

KENYAN CONSTITUTIONAL AGENT

LEVERAGING AI FOR CIVIL EDUCATION

PRESENTED BY:

**MATHEWS ODONGO
PACIFICAH ASAMBA
FRANKLINE ONDIEKI
NIGHTINGALE JEPTOO
TINAH MWIKALI NGEI
HELLEN DIANA MACHARIA**

THE
Constitution
OF
Kenya
2010

Laws of Kenya

ALG

PROJECT OVERVIEW

Goal:

Create an AI-powered agent to make the Kenya Constitution (2010) easy to access and understand.

Approach:

- Convert constitutional text into structured, machine-readable data.
- Use AI and semantic search to answer natural language queries.
- Provide accurate references to specific chapters and articles.

Outcome:

A user-friendly tool that promotes civic literacy and empowers citizens with accessible legal knowledge.

BUSINESS UNDERSTANDING

- **Context:** Constitution is long, complex, and legalistic
→ hard for citizens to navigate.
- **Problem:** Traditional consultation is costly & time-consuming.
- **Objective:** AI agent that supports English & Swahili, gives accurate, clear answers, and cites articles.
- **Stakeholders:** Citizens (youth focus), NGOs, government institutions.
- **Impact:** Empowerment, accessibility, and scalability.

DATASET OVERVIEW

- **Source:** Kenya Constitution (2010).
- **Structure:** 26 chapters, 260 articles.
- **Format:** Preprocessed into JSON/CSV with metadata.
- **Example fields:** chapter, article, content, metadata.

DATA PREPARATION

- **Cleaning:** Removed formatting & footnotes.
- **Segmentation:** Split long articles into chunks.
- **Metadata:** Added chapter/article references.
- **Embeddings:** Generated semantic vectors for each article.

EXPLORATORY DATA ANALYSIS

In this step, we aim to understand the structure and content of the cleaned Kenyan Constitution dataset.

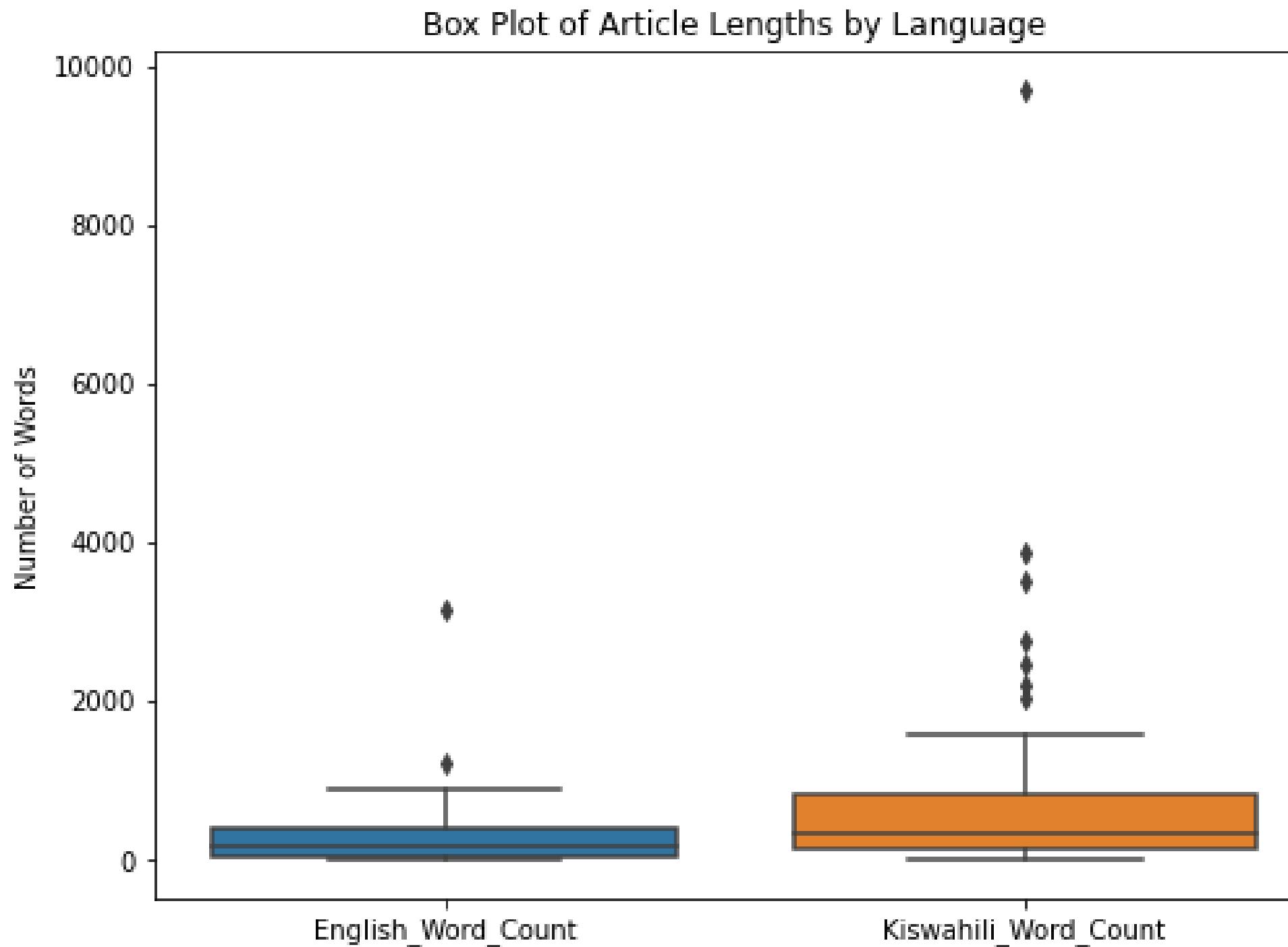
We will explore both English and Kiswahili text to identify:

