A REPORT

ON

AIR POLLUTION CORRELATION WITH URBANISATION

BY

Rishi Bharadwaj 2022A7PS2001H

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**

(September,2024)

Acknowledgements

I would like to express my gratitude to Prof. Manik Gupta for giving me the opportunity to contribute to this project and for her mentorship and guidance.

I would also like to thank Mr. Ravi Bhushan for his assistance and weekly follow-ups which gave me clear ideas on how to proceed with this project.

**Table Of Contents**

1. **Cover** ................................................................................. 1
2. **Acknowledgments** ........................................................................... 2
3. **Table of Contents** .......................................................... 5
4. **Abstract** ..................................................................... 6
5. **Main Text** .......................................................................... 7
6. 7.1 Intent Detection Using Sentence Similarity................... 7
7. 7.2 Transformer Architecture Used in Embedding Models……………….8
8. 7.3 Choosing an Appropriate Embedding Model ............................................... 9
9. 7.4 Implementation of Intent Detection Model .................................................. 10
10. 7.5 Results after Deployment .................................................... 11
11. **Conclusions**................................................................ 12
12. **References** ..................................................................... 13
13. **Glossary** ……………………………………………...14

# Abstract

Air pollution related studies have become a vital part of climate science due to pollutions impact on the environment and on human well-being. The effect of air pollution on human health has been well documented and researched. Various pollutants like Nitrogen Dioxide (NO2), Carbon Monoxide (CO), Carbon Dioxide (CO2), Sulphur dioxide (SO2) etc. when inhaled can cause severe damage to the lungs and other organs over time. They can also cause breathing issues like asthma and contribute to allergies.

On a larger scale these pollutants also contribute to global warming and phenomena such as acid rain. As climate change is becoming a larger and larger issue, tracking what factors contribute to air pollution has become more and more vital. One of the largest factors contributing to climate change is human activity. As our population has increased and our cities have rapidly developed, our use of fossil fuels for energy has also skyrocketed. For this project, the region of interest is Hyderabad, a growing metropolitan city in India. As it has grown and expanded over the previous few decades, air pollution has become a significant issue, caused by several factors including increasing population density, industrial activity, and vehicular traffic.

This project aims to prove a direct correlation between air pollution and urbanisation. The region of study is Hyderabad, a growing metropolitan city in India. The region includes urban, peri-urban and rural areas. The study uses data from Sentinel 5-P satellite’s TROPOMI sensors for air quality. For data on urbanisation data Normalized Difference Vegetation Index (NDVI) and Land Surface Temperature (LST) data from the MODIS (Moderate Resolution Imaging Spectroradiometer) sensor were used as well as nighttime lights (NTL) data obtained from the Visible Infrared Imaging Radiometer Suite (VIIRS). Using these datasets, it is possible to show a direct correlation between air pollution and increasing urbanisation.

# Literature Survey

Remote sensing and machine learning using air pollution data has become quite widespread. Many studies have been done using machine learning models to predict future levels of pollutants using different methods. These include classical methods such as Kth Nearest Neighbours(KNN) and Support Vector Regression1 , as well as deep learning methods such as Artificial Neural Networks2 .

Studies have also been done on the relationships between air pollution and increasing urbanisation in growing cities3.

# Citations

[1] S. Simu *et al.*, "Air Pollution Prediction using Machine Learning," *2020 IEEE Bombay Section Signature Conference (IBSSC)*, Mumbai, India, 2020, pp. 231-236. doi: 10.1109/IBSSC51096.2020.9332184.

[1] H. Maleki, A. Sorooshian, G. Goudarzi, *et al.*, "Air pollution prediction by using an artificial neural network model," *Clean Techn. Environ. Policy*, vol. 21, pp. 1341–1352, 2019. doi: <https://doi.org/10.1007/s10098-019-01709-w>

[2] ] M. Khirwar and A. Narang, “GeoFormer: A vision and sequence transformer-based approach for greenhouse gas monitoring,” \*arXiv\*, vol. abs/2402.07164, 2024. [Online]. Available: <https://arxiv.org/abs/2402.07164>.

[3] [2] S. Wang, S. Gao, S. Li, and K. Feng, "Strategizing the relation between urbanization and air pollution: Empirical evidence from global countries," \*Journal of Cleaner Production\*, vol. 243, p. 118615, 2020. [Online]. Available: https://doi.org/10.1016/j.jclepro.2019.118615. [Accessed: Sep. 23, 2024].

