AQSA: AI for Startups

Mohamed Yosef [1] Sherif Masoud Abdullah Abdelsalam
Mohamed Emam Mohamed Shaban Nada Ahmed
Mariam Ahmed Hager Ali Alaa Nasr

April 2024

[1]Mohamed Yosef is the Lead Researcher for this project.

Abstract

AQSA is a comprehensive AI-driven solution for startups and small businesses. We created AQSA with a goal to reach the gap between research and real-world applications. Our project contains three components, each designed to address critical challenges in business world.The first component is a generative chatbot, capable of of answering queries related to various aspects of the startup ecosystem, including other parts of the project and general principles of project management. The second part focuses on automated data analysis, using preprocessing and machine learning techniques to extract meaningful insights from raw data–enabling startups to make better decisions. Lastly, the third component is the forecasting module, which combines historical data, market trends, and external factors like US dollar verses EG pound change rates and oil prices to generate reliable predictions for sales or interest rates in the next month or year. So, you can say that AQSA aims to provide information, enhance decision-making, and foster innovation within the startup ecosystem.

Contents

Chapter 1

Introduction

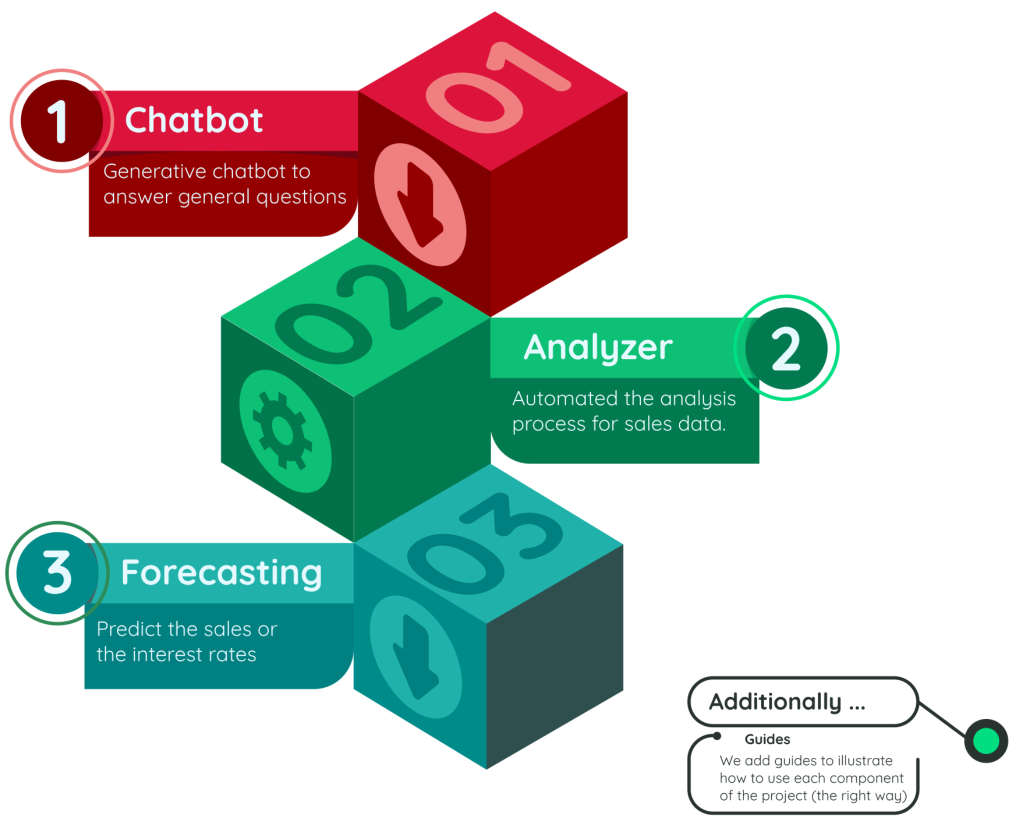
In the dynamic landscape of entrepreneurship, everything is about time and data. So, we hope to provide some helpful products to make use of both data and time. For sure this is not the first project to focus on AI as discussed in chapter 2. Instead we focus on simple, easy techniques to do the boring and time consuming tasks by brining AI to the table. This is where AQSA steps in. AQSA stands for Artificial Qualified Startup Assistance. It consists of three parts; generative chatbot, automated data analyzer, and forecasting engine.

Firstly, we'll explore the **Generative Chatbot**, your friendly and informative virtual assistant. This ever-present companion will be at your beck and call, ready to answer any query you have. Feeling unsure about the best project management approach for your specific needs? The chatbot can provide guidance on various methodologies, from Agile to Waterfall, and even suggest resources for further exploration. Encountering a common startup hurdle, like managing a remote team or keeping marketing campaigns on track? The chatbot can offer solutions and best practices gleaned from the experiences of countless startups. And of course, it can answer any questions you have about AQSA's functionalities, ensuring you get the most out of this powerful AI assistant.

Next, we'll unpack the **Automated Data Analyzer**. No longer will deciphering mountains of data be a daunting task. This powerful tool will become your data guru, automatically analyzing your startup's information to reveal hidden patterns and trends. With these insights at your fingertips, you'll be able to make informed decisions based on what the data is truly telling you. For example, the Automated Data Analyzer can identify customer demographics most receptive to your marketing campaigns, helping you refine your target audience and maximize your return on investment. Similarly, it can analyze sales data to pinpoint peak sales periods and identify any seasonal trends that might impact your inventory management. By automating these tasks and providing clear, actionable insights, the Automated Data Analyzer frees you up to focus on what matters most - growing your business.

Finally, we'll unveil the **Forecasting Engine**, your window into the future of your business. Leveraging the power of machine learning, this engine will analyze your historical data to identify trends and seasonality. It will then use this knowledge to predict potential sales and income for the coming months and year. Armed with this invaluable foresight, you can be well-prepared to navigate financial decisions with greater confidence. For instance, the Forecasting Engine can help you anticipate periods of high cash flow, allowing you to strategically invest in marketing campaigns or product development. Conversely, it can also warn you of potential dips in revenue, enabling you to proactively adjust your budget or implement cost-saving measures. In essence, the Forecasting Engine becomes your partner in financial planning, empowering you to make data-driven decisions that propel your startup forward.

Throughout this book, we'll delve deeper into each of these functionalities, providing a comprehensive guide to utilizing AQSA's full potential.



Chapter 2

AI Applications: a review

In 2016, the world watched AlphaGo (Silver and Hassabis, 2015), an AI system, defeated Lee Sedol, the world champion of Go. This wasn't just a victory for AI; it was a turning point. AlphaGo didn't rely on brute force calculations, but on a new approach called deep reinforcement learning that uses neural networks and deep learning.

Just a year later, Google researchers introduced "Attention is all you need" transformer architecture (Vaswani et al., 2017). This innovation revolutionized natural language processing (NLP). Imagine a translator that doesn't just convert words, but understands their context and relationship within a sentence. If you take a deeper look at ChatGPT (OpenAI, 2022), you will find our that the **T** in GP**T** stands for "Transformer". What I want to say is that the rise of transformers was not an ordinary thing, it was the start of the biggest large language models (LLMs) nowadays like ChatGPT, Gemini, Claude, etc.

These LLMs can churn through massive amounts of text data, learning intricate patterns and generating human-quality writing, translation, and even creative text formats. We're not just parlor tricks though. LLMs are transforming industries. Take business for instance. Machine learning has fundamentally changed how we analyze data. Traditionally, time series data (think stock prices or sales figures) was analyzed through rigid statistical models. But machine learning allows us to uncover hidden patterns and predict future trends with far greater accuracy. Imagine being able to forecast sales fluctuations or market shifts with uncanny precision – that's the power of AI.

The story has not ended yet. In fact, as I am writing this now, there are hundreds of AI startups working towards more capable models, architectures, systems, and more importantly ethics. So, it is a pleasure to be part of the future of intelligence.

2.1 Deep Learning

Deep learning drives many applications and services, including digital assistants, voice-enabled TV remotes, credit card fraud detection, self-driving cars, and generative AI. Unlike traditional machine learning, deep learning can handle unstructured data, such as text and images, without extensive preprocessing, also deep learning algorithms automate feature extraction, which reduces dependency on human expertise. As shown in Figure 2.1, the core of deep learning is a neural network that is made up of layers (input, hidden, and output) that contain nodes. Each node computes its output based on a set of weights (or parameters) applied to the output of the previous layer.

