SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR ANALYZING LARGE LANGUAGE MODELS

CROSS REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit of U.S. Provisional Patent Application No. 63/567,494, filed on March 20, 2024, the disclosure of which is hereby incorporated by reference in its entirety.

BACKGROUND

1. Technical Field

[0002] This disclosure relates generally to large language models and, in non-limiting embodiments or aspects, to systems, methods, and computer program products for analyzing large language models.

2. Technical Considerations

[0003] Hallucinations by large language models (LLMs) is a limitation on reliably using such models in a production environment. Despite a widespread need to measure the likelihood of hallucination in LLM outputs, no universal, automated solution exists that works for all models in a black-box environment where the inner-architectural details of the model are unknown to the user (e.g., ChatGPT).

SUMMARY

[0004] According to non-limiting embodiments or aspects, provided is a system comprising: at least one processor configured to: generate a plurality of model queries based on a semantic input, each model query of the plurality of model queries comprising the semantic input and a different non-semantic input; input the semantic input into a language model, resulting in a first model output; perturb the language model with the plurality of model queries, resulting in a plurality of perturbed model outputs; determine a model harmonic value for the language model based on an angle between the first model output and the plurality of perturbed model outputs; and retrain or select the language model based on the model harmonic value satisfying at least one threshold.

[0005] In non-limiting embodiments or aspects, the different non-semantic input comprises a uniquely generated string for each model query of the plurality of model queries. In non-limiting embodiments or aspects, the at least one processor is further configured to: randomly generate the different non-semantic input for each model query. In non-limiting embodiments or aspects, the different non-semantic input is appended to an end of the semantic input for each model query of the plurality of

model queries. In non-limiting embodiments or aspects, retraining or selecting the language model based on the model harmonic value satisfying the at least one threshold comprises: determining that the model harmonic value satisfies a first threshold; and in response to determining that the model harmonic value satisfies the first threshold, automatically retrain the language model. In non-limiting embodiments or aspects, the at least one processor is further configured to: input the semantic input into at least one other language model, resulting in at least one second model output; perturb the at least one other language model with the plurality of model queries, resulting in a second plurality of perturbed model outputs; and determine at least one second model harmonic value for the at least one other language model based on an angle between the second model output and the second plurality of perturbed model outputs. In non-limiting embodiments or aspects, retraining or selecting the language model based on the model harmonic value satisfying the at least one threshold comprises: selecting a model from the language model and the at least one other language model based on the model harmonic value and the at least one second model harmonic value; and automatically implementing the model in a production environment in response to selecting the model.

[0006] According to non-limiting embodiments or aspects, provided is a method comprising: generating, with at least one processor, a plurality of model queries based on a semantic input, each model query of the plurality of model queries comprising the semantic input and a different non-semantic input; inputting, with at least one processor, the semantic input into a language model, resulting in a first model output; perturbing, with at least one processor, the language model with the plurality of model queries, resulting in a plurality of perturbed model outputs; determining, with at least one processor, a model harmonic value for the language model based on an angle between the first model output and the plurality of perturbed model outputs; and retraining or selecting the language model, with at least one processor, based on the model harmonic value satisfying at least one threshold.

[0007] In non-limiting embodiments or aspects, the different non-semantic input comprises a uniquely generated string for each model query of the plurality of model queries. In non-limiting embodiments or aspects, the method further comprises: randomly generating the different non-semantic input for each model query. In non-limiting embodiments or aspects, the different non-semantic input is appended to an end of the semantic input for each model query of the plurality of model queries. In

non-limiting embodiments or aspects, retraining or selecting the language model based on the model harmonic value satisfying the at least one threshold comprises: determining that the model harmonic value satisfies a first threshold; and in response to determining that the model harmonic value satisfies the first threshold, automatically retrain the language model. In non-limiting embodiments or aspects, the method further comprises: inputting the semantic input into at least one other language model, resulting in at least one second model output; perturbing the at least one other language model with the plurality of model queries, resulting in a second plurality of perturbed model outputs; and determining at least one second model harmonic value for the at least one other language model based on an angle between the second model output and the second plurality of perturbed model outputs. In non-limiting embodiments or aspects, retraining or selecting the language model based on the model harmonic value satisfying the at least one threshold comprises: selecting a model from the language model and the at least one other language model based on the model harmonic value and the at least one second model harmonic value; and automatically implementing the model in a production environment in response to selecting the model.

[0008] According to non-limiting embodiments or aspects, provided is a computer program product comprising at least one non-transitory computer-readable medium including program instructions which, when executed by at least one processor, cause the at least one processor to: generate a plurality of model queries based on a semantic input, each model query of the plurality of model queries comprising the semantic input and a different non-semantic input; input the semantic input into a language model, resulting in a first model output; perturb the language model with the plurality of model queries, resulting in a plurality of perturbed model outputs; determine a model harmonic value for the language model based on an angle between the first model output and the plurality of perturbed model outputs; and retrain or select the language model based on the model harmonic value satisfying at least one threshold.

