

# CS410 FALL 2023 PROJECT Testing Data

Textual Spotlight is a Chrome extension designed to enhance the browsing experience by leveraging the capabilities of large language models (LLMs) for contextual understanding of webpages text.

The testing data for this project compares the performances of several leading LLMs—specifically OpenAI GPT-4 TURBO, Cohere Command, and LLama 13B Chat. The primary focus of this comparative analysis is to examine how each model responds to the same user input and to evaluate the response's relevancy.

## Action: EXPLAIN – Test Case #1

Input (Highlighted Text)	Output	Model	Cosine Similarity
<p>A large language model (LLM) is a deep learning algorithm that can perform a variety of natural language processing (NLP) tasks. Large language models use transformer models and are trained using massive datasets — hence, large. This enables them to recognize, translate, predict, or generate text or other content.</p> <p>Large language models are also referred to as neural networks (NNs), which are computing systems inspired by the human brain. These neural networks work using a network of nodes that are layered, much like neurons.</p>	<p>A large language model (LLM) is a deep learning algorithm that can perform natural language processing (NLP) tasks. It uses transformer models and is trained with massive datasets, enabling it to recognize, translate, predict, or generate text or other content. Large language models are also known as neural networks (NNs), which are computing systems inspired by the human brain, working with layered nodes, similar to neurons.</p>	OpenAI - GPT4 TURBO	99.54%
	<p>A large language model is a powerful tool powered by deep learning algorithms and neural networks. These models use transformer models and are trained on huge datasets, making them capable of various natural language processing tasks like text recognition, translation, prediction, and generation. Their structure, inspired by the human brain, allows them to process information in a way that resembles how humans think and communicate."</p>	Cohere - Command	97.47%

	A large language model (LLM) is a type of deep learning algorithm that can do many natural language processing tasks, like recognizing, translating, predicting, or generating text. It uses something called a transformer model and is trained on huge amounts of data, earning its name "large". Think of it like a computer program that mimics how our brains process language!	LLama - 13B chat	85.63%
--	--	------------------	--------

Highlighted Text link reference: [What is a Large Language Model? | A Comprehensive LLMs Guide | Elastic.](#)

## Action: EXPLAIN – Test Case #2

Input (Highlighted Text)	Output	Model	Cosine Similarity
<p>Python supports optional type annotations[4] (the annotations/hints are not "mandatory, even by convention"[102]). That said, the support for these optional annotations is (currently) non-existent in the (default/reference) language implementation (CPython),[103] except for the parser reading them and overlooking. Since the annotations are there in the source code, they can be described as hints to the programmer only, for readability, while non-default implementations, or other tools, can respect them as more than hints. This optional Python's syntax allows specifying static types, and an experimental optional static type-checker, mypy, supports compile-time type checking, and the "aim is to support almost all Python language constructs in mypy."</p>	<p>Python supports optional type annotations, which are hints to the programmer for readability. These annotations are not mandatory and are currently not supported in the default implementation of Python (CPython), except for the parser reading them. However, non-default implementations or other tools can respect these annotations as more than hints. Python's optional syntax allows specifying static types, and a type-checker called mypy supports compile-time type checking. The goal of mypy is to support almost all Python language constructs.</p>	OpenAI - GPT4 TURBO	97.91%
	<p>The optional type annotations in Python may not be widely supported in the language's default implementation, but they serve as important hints to programmers to improve code readability. These annotations can specify static types and are intended to be respected by tools other than the default interpreter. One such tool is mypy, an optional static type checker for Python, which aims to support almost all Python language constructs and enables compile-time type checking to help catch type-related errors before runtime.</p>	Cohere - Command	97.95%
	<p>In Python, type annotations are optional and do not have to be included in every function or class definition. These annotations serve as hints for the programmer, providing information about the expected data types of variables. While CPython, the default Python interpreter, does not currently support type annotations, other tools and non-default implementations may use them to perform static type checking. Additionally, the experimental static type checker mypy supports compile-time type checking and aims to support most Python language constructs.</p>	LLama - 13B chat	69.62%