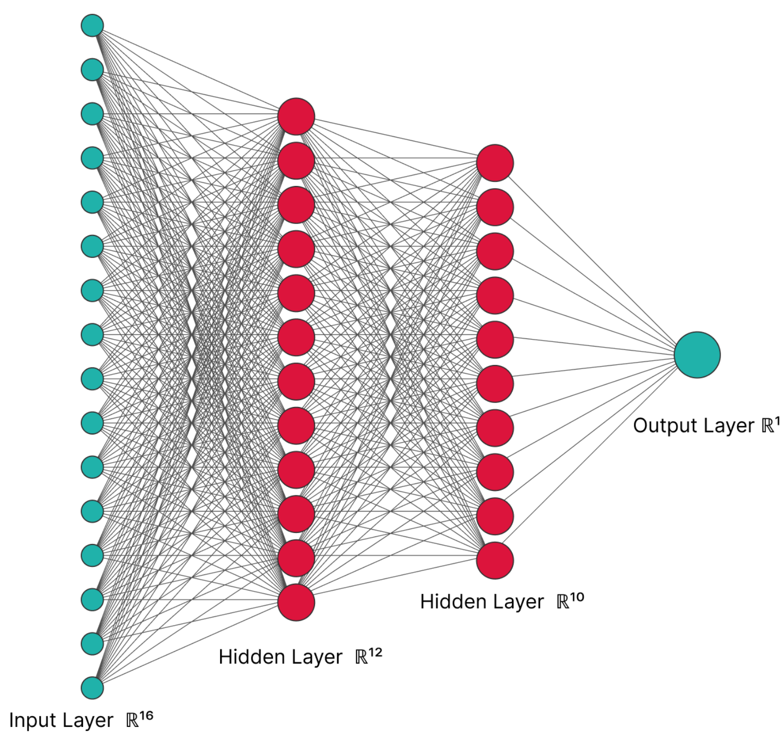


Figure 2.1: A simple neural network with two hidden layers (in red), and a one-dimensional output layer.

This is only one type of neural networks architectures which is a fully-connected network or feed-forward neural network. If you want to deal with an image classification problem like MNIST (Lecun et al., 1998) where we have only 28 by 28 pixels which means using 784 neurons in the input layer, but what if we have a bigger image say 100 by 100 pixels, this will make our hidden layer contains 10 000 neurons and if we have one hidden layer with 100 neurons, it will need one million parameters which is a huge number of parameters to train for just an image. So, to be more flexible, we can use partially connected layers like in convolutional neural networks (CNNs) (Lecun et al., 1998);(Fukushima, 1980). These networks contain learned **filters** that are applied across all parts of the input, which is typically an image. In this way, the networks can learn functions which are **translation invariant**. For example, the network can learn a filter to detect a cat, and because it will be applied across many positions in the input image, the network can detect cats in any part of the image.

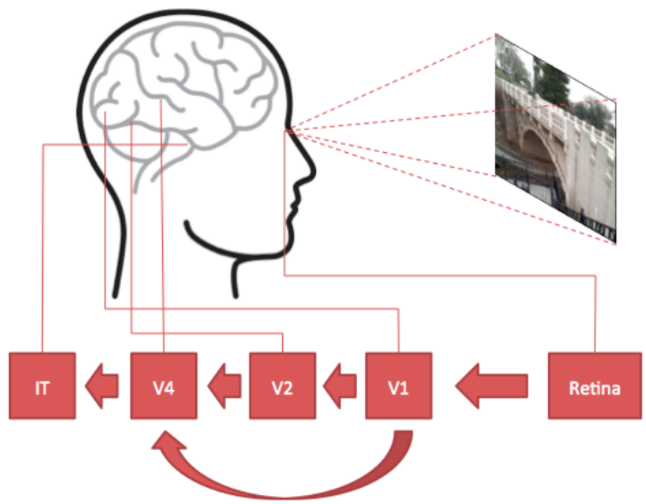


Figure 2.2: A brief illustration of a ventral stream of the visual cortex in the human vision system. It consists of primary visual cortex (V1), visual areas (V2 and V4) and inferior temporal gyrus (Wang and Raj, 2017).

Another type of architecture is **recurrent neural networks** (RNNs) which model **sequential data**, meaning they have **sequential memory**. An RNN takes in different kinds of inputs (text, words, letters, parts of an image, sounds, etc.) and returns different kinds of outputs (the next word/letter in the sequence, paired with a fully-connected network it can return a classification, etc.). While this can give an RNN a rudimentary form of **memory**, it also exacerbates problems with **vanishing and exploding gradients**. Because computing the gradient depends on multiplying by the same parameter values repeatedly, this can cause the gradients to explode (if the parameter is greater than one) or vanish (if the parameter is less than one). **Long Short-Term Memory** (LSTM) networks (Hochreiter and Schmidhuber, 1997) help to address this problem by adding an **input, output, and forget gate** to each recurrent cell. These gates allow the network to learn when to update the information in the cell and when to erase it, rather than simply multiplying by the same parameters each time.

Recently, **transformers** have emerged as an alternative to RNNs (Vaswani et al., 2017). These models make use of an **attention** mechanism to summarize inputs of varying

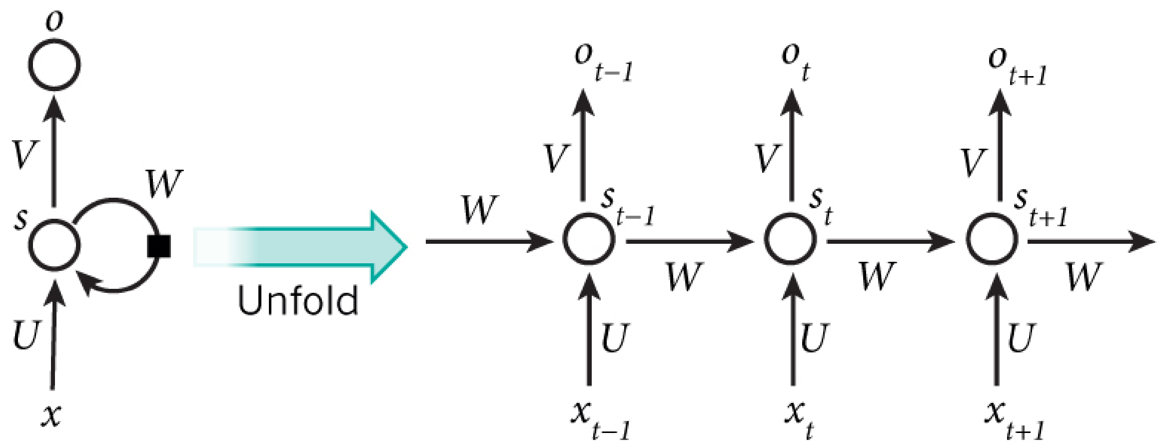


Figure 2.3: A recurrent neural network with one hidden unit (left) and its unrolling version in time (right). The unrolling version illustrates what happens in time: S_{t-1}, S_t, and S_{t+1} are the same unit with different states at different time steps. (LeCun et al., 2015)

lengths based on dynamically changing, learned attention weights. Transformers have been shown to be highly effective at modeling sequences of data, and consequently have led to impressive results in music generation (Huang et al., 2018) and text generation (Radford et al., 2019).

2.2 Generative AI

What is would be like if machines can not only understand language, creativity, and innovation, but can create create it themselves. This is what generative AI is trying to do – they have really done a huge, great work towards it. For AI researcher, they are not there yet because they want to reach what they called Artificial General Intelligence (AGI) which is an AI that can do anything like humans or even better than humans.

While researcher still have a lot of work to do, Generative AI has captured interest across the business population. According to a survey by (McKinsey, 2023); seventy-nine percent of individuals across regions, industries, ans seniority levels are using generative AI for work and outside of work. Organizations, too, are now commonly using gen AI. One-third of all respondents say their organizations are already regularly using generative AI in at least one function—meaning that 60 percent of organizations with reported AI adoption are using gen AI. What's more, 40 percent of those reporting AI adoption at their organizations say their companies expect to invest more in AI overall thanks to generative AI, and 28 percent say generative AI use is already on their board's agenda.

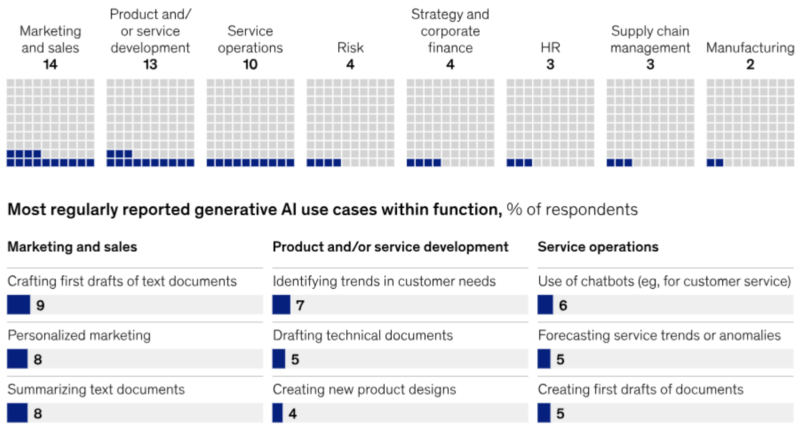
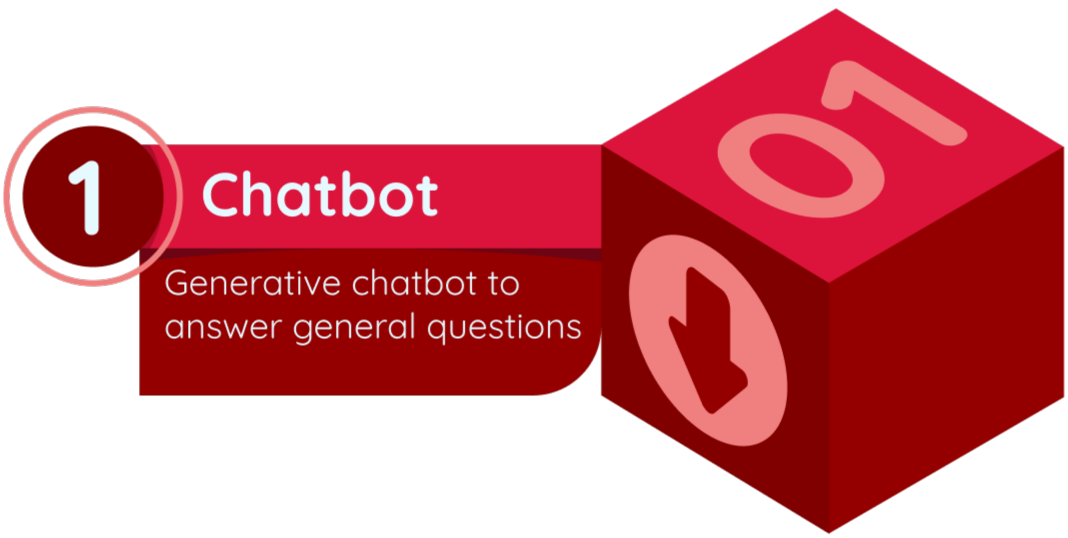


Figure 2.4: The most commonly reported uses of generative AI tools are in marketing and sales, product and service development, and services operations (McKinsey, 2023).