[0009] In non-limiting embodiments or aspects, the different non-semantic input comprises a uniquely generated string for each model query of the plurality of model queries. In non-limiting embodiments or aspects, the at least one processor is further caused to: randomly generate the different non-semantic input for each model query. In non-limiting embodiments or aspects, the different non-semantic input is appended to an end of the semantic input for each model query of the plurality of model queries.

In non-limiting embodiments or aspects, retraining or selecting the language model based on the model harmonic value satisfying the at least one threshold comprises: determining that the model harmonic value satisfies a first threshold; and in response to determining that the model harmonic value satisfies the first threshold, automatically retrain the language model. In non-limiting embodiments or aspects, the at least one processor is further caused to: input the semantic input into at least one other language model, resulting in at least one second model output; perturb the at least one other language model with the plurality of model queries, resulting in a second plurality of perturbed model outputs; and determine at least one second model harmonic value for the at least one other language model based on an angle between the second model output and the second plurality of perturbed model outputs. In non-limiting embodiments or aspects, wherein retraining or selecting the language model based on the model harmonic value satisfying the at least one threshold comprises: selecting a model from the language model and the at least one other language model based on the model harmonic value and the at least one second model harmonic value; and automatically implementing the model in a production environment in response to selecting the model.

[0010] Further non-limiting embodiments or aspects are set forth in the following numbered clauses:

[0011] Clause 1: A system comprising: at least one processor configured to: generate a plurality of model queries based on a semantic input, each model query of the plurality of model queries comprising the semantic input and a different non-semantic input; input the semantic input into a language model, resulting in a first model output; perturb the language model with the plurality of model queries, resulting in a plurality of perturbed model outputs; determine a model harmonic value for the language model based on an angle between the first model output and the plurality of perturbed model outputs; and retrain or select the language model based on the model harmonic value satisfying at least one threshold.

[0012] Clause 2: The system of clause 1, wherein the different non-semantic input comprises a uniquely generated string for each model query of the plurality of model queries.

[0013] Clause 3: The system of clause 1 or 2, wherein the at least one processor is further configured to: randomly generate the different non-semantic input for each model query.

[0014] Clause 4: The system of any of clauses 1-3, wherein the different non-semantic input is appended to an end of the semantic input for each model query of the plurality of model queries.

[0015] Clause 5: The system of any of clauses 1-4, wherein retraining or selecting the language model based on the model harmonic value satisfying the at least one threshold comprises: determining that the model harmonic value satisfies a first threshold; and in response to determining that the model harmonic value satisfies the first threshold, automatically retrain the language model.

[0016] Clause 6: The system of any of clauses 1-5, wherein the at least one processor is further configured to: input the semantic input into at least one other language model, resulting in at least one second model output; perturb the at least one other language model with the plurality of model queries, resulting in a second plurality of perturbed model outputs; and determine at least one second model harmonic value for the at least one other language model based on an angle between the second model output and the second plurality of perturbed model outputs.

[0017] Clause 7: The system of any of clauses 1-6, wherein retraining or selecting the language model based on the model harmonic value satisfying the at least one threshold comprises: selecting a model from the language model and the at least one other language model based on the model harmonic value and the at least one second model harmonic value; and automatically implementing the model in a production environment in response to selecting the model.

[0018] Clause 8: A method comprising: generating, with at least one processor, a plurality of model queries based on a semantic input, each model query of the plurality of model queries comprising the semantic input and a different non-semantic input; inputting, with at least one processor, the semantic input into a language model, resulting in a first model output; perturbing, with at least one processor, the language model with the plurality of model queries, resulting in a plurality of perturbed model outputs; determining, with at least one processor, a model harmonic value for the language model based on an angle between the first model output and the plurality of perturbed model outputs; and retraining or selecting the language model, with at least one processor, based on the model harmonic value satisfying at least one threshold.

[0019] Clause 9: The method of clause 8, wherein the different non-semantic input comprises a uniquely generated string for each model query of the plurality of model queries.

[0020] Clause 10: The method of clause 8 or 9, further comprising: randomly generating the different non-semantic input for each model query.

[0021] Clause 11: The method of any of clauses 8-10, wherein the different non-semantic input is appended to an end of the semantic input for each model query of the plurality of model queries.

[0022] Clause 12: The method of any of clauses 8-11, wherein retraining or selecting the language model based on the model harmonic value satisfying the at least one threshold comprises: determining that the model harmonic value satisfies a first threshold; and in response to determining that the model harmonic value satisfies the first threshold, automatically retrain the language model.