Highlighted Text link reference: [Python \(programming language\) - Wikipedia](#)

## Action: SUMMARIZE – Test Case #1

Input (Highlighted Text)	Output	Model	Cosine Similarity
<p>A large language model (LLM) is a deep learning algorithm that can perform a variety of natural language processing (NLP) tasks. Large language models use transformer models and are trained using massive datasets — hence, large. This enables them to recognize, translate, predict, or generate text or other content.</p> <p>Large language models are also referred to as neural networks (NNs), which are computing systems inspired by the human brain. These neural networks work using a network of nodes that are layered, much like neurons.</p>	<p>A large language model (LLM) is a deep learning algorithm used for natural language processing (NLP) tasks. It uses transformer models and massive datasets to recognize, translate, predict, or generate text. LLMs are also called neural networks (NNs) and are inspired by the human brain with layered nodes.</p>	OpenAI - GPT4 TURBO	99.54%
	<p>Large language models, also known as neural networks, are deep learning algorithms that utilize transformer models and vast datasets to perform various natural language processing tasks. These tasks include text recognition, translation, prediction, and generation. Neural networks are structured like the human brain, consisting of a network of interconnected nodes layered atop each other.</p>	Cohere - Command	96.74%
	<p>Large language models (LLMs) are deep learning algorithms that perform various NLP tasks through transformer models and massive dataset training. They can recognize, translate, predict, or generate text/content, similar to neural networks (NNs) that mimic the human brain with layered nodes.</p>	LLama - 13B chat	85.63%

Highlighted Text link reference: [What is a Large Language Model? | A Comprehensive LLMs Guide | Elastic.](#)

## Action: SUMMARIZE – Test Case #2

Input (Highlighted Text)	Output	Model	Cosine Similarity
<p>Python supports optional type annotations[4] (the annotations/hints are not "mandatory, even by convention"[102]). That said, the support for these optional annotations is (currently) non-existent in the (default/reference) language implementation (CPython),[103] except for the parser reading them and overlooking. Since the annotations are there in the source code, they can be described as hints to the programmer only, for readability, while non-default implementations, or other tools, can respect them as more than hints. This optional Python's syntax allows specifying static types, and an experimental optional static type-checker, mypy, supports compile-time type checking, and the "aim is to support almost all Python language constructs in mypy."</p>	<p>Python supports optional type annotations, but the default implementation of the language (CPython) does not fully support them. However, these annotations can be helpful hints for programmers, and alternative implementations or tools may respect them more strongly. The experimental static type-checker mypy aims to support most Python language constructs.</p>	OpenAI - GPT4 TURBO	96.50%
	<p>The text discusses Python's support for optional type annotations, which are not mandatory even by convention, and currently have limited support in the default language implementation (CPython). However, these annotations can serve as hints to programmers for improving readability, and tools such as non-default implementations or the experimental static type-checker mypy can utilize them for static type-checking. mypy aims to support almost all Python language constructs in its type-checking functionality.</p>	Cohere - Command	97.96%
	<p>Python has optional type annotations that can improve code readability but are not strictly enforced by default language implementation CPython. Third-party tools like mypy can provide compile-time type checking for added stability.</p>	LLama - 13B chat	78.50%

Highlighted Text link reference: [Python \(programming language\) - Wikipedia](#)

## Action: PARAPHRASE – Test Case #1

Input (Highlighted Text)	Output	Model	Cosine Similarity
<p>A large language model (LLM) is a deep learning algorithm that can perform a variety of natural language processing (NLP) tasks. Large language models use transformer models and are trained using massive datasets — hence, large. This enables them to recognize, translate, predict, or generate text or other content.</p> <p>Large language models are also referred to as neural networks (NNs), which are computing systems inspired by the human brain. These neural networks work using a network of nodes that are layered, much like neurons.</p>	<p>A large language model is an advanced AI algorithm capable of various natural language processing tasks. It utilizes transformer models and is trained on extensive datasets to recognize, translate, predict, or generate textual content. These models are often called neural networks as they are inspired by the human brain, functioning as a network of layered nodes.</p>	OpenAI - GPT4 TURBO	96.06%
	<p>The large language model (LLM) is a powerful deep learning algorithm that handles various natural language processing (NLP) tasks. This is achieved through the use of transformer models and colossal training datasets, which allow LLMs to perform tasks like text generation, prediction, recognition, translation, or content generation.</p> <p>Neural networks (NNs) are another name for large language models, and they are computing systems designed to mimic the human brain. These NNs operate via a node network layered similarly to the neurons in the human brain.</p>	Cohere - Command	99.25%
	<p>Deep learning algorithms known as large language models (LLMs) excel at natural language processing tasks like translation, prediction, and content generation. These powerful models are trained on vast datasets and mimic the structure of the human brain through interconnected nodes, making them highly effective at understanding and generating text.</p>	LLama - 13B chat	85.63%