# Transformer Architecture used in Embedding Models

The transformer architecture, introduced in the paper "Attention is All You Need" by Vaswani et al. in 2017, has revolutionized the field of Natural Language Processing (NLP). It has become the foundation for many state-of-the-art NLP models, including BERT, GPT, and their variants. The architecture is based on self-attention mechanisms and is designed to handle long-range dependencies in text more effectively than previous models like RNNs (Recurrent Neural Networks) and LSTMs (Long Short-Term Memory networks).

At its core, the transformer model consists of an encoder-decoder structure, where both the encoder and decoder are made up of stacked layers of self-attention and feed-forward neural networks. Each layer in the encoder processes the input sequence by weighing the importance of each word relative to others, allowing the model to focus on different parts of the input for each word.

The encoder processes the input sequence and outputs a set of continuous representations. The decoder generates the output sequence based on the encoder’s representations and the already generated tokens in the target sequence. This self-attention mechanism helps capture long-range dependencies and contextual information, which is crucial for understanding complex language structures.

# Choosing an Appropriate Embedding Model

To carry out the task of semantic textual similarity we need an embedding model. I was instructed to pick an open-source model from the Hugging Face leaderboard, a popular collection of ML models. The factors to consider for choosing a model are the number of parameters the model has and the accuracy of the model. The larger the number of parameters, the more resources the model requires to run and hence the more expensive it will be.

I picked an open-source model called *bge-small-en-v1.5* which had a very good parameter to performance ratio. It is based on the transformer architecture explained above. It used 33 million parameters but performed on par too other models with more than 100 million.

The model also had an MIT license ensuring we were free to use the model for commercial purposes.

This model was first picked for my task on phrase semantic matching, but was eventually deployed on a wide variety of NLP tasks including agent detection and intent detection.

# Implementation of Intent Detection Model

I developed an intent detection bot by leveraging an embedding model to transform sentences into vector embeddings. The core idea was to utilize cosine similarity to measure the semantic closeness between incoming test sentences and a set of example phrases, which had been pre-sorted into predefined categories. This method allowed for a nuanced understanding of semantic similarities, which is crucial for accurate intent detection in various Natural Language Processing (NLP) applications.

For the embedding model I utilized *bge-small-en-v1.5* from my previous project. Once the model was chosen, I focused on converting each sentence, whether a user query or an example phrase, into a high-dimensional vector representation. These vectors were then used to calculate the cosine similarity between the test sentences and the example phrases. The cosine similarity score provided a measure of how closely related the meanings of the sentences were.

To refine the bot's performance, I conducted a series of experiments to determine an optimal threshold score for cosine similarity. This involved adjusting the threshold and evaluating the bot's accuracy in categorizing sentences. When a test sentence exceeded this threshold, it was confidently matched to the corresponding category. This approach ensured that only sentences with a high degree of similarity to the example phrases were categorized automatically.

However, not all sentences fit neatly into the predefined categories based solely on cosine similarity. For those test sentences that did not meet the threshold, they were sent to an already implemented fallback mechanism using a GPT wrapper. This mechanism allowed the bot to utilize advanced NLP capabilities to analyse the sentence in more depth and assign it to the most appropriate category. This step was crucial for handling more complex or ambiguous sentences that the initial embedding model might struggle with.

My supervisor then completed the bot's implementation by integrating it with Google Firestore. This integration provided a robust and scalable backend solution for storing and retrieving data efficiently. With Firestore, the bot could handle large volumes of data and maintain high performance, even as the number of user queries increased. The database integration also allowed for real-time updates and easy management of the example phrases and categories. Eventually, after collecting enough phrases from users, almost any example phrase should be matched to the correct category.

# Results after Deployment

After deployment, the intent detection bot performed above expectations. Its ability to accurately categorize user inputs allowed it to replace the previous model that relied solely on sending all sentences to GPT for processing. This new approach offered significant advantages in terms of speed and cost-effectiveness. One of the key benefits of this new model was its speed. By using an embedding model to convert sentences into vector embeddings and calculating cosine similarity for categorization, the bot was able to process and classify sentences almost instantaneously. The initial classification was computationally light and fast, enabling the bot to handle a high volume of user queries in real-time. This rapid response time greatly improved the user experience, as it minimized the delay in providing relevant responses or actions based on the detected intents. In contrast, the previous model, which sent all sentences directly to GPT for processing, was significantly slower. GPT models, while powerful and accurate, require substantial computational resources and time to generate responses. Each query processed by GPT involved complex language modelling and context understanding, which, although highly accurate, introduced latency. By handling most of the intent detection through the embedding model and cosine similarity threshold, the new approach reduced the number of queries needing GPT's intensive processing, thus speeding up the overall response time.