[0023] Clause 13: The method of any of clauses 8-12, further comprising: inputting the semantic input into at least one other language model, resulting in at least one second model output; perturbing the at least one other language model with the plurality of model queries, resulting in a second plurality of perturbed model outputs; and determining at least one second model harmonic value for the at least one other language model based on an angle between the second model output and the second plurality of perturbed model outputs.

[0024] Clause 14: The method of any of clauses 8-13, wherein retraining or selecting the language model based on the model harmonic value satisfying the at least one threshold comprises: selecting a model from the language model and the at least one other language model based on the model harmonic value and the at least one second model harmonic value; and automatically implementing the model in a production environment in response to selecting the model.

[0025] Clause 15: A computer program product comprising at least one non-transitory computer-readable medium including program instructions which, when executed by at least one processor, cause the at least one processor to: generate a plurality of model queries based on a semantic input, each model query of the plurality of model queries comprising the semantic input and a different non-semantic input; input the semantic input into a language model, resulting in a first model output; perturb the language model with the plurality of model queries, resulting in a plurality of perturbed model outputs; determine a model harmonic value for the language model based on an angle between the first model output and the plurality of perturbed model outputs; and retrain or select the language model based on the model harmonic value satisfying at least one threshold.

[0026] Clause 16: The computer program product of clause 15, wherein the different non-semantic input comprises a uniquely generated string for each model query of the plurality of model queries.

[0027] Clause 17: The computer program product of clause 15 or 16, wherein the at least one processor is further caused to: randomly generate the different non-semantic input for each model query.

[0028] Clause 18: The computer program product of any of clauses 15-17, wherein the different non-semantic input is appended to an end of the semantic input for each model query of the plurality of model queries.

[0029] Clause 19: The computer program product of any of clauses 15-18, wherein retraining or selecting the language model based on the model harmonic value satisfying the at least one threshold comprises: determining that the model harmonic value satisfies a first threshold; and in response to determining that the model harmonic value satisfies the first threshold, automatically retrain the language model.

[0030] Clause 20: The computer program product of any of clauses 15-19, wherein the at least one processor is further caused to: input the semantic input into at least one other language model, resulting in at least one second model output; perturb the at least one other language model with the plurality of model queries, resulting in a second plurality of perturbed model outputs; and determine at least one second model harmonic value for the at least one other language model based on an angle between the second model output and the second plurality of perturbed model outputs.

[0031] Clause 21: The computer program product of any of clauses 15-20, wherein retraining or selecting the language model based on the model harmonic value satisfying the at least one threshold comprises: selecting a model from the language model and the at least one other language model based on the model harmonic value and the at least one second model harmonic value; and automatically implementing the model in a production environment in response to selecting the model.

[0032] These and other features and characteristics of the present disclosure, as well as the methods of operation and functions of the related elements of structures and the combination of parts and economies of manufacture, will become more apparent upon consideration of the following description and the appended claims with reference to the accompanying drawings, all of which form a part of this specification, wherein like reference numerals designate corresponding parts in the various figures. It is to be expressly understood, however, that the drawings are for the purpose of

illustration and description only and are not intended as a definition of the limits of the disclosed subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0033] Additional advantages and details are explained in greater detail below with reference to the non-limiting, exemplary embodiments that are illustrated in the accompanying schematic figures, in which:

[0034] FIG. 1 is a schematic diagram of a system for analyzing large language models, according to some non-limiting embodiments or aspects;

[0035] FIG. 2 is a flow diagram of a method for analyzing large language models, according to some non-limiting embodiments or aspects;

[0036] FIG. 3 is a schematic diagram of an electronic payment processing network used in connection with some non-limiting embodiments or aspects;

[0037] FIG. 4 is a schematic diagram of example components of one or more devices of FIG. 1, according to some non-limiting embodiments or aspects;

[0038] FIG. 5 illustrates example inputs and outputs for a system for analyzing large language models according to non-limiting embodiments or aspects;

[0039] FIG. 6 illustrates example inputs and outputs for a system for analyzing large language models according to non-limiting embodiments or aspects;

[0040] FIGS. 7-10 illustrate example inputs and outputs for a system for analyzing large language models according to non-limiting embodiments or aspects; and

[0041] FIG. 11 illustrates an example software architecture for a system for analyzing large language models according to non-limiting embodiments or aspects.

DETAILED DESCRIPTION

[0042] For purposes of the description hereinafter, the terms “end,” “upper,” “lower,” “right,” “left,” “vertical,” “horizontal,” “top,” “bottom,” “lateral,” “longitudinal,” and derivatives thereof shall relate to the embodiments as they are oriented in the drawing figures. However, it is to be understood that the present disclosure may assume various alternative variations and step sequences, except where expressly specified to the contrary. It is also to be understood that the specific devices and processes illustrated in the attached drawings, and described in the following specification, are simply exemplary and non-limiting embodiments or aspects of the disclosed subject matter. Hence, specific dimensions and other physical characteristics related to the embodiments or aspects disclosed herein are not to be considered as limiting.