- Number of articles/sections
- Distribution of text length per article
- Common words and phrases
- Coverage of topics across the Constitution

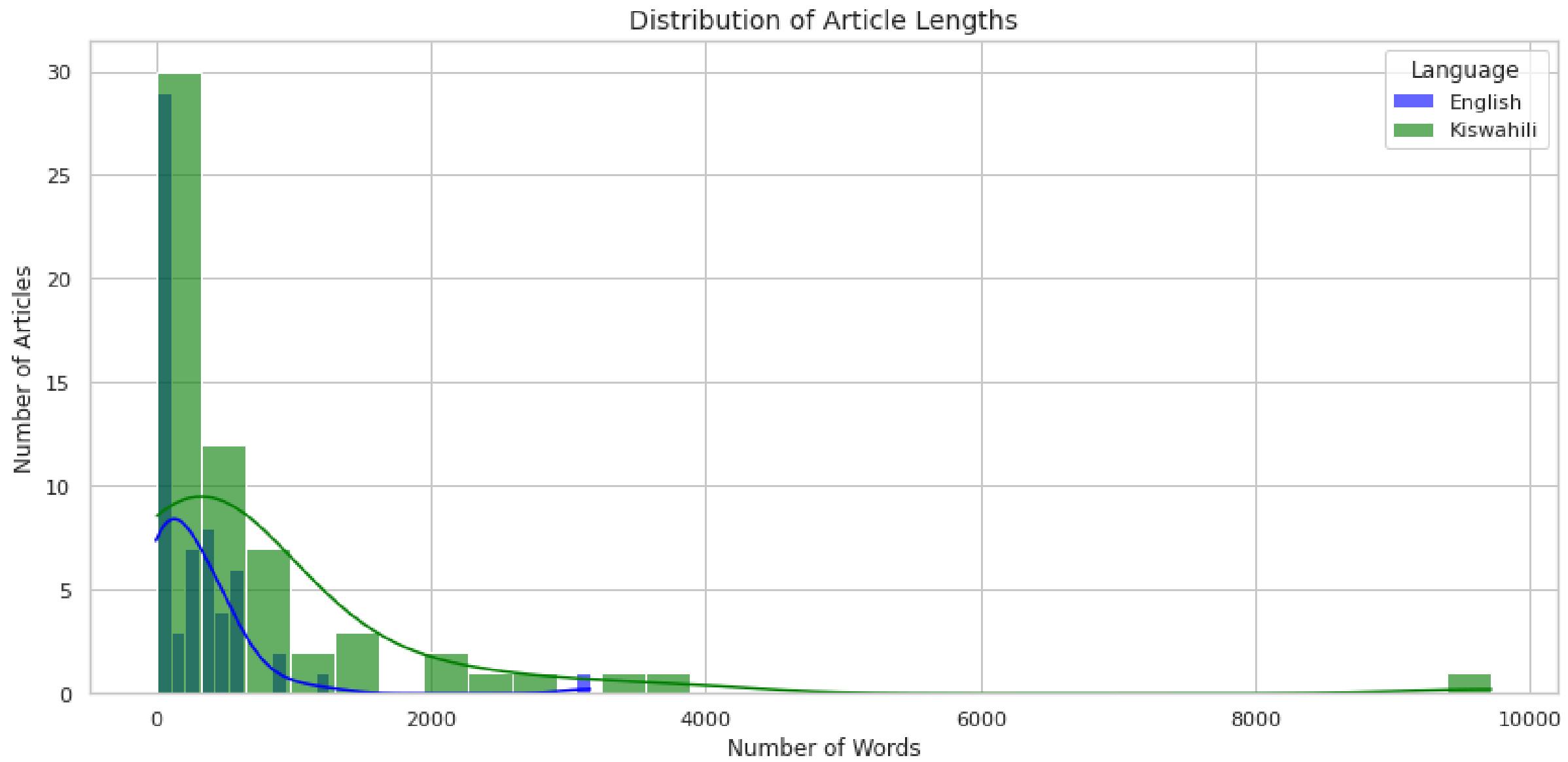
This helps us identify potential preprocessing needs and informs the feature engineering and NLP pipeline.

