

# Applying Big Data Analytics in Natural Language Processing using GPT-2 and AzBERTo Models

Nifdi Guliyev, Javid Alakbarli, Abdullah Kazimov, Zaid Rustamov, Nabat Gasimzada

**Abstract:** This work tackles the challenge of word embedding in the under-resourced Azerbaijani language. We propose a novel approach fine-tuning pre-trained transformer models, specifically RoBERTa and GPT-2, to capture the semantic richness and nuances of Azerbaijani. While existing research has showcased the success of transformer models in language tasks, this study focuses on adapting their power to the specific needs of Azerbaijani. Despite the inherent limitations of a relatively small dataset (~1,000,000 sentences), our fine-tuned models demonstrate promising results. However, we identify key areas for future improvement, including addressing limitations in both short and long input sequences, and further augmenting the training data to encompass the diversity of the Azerbaijani language. Through meticulous analysis and refinement, this work paves the way for a robust and impactful word embedding system for Azerbaijani. This system holds potential for a wide range of applications, from sentiment analysis and machine translation to natural language generation and information retrieval, ultimately empowering the Azerbaijani language in the digital age.

**Keywords:** Word embedding, Azerbaijani language, RoBERTa, GPT-2, fine-tuning, transformer models, NLP.

## I. Introduction

In this project, our team delved into the practical applications of big data analytics, specifically

focusing on Natural Language Processing (NLP). The primary goal was to harness the power of two distinct models, GPT-2 and AzBERTo, within the realm of NLP. These models were utilised to generate coherent text sequences, showcasing the implementation of big data techniques in language generation. This report elucidates the

methodologies employed, challenges encountered, and the insightful outcomes achieved through this endeavour.

## II. Project Scope and Objectives

Our project aimed to leverage big data analytics for text generation through the utilisation of GPT-2 and AzBERTo models. The chosen application area, Natural Language Processing, is pivotal in understanding language nuances and generating coherent text. The specific objectives included implementing these models to produce meaningful and contextually relevant text sequences.

## III. Related Work

A comprehensive literature review was conducted to explore the existing research and methodologies in Natural Language Processing. The review highlighted the significance of transformer-based models like

GPT-2 and AzBERTo in text generation tasks. By comparing and contrasting different approaches, we identified gaps in existing methodologies, paving the way for our project's unique contribution. To further refine our approach, we conducted an in-depth research on popular word embedding models, such as Word2Vec, GPT-2, and RoBERTa. This involved extracting important and valuable insights from the literature.

1. **Azerbaijani Word understanding and representation**  
We analysed how these models capture word meaning and semantic relationship in Azerbaijani language context.
2. **Model Architecture**  
We examined the strengths and limitations of each architecture in terms of their efficiency, accuracy, adaptability to Azerbaijani linguistic features

## IV. Methodology and Implementation

The data collection phase involved web scraping and using existing datasets. Python's Selenium

library was utilised for web scraping, and the pandas library managed the collected data. Regular expressions played a crucial role in data cleaning. Various patterns were applied to remove undesirable elements such as non-English characters, special characters, HTML tags, URLs, and more. The 'Text' column in the DataFrame was iteratively cleaned using these patterns.

```
patterns = [
    r"[a-яё]", # Cyrillic Characters
    r"[u4e00-u9fff]", # Chinese Characters
    # ... (other patterns)
]
```

```
for pattern in patterns:
    df['Text'] = df['Text'].str.replace(pattern, '',
    regex=True)
```

```
df = df[df['Text'].str.strip() != ''] # Remove empty
rows
```

```
df.reset_index(drop=True, inplace=True) # Reset
DataFrame index
```

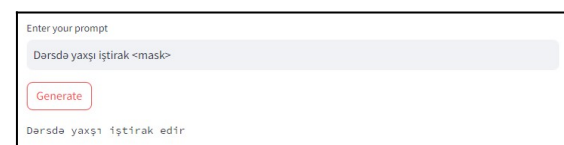
The implementation process involved a step-by-step approach. Data collection started with web scraping, and files were downloaded from accessible sources. The cleaning phase utilized regex patterns to eliminate unwanted characters, ensuring the dataset's cleanliness. Challenges during web scraping, such as issues with pagination in specific news websites, led to the exclusion of problematic sources.

