*Negative Samples\*\*

- Randomly sample 1000–2000 documents from ClueWeb22 (non-energy content).

- Ensure no obvious energy keywords are present (e.g., "energy", "solar", "power plant", "carbon").

## Step 2: Preprocess Text

- Extract plain text from PDFs/HTML using tools like `pdfminer`, `beautifulsoup`, or similar.

- Normalize all text:

- Lowercase everything

- Remove HTML tags, special characters, headers, footers

- Optionally remove stopwords

## Step 3: Format for FastText

Create a `.txt` file where each line follows this structure:

```

\_\_label\_\_energy this is a document about wind energy production...

\_\_label\_\_nonenergy this is a blog post about cooking recipes...

```

- Use `\t` (tab) or a single space between label and text

- Save as `train.txt` and `valid.txt` (80/20 split recommended)

## Step 4: Train FastText Classifier

Use the official FastText library: https://github.com/facebookresearch/fastText

\*\*Training command:\*\*

```bash

./fasttext supervised -input train.txt -output model\_energy -epoch 10 -lr 0.5 -wordNgrams 2 -dim 100

```

- `-epoch 10`: number of training passes

- `-lr 0.5`: learning rate

- `-wordNgrams 2`: include bigram features

- `-dim 100`: vector dimensions

\*\*Evaluate on the validation set:\*\*

```bash

./fasttext test model\_energy.bin valid.txt

```

## Step 5: Classify ClueWeb22 Documents

1. Apply the trained model to ClueWeb22 documents.

2. For each document:

- Extract clean plain text

- Use the model to predict label:

```bash

./fasttext predict model\_energy.bin document.txt

```

3. If prediction is `\_\_label\_\_energy`, save the document to the final dataset.

## Step 6: Output

- A folder containing all energy-classified documents

- A summary file:

- Total processed

- Number predicted as energy

- Sample outputs with prediction scores (optional)

## Optional Evaluation (Advanced)

- Calculate precision, recall, F1-score on test data

- Visualize keyword distributions or word embeddings using t-SNE

- Try paragraph-level splitting if document-level classification is noisy