Length categories

- Kiswahili articles show greater variation and tend to be longer.
- Both languages contain a few very long outliers, with Kiswahili reaching nearly 10,000 words.



Length category by word count



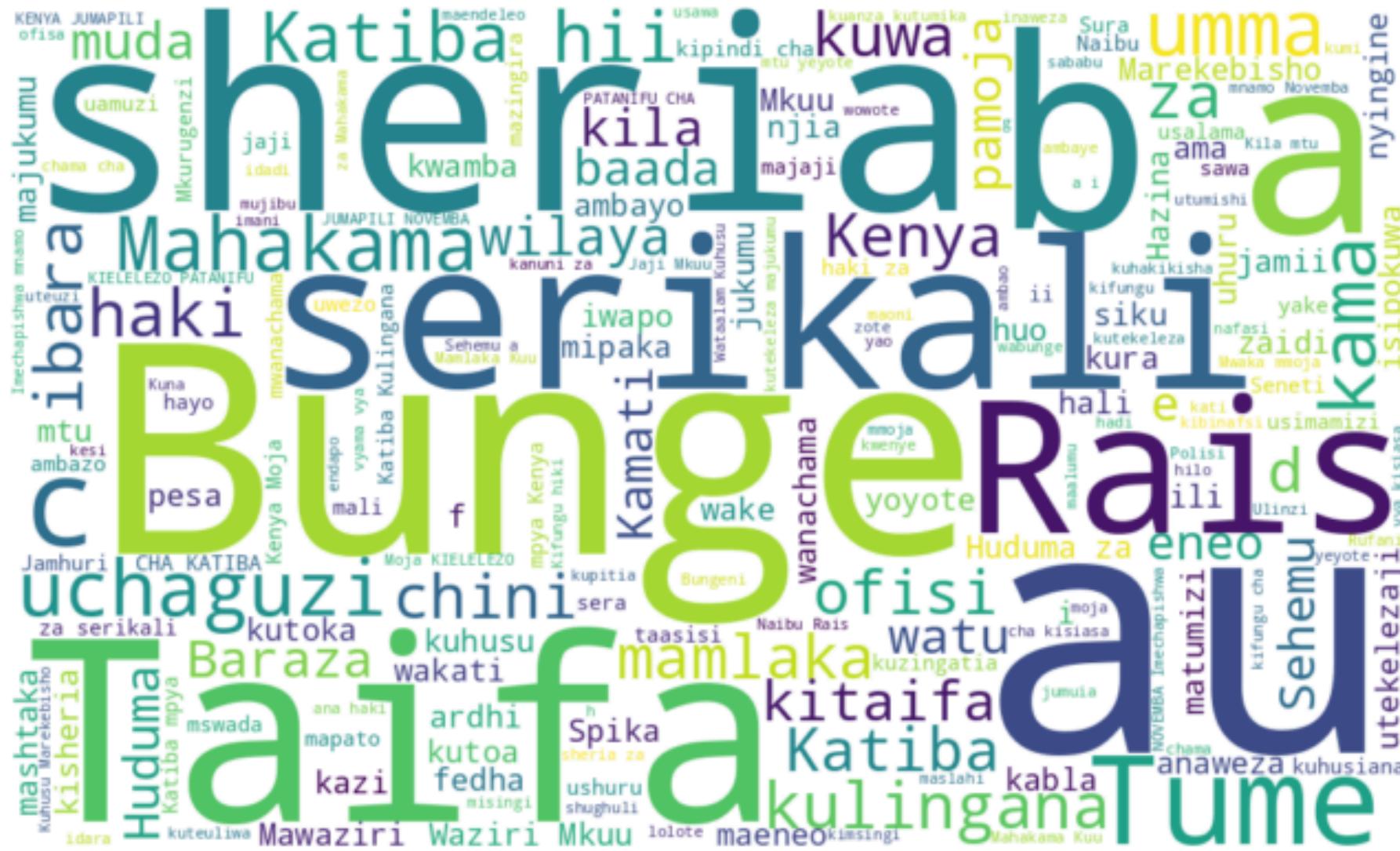
- Most articles in both English and Kiswahili are short (under 500 words).
- A few articles are much longer, creating a right-skewed distribution.
- Kiswahili articles show greater variability and some extreme outliers nearing 10,000 words.
- English articles are generally more concise and consistent in length.