While the use of AI and generative chatbot is still limited; according to figure 2.4 only 14% in the best case. And, for sure, it will be less and less in the case of startups where there is no resources or money to waste in external technologies.

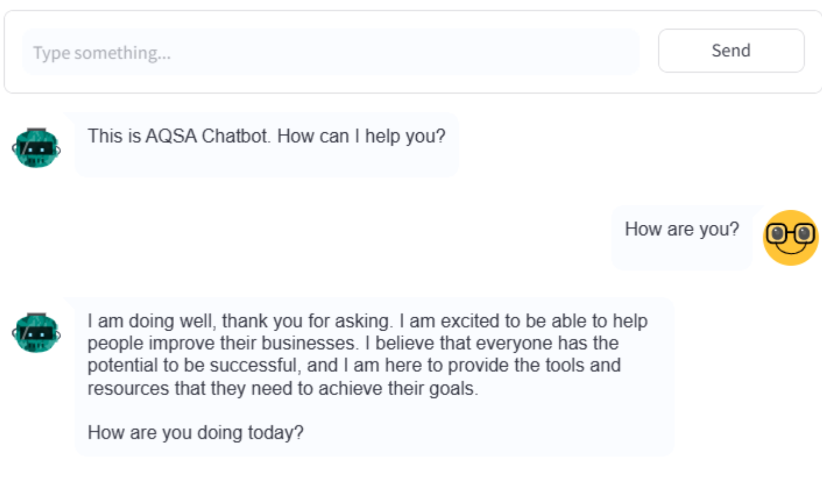
Chapter 3

AQSA Chatbot



The initial stages of a startup venture are often characterized by a dynamic and information-intensive environment. Founders and their teams must navigate a multitude of challenges, from selecting the most appropriate project management methodology to troubleshooting technical hurdles. In this context, readily available access to relevant information and guidance becomes crucial for informed decision-making and efficient problem-solving. The Generative Chatbot component of AQSA fulfills this critical role by serving as a conversational knowledge base and virtual assistant.

This chapter explores the design and functionalities of the Generative Chatbot within the AQSA framework. We will delve into the chatbot's architecture, focusing on its natural language processing (NLP) capabilities and its knowledge base management system.



3.1 How it works

The chatbot uses a combination of techniques for robust intent recognition. Named Entity Recognition (NER) allows the chatbot to identify and classify specific entities within the user's query, such as project management methodologies (e.g., Agile, Waterfall) or specific AQSA functionalities. Part-of-Speech tagging further refines the understanding of the user's intent by identifying grammatical elements like verbs and nouns within the query. Finally, Syntactic analysis examines the sentence structure to grasp the overall meaning and context of the user's question.

To provide insightful and informative responses, the Generative Chatbot relies on a comprehensive knowledge base. This knowledge base houses a structured collection of information relevant to startup operations, project management best practices, and the functionalities offered within the AQSA platform itself. The knowledge base can encompass various information formats, including text documents, multimedia resources, and structured data.

3.2 First Version: PaLM

In the first version, we used PaLM – a powerful factual language model from Google AI. PaLM stands out for its versatility and ability to handle a vast array of prompts and questions. This versatility stems from its underlying architecture, the Pathway System (Barham et al., 2022). This innovative system facilitates highly efficient training across multiple TPU (Tensor Processing Unit) Pods. PaLM exhibits exceptional proficiency in understanding complex queries and generating informative responses across a wide range of domains, making it a compelling choice for the initial development phase of the AQSA chatbot.

When Google team (Narang and Chowdhery, 2022) compares the performance of PaLM to Gopher (Rae et al., 2022) and Chinchilla (Hoffmann et al., 2022), averaged across a common subset of 58 of the tasks. Interestingly, the team notes that PaLM's performance as a function of scale follows a log-linear behavior similar to prior models, suggesting that performance improvements from scale have not yet plateaued. PaLM 540B 5-shot also does better than the average performance of people asked to solve the same tasks.

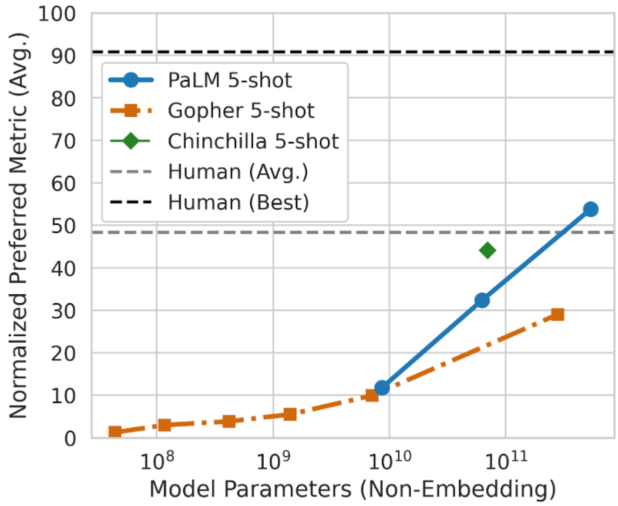


Figure 3.1: Scaling behavior of PaLM on a subset of 58 BIG-bench tasks. (Narang and Chowdhery, 2022)

By combining model scale with chain-fo-thought prompting (Wei et al., 2023) ,PaLM also shows breakthrough capabilities on reasoning tasks that require multi-step arithmetic or common-sense reasoning. Prior LLMs, like Gopher (Rae et al., 2022), saw less benefit from model scale in improving performance.

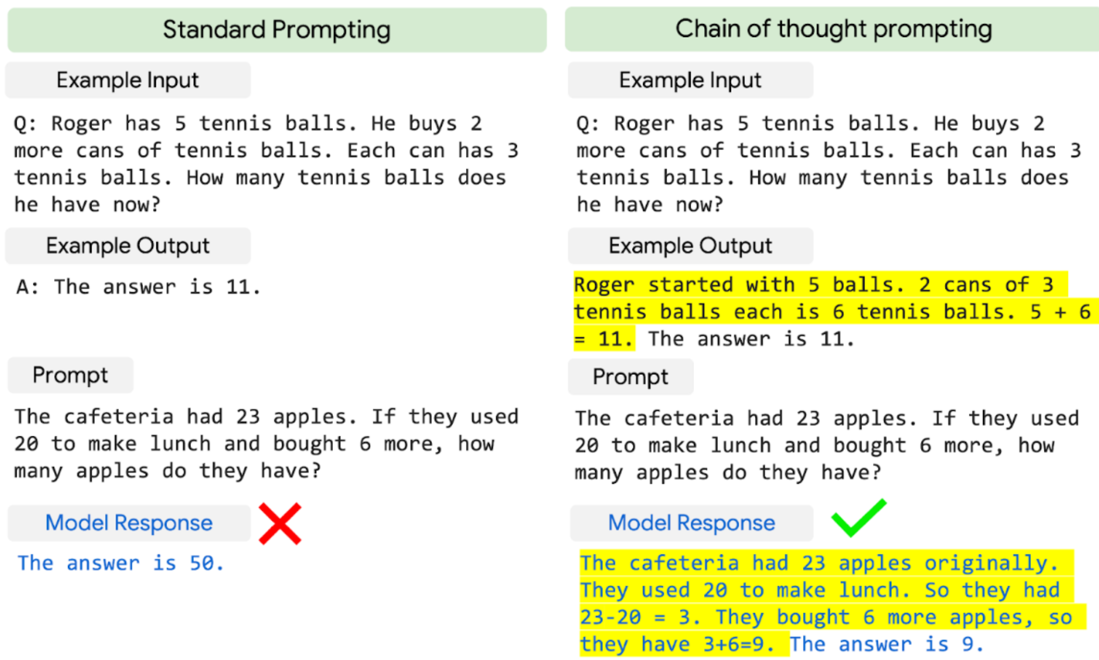


Figure 3.2: Standard prompting versus chain-of-thought prompting for an example grade-school math problem. Chain-of-thought prompting decomposes the prompt for a multi-step reasoning problem into intermediate steps (highlighted in yellow), similar to how a person would approach it (Narang and Chowdhery, 2022).

3.3 Add Arabic Language: AraBert

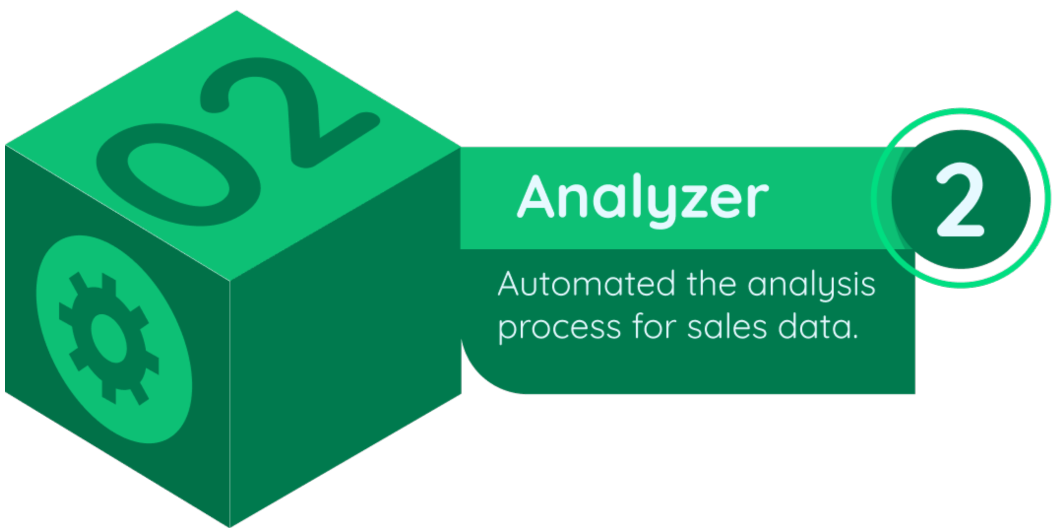
The initial version of the Generative Chatbot within AQSA leveraged PaLM (Pathway Language Model) for natural language processing tasks. PaLM is a factual language model from Google AI, trained on a massive dataset of text and code. Its versatility and ability to handle a wide range of prompts and questions made it a suitable choice for the initial development phase. However, PaLM is primarily designed for English language processing and may not perform optimally when handling user queries in Arabic.

To address this limitation and cater to a broader user base, the second version of the Generative Chatbot adopted AraBERT, a pre-trained Arabic language model. AraBERT is based on the BERT (Bidirectional Encoder Representations from Transformers) architecture, specifically fine-tuned on a large corpus of Arabic text data. This fine-tuning process allows AraBERT to capture the nuances of Arabic grammar, morphology, and syntax, leading to superior performance in understanding and responding to Arabic user queries.

The selection of AraBERT reflects the growing body of research on Arabic language models. While PaLM represents a powerful general-purpose language model, other Arabic models like mBERT (Arabic BERT) and ALBERT (A Lite BERT) offer strong alternatives for specific tasks. The choice between these models depends on factors such as the size and domain of the training data, as well as the desired balance between accuracy and computational efficiency.

Chapter 4

Automated Data Analyzer



Data is the lifeblood of any successful startup. It provides invaluable insights into customer behavior, market trends, and operational efficiency. However, for many startups, the sheer volume of data can be overwhelming. Manually sifting through this data to identify meaningful patterns and trends is not only time-consuming but also prone to human error. This is where the Automated Data Analyzer within the AQSA steps in. It has two main assumptions; data is already clean and business understand its own data.

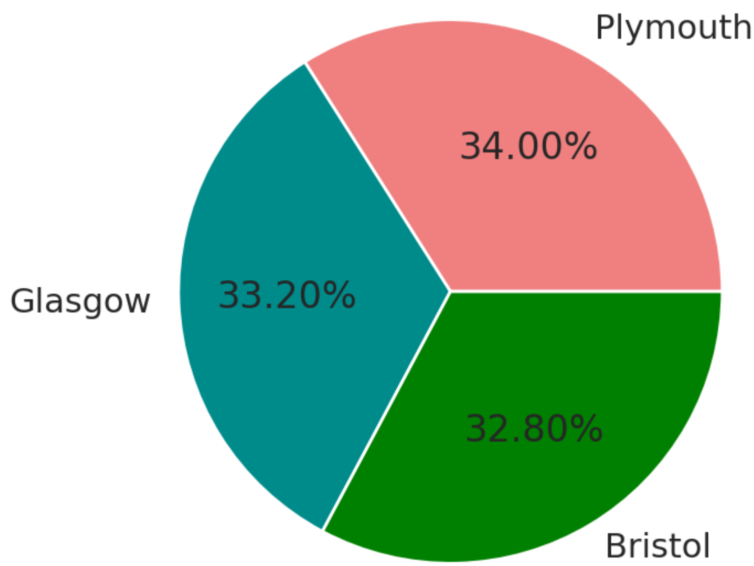
Our Automated Data Analyzer consists of three parts; target analysis which analyze the data around one main categorical variable like branch for a supermarket as shown in figure ?. Also, we have categorical variable analysis which plots the distribution of the categorical columns while the third part is numerical data analyzer.

4.1 Target Analysis

This module focuses on analyzing data in relation to a single, pre-defined categorical variable. It's akin to drilling down into the data and examining it through the lens of that specific variable. This allows users to gain granular insights into how different categories perform on various metrics. The example you provided, analyzing supermarket data by branch, exemplifies this type of analysis. In this scenario, target analysis can reveal a wealth of information that can be used to optimize operations and improve customer satisfaction. For instance, it can expose which branches have the highest sales figures, allowing for the identification of best practices that can be replicated across the supermarket chain. Conversely, it can also pinpoint branches that are consistently underperforming. Delving deeper into the data for these underperforming branches, the analysis might reveal that they cater to a different customer demographic with distinct purchasing habits. This information can then be used to tailor marketing strategies and product offerings to better resonate with the local customer base, potentially improving sales performance and overall customer satisfaction at that branch.

```python
st.markdown('### Target Analysis...')
counts = data[target].value_counts()
counts
trans_counts = []
trans_labels = data[target].unique()
for i, count in enumerate(counts):
    trans_counts.append(counts[i])

# Create the pie chart
plt.figure(figsize=(4,4))
plt.pie(trans_counts, labels=trans_labels, autopct="%1.2f%%", colors=['lightcoral', 'darkcyan',
                                                                       'green', 'red', 'blue', 'gray'])
st.pyplot(plt.gcf())
```



Target analysis can also be applied to a multitude of other business scenarios beyond supermarkets. For example, a ride-sharing company might utilize target analysis to examine ride data in relation to driver demographics (e.g., full-time drivers, part-time drivers). This analysis could reveal variations in customer satisfaction ratings or average trip fares associated with different driver demographics.

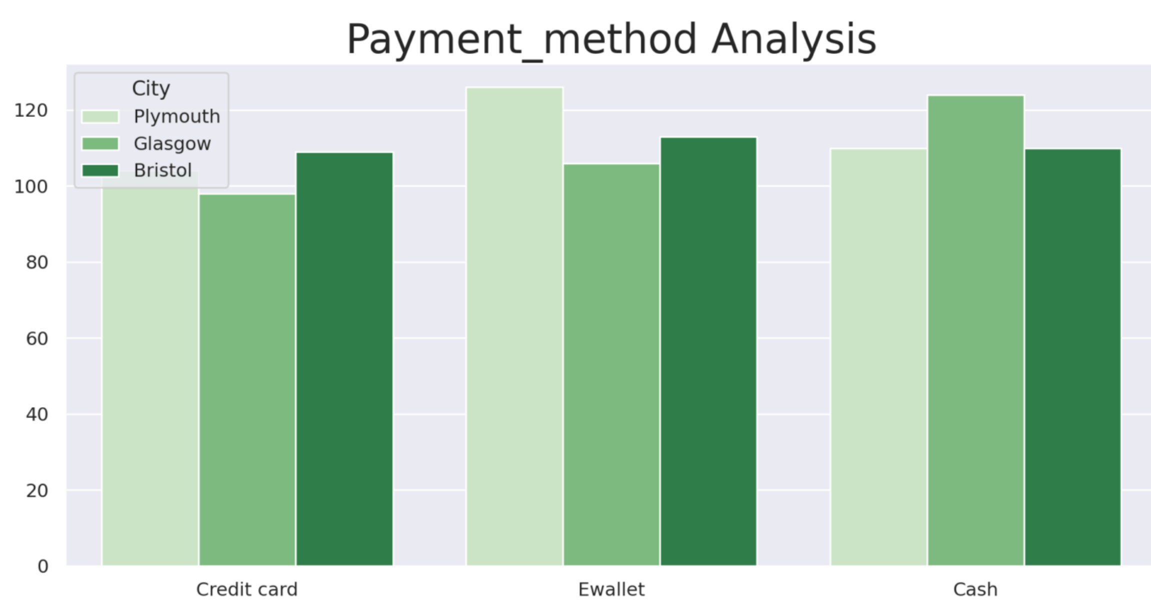
4.2 Categorical Variables Analysis

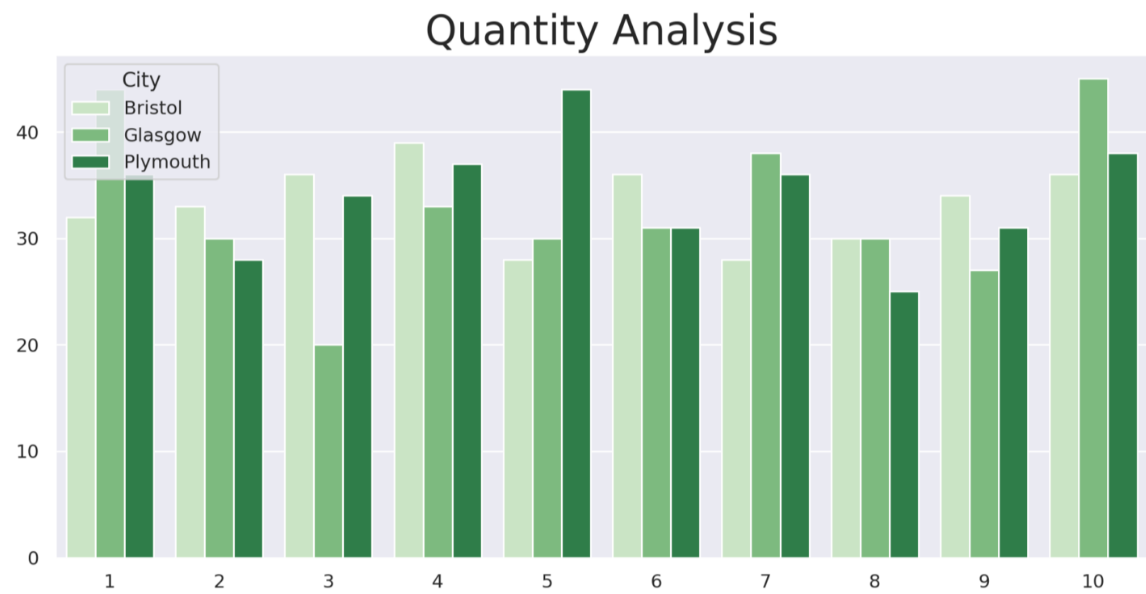
This component delves into the distribution of categorical variables within the data. Categorical variables represent data that can be classified into distinct groups or categories, such as customer product preferences (e.g., dairy, produce, bakery) or customer acquisition channel (e.g., online marketing, social media, in-store promotions). Categorical variable analysis helps users visualize the distribution of data across these categories. This can be achieved through various techniques, including bar charts, pie charts, and heatmaps. By examining these visualizations, users can identify dominant categories, potential outliers, and any relationships that may exist between different categorical variables.

```
st.markdown('### Categorical Variables Analysis...')
fig = plt.figure(figsize=(10, 5*len(cat_cols)))
for i, col in enumerate(cat_cols):
    ax=fig.add_subplot(len(cat_cols), 1, i+1)
    sns.countplot(data=data, x=col, axes=ax, hue=target, palette='Greens')
    plt.title(f"{col} Analysis", fontsize=24)
    ax.set(xlabel=None, ylabel=None)

fig.tight_layout()
st.pyplot(plt.gcf())

st.divider()
```





4.3 Numerical Variables Analysis

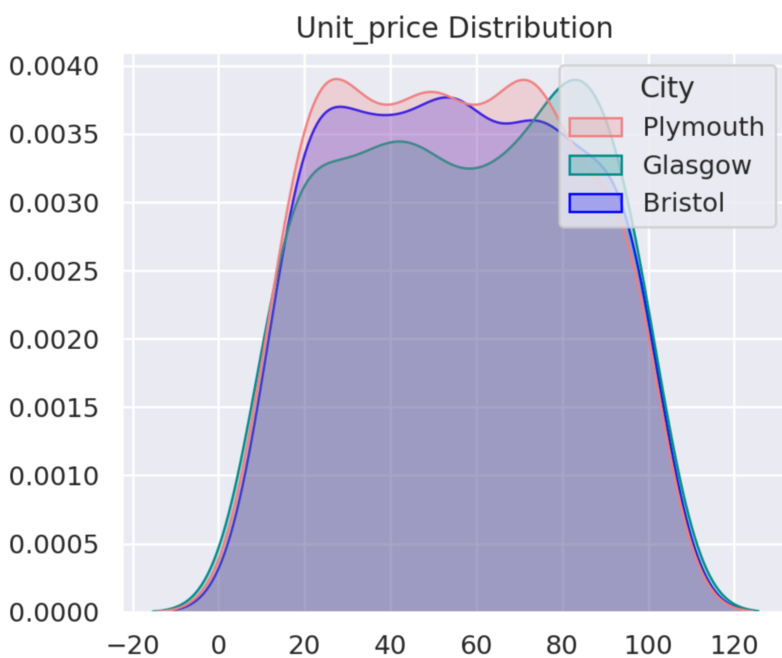
By visualizing the distribution of numerical variables (columns), AQSA helps users understand central tendency (center of the data) and dispersion (spread of the data). Distribution plots like histograms reveal if the data is clustered around a central point (normal distribution) or skewed towards one side. Additionally, the spread of the data points highlights variability and can help identify outliers.

In addition to understanding central tendency, distribution plots also reveal the variability of the data, often referred to as dispersion. Histograms with a narrow spread suggest that the data points are clustered closely around the central tendency, indicating low variability. Conversely, wider histograms with a larger spread depict higher data variability. Understanding the level of dispersion is crucial for tasks like identifying outliers, which are data points that deviate significantly from the majority of the data. AQSA's visualizations can highlight potential outliers, prompting users to investigate them further and determine if they represent genuine data points or errors.

Furthermore, AQSA's focus on distribution visualization can facilitate basic comparative analysis. By generating side-by-side distribution plots for multiple numerical variables, users can visually compare their distributions. This can be particularly insightful for identifying potential relationships between variables. For instance, a startup selling sports equipment might compare the distribution of customer purchase amounts for different product categories (e.g., basketball equipment, baseball equipment). The side-by-side histograms might reveal that basketball equipment purchases tend to have a higher average value compared to baseball equipment purchases.

```python
st.markdown('### Continuous Variables Analysis...')
fig = plt.figure(figsize=(5, 4*len(cont_cols)))
for i, col in enumerate(cont_cols):
    ax=fig.add_subplot(len(cont_cols), 1, i+1)
    sns.kdeplot(data=data, x=col, axes=ax, hue=target, fill=True, palette=['lightcoral', 'darkcyan',
                                                                            'blue', 'red', 'gray',
                                                                            'orange', 'yellow'])
    ax.set_title(f"{col} Distribution")
    ax.set(xlabel=None, ylabel=None)

fig.tight_layout()
st.pyplot(plt.gcf())
```



4.4 Why not to analyze based on time?

Time series data is everywhere and almost any data that collected online has a temporal axis but you can analyze it easily using Google Analytics which is a popular tool globally and in Egypt. But what is not popular is Time Series Forecasting which we will talk about in the next chapter.

Chapter 5

Forecasting Module

In the dynamic world of startups, informed decision-making often hinges on the ability to anticipate future trends. The Forecasting Engine within the AQSA framework addresses this critical need by leveraging the power of machine learning to predict future sales, income, or other relevant metrics. This chapter delves into the functionalities and design principles of the Forecasting Engine. We will explore the various forecasting models employed by the engine, along with factors considered when selecting the most suitable model for a given dataset.

5.1 The origins of time

Time is considered the fourth dimension, along with the three dimensions of space (width, height, and depth). This means that for something to happen, it needs to occur at a specific location in space and at a specific point in time (Minkowski, 1908). On a basic level, time is the experience of events happening in a sequence, from past to present to future. Some events happen very fast like processes in atoms which take femtoseconds (Zewail et al., 1992). Also, there is a more faster event that takes 247 zeptoseconds in which a light particle cross a hydrogen molecule (Grundmann et al., 2020) which broke the record of Ahmed Zewail's femtosecond.

If someone came up to you on street and asked you to draw time, what would you draw? There are many options here like drawing a watch, a calender, or even a forward arrow $\rightarrow$ BUT all of those drawing would be just a representation of linear time. What I want to say is that time cannot be explained by even the smartest being in Earth. Theories like relativity (Einstein, 1915) and quantum mechanics highlight inconsistencies in our classical understanding of time. Physicists like Sabine Hossenfelder argue that we might not fully grasp the nature of time because our current frameworks struggle to reconcile these theories. However, data scientists and researchers like me only care about linear time; a time that can be represented with `datetime` function. With that said, we can move on to our second question: what is time-series data?

5.1.1 What is time series

A time-series data is a series or sequence of data points taken at equally intervals (as shown in figure 5.1). I like to call these "equally intervals", or whatever the thing you usually see in x-axis, a **temporal axis**.This temporal axis does not have to be timed, it just need to be a well-ordered, well-spaced axis (Nielsen, 2019). For example, distance for a self-driving car where you collect data on various aspects of the car's environment(speed, obstacles) every meter it travels. In this scenario, the data points are ordered by distance traveled, not by actual time. Another example is customer journey in an e-commerce website where you track user interactions like product views and clicks. The data can be organized based on the steps a customer take in their buying journey (e.g., browsing an item, adding to cart, checkout). Here, the "time" is not date or hours instead it is the order of actions.

Even this text that you are reading now could also be represented as a sequence of words, phrases, or paragraphs. If you go back to the table of contents, you easily see that you measure your reading by sections not by minutes or hours. With that said, we can apply time-series methods in all of these cases and still have a great performance.

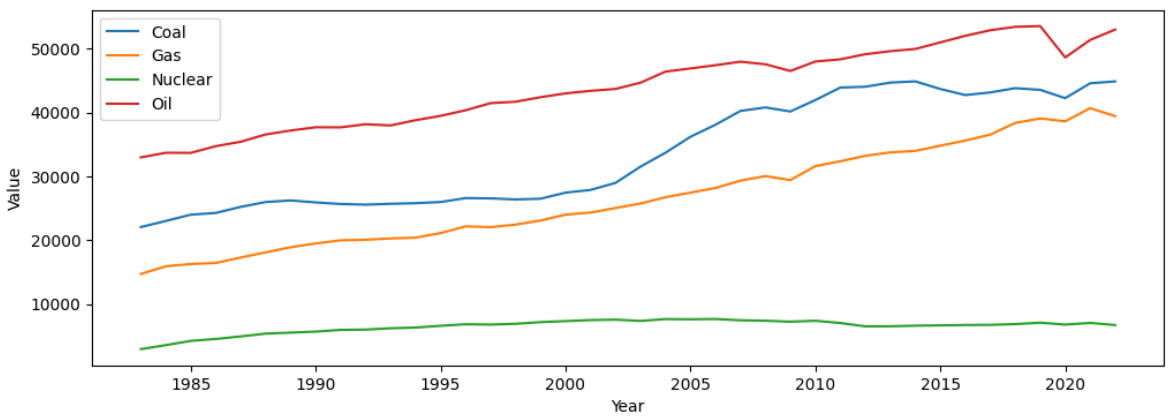


Figure 5.1: Time-series data where the temporal axis is date (in years). Visualizing the change in Non-Renewable energy consumption from 1983 to 2022 (Yosef, 2023)

5.1.2 Components of time series

A common component is **trend** which is a general direction of the data over a long period of time. A trend can be increasing (upward) as in figure 5.2, decreasing (downward), or horizontal (stationary). **Seasonality** on the other hand, is a trend that repeats with respect to time (e.g., an increasing in water consumption in summer due to hot weather conditions). There is a bigger pattern that seasonality called **cyclical component**; say that we tracked your weight in the last couple of years and we found out that you tend to lose weight in the summer and gain it back in the winter. This long, repeating pattern is the cyclical component in time series data. We also have **irregular variation** or random variation (also called error or residual as in figure 5.2) which is the unpredictable wiggle in the data. These unpredictable variations can be caused by one-time event like unexpected storms, new announcements, etc.

To extract some of these components from the data, we use **ETS Decomposition**. When something is composed of other things, it contains those elements. The same thing with time series data which contains the **ETS**; where **E** stands for error or irregular variations, **T** stands for trends, and **S** stands for seasons. By decomposing the data or breaking down complex time series data into simpler components, ETS, we gain insights and clarity. Whether it's analyzing time series data or solving a big problem, decomposition helps me see the bigger picture.

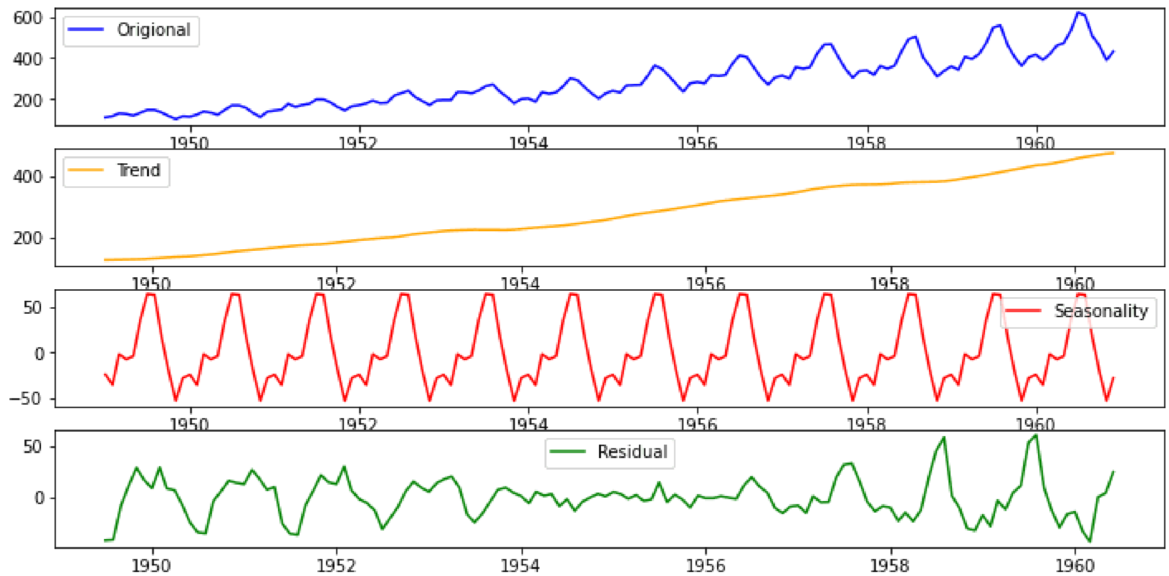
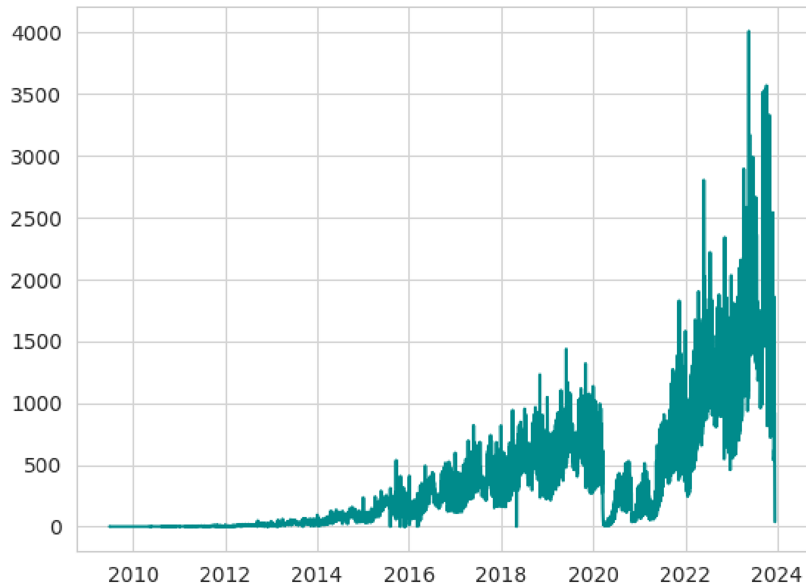


Figure 5.2: Decomposition of time series components (Bhat, 2023)



5.2 Stationarity

A stationary time series is one in which a time series measurement reflects a system in a steady state. Sometimes it is difficult to assert what exactly this means, and it can be easier to rule things out as not being stationary rather than saying something is stationary. An easy example of data that is not stationary is the airline passengers data set which plotted in Figure 5.2. There are several traits that show this process is not stationary. First, the mean value is increasing over time, rather than remaining steady. Second, the distance between peak and trough on a yearly basis is growing, so the variance of the process is increasing over time. Third, the process displays strong seasonal behavior, the antithesis of stationarity (Nielsen, 2019).

Tests for determining whether a process is stationary are called hypothesis tests. The Augmented Dickey-Fuller (ADF) test is the most commonly used metric to assess a time series for stationarity problems. This test posits a null hypothesis that a unit root is present in a time series. Depending on the results of the test, this null hypothesis can be rejected for a specified significance level, meaning the presence of a unit root test can be rejected at a given significance level.

```python
from statsmodels.tsa.stattools import adfuller

def test_stationarity(df, ts):
  # determing rolling statistics
  rolmean = df[ts].rolling(window=12, center=False).mean()
  rolstd = df[ts].rolling(window=12, center=False).std()

  # plot
  original = plt.plot(df[ts], color='lightgray', label='Original')
  mean = plt.plot(rolmean, color='crimson', label='Rolling mean')
  std = plt.plot(rolstd, color='darkcyan', label='Rolling std')

  plt.legend(loc='best')
  plt.title(f'Rolling mean and std for {ts}')
  plt.show()

  # perform dickey-fuller test:
  # H_0: data is not stationary
  # H_1: data is stationary

  print('\nResults of Dickey-Fuller Test:')
  adftest = adfuller(df[ts], autolag='AIC')
  dfoutput = pd.Series(adftest[0:4],
                      index=['Test Statistic',
                             'p-value',
                             'Num of Lags used',
                             'Num of observations used'])
  for key, value in adftest[4].items():
    dfoutput[f'Critical Value {key}'] = value
  print(dfoutput, '\n\n')

  if dfoutput['p-value'] > 0.05:
    print("Failed to reject H_0 and data is not stationary")
  else:
    print("Reject H_0 and data is stationary ✅")
```

5.3 AutoARIMA

AutoARIMA, a cornerstone of the AQSA Forecasting Engine, represents a powerful and versatile statistical method for time series forecasting. It builds upon the ARIMA model, which forecasts future values in a time series by considering past values (Autoregressive component), differencing the data to achieve stationarity (Integrated component), and incorporating the residuals or errors from past forecasts (Moving Average component).