Highlighted Text link reference: [What is a Large Language Model? | A Comprehensive LLMs Guide | Elastic.](#)

## Action: PARAPHRASE – Test Case #2

Input (Highlighted Text)	Output	Model	Cosine Similarity
<p>Python supports optional type annotations[4] (the annotations/hints are not "mandatory, even by convention"[102]). That said, the support for these optional annotations is (currently) non-existent in the (default/reference) language implementation (CPython),[103] except for the parser reading them and overlooking. Since the annotations are there in the source code, they can be described as hints to the programmer only, for readability, while non-default implementations, or other tools, can respect them as more than hints. This optional Python's syntax allows specifying static types, and an experimental optional static type-checker, mypy, supports compile-time type checking, and the "aim is to support almost all Python language constructs in mypy."</p>	<p>Python offers the ability to include optional type annotations, but they are not mandatory and currently not fully supported in Python's default implementation (CPython). These annotations serve as hints to the programmer for readability purposes, while alternative implementations or tools may treat them as more than hints. A specific optional static type-checker called mypy aims to support comprehensive compile-time type checking for almost all Python language constructs.</p>	OpenAI - GPT4 TURBO	96.90%
	<p>Python supports optional type annotations, although they are not mandatory by convention. The default language implementation, CPython, does not currently utilize these annotations, except to parse them. These annotations serve as hints to the programmer for improved readability, but they can be acted upon by non-default implementations or other tools. Python's syntax allows for the specification of static types, and an experimental type-checker, mypy, supports static type-checking. mypy aims to support almost all Python language constructs.</p>	Cohere - Command	98.67%
	<p>Python allows optional type hints (not mandatory, but useful for readability). Currently, CPython doesn't support these hints, but other tools may recognize them as more than just hints. The syntax enables specifying static types, and the experimental type-checker mypy offers compile-time type checking for most Python constructs.</p>	LLama - 13B chat	79.55%

Highlighted Text link reference: [Python \(programming language\) - Wikipedia](#)

## Conclusive Insights from Test Cases

**Variation in Model Performance:** The test cases demonstrate that different models have varying degrees of effectiveness in understanding and responding to the same input; this variation is quantitatively reflected in the cosine similarity scores.

For instance, one model may excel at providing explanations (higher cosine similarity scores in *"Explain"* actions) but may not perform as well in *"Paraphrasing"*.

**Impact of Action Type on Model Output:** Each action type—Explain, Summarize, Paraphrase—elicits different responses from the models; this is likely due to how each model has been trained and its inherent capabilities in handling specific types of language tasks.

The cosine similarity scores indicate that specific models may be better suited for summarization explanation, or vice versa, based on their architecture and training.

**Practical Application in Textual Spotlight:** The Textual Spotlight extension leverages these differences effectively by offering users a choice of models. Users can select the model that best suits their current needs based on the insights these test cases provide.

By calculating and presenting the cosine similarity scores, Textual Spotlight aids users in making an informed decision about which model's output is more semantically aligned with the original text for a given action.

To conclude, these test cases highlight the utility of Textual Spotlight in evaluating and comparing AI models based on their performance in real-world text-processing tasks. By leveraging embeddings and cosine similarity scores to quantitatively assess the model outputs, the extension clearly measures how closely each AI model aligns with the user's intent, as represented by the original text.