Word cloud of most popular words (English)

- The distribution is right-skewed with a few very long articles.
 - English articles are generally shorter and more consistent.

Word Cloud - Kiswahili Text



Word cloud of most popular words (Swahili)

- Kiswahili articles show more variation and longer extremes.

SYSTEM ARCHITECTURE

steps:

- Input user query.
- Convert to embedding.
- Retrieve relevant article(s) from vector DB.
- AI generates context-aware answer + citation.

Tools: Python, LangChain, OpenAI embeddings, FAISS, Streamlit.

AI AGENT WORKFLOW

Example Query: “What are my rights under Article 43?”



QUERY
RETRIEVAL

EMBEDDING-
ARTICLE



RETRIEVED CONTENT: ARTICLE 43



ANSWER GENERATED WITH SUMMARY
AND REFERENCE

FINDINGS

Model Training & Performance

- Deep learning model learned patterns effectively.
- Training & validation accuracy improved steadily.
- Dropout reduced overfitting → stable results.
- High accuracy on test set → NN complements ML.

Comparison with Previous Models

- Naive Bayes & AdaBoost achieved near-perfect results.
- Deep learning comparable but more resource-heavy.
- Neural networks scale better for complex/unstructured text.

Interpretability

- Traditional ML = more interpretable.
- Deep learning = better at capturing non-linear relationships.

RECOMMENDATIONS

- Use embeddings (Word2Vec/BERT) for scalability.
- Retrain with amendments & feedback.
- Deploy via FastAPI/Flask, expose as service.

NEXT STEPS

- **API Development**
 - Build with FastAPI (/predict, /answer).
 - Lightweight, scalable, easy to integrate.
- **User Interface**
 - Create a Streamlit web app for interactive Q&A.
 - Show query results: ML classification + DL contextual response + Article reference.
- **Deployment**
 - Host on Render / Heroku / AWS EC2.
 - Combine FastAPI backend + Streamlit frontend.
 - Make the AI agent publicly accessible online.

QUESTIONS?

“Your Constitution, simplified through AI.”

Thank you

EMPOWERING KENYANS THROUGH AI DRIVEN CIVIC EDUCATION