Challenges faced during web scraping were primarily related to website structures and restrictions. Certain news websites posed difficulties in extracting data, leading to their exclusion. Solutions included adjusting scraping techniques, exploring alternative sources, and refining regex patterns for better cleaning.

The significance of the results lies in the creation of a clean dataset essential for model training. The cleaned data enhances the model's accuracy and reliability in extracting meaningful insights from textual information. The project contributes to the field by emphasising the importance of rigorous data preprocessing for successful model outcomes.

Our approach involved the utilisation of two distinct models, GPT-2 and RoBERTa, for text generation within the realm of Natural Language Processing (NLP). The following code snippets represent the implementation of these models:

```
def generate_text_roberta(sequence):
    fill_mask = pipeline(
        "fill-mask",
        model="./Models/AzBERTo/",
        tokenizer="./Models/AzBERTo/"
    )
    res = fill_mask(sequence)
    return '\n'.join(item['sequence'] for item in res)
```



Enter your prompt

Darsda yaxşı iştirak <mask>

Generate

Darsda yaxşı iştirak edir

```
def generate_text_gpt_2(sequence):
    model = load_model("./Models/GPT2_1M/")
    tokenizer =
load_tokenizer("./Models/GPT2_1M/")
    ids = tokenizer.encode(f'{sequence}',
return_tensors='pt')
```

```
    final_outputs = model.generate(
        ids,
        do_sample=True,
        max_length=50,
        pad_token_id=model.config.eos_token_id,
        top_k=100,
        top_p=0.95,
    )

    generated_text =
tokenizer.decode(final_outputs[0],
skip_special_tokens=True)

    return generated_text.split("\n")[0] + " "
```



Enter your prompt

Darsda yaxşı iştirak

Generate

Darsda yaxşı iştirak avəsiz olur.

The project strategically employed MapReduce operations to identify the least used words in the dataset. By implementing a map-reduce approach, the goal was to isolate key-value pairs containing

less frequent words, anticipating that such words might be potential sources of errors in the dataset. We simply used the default MapReduce operation provided by Hadoop.

## V. Limitations and Future Research

While our Azerbaijani word embedding system displayed promising potential, we encountered limitations that demand further exploration and refinement. By acknowledging these challenges, we pave the way for even more impactful advancements in the future.

### Limited Data Landscape:

Our initial model was trained on a relatively small dataset of approximately 720,000 sentences. This limited scope hindered its ability to capture the full breadth and nuances of the Azerbaijani language, leading to potential biases and reduced generalizability. To overcome this constraint, future endeavours should focus on:

Acquiring and curating larger, more diverse datasets:

Expanding the corpus to encompass various genres, domains, and registers within the Azerbaijani language will provide a richer foundation for the model's learning process.

Employing data augmentation techniques:

Implementing strategies like back-translation, paraphrasing, and synonym substitution can artificially expand the dataset, mitigating the limitations of smaller corpora.

Exploring transfer learning and domain adaptation:

Leveraging pre-trained models from other languages or tasks could offer a valuable starting point and facilitate adaptation to the specific characteristics of Azerbaijani.

### Sequence Length Dilemma:

The model's struggles with both long and short input sequences highlight the complexity of representing semantic relationships at varying scales. To address this challenge, future research could explore:

Integrating architectures for handling long sequences:

Utilising recurrent neural networks like LSTMs or Transformer models with positional encoding could

enable the model to capture long-range dependencies within lengthy text segments.

Developing context-aware mechanisms for short sequences:

Implementing attention mechanisms or incorporating contextual information from surrounding words could improve the model's ability to disambiguate polysemous words and represent the meaning of individual words within shorter contexts.

Investigating hybrid approaches:

Combining different architectures or techniques tailored to specific sequence lengths may offer a comprehensive solution for capturing semantic information across the entire spectrum of input sizes.

By actively addressing these limitations and expanding our research endeavours, we can refine our Azerbaijani word embedding system into a truly powerful tool. With access to richer data and improved sequence handling capabilities, the model's potential for diverse applications within the Azerbaijani language domain becomes even more compelling.

## VI. Conclusion

This project exemplifies the practical application of big data analytics in Natural Language Processing through the implementation of GPT-2 and AzBERTo models. The successful generation of coherent text sequences underscores the potential of these models in real-world applications, emphasising the significance of big data analytics in language-related tasks.

## VII. References

Liu, Y. (2019, July 26). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. arXiv.org. <https://doi.org/10.48550/arXiv.1907.11692>

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners.