LLM Powered Assistant to Facilitate the Study of Academic Articles

Ana Negru, 12628859

Assignment intro:

This assignment entailed building an assistant to understand, learn and study the content of web pages related to academic articles for the purpose of facilitating learning about subfields of AI, such as NLP. The assistant leverages capabilities of the large language model (LLM) GPT3.5 in order to allow users to input a url belonging to an academic article and then be able to ask questions to the assistant to explain or clarify specific parts of the article. The assistant can also provide summaries or explain concepts using simpler language. All information that is explained is taken directly from the article.

Many courses that students take require us to read academic articles, and even if they are not mandatory as a part of the syllabus, many professors do share links to these articles as supplementary resources in case we would like to go deeper into these topics out of our own interest. Such papers can be time-consuming to read, can have convoluted language making them difficult to interpret, or require a base of technical knowledge around the topic. Hence, it would be of use to have an agent that can explain and interpret the content of such articles in a clear, concise and easily digestible manner. This aspect of providing easily understandable answers was incorporated in the prompt template for the assistant.

This assistant app also serves as reinforcement towards the idea that large language models can be used as a valuable tool to accelerate and facilitate learning instead of hindering it, as the services provided by such agents can be leveraged to help students gain understanding around course materials.

Linguistic meaning of the academic article assistant app:

This academic webpage article app is linguistically meaningful as it is designed to analyze and understand/interpret natural language. It utilizes a combination of NLP techniques and tools including document loading, text splitting, embeddings, similarity search, large language models and prompt engineering. This is all done with the aim of helping users understand and learn the content of academic papers related to the topic of natural language processing. The components of this app work together to give computers the ability to understand text in a similar way that humans can by combining computational linguistics with statistical methods. By processing natural language, the assistant is able to answer questions, provide explanations and offer insights in a way that is more accessible and informative than a simple keyword (ctrl f) search in a pdf would. This linguistic understanding further allows the app to tailor its responses to the user's specific question and assumes a lower level of understanding by default, resulting in a more effective and personalized learning experience. The tasks that the app accomplishes incorporate 3 important facets of NLP: Representing input (article from the web page serves as input), generating output (creating responses based on input data & queries) and computational modeling (understanding language structure and use).

Quickstart guide:
Included in the code is a short quickstart guide to aid in familiarization with the Langchain library used for this project as well as NLP concepts that are integrated within this framework. Examples are provided to demonstrate how certain functions work.

First of all, Langchain supports a variety of LLMs, such as Hugging Face Hub, as well as OpenAI, which was used in this assignment. For this we load the OpenAI model from the langchain library (which needs to be accessed and initialized with an API key), and provide it with a prompt. This is the starting point in order to further interact with the language model.

Next, there is a prompts module, which can be used to manage, optimize and serialize prompts. The prompt should be provided with an input variable and a template, to ask for user information or get some kind of variable information and input it into a specific prompt. The way it works in practice is similar to how f-strings in python are used.

The third component is memory– we can provide the app with both long-term and short-term memory, essentially to make it smarter so it does not forget previous interactions with the user. Here you can import ConversationChain and provide it with the model, followed by calling the .predict() method on the conversation. Predictions can be added to the output, which can change based on user input.

Indexes essentially allow us to build smart applications for a wide array of purposes, using our own data, and include document loaders, textsplitters, vectorstores and retrievers.

Another core component of Langchain are chains, which is where things start to come together. Up until now, the previously mentioned aspects of LangChain such as models, prompts, memory etc are not necessarily new phenomena, and are encountered in more basic user interactions with Chatgpt. The chains go further than one single LLM call and can be used to put together a sequence of calls as well as integrate other tools. For instance, with this we can chain together the model (which makes a connection to the API) and the memory & prompts. The model and prompt are provided as input parameters for the chain.

The last component is agents, which involve an LLM making decisions about actions to take, taking the actions and applying them on observations. Within the agents we have tools such as a Wikipedia API or Pandas dataframe agent. These agents will use the LLMs to assess which of these tools to use and will subsequently use them, returning with the relevant information.

Below are screenshot examples demonstrating the output in relation to the quickstart guide:

Defining the LLM & setting the prompt to “Write a haiku about natural language processing”:

```
NLP, so vast and wide
A tool for understanding
The human language's tides
(base) valentinnegru@Anas-macbook A3 %
```

Prompts, memory & chains:
The first few lines are a general prompt which is engineered within the library. It demonstrates human interaction with the AI and what kind of responses one might expect.
The input to the next conversation by the human was “I’m doing well! The weather is great and I’m having a conversation with an AI”.

Continuing with the code and going through to the chain step, the input variable was {product}, and the template was made to find a good name for a company that makes the input variable. Chaining this together and asking it to come up with a name for the app created in this assignment (an AI assistant to help with studying the content of web-pages related to academic articles) gives the output AI Scholarly Analyzer. This demonstrates generative capabilities of the AI.

```
> Entering new ConversationChain chain...
Prompt after formatting:
The following is a friendly conversation between a human and an AI. The AI is talkative and provides lots of specific details from its context. If the AI does not know
the answer to a question, it truthfully says it does not know.

Current conversation:

Human: Hi there!
AI:

> Finished chain.
 Hi there! I'm an AI. It's great to meet you. What can I do for you?


> Entering new ConversationChain chain...
Prompt after formatting:
The following is a friendly conversation between a human and an AI. The AI is talkative and provides lots of specific details from its context. If the AI does not know
the answer to a question, it truthfully says it does not know.

Current conversation:
Human: Hi there!
AI:  Hi there! I'm an AI. It's great to meet you. What can I do for you?
Human: I'm doing well! The weather is great and I'm having a conversation with an AI.
AI:

> Finished chain.
 That's wonderful! I'm so glad to hear that you're doing well and that you're enjoying the weather. It's a great day for a conversation. What would you like to talk abo
ut?


AI Scholarly Analyzer.
```

Agents:
When initializing the agent, it was given access to a list of tools, and it can then choose which ones to use to arrive at the final answer. It uses the agent type ZERO_SHOT_REACT_DESCRIPTION to do this and thus automatically selects the appropriate tool(s) from the given set to answer the question posed in the input prompt.
With the query: “When was python released, who was the creator, multiply the year by 3”, we get the following response:

```
Current conversation:
...
Final Answer: Python was released in 1991 and was created by Guido van Rossum, and 1991 multiplied by 3 is 5973.

> Finished chain.
Python was released in 1991 and was created by Guido van Rossum, and 1991 multiplied by 3 is 5973.
```

With the query: “In what year was the movie The Shining released and who was the director? Multiply the number corresponding to the year that The Shining was released in by 5”

```
> Entering new AgentExecutor chain...
 I need to know the year the movie The Shining was released and who the director was
Action: Wikipedia
Action Input: "The Shining"/Users/valentinnegru/opt/anaconda3/lib/python3.9/site-packages/wikipedia/wikipedia.py:389: GuessedAtParserWarning: No parser was explicitly s
pecified, so I'm using the best available HTML parser for this system ("lxml"). This usually isn't a problem, but if you run this code on another system, or in a differ
ent virtual environment, it may use a different parser and behave differently.

The code that caused this warning is on line 389 of the file /Users/valentinnegru/opt/anaconda3/lib/python3.9/site-packages/wikipedia/wikipedia.py. To get rid of this w
arning, pass the additional argument 'features="lxml"' to the BeautifulSoup constructor.

  lis = BeautifulSoup(html).find_all('li')

Observation: Page: The Shining (film)
Summary: The Shining is a 1980 psychological horror film produced and directed by Stanley Kubrick and co-written with novelist Diane Johnson. The film is based on Steph
en King's 1977 novel of the same name and stars Jack Nicholson, Shelley Duvall, Scatman Crothers, and Danny Lloyd. The film's central character is Jack Torrance (Nichol
son), an aspiring writer and recovering alcoholic who accepts a position as the off-season caretaker of the isolated historic Overlook Hotel in the Colorado Rockies, wi
th his wife, Wendy Torrance (Duvall), and young son, Danny Torrance (Lloyd). Danny is gifted with psychic abilities named "shining". After a winter storm leaves the Tor
rances snowbound, Jack's sanity deteriorates due to the influence of the supernatural forces that inhabit the hotel.
Production took place almost exclusively at EMI Elstree Studios, with sets based on real locations. Kubrick often worked with a small crew, which allowed him to do many
 takes, sometimes to the exhaustion of the actors and staff. The new Steadicam mount was used to shoot several scenes, giving the film an innovative and immersive look
and feel. There has been much speculation about the meanings and actions in the film because of inconsistencies, ambiguities, symbolism, and differences from the book.
The film was released in the United States on May 23, 1980, and in the United Kingdom on October 2 by Warner Bros. There were several versions for theatrical releases,
each of which was cut shorter than the preceding cut; about 27 minutes was cut in total. Reactions to the film at the time of its release were mixed; Stephen King criti
cized the film due to its deviations from the novel. The film received two controversial nominations at the Razzies—Worst Director and Worst Actress (for Duvall)—the la
tter of which was later rescinded due to Kubrick's treatment of Duvall on set. Critical response to the film has since become more favorable.
In 2012, The Shining was ranked the 75th greatest film of all time in the Sight & Sound directors' poll. In 2018, the film was selected for preservation in the United S
tates National Film Registry by the Library of Congress as being "culturally, historically, or aesthetically significant". Thirty-nine years after the original film, a
sequel, Doctor Sleep, was released on November 8, 2019.



Page: The Shining (novel)
Summary: The Shining is a 1977 horror novel by American author Stephen King. It is King's third published novel and first hardback bestseller; its success firmly establ
ished King as a preeminent author in the horror genre. The setting and characters are influenced by King's personal experiences, including both his visit to The Stanley
 Hotel in 1974 and his struggle with alcoholism. The novel was adapted into a 1980 film and a 1997 miniseries. The book was followed by a sequel, Doctor Sleep, publishe
d in 2013, which in turn was adapted into a film of the same name in 2019.
The Shining centers on Jack Torrance, a struggling writer and recovering alcoholic who accepts a position as the off-season caretaker of the historic Overlook Hotel in
the Colorado Rockies. His family accompanies him on this job, including his young son Danny, who possesses "the shining", an array of psychic abilities that allow the c
hild to glimpse the hotel's horrific true nature. Soon, after a winter storm leaves the family snowbound, the supernatural forces inhabiting the hotel influence Jack's
sanity, leaving his wife and son in grave danger.
Thought: I now know the final answer
Final Answer: The Shining was released in 1980 and directed by Stanley Kubrick. The number corresponding to the year The Shining was released, 1980, multiplied by 5 is
10,000.

> Finished chain.
The Shining was released in 1980 and directed by Stanley Kubrick. The number corresponding to the year The Shining was released, 1980, multiplied by 5 is 10,000.
```

We can see the AI taking in the task “The Shining” as input and using the source Wikipedia to find out the information. It returns 2 observations related to 2 pages. It is possible that it is generating 2 different outputs based on the input it received, indicating that the AI is recognizing the input, “The Shining” as either related to a film or a novel and is generating different outputs accordingly. This is true because The Shining is indeed both a film and a novel. In the end, the AI distinguishes between the two and picks the film, that the director is Stanley Kubrick and the release year was 1980. However, there is an error in the calculation when the year 1980 is multiplied by 5. The answer it is giving says 10 000, but the correct answer of 1980*5 = 9 900. This may be due to a malfunction in the calculator tool or wrong values being applied.

Langchain & coding process:

For the purpose of this project, the open source framework Langchain was utilized in order to combine the LLM GPT-3 with an external source of data- in this case webpages pertaining to academic articles. This is different from pasting a snippet of a text document into a chatgpt prompt, as it is about referencing a constructed database filled with the data of our choosing.

The main value proposition of Langchain is divided into 3 concepts: Components, chains and agents. For components, there are LLM wrappers that allow us to connect to large language models like GPT-3, prompt templates allow us to avoid having to hard code text as the input to LLMs. There are also indexes, which allow us to extract relevant information. Secondly, chains facilitate the combination of multiple components together to solve a specific task and build an LLM application. Finally there are agents, which allow LLMs to interact with their environment, e.g. make API requests with a specific action.

Once the information needed is obtained, Langchain is used to power the action we want to take. It works by taking the document that we want our language model to reference, slicing it up into smaller chunks which are stored in a vector database. The chunks are stored as embeddings, meaning they are vector representations of the text. This allows us to build language model applications which follow the following general pipeline: A user asks an initial question. This question is then sent to the language model and a vector representation of that question is used to do a similarity search in the vector database, which allows us to fetch the relevant chunks of information from the vector database and feed it to the language model. Now the language model has both the initial question and the relevant information from the vector database and is capable of providing an answer and/or taking an action. These applications are both data aware (we can reference our own data in a vector store) and agentic– can take actions.

The Langchain library tools that were used for this project include document loaders, text splitters and vector stores. Firstly, a function to create a database out of a web url was defined using the document loader WebBaseLoader, and then a transcript was obtained given a website url input. However, looking at the GPT-3 API documentation for the gpt-3.5-turbo model handles 4096 tokens as input, and the tokens of the transcript were much larger than that amount. To overcome this, text_splitter was used to split the transcript up in chunk sizes of 1000 characters each. This achieved a list containing documents split up based on character number. In order to provide it to the API, the embeddings from OpenAI were used to convert the text splits that were just created into vectors– a numerical representation of the text itself. Afterwards the FAIS library was used for a similarity search. This was combined to create the database of all the documents, and when a user wants to ask a question with regards to the transcript, a similarity search is first performed to find the chunks that are most similar to the prompt that the user is asking. In summary, we have this database with the vectors, and a similarity search is done on them to find the relevant pieces of information needed for the specific query at hand. This is brought together with the function create_db.

Next, this is provided to another function, query_response, where the database previously created is used to answer specific questions. The database and the query is provided to the function, the parameter k defaults to 4 in order to maximize the amount of tokens sent to the API. A similarity search is performed on the database using the query, and k documents are returned. Given the question, it goes through all the documents created from the initial input transcript, and does a similarity search. Once all the documents are obtained, they are joined into one single string. Then a model is created (with gpt-3.5-turbo). Afterwards, a template for the prompt is defined for the specific task at hand, providing the input parameter docs.

The next step is to chain everything together with LLMChain using a chatmodel ChatOpenAI. Here there is a system message prompt as well as a human message prompt. Firstly, the system message prompt explains what the agent should do, and a prompt to alter the question/input that the human is using follows. These are combined into a chat_prompt, which is put into a chain with chat and chat_prompt. The response to that contains the query and previously defined docs.

Prompt engineering:

As explained previously, the code written to generate this academic article webpage assistant demonstrates how to use a language model to output responses based on a user's query and set of web pages related to a specific topic. It also incorporates several techniques and tools used in prompt engineering, in which prompts are optimized and designed for NLP models. The user's query is first used to search for relevant documents based on their content. Based on the content of the documents, a response to the user's query using the language model ChatOpenAI is generated. The response is created using a pre-defined template for the system message prompt as well as the human message prompt, which are combined in order to create a chat prompt template. This prompt template is then used to return the response utilizing LLMChain, producing responses based on a prompt.

The template for the system message prompt demonstrates several important aspects related to prompt engineering, and provides context and guidelines for the language model to generate relevant responses. First of all, the template defines the task of the assistant, i.e answer questions about web pages based on content. Then it defines the type of content it's working with (AI & NLP related) to provide the model context, as well as the criteria for generating responses. The goal of the task is also clearly outlined in order to emphasize the importance of the role of the prompt so that the assistant has background knowledge on how it should answer as well as its purpose. In this case it's explained that the role of the assistant is to explain complex academic concepts from articles in a clear and easily digestible way. Lastly, the prompt also provides guidance on how the response should be structured and presented, referring to the fact that information should be expressed clearly, assuming that the user does not have much domain knowledge/expertise in the area.

Screenshots & example scenarios:

As we have been assigned/given out readings for this NLP course, the assistant will be applied to 3 of the academic papers that have been referenced to us, and we will see how well it works in providing questions to help us learn and understand the content of these articles.

BERT Rediscovers the Classical NLP Pipeline (Tenney & Pavlick, 2019):

Input prompt: “What are the 3 main conclusions to be drawn from this article?”

```
The article discusses the BERT model and aims to
identify where linguistic information is captured
within the network. The three main conclusions
drawn from the article are:1. The BERT model
represents the steps of the traditional NLP
pipeline in an interpretable and localizable
way.2. The regions responsible for each step of
the pipeline appear in the expected sequence: POS
tagging, parsing, NER, semantic roles, then
coreference.3. The model can dynamically adjust
the pipeline by revising lower-level decisions
based on disambiguating information from higher-
level representations. Overall, the article
provides insights into the inner workings of the
BERT model and its ability to capture linguistic
information in an interpretable and dynamic way.
```

Input prompt: “Explain the 5 regions responsible for each step of the pipeline using language that is easy to understand.”

```
The article discusses a model called BERT that is
used for natural language processing. The model
represents the steps of the traditional NLP
pipeline in an interpretable and localizable way.
The regions responsible for each step appear in
the expected sequence: 1. POS tagging: This is
where the model identifies the parts of speech of
each word in a sentence. For example, it can
identify whether a word is a noun, verb,
adjective, etc.2. Parsing: This is where the model
analyzes the grammatical structure of a sentence.
It can identify the subject, verb, and object of a
sentence, as well as the relationships between
different parts of the sentence.3. NER: This
stands for Named Entity Recognition. This is where
the model identifies and categorizes named
entities in a sentence, such as people,
organizations, and locations.4. Semantic roles:
This is where the model identifies the roles that
different words play in a sentence. For example,
it can identify the agent of an action, the
patient of an action, and the instrument used to
perform an action.5. Coreference: This is where
the model identifies when two or more words in a
sentence refer to the same entity. For example, if
a sentence says "John went to the store. He bought
some milk", the model can identify that "he"
refers to John. Overall, the model uses these
regions to analyze and understand the meaning of
sentences in a way that is similar to how humans
process language.
```

As we can see, questions that we ask can build off of eachother. For instance, if we initially were aiming to obtain a high-level summary of the article content with the 3 main ideas presented but then we want more information on main idea #2 relating to the different steps in the pipeline we can make a query relating to that. It is also possible to play around with the queries and what we’re asking. For instance if we’re just starting out learning about NLP and aren’t already familiar with concepts, that can be incorporated in the query and we can ask for the concepts to be explained using more simple language and the assistant will adapt to the query request. Additionally, the assistant provides simple examples to explain each part of the pipeline, further enhancing explainability.

Gender bias in Word Embeddings: A Comprehensive Analysis of Frequency, Syntax, and Semantics (Caliskan et al., 2022):

In this toy setting we know that the article is based on gender bias within word embeddings and would like a high-level overview/summary of the main things the article brings up in relation to this topic, therefore the query "Give me a short and simple summary on what this article says about AI bias related to gender in word embeddings" yields the following output:

```
This article provides a comprehensive analysis of
gender bias in word embeddings, which are a type
of artificial intelligence algorithm used in
natural language processing. The study found that
word embeddings tend to reinforce gender
stereotypes and biases, such as associating
certain professions with one gender over the
other. The article highlights the importance of
addressing and mitigating these biases in AI
algorithms to ensure fair and equitable outcomes.
```

Now for instance we see that the last part of the summary points to the fact that the article focuses on how important it is to mitigate such biases. If we would want to know how the article specifies that this should be done, we can ask the assistant with the following query: "What can be done to address and mitigate these biases in AI in order to ensure fair and equitable outcomes?" which generates this output:

```
The article mentions several approaches that can
be taken to address and mitigate biases in AI and
ensure fair and equitable outcomes. One approach
is to use word embeddings that are trained on more
diverse and representative data sets. This can
help to reduce the impact of biases that may be
present in the data. Another approach is to use
techniques from psycholinguistics to identify and
measure biases in language use, and to develop
methods for correcting these biases. Additionally,
it may be helpful to consider the representation
of gender and other social categories in AI
systems, and to ensure that these systems are
designed to be inclusive and equitable. Finally,
it is important to recognize that biases in AI are
often the result of broader societal biases, and
that addressing these biases will require a
concerted effort across multiple domains,
including education, policy, and culture.
```

Attention Is All You Need (Vaswani et al., 2017) :

Imagine a setting in which we want to learn specifically about the architecture of neural sequence transduction models. In the article, section 3 addresses this, but we want a shorter summary that explains all components described in that section in one place. We also want to limit the words because this section is pretty spread out and can be difficult to digest. This can be achieved by the following query: "I would like to learn about the architecture of neural sequence transduction models, explain it to me in no more than 300 words". This consequently produces the following summary:

```
Neural sequence transduction models are a type of
neural network that is used to process sequential
data, such as natural language text. These models
are designed to take in a sequence of input data
and produce a corresponding sequence of output
data. The architecture of neural sequence
transduction models typically consists of an
encoder and a decoder. The encoder takes in the
input sequence and produces a fixed-length
representation of it, which is then fed into the
decoder. The decoder then generates the output
sequence based on the encoder's representation and
any previous output it has generated.One common
type of neural sequence transduction model is the
sequence-to-sequence (seq2seq) model. In a seq2seq
model, the encoder and decoder are typically
implemented using recurrent neural networks
(RNNs), such as long short-term memory (LSTM) or
gated recurrent unit (GRU) networks. These
networks are designed to handle sequential data by
maintaining a hidden state that captures
information about the previous inputs.During
training, the model is typically optimized using a
loss function that measures the difference between
the predicted output sequence and the true output
sequence. This loss function is backpropagated
through the model to update the weights and
improve the model's performance.Overall, neural
sequence transduction models are a powerful tool
for processing sequential data, and have been used
in a wide range of applications, including machine
translation, speech recognition, and text
summarization.
```

If we wanted to study that part of the article and use the assistant to aid us in making us practice/mock up questions that take into account that section of the article, we can do so with the following prompt: “Can you make a quiz consisting of 10 questions for me to study related to sequence transduction model architecture?” which yields the following output:

```
quiz.Sequence transduction model architecture is a
type of neural network architecture used in
natural language processing tasks such as machine
translation and speech recognition. It involves
mapping an input sequence to an output sequence,
where the input and output sequences can have
different lengths. Here are some potential quiz
questions related to sequence transduction model
architecture:1. What is sequence transduction
model architecture used for?2. What is the
difference between the input and output sequences
in sequence transduction model architecture?3.
What is the purpose of the encoder in sequence
transduction model architecture?4. What is the
purpose of the decoder in sequence transduction
model architecture?5. What is attention in
sequence transduction model architecture?6. What
is the difference between global and local
attention in sequence transduction model
architecture?7. What is beam search in sequence
transduction model architecture?8. What is the
purpose of teacher forcing in sequence
transduction model architecture?9. What is the
purpose of scheduled sampling in sequence
transduction model architecture?10. What are some
potential applications of sequence transduction
model architecture beyond natural language
processing?
```

The prompts that have been input for the 3 articles focused on seeing how the AI summarizes the content, simplifies its answers, adapts to constraints set on it (e.g. limit the number of words), how subsequent questions can build on previously obtained information and finally, creating quizzes based on the content referenced in the prompt. This demonstrates that the AI assistant built for this assignment can be used as a viable tool to extract specific use-case information from academic articles, and can even be utilized dynamically in order to help us learn the content it explains.

Ethical considerations:

Given the rapid advancement of LLMs and how quickly they become so widespread, the scope of ethical considerations in relation to this topic has broadened significantly. A wide array of relevant digital ethics and responsible AI topics are encompassed, such as bias, fairness, privacy, transparency, etc.

First to consider is data privacy, as virtual assistants like the one in this assignment can rely on personal data to function effectively. This can include information about a user's preferences, behavior, location and needs (can be inferred from the prompts which users input). Hence it is crucial to ensure that the data is collected and stored securely, and that users are informed about how their data is being used.

Another aspect to consider is bias and fairness. Generally if data is biased or not fully representative of the subject it depicts, it can lead to unjust or discriminatory outcomes. This aspect is not as applicable to the academic paper assistant developed in this assignment, as the main purpose is solely to use the data coming from a specific article that the user wishes to look into. By doing so, the scope of this assignment inherently allows bias towards the concepts represented in that particular article. However, the topic of bias is still worth mentioning and bearing in mind nonetheless.

Further aspects concerning ethics and virtual assistants would be transparency and explainability, as such systems can be opaque regarding the way in which they arrive at their responses or recommendations. Therefore, it is important to ensure that these systems, particularly with regards to the agents, are transparent in terms of how decisions are made and that users can understand/challenge the meaning behind the decisions.

References:

Caliskan, A., Ajay, P. P., Charlesworth, T., Wolfe, R., & Banaji, M. R. (2022, July). Gender bias in word embeddings: a comprehensive analysis of frequency, syntax, and semantics. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society* (pp. 156-170).

Tenney, I., Das, D., & Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. *arXiv preprint arXiv:1905.05950*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.