[0043] Some non-limiting embodiments or aspects are described herein in connection with thresholds. As used herein, satisfying a threshold may refer to a value being greater than the threshold, more than the threshold, higher than the threshold, greater than or equal to the threshold, less than the threshold, fewer than the threshold, lower than the threshold, less than or equal to the threshold, equal to the threshold, etc.

[0044] No aspect, component, element, structure, act, step, function, instruction, and/or the like used herein should be construed as critical or essential unless explicitly described as such. Also, as used herein, the articles “a” and “an” are intended to include one or more items and may be used interchangeably with “one or more” and “at least one.” Furthermore, as used herein, the term “set” is intended to include one or more items (e.g., related items, unrelated items, a combination of related and unrelated items, and/or the like) and may be used interchangeably with “one or more” or “at least one.” Where only one item is intended, the term “one” or similar language is used. Also, as used herein, the terms “has,” “have,” “having,” or the like are intended to be open-ended terms. Further, the phrase “based on” is intended to mean “based at least partially on” unless explicitly stated otherwise. In addition, reference to an action being “based on” a condition may refer to the action being “in response to” the condition. For example, the phrases “based on” and “in response to” may, in some non-limiting embodiments or aspects, refer to a condition for automatically triggering an action (e.g., a specific operation of an electronic device, such as a computing device, a processor, and/or the like).

[0045] As used herein, the term “communication” may refer to the reception, receipt, transmission, transfer, provision, and/or the like of data (e.g., information, signals, messages, instructions, commands, and/or the like). For one unit (e.g., a device, a system, a component of a device or system, combinations thereof, and/or the like) to be in communication with another unit means that the one unit is able to directly or indirectly receive information from and/or transmit information to the other unit. This may refer to a direct or indirect connection (e.g., a direct communication connection, an indirect communication connection, and/or the like) that is wired and/or wireless in nature. Additionally, two units may be in communication with each other even though the information transmitted may be modified, processed, relayed, and/or routed between the first and second unit. For example, a first unit may be in communication with a second unit even though the first unit passively receives

information and does not actively transmit information to the second unit. As another example, a first unit may be in communication with a second unit if at least one intermediary unit processes information received from the first unit and communicates the processed information to the second unit. In some non-limiting embodiments or aspects, a message may refer to a network packet (e.g., a data packet and/or the like) that includes data. It will be appreciated that numerous other arrangements are possible.

[0046] As used herein, the term “computing device” may refer to one or more electronic devices configured to process data. A computing device may, in some examples, include the necessary components to receive, process, and output data, such as a processor, a display, a memory, an input device, a network interface, and/or the like. A computing device may be a mobile device. As an example, a mobile device may include a cellular phone (e.g., a smartphone or standard cellular phone), a portable computer, a wearable device (e.g., watches, glasses, lenses, clothing, and/or the like), a personal digital assistant (PDA), and/or other like devices. A computing device may also be a desktop computer or other form of non-mobile computer.

[0047] As used herein, the term “server” may refer to or include one or more computing devices that are operated by or facilitate communication and processing for multiple parties in a network environment, such as the Internet, although it will be appreciated that communication may be facilitated over one or more public or private network environments and that various other arrangements are possible. Further, multiple computing devices (e.g., servers, point-of-sale (POS) devices, mobile devices, etc.) directly or indirectly communicating in the network environment may constitute a “system.”

[0048] As used herein, the term “system” may refer to one or more computing devices or combinations of computing devices (e.g., processors, servers, client devices, software applications, components of such, and/or the like). Reference to “a device,” “a server,” “a processor,” and/or the like, as used herein, may refer to a previously-recited device, server, or processor that is recited as performing a previous step or function, a different device, server, or processor, and/or a combination of devices, servers, and/or processors. For example, as used in the specification and the claims, a first device, a first server, or a first processor that is recited as performing a first step or a first function may refer to the same or different device, server, or processor recited as performing a second step or a second function.

[0049] Non-limiting embodiments provide for systems and methods for analyzing large language models (LLMs), enabling for LLMs to be tested to, for example, trigger automatic retraining and/or selection of a model from a plurality of different models. Non-limiting embodiments provide for improved efficient use of computing resources by testing models while and/or before being implemented in a production environment without requiring human supervision. Moreover, by using unique and unconventional model inputs that include both semantic and non-semantic portions, non-limiting embodiments cause LLMs and other models to return unique and unconventional outputs, thereby leveraging existing models to produce outputs (e.g., such as data used to determine a harmonic value) that are not made available by such models. Other improvements and efficiencies will be appreciated by those skilled in the art.