The cost-effectiveness of the new model was another critical improvement. Utilizing GPT for every query can be expensive due to the high computational power and resources required to run such advanced models. By reducing the reliance on GPT and only invoking it for sentences that did not meet the cosine similarity threshold, the new system significantly lowered operational costs. The embedding model and cosine similarity calculations are far less resource-intensive, allowing for a more economical solution without sacrificing accuracy.

# Conclusions

My PS-1 internship at Voicegain.ai provided a comprehensive experience in the field of speech analytics and machine learning. I was able to work on multiple impactful projects involving Sentence Similarity, Agent Detection, Intent Detection and Named-Entity Recognition, crucial components of modern Natural Language Processing (NLP) applications.

Throughout the internship, I had the opportunity to deeply explore intent detection and improve the existing algorithm used by Voicegain.ai. The process involved selecting a state-of-the-art model from the Hugging Face leaderboard, adapting it to meet our specific requirements, and deploying it on NVIDIA's Triton Inference Server. The chosen model for sentence similarity, bge-small-en-v1.5, delivered superior performance and ensured efficient resource utilization, making it a cost-effective solution for the company. I later used this model in other projects including agent detection and intent detection.

During this project, I learned the critical importance of speed and cost-effectiveness in deploying machine learning models through the trade-offs associated with using GPT. Initially, our intent detection system relied heavily on GPT for processing every query, which, while accurate, proved to be both time-consuming and expensive due to the high computational demands of GPT models. This experience underscored the need for a more efficient solution. By integrating a state-of-the-art embedding model to handle the majority of queries and reserving GPT for more complex cases, we significantly reduced processing times and operational costs. This hybrid approach not only maintained high accuracy but also demonstrated the value of optimizing model deployment for both speed and cost, ensuring a scalable and economically viable solution.

The supportive and collaborative work environment at Voicegain.ai, including daily standup meetings and guidance from experienced professionals like the Chief AI Officer, Mr. Kuo Zhang, played a crucial role in the successful completion of the project. The welcoming culture and the presence of fellow BITS graduates added to the overall positive experience.

In conclusion, my internship at Voicegain.ai not only allowed me to contribute meaningfully to an ongoing project but also provided me with valuable insights and skills in the field of speech analytics and machine learning. The successful deployment of the enhanced intent detection model into production was a testament to the collaborative efforts of the team, and I am grateful for the opportunity.

# References

1. Natural Language Processing:

<https://en.wikipedia.org/wiki/Natural_language_processing>

1. Attention Is All You Need: [arXiv:1706.03762](https://arxiv.org/abs/1706.03762)
2. HuggingFace Leaderboard:

<https://huggingface.co/spaces/mteb/leaderboard>

1. Bge-small-en-v1.5:

<https://huggingface.co/BAAI/bge-small-en-v1.5>

1. Cosine Similarity:

<https://en.wikipedia.org/wiki/Cosine_similarity>

1. Nvidia Triton Inference Service:

<https://developer.nvidia.com/triton-inference-server>

# Glossary

**BGE-Small-EN-V1.5:** A specific open-source embedding model chosen for its balance of performance and computational efficiency.

**Cosine Similarity**

A measure of similarity between two non-zero vectors of an inner product space. It is defined as the cosine of the angle between the vectors, providing a value between -1 and 1.

**Embedding**

A numerical representation of text that captures the semantic meaning of words or phrases. In NLP, embeddings are used to convert text data into vectors that can be processed by machine learning models.

**Google Firestore:** A scalable, real-time NoSQL database provided by Google for storing and syncing data.

**GPT (Generative Pre-trained Transformer):** A state-of-the-art language model developed by OpenAI that generates human-like text based on the input it receives.

**Hugging Face**

An AI company that provides a wide range of NLP models and tools, including a popular model repository used for various NLP tasks.

**Intent Detection:** The process of identifying the underlying purpose or intention behind a user's input in natural language.

**Long Short-Term Memory (LSTM) Networks:** A type of RNN designed to remember long-term dependencies and alleviate the vanishing gradient problem.

**Machine Learning (ML)**

A subset of AI focused on the development of algorithms that enable computers to learn from and make predictions or decisions based on data.

**MIT License:** A permissive free software license originating at the Massachusetts Institute of Technology (MIT), allowing for the use, modification, and distribution of software.

**Natural Language Processing (NLP)**

A field of AI that focuses on the interaction between computers and humans through natural language. It involves tasks such as language translation, sentiment analysis, and speech recognition.

**Parameter:** A variable in a machine learning model that is adjusted during training to minimize error in the model's predictions.

**Self-Attention Mechanism:** A component of the transformer architecture that allows the model to weigh the importance of different parts of the input sequence dynamically.

**Vector**

A numerical representation of data in machine learning. In NLP, vectors are used to represent words, phrases, or sentences in a way that captures their semantic meaning.