AutoARIMA streamlines the ARIMA approach by automating the process of identifying the optimal model parameters. This includes determining the order of the autoregressive (AR) terms, the degree of differencing (I), and the order of the moving average (MA) terms. By analyzing the characteristics of the time series data, AutoARIMA can automatically select the most suitable combination of these parameters, resulting in a robust and accurate forecasting model.

Here are some key advantages of using AutoARIMA for time series forecasting. **Interpretability**: AutoARIMA models are relatively interpretable, meaning that the factors influencing the forecasts can be understood and explained. This can be particularly valuable for startups, as it allows them to gain insights into the rationale behind the predicted values. **Efficiency**: AutoARIMA automates the model selection process, saving time and resources compared to manual ARIMA parameter identification. This is particularly beneficial for startups with limited data science expertise. **Versatility**: AutoARIMA can handle a wide range of time series data, including those with trends, seasonality, and varying degrees of stationarity.

It's important to note that AutoARIMA, like all statistical models, has limitations. Its effectiveness relies on the presence of historical data that exhibits clear patterns and trends. For highly volatile or non-linear time series data, deep learning models may offer superior forecasting performance.

5.4 Deep Learning for Time Series

While AutoARIMA offers a robust and interpretable solution for many forecasting tasks, deep learning models have emerged as a powerful alternative for complex time series data. This section delves deeper into the specific deep learning architectures employed within the AQSA Forecasting Engine and explores the advantages and considerations associated with their use.

5.4.1 Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTMs):

A cornerstone of deep learning for time series forecasting lies in Recurrent Neural Networks (RNNs). RNNs are a class of artificial neural networks specifically designed to handle sequential data, such as time series. Unlike traditional feedforward neural networks, RNNs incorporate a feedback loop that allows them to process information from previous time steps in the sequence. This enables RNNs to capture temporal dependencies within the data, a crucial aspect for effective time series forecasting.

However, RNNs can struggle with capturing long-term dependencies within time series data. This is where Long Short-Term Memory (LSTM) networks come into play. LSTMs are a specific type of RNN architecture that incorporates memory cells to explicitly learn and retain information from past time steps. These memory cells allow LSTMs to effectively model long-term dependencies within time series data, making them particularly well-suited for forecasting tasks where historical data has a significant influence on future values.

5.4.2 Advantages of Deep Learning for Time Series Forecasting:

- **Non-linearity:** Deep learning models excel at capturing complex non-linear relationships within data. This is particularly advantageous for time series data that exhibits trends, seasonality, or other non-linear patterns that may not be adequately captured by traditional statistical methods.
- **Feature Learning:** Deep learning models can automatically learn relevant features from raw data, eliminating the need for manual feature engineering. This can be particularly beneficial for complex time series data where identifying the most informative features can be challenging.
- **High Accuracy:** When trained on large amounts of high-quality data, deep learning models can achieve superior forecasting accuracy compared to traditional statistical methods, especially for complex or volatile time series data.

5.4.3 Considerations for Deep Learning in AQSA:

- **Computational Cost:** Deep learning models often require significant computational resources for training. The AQSA framework employs techniques like model optimization and distributed computing to address this challenge, ensuring efficient training and forecasting even for large datasets.

- **Data Availability:** Deep learning models typically perform best when trained on vast amounts of data. For startups with limited historical data, AutoARIMA or other statistical methods may be more suitable.

- **Interpretability:** Unlike AutoARIMA, deep learning models can be less interpretable, making it challenging to understand the rationale behind the forecasts. The AQSA framework incorporates techniques like visualization and saliency maps to enhance the interpretability of deep learning forecasts, empowering users to make informed decisions with greater confidence.

Chapter 6

Guides

AQSA

There is no doubt that every new tool that the user interacts with must have instructions to make the user understand and interact with the tool more easily and to know what the tool is basically doing, so we have given guidance to this tool so that the user can understand its role and how to use it and deal with it in order to give it the desired results.

Guidelines are a set of directions and recommendations that serve as a guide for performing a specific task correctly and effectively. Guidelines are an essential part of various fields, ranging from business and education to health and technology. The purpose of guidelines is to provide a framework that guides individuals or teams in executing tasks efficiently and achieving desired outcomes with minimal difficulty and risk.

6.1 Importance of Guidelines

- **Guiding Processes:** Guidelines help to guide processes and identify the necessary steps for executing the task correctly. Guidelines are considered a roadmap, guiding individuals, and teams through the stages of the task.
- **Data Quality Assurance:** The Guidelines provide a practical framework for verifying the quality of data used in the project, including validation, cleaning, and optimization.
- **Achieving Maximum Benefit:** The Guidelines help in using resources and time efficiently, achieving maximum results with minimal effort.
- **Ensuring quality and accuracy:** The guidelines contribute to ensuring the quality and accuracy of the work, as they specify the correct standards that must be followed.
- **Reducing errors:** By providing correct guidance, guidelines can reduce errors that may occur during the task execution.

Finally, guidelines are a valuable and necessary tool in various fields, as they help to guide individuals and teams towards achieving goals efficiently and effectively. With guidelines, performance can be improved, and errors can be reduced, leading to success and development in different fields.

6.2 Nature of data given

Here we will focus on understanding the nature of the data that we are working on in our project. We're going to analyze the type and source of data, and we're going to identify variables and columns in the data. We will provide examples of types of data that may be part of data analysis projects, such as sales data and financial statements. We will explain how the data are checked and whether they meet the required standards, as well as how they are properly organized for use in analysis processes. The statements given must relate to sales or financial data. This data can include information such as monthly or annual sales, revenues, costs, profits, and any additional information you wish to analyze. Make sure to explain the nature of this data in the instructions to users to ensure they understand the nature of the data they will be working with. When we talk about "sales data" or "financial data," we are referring to a set of data that focuses on the financial operations of a company or organization. This data can include information about sales, revenues, costs, and profits, and is often broken down by specific time periods such as monthly, quarterly, or annually.

- **Sales:** This includes shared data collected from sales over a specific period of time and can be divided into different types such as products or services.
- **Revenues:** This data includes the total income which was achieved by the company from sales, and this includes sales in addition to any other revenues such as additional income from fees or benefits.
- **Costs:** This data includes costs associated with providing products or services and can be divided into raw material costs and wages.
- **Profits:** Representing the difference between revenues and costs, profits indicate the net profit earned by the company during the specified period.

  While explaining to users how to use a data analyzer, we should explain that the data they are going to analyze and explore whether this data is "sales data" or "financial data". This will help them understand the context and purpose of using the analyzer and how to apply it to the data provided.

6.3 No errors

Ensuring that there are no errors in the given data and that there are no empty values is an essential step to ensure the accuracy and validity of the analyzes and reports that will be generated. This is done through a set of steps:

1. **Check data for errors**

   Before importing data into the analyzer, you should check it carefully to make sure there are no formatting errors or missing data.

2. **Dealing with empty values**

   Empty values in the data must be identified and handled before the analysis begins. You can fill empty values with specific values such as zero, average, or previous value, or even delete rows or columns that contain empty values depending on the nature of the data and the analysis method.

3. **Verification of data accuracy**

   Check the data for accuracy and validity. Verify that the input values are suitable with the expectation, and make sure that there are no unreasonable or incorrect values.

4. **Use automated testing tools**

   Automated testing tools and data analysis utilities can be used to detect errors and null values automatically, which helps speed up the data verification process.

5. **Data documents**

   It would be preferable to create documentation detailing the structure of the data used, handling possible errors, and how to deal with empty values. This helps to keep records and simplify analysis processes in the future.

6. **Final test**

   Once the data is imported and handling possible errors are processed, final tests must be performed to ensure that the data is ready for use in analysis and reporting.

   By following these steps, you can ensure the quality and validity of the data, leading to accurate and reliable analyses and more effective reports.

6.4 Quality of given data

In this section, we will focus on methods for ensuring data quality. We will talk about how to check for errors in the data, such as missing values, illogical values, and excess or missing values. We will present methods and tools for cleaning and optimizing data, including removing duplicate data, correcting grammatical errors, and setting standard values. We'll also learn the best techniques for documenting data and keeping accurate records of analysis. The quality of the input data into the data analyst plays a crucial role in the accuracy of the analyzes and conclusions reached. This is done through several steps, which are as follows:

- **Reliability**

  Make sure that the data you provide is reliable and taken from reliable sources. Shared data may be from the company's financial system or from approved external sources.

- **Quantity and quality**

  Make sure that you receive a sufficient quantity of data and that it covers all important aspects of the analysis. For example, if you are analyzing sales, that data should include information about all products or services offered.

- **Completeness**

  Make sure there is no missing or incomplete data, and that all necessary fields are present and completed correctly. If there is missing data, this may distort the analyzes and reduce their accuracy.

- **Keep history**

  The given data must contain historical information for a sufficient period of time. This can allow for long-term trend analysis and understanding of past patterns of performance**.**

- **Cleaning and optimization**

  Before using the data in analytics, you may need cleaning and optimization processes to correct grammatical errors, remove duplicate data, standardize formatting, and other actions that enhance data quality.

- **Validation**

  Before starting analyses, it is a good idea to validate the data through comparisons with independent sources or through the use of automated validation tools.

  By considering these points and ensuring the quality of the input data, you can increase the accuracy and reliability of the analyses and conclusions you extract from the data analyst.

6.5 Convert your data into CSV

To convert your data to CSV format, you can follow these general steps:

1. **Organize Your Data**

   Make sure your data is organized in a tabular format, with rows representing individual records and columns representing different attributes or variables.

2. **Export or Save as CSV**

   Many software applications, such as Microsoft Excel or Google Sheets, allow you to directly export or save your data as a CSV file. Look for the "Save As" or "Export" option in the File menu and choose CSV as the file format.

3. **Check Settings**

   When saving/exporting as CSV, ensure that the settings are configured appropriately. This may include specifying the delimiter (usually a comma for CSV), encoding, and any other relevant options.

4. **Review the Output**

   Once the CSV file is generated, open it in a text editor or spreadsheet program to review the formatting and ensure that the data has been properly converted.

5. **Make Adjustments if Necessary**

   Depending on the software you used and the complexity of your data, you may need to adjust the CSV file, such as handling special characters or formatting problems.

6. **Save the CSV File**

   Once you're satisfied with the CSV file, save it in a location of your choice, and it will be ready for use.

Following these instructions, you can easily convert the data into the CSV format and prepare it for use in the data analysis tools to carry out the required analyses.

Chapter 7

Future Steps

7.1 Personalized Chatbots: Tailoring the Experience

The current iteration of the AQSA chatbot provides a valuable one-size-fits-all solution for startup inquiries. However, future iterations could explore the concept of personalized chatbots. These chatbots would be tailored to the specific needs and industry of each startup. Imagine an AQSA chatbot for a fintech startup that can answer questions about regulatory compliance or secure payment processing. Conversely, a chatbot for a fashion startup might engage in discussions about trend forecasting or social media marketing strategies.

Personalized chatbots require a deeper understanding of each startup's unique domain and challenges. This could be achieved through a combination of initial user onboarding surveys, ongoing interactions with the chatbot, and integration with other data sources within the AQSA framework. By leveraging these insights, personalized chatbots can offer a more relevant and engaging user experience, fostering a stronger connection between AQSA and the startups it serves.

7.2 Scaling the Analyzer: Embracing Big Data

The current capabilities of the Automated Data Analyzer within AQSA are well-suited for startups with moderate data volumes. However, as startups grow and accumulate more data, the analyzer may require enhancements to handle larger and more complex datasets. This could involve exploring new server configurations with increased processing power or potentially migrating the AQSA framework to a cloud-based solution.

Cloud platforms offer several advantages for handling big data. They provide virtually limitless scalability, allowing the AQSA framework to seamlessly adapt to the growing data demands of startups. Additionally, cloud platforms offer access to on-demand computational resources, ensuring efficient data processing even for massive datasets.

Furthermore, the future of the Data Analyzer lies in exploring advanced data mining techniques. By incorporating natural language processing (NLP) capabilities, the analyzer could automatically identify key insights and trends within unstructured data sources like emails, customer reviews, or social media posts. This comprehensive data analysis would empower startups to gain a holistic understanding of their operations and customer behavior, ultimately driving more informed decision-making.

7.3 Reinforcement Learning for Intelligent Forecasting

The AQSA Forecasting Engine currently leverages a combination of statistical and deep learning models to generate future predictions. Future iterations could explore the integration of reinforcement learning techniques. Reinforcement learning involves training an AI model through a process of trial and error, where the model receives rewards for making accurate predictions and penalties for inaccurate ones.

By incorporating reinforcement learning, the AQSA Forecasting Engine could continuously improve its forecasting accuracy over time. The model would be able to learn from past forecasting errors and adjust its approach to generate more reliable predictions in the future. This would be particularly valuable for startups operating in dynamic and unpredictable environments, where the ability to anticipate future trends is critical for success.

7.4 Beyond the Horizon: The Evolving Landscape of AQSA

The roadmap for AQSA extends beyond the advancements outlined in this chapter. The framework can potentially integrate with other AI-powered tools to offer a more comprehensive suite of services for startups. For instance, AQSA could integrate with design thinking tools or project management platforms, further streamlining the startup journey.

Furthermore, AQSA could explore the potential of generative AI to assist startups in various creative tasks. Imagine an AQSA module that can generate marketing copy, design prototypes, or even write code snippets based on user input. These capabilities would empower startups to operate more efficiently and bring their innovative ideas to life faster.

Final Words

First and foremost, we want to thank our esteemed supervisor, Dr. Mohamed Ibrahim, for his unwavering support and guidance throughout the development of this project. He has been instrumental in shaping AQSA into the robust platform it is today.

The journey, however, does not end here. AQSA's potential for growth and evolution is vast. We are eager to witness the ways in which startups leverage this framework to cultivate groundbreaking ideas and propel their ventures towards success. The future of innovation is bright, and AQSA stands ready to illuminate the path forward.

Bibliography

Paul Barham, Aakanksha Chowdhery, Jeff Dean, Sanjay Ghemawat, Steven Hand, Dan Hurt, Michael Isard, Hyeontaek Lim, Ruoming Pang, Sudip Roy, Brennan Saeta, Parker Schuh, Ryan Sepassi, Laurent El Shafey, Chandramohan A. Thekkath, and Yonghui Wu. Pathways: asynchronous distributed dataflow for ml, March 2022. URL http://arxiv.org/abs/2203.12533. arXiv:2203.12533 [cs].

Harshini Bhat. Understanding time series analysis: basic guide & examples, June 2023. URL https://www.almabetter.com/bytes/articles/time-series-analysis.

Albert Einstein. On the general theory of relativity. *Prussian Academy of Science*, 1915.

Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, April 1980. ISSN 1432-0770. doi: 10.1007/BF00344251. URL https://doi.org/10.1007/BF00344251.

Sven Grundmann, Daniel Trabert, Kilian Fehre, Nico Strenger, Andreas Pier, Leon Kaiser, Max Kircher, Miriam Weller, Sebastian Eckart, Lothar Ph. H. Schmidt, Florian Trinter, Till Jahnke, Markus S. Schöffler, and Reinhard Dörner. Zeptosecond birth time delay in molecular photoionization. *Science*, 370(6514):339–341, October 2020. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.abb9318. URL https://www.science.org/doi/10.1126/science.abb9318.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, November 1997. ISSN 0899-7667, 1530-888X. doi: 10.1162/neco.1997.9.8.1735. URL https://direct.mit.edu/neco/article/9/8/1735-1780/6109.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, March 2022. URL http://arxiv.org/abs/2203.15556. arXiv:2203.15556 [cs].

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer, 2018. URL https://arxiv.org/abs/1809.04281.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. ISSN 1558-2256. doi: 10.1109/5.726791. URL https://ieeexplore.ieee.org/document/726791.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, May 2015. ISSN 1476-4687. doi: 10.1038/nature14539. URL https://www.nature.com/articles/nature14539.

McKinsey. The state of ai in 2023: Generative ai's breakout year. *QuantumBlack*, August 2023. URL https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai-in-2023-generative-ais-breakout-year.

Hermann Minkowski. Space and time, 1908. URL https://mathweb.ucsd.edu/~b3tran/cgm/Minkowski_SpaceAndTime_1909.pdf.

Sharan Narang and Aakanksha Chowdhery. Pathways language model (palm): Scaling to 540 billion parameters for breakthrough performance. *Google Research*, April 2022. URL https://research.google/blog/pathways-language-model-palm-scaling-to-540-billion-parameters-for-breakthrough-performance/.

Aileen Nielsen. *Practical Time Series Analysis*. O'Reilly Media, Inc, October 2019. ISBN 978-1-4920-4165-8. URL https://www.oreilly.com/library/view/practical-time-series/9781492041641/.

OpenAI. Introducing chatgpt. *OpenAI Blog*, November 2022. URL https://openai.com/blog/chatgpt.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019. URL https://api.semanticscholar.org/CorpusID:160025533.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki

Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. Scaling language models: methods, analysis & insights from training gopher, January 2022. URL http://arxiv.org/abs/2112.11446. arXiv:2112.11446 [cs].

David Silver and Demis Hassabis. Alphgo mastered the ancient game of go. *DeepMind*, October 2015. URL https://deepmind.google/technologies/alphago/.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv*, 2017. doi: 10.48550/ARXIV.1706.03762. URL https://arxiv.org/abs/1706.03762.

Haohan Wang and Bhiksha Raj. On the origin of deep learning, 2017. URL https://arxiv.org/abs/1702.07800.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, January 2023. URL http://arxiv.org/abs/2201.11903. arXiv:2201.11903 [cs].

Mohamed Yosef. Global energy consumption from 1983 to 2022. *MY101*, 2023. URL https://mohamedyosef101.github.io/publication/global-energy/.

Ahmed Zewail, M. Dantus, R.M. Bowman, and A. Mokhtari. Femtochemistry: recent advances and extension to high pressures. *Journal of Photochemistry and Photobiology A: Chemistry*, 62(3):301–319, 1992. ISSN 1010-6030. doi: https://doi.org/10.1016/1010-6030(92)85061-X. URL https://www.sciencedirect.com/science/article/pii/101060309285061X.

تم بحمد الله