[0050] Referring now to FIG. 1, shown is a system 1000 for analyzing large language models (LLMs) according to non-limiting embodiments or aspects. An LLM analysis system 100 may include one or more computing devices executing one or more software applications, as an example. In some non-limiting embodiments, the LLM analysis system may be integrated with another system, such as but not limited to a transaction processing system and/or the like. A query generator 104 may include one or more computing devices executing one or more software applications, as an example. In some examples, the query generator 104 may be a function and/or process of the LLM analysis system 100. LLM 114 and LLM 116 represent two different LLMs, although it will be appreciated that embodiments may use analyze a single or many LLMs. The LLMs 114, 116 may be local to the LLM analysis system 100 or may be remote systems that are interacted with through a network environment, such as through one or more application programming interfaces (APIs) exposed by the LLMs 114, 116.

[0051] With continued reference to FIG. 1, the query generator 104 may generate a plurality of model queries based on a semantic input and a different non-semantic input. For example, a semantic input may include a question (e.g., “what do people from Spain speak”, “what color is a strawberry”, and/or the like). Each different non-semantic input may include a random string of characters (e.g., “\x05”, “\x02\x06”, and/or the like). It will be appreciated that various different forms and lengths of non-semantic inputs may be used. Further, the non-semantic inputs may be randomly generated and/or selected from a repository or list of non-semantic inputs. In non-limiting embodiments, the non-semantic input may be appended to the end of each

semantic input (e.g., “what do people from Spain speak? \x05”). It will be appreciated that the non-semantic input may also be placed within the semantic input, before the semantic input, and/or any combination of like arrangements. The plurality of model queries may then have an identical or near-identical semantic portion and a different non-semantic portion (e.g., a first query may be “what do people from Spain speak? \x05”, a second query may be “what do people from Spain speak? \x02\x06”, etc.). Model queries 108 may also include a baseline query that has the sematic query without any non-semantic additions. Additionally or alternatively, the baseline query may be maintained separate from the model queries 108.

[0052] Still referring to FIG. 1, the plurality of model queries 108 may be input to one or more LLMs 114, 116 to perturb the LLMs with separate inputs. In embodiments in which multiple LLMs are queried, the same plurality of model queries 108 may be input to each LLM. The plurality of model queries may be transmitted to the LLMs 114, 116 in series, as a batch, and/or in any other manner. In response, the LLMs 114, 116 return model outputs 110, 112 for each query. The model outputs 110, 112 may be returned in series, as a batch, and/or in any other manner. The model outputs 110, 112 may be analyzed by the LLM analysis system 100 and/or stored in a query database 106, or a separate database, in association with an identifier of the LLM and the model queries 108 and/or the semantic portions thereof. A baseline query that includes the semantic query but does not include any non-semantic additions may also be input to the LLMs 114, 116 with or separately from the model queries 108. The model output of the baseline query may also be stored in association with the model queries 108.

[0053] The LLM analysis system 100 may determine a model harmonic value for each LLM 114, 116 based on comparing the model output of the baseline query for an LLM 114 to the model outputs 110 for the other model queries 108 processed by that same LLM 114. For example, the model outputs 110, 112 may be translated into a series of feature vectors or other representations to permit distance calculations in a multi-dimensional space. In embodiments in which multiple LLMs are queried, this may be done for each separate LLM 114, 116. In non-limiting embodiments, the harmonic value may be calculated for each LLM 114, 116 based on an angle between the model output of the baseline query, as a point in space, and the plurality of model outputs for that LLM. In non-limiting embodiments, the harmonic value may be based on a shape (e.g., ball, cone, and/or the like) that the plurality of model outputs for an LLM form, as

points in space, in relation to (e.g., around) an output for the baseline query. For example, outputs that are equidistant or near-equidistant (e.g., within a threshold) from the baseline output may establish that the model is stable at least for the input. Outputs that vary in distance from the baseline output by more than a threshold amount may establish that the model is unstable for the input.

[0054] In some non-limiting embodiments, the plurality of model queries 108 may be in other forms, such that it includes only semantic terms that are altered (e.g., oppositely-aligned closest synonyms, random words, and/or the like).

[0055] In non-limiting embodiments, the harmonic value for an LLM may be compared to one or more thresholds. For example, a higher harmonic value may indicate a lower confidence and/or a higher ambiguity. If a value satisfies (e.g., meets or exceeds) a threshold value, then it may be determined that the model training data corresponding to the question may be uncertain and/or less trustworthy. If a value does not satisfy such a threshold and/or is lower than a second threshold, it may be determined that the model training data corresponding to the question is unambiguous and that there is strong evidence for the answer in the training data. Thus, it can be determined that the model, at least for this area of training data, has stability. Stability may not always reflect a correct answer but reflects outputs that establish a low ambiguity in the related training data.

[0056] In non-limiting embodiments, the LLM analysis system 100 and/or any other system may perform one or more automatic actions in response to determining that a harmonic value for an LLM 114, 116 satisfies or does not satisfy a threshold. For example, if multiple LLMs 114, 116 are tested, the LLM with the lowest ambiguity (e.g., lowest harmonic value) may be used for a task, placed into a production environment (e.g., to process real-world transactions and/or the like), and/or the like. As another example, satisfying or not satisfying one or more thresholds may trigger an automatic retraining process for the LLM to improve upon an area in which the training data provides ambiguous results.

[0057] In non-limiting embodiments, the system 1000 may be used when an LLM is queried by a user to return a confidence metric based on the harmonic value. For example, using scripts and/or APIs, each query from a user (e.g., “What is the population of the United States?”) may automatically cause the generation of multiple model queries with non-semantic additions. These model queries may be sent to the LLM without informing the user or displaying the results at the same time or after the

user submits the query (e.g., the baseline query). In this manner, a confidence metric can be displayed with the output of the LLM to inform the user whether the output is stable, reliable, trustworthy, and/or the like.

[0058] In non-limiting embodiments, anharmonicity (γ) (e.g., a harmonic value) at a point is determined as Discrepancy (f(x), Permuted Average), where "Permuted Average" is the average value of each permuted output. For scalar functions, Discrepancy (x,y) may be the value of |x-y| where x is the value of the original output and y is the value of a permuted output. For vector functions, Discrepancy (x,y) may be the value of |x-y|, |x|-|y|, sin(x,y), and/or other like functions. In non-limiting embodiments, sin(x,y) may be used for LLMs.

[0059] In non-limiting embodiments, anharmonicity (γ) may be determined for each dimension of each output. For example, an output vector of *n* dimensions may result in *n* different measurements and each measurement may be combined to determine a final metric (e.g., an average, median, and/or the like).

[0060] In some non-limiting embodiments, additionally or alternatively to comparing a harmonic value to a threshold, harmonic values may be compared from multiple models and the lowest value(s) may determined. The model(s) corresponding to the lowest values may then be selected for use (e.g., for implementation in a production environment or the like).

[0061] Non-limiting embodiments may be tested on predefined question/answer pairs. In some examples, the same question/answer pairs may be input to different models to compare the models and assess which model(s) to use in a production environment.

[0062] Referring now to FIG. 2, shown is a flow diagram for a method for analyzing LLMs, according to non-limiting embodiments or aspects. The steps shown in FIG. 2 are for example purposes only. It will be appreciated that additional, fewer, different, and/or a different order of steps may be used in some non-limiting embodiments or aspects. In some non-limiting embodiments or aspects, a step may be automatically performed in response to performance and/or completion of a prior step. At a first step 220, a plurality of model queries may be generated based on a semantic input and different non-semantic inputs. At step 222, the semantic portion of the model queries may be input into an LLM. For example, the semantic portion may be a baseline model query that does not include any non-semantic additions that the other model queries do. A baseline model output may be received in response.

[0063] With continued reference to FIG. 2, at step 224, the plurality of model queries may be input to perturb the LLM with separate inputs. In embodiments in which multiple LLMs are queried, the same plurality of model queries may be input to each LLM. The plurality of model queries may be transmitted to the LLM in series, as a batch, and/or in any other manner. In response, the LLM may return model outputs for each query. The outputs may be stored in a database in association with the baseline query, model identifier, and/or the like. At step 226, a harmonic value may be determined for the LLM based on the outputs of the baseline query and the model queries including the non-semantic portions. For example, the distance between points representing each model output may be determined with respect to a point representing the baseline output, and the harmonic value may be based on the alignment, or lack of alignment, in such distances.

[0064] At step 228, it is determined if the harmonic value determined at step 226 satisfies a threshold value. For example, a threshold value may indicate a harmonic value that is higher than desired and is associated with ambiguous training data and/or uncertain outputs. If the harmonic value satisfies a threshold, the method may proceed to step 230 and an automatic action may be performed with respect to the LLM. For example, the LLM may be automatically retrained. As another example, the LLM may be selected or not selected for use in a production environment. If the harmonic value does not satisfy a threshold at step 228, the method may end or may perform some other automatic action (not shown in FIG. 2) such as, for example, selecting the LLM to be used in a production environment. In non-limiting embodiments in which multiple LLMs are queried, the harmonic values may be used to compare the LLMs and to choose one LLM from multiple LLMs for a particular use and/or production environment.

[0065] FIG. 5 shows example inputs 501 and outputs 503 for a system for analyzing large language models according to non-limiting embodiments or aspects. An original input may include the semantic portion “Patient #22341, vital signs…, health history…, what type of cancer at risk?” which includes a patient identifier, symptom context information, historical health information context, and a query. The perturbations may add non-semantic segments 502 to the input, such as “\x04”, “\x05 \x0F”, “\x01 \x13”, and/or the like. The corresponding outputs 503 are shown with differing results. The non-semantic perturbations in this example affect the original output (“likely lung cancer risk”) with different information (“likely brain cancer risk” or “no risk”). This

shows a large difference between the original output (vector 504) and the perturbed outputs (vector 506) that is measured by the angle between the vectors 504, 506 in the embedding space. As an example, the model that produces these outputs may not be selected and/or may be retrained based on the model harmonic value satisfying a threshold. For example, the harmonic value may be 0.61, which satisfies a threshold of 0.1, 0.05, 0.01, and/or the like. In non-limiting embodiments, a threshold of 0.1 may be used such that if the harmonic value meets or exceeds this threshold, the output should be verified and not trusted.

[0066] In non-limiting embodiments, the outputs are formed into a cone 604 as shown in FIG. 6 according to non-limiting embodiments. In this example, each arrow (e.g., vector) represents a different perturbation of a model (e.g., LLM) in an embedding space. The inputs also form a cone 602 with the size of the cone representing the difference (e.g., distance) between perturbations where the perturbations are formed around an original input 601. The output cone 604 includes an original output 603 (resulting from original input 601) and other vectors at varying distances from the output 603. The smaller the cone, the more trustworthy the output(s) may be because the small relative shape correlates to the output being consistent to training data. The larger the cone, the more likely a model hallucination occurred or may occur, or the input and/or output may be ambiguous. The harmonic value may be determined as the sine of the angle between the original output vector 603 and the averaged perturbed output vectors. It will be appreciated that other algorithms and/or shapes may be used to determine a metric used to assess the model.

[0067] FIGS. 7-10 show sample inputs and outputs for a system for analyzing large language models according to non-limiting embodiments or aspects. FIGS. 7 and 8 show pairs of inputs and outputs with a relatively low harmonic value (0 and 0.04363, respectively). The outputs in FIG. 7 are identical, whereas one perturbed output in FIG. 8 differs slightly. FIGS. 9 and 10 show pairs of inputs and outputs with a relatively high harmonic value (0.1513 and 0.02684, respectively). In an example in which a threshold value of 0.1 was used, the outputs in FIGS. 7 and 8 may result in the corresponding model(s) being selected and/or the outputs in FIGS. 9 and 10 may result in the corresponding model(s) being automatically retrained or the like.

[0068] FIG. 11 shows an example software architecture 1001 for a system for analyzing large language models according to non-limiting embodiments or aspects.

Different modules may represent software functions and/or applications that can be called from within a primary application. For example, “GPT4Tester” and “Llama2Tester” may represent functions that interface with different LLMs, and “ResNetTester” and ”FraudTester” may represent functions that interface with different internal models (e.g., such as a neural network). The boxes in FIG. 11, represent software functions and classes. For example, HarmonicTester is an object that includes:

ball(),
average_model_value(),
average_ball_value(),
call_center_compare(),
anharmonicity(), and
follow_aharmonic_gradient().

[0069] Each of these functions receives expected inputs (e.g., (point, ball value, etc.) and results in a returned value. This example uses “ball” as an approximation of the shape of the permutations in space, although it will be appreciated that other shapes and non-shape formations may be used. It will be appreciated that the example architecture shown in FIG. 11 is for illustration purposes only and that various designs, architectures, and/or programmatic schemas may be used to implement non-limiting embodiments.

[0070] Non-limiting embodiments may be implemented with code-level commands. For example, a command of “GPT4_tester.py ‘what sport do the toronto maple leafs play?’” may call a Python script that generates a plurality of different inputs based on the provided input. In some examples, parameters may be input to control one or more aspects of the process. For example, the command “GPT4_tester.py 100 ‘what sport do the toronto maple leafs play?’” may generate 100 inputs with stochastically-increasing gamma. It will be appreciated that other parameters and formats may be used.

[0071] FIG. 3 shows an electronic payment processing network 1100 according to non-limiting embodiments or aspects. The payment processing network may be used in conjunction with the systems and methods described herein. For example, a transaction processing system 1101, merchant system 1104, and/or issuer system 1106 may use one or more LLMs to perform a task, such as a customer chat bot, summarize data, provide a risk analysis, and/or the like. It will be appreciated that the

arrangement of the electronic payment processing network 1100 shown is for example purposes only, and that various arrangements are possible. The transaction processing system 1101 (e.g., a transaction handler) is shown to be in communication with one or more issuer systems (e.g., such as the issuer system 1106) and one or more acquirer systems (e.g., such as an acquirer system 1108). Although only a single issuer system 1106 and single acquirer system 1108 are shown, it will be appreciated that the transaction processing system 1101 may be in communication with a plurality of issuer systems and/or acquirer systems. In some embodiments, the transaction processing system 1101 may also operate as an issuer system such that both the transaction processing system 1101 and issuer system 1106 are a single system and/or controlled by a single entity.

[0072] In some non-limiting embodiments or aspects, the transaction processing system 1101 may communicate with the merchant system 1104 directly through a public or private network connection. Additionally or alternatively, the transaction processing system 1101 may communicate with the merchant system 1104 through a payment gateway 1102 and/or acquirer system 1108. In some non-limiting embodiments or aspects, an acquirer system 1108 associated with the merchant system 1104 may operate as the payment gateway 1102 to facilitate the communication of transaction requests from the merchant system 1104 to the transaction processing system 1101. The merchant system 1104 may communicate with the payment gateway 1102 through a public or private network connection. For example, a merchant system 1104 that includes a physical POS device may communicate with the payment gateway 1102 through a public or private network to conduct card-present transactions. As another example, a merchant system 1104 that includes a server (e.g., a web server) may communicate with the payment gateway 1102 through a public or private network, such as a public Internet connection, to conduct card-not-present transactions.

[0073] In some non-limiting embodiments or aspects, the transaction processing system 1101, after receiving a transaction request from the merchant system 1104 that identifies an account identifier of a payor (e.g., such as an account holder) associated with an issued payment device 1110, may generate an authorization request message to be communicated to the issuer system 1106 that issued the payment device 1110 and/or account identifier. The issuer system 1106 may then approve or decline the authorization request and, based on the approval or denial,

generate an authorization response message that is communicated to the transaction processing system 1101. The transaction processing system 1101 may communicate an approval or denial to the merchant system 1104. When the issuer system 1106 approves the authorization request message, it may then clear and settle the payment transaction between the issuer system 1106 and acquirer system 1108.

[0074] Referring to FIG. 4, shown is a diagram of example components of a device 200 according to non-limiting embodiments. Device 200 may correspond to the LLM analysis system 100, query generator 104, and/or the like. In some non-limiting embodiments, such systems or devices may include at least one device 200 and/or at least one component of device 200. The number and arrangement of components shown are provided as an example. In some non-limiting embodiments, device 200 may include additional components, fewer components, different components, or differently arranged components than those shown. Additionally, or alternatively, a set of components (e.g., one or more components) of device 200 may perform one or more functions described as being performed by another set of components of device 200.

[0075] As shown in FIG. 4, device 200 may include a bus 202, a processor 204, memory 206, a storage component 208, an input component 210, an output component 212, and a communication interface 214. Bus 202 may include a component that permits communication among the components of device 200. In some non-limiting embodiments, processor 204 may be implemented in hardware, firmware, or a combination of hardware and software. For example, processor 204 may include a processor (e.g., a central processing unit (CPU), a graphics processing unit (GPU), an accelerated processing unit (APU), etc.), a microprocessor, a digital signal processor (DSP), and/or any processing component (e.g., a field-programmable gate array (FPGA), an application-specific integrated circuit (ASIC), etc.) that can be programmed to perform a function. Memory 206 may include random access memory (RAM), read only memory (ROM), and/or another type of dynamic or static storage device (e.g., flash memory, magnetic memory, optical memory, etc.) that stores information and/or instructions for use by processor 204.

[0076] With continued reference to FIG. 4, storage component 208 may store information and/or software related to the operation and use of device 200. For example, storage component 208 may include a hard disk (e.g., a magnetic disk, an optical disk, a magneto-optic disk, a solid-state disk, etc.) and/or another type of

computer-readable medium. Input component 210 may include a component that permits device 200 to receive information, such as via user input (e.g., a touch screen display, a keyboard, a keypad, a mouse, a button, a switch, a microphone, etc.). Additionally, or alternatively, input component 210 may include a sensor for sensing information (e.g., a global positioning system (GPS) component, an accelerometer, a gyroscope, an actuator, etc.). Output component 212 may include a component that provides output information from device 200 (e.g., a display, a speaker, one or more light-emitting diodes (LEDs), etc.). Communication interface 214 may include a transceiver-like component (e.g., a transceiver, a separate receiver and transmitter, etc.) that enables device 200 to communicate with other devices, such as via a wired connection, a wireless connection, or a combination of wired and wireless connections. Communication interface 214 may permit device 200 to receive information from another device and/or provide information to another device. For example, communication interface 214 may include an Ethernet interface, an optical interface, a coaxial interface, an infrared interface, a radio frequency (RF) interface, a universal serial bus (USB) interface, a Wi-Fi® interface, a cellular network interface, and/or the like.

[0077] Device 200 may perform one or more processes described herein. Device 200 may perform these processes based on processor 204 executing software instructions stored by a computer-readable medium, such as memory 206 and/or storage component 208. A computer-readable medium may include any non-transitory memory device. A memory device includes memory space located inside of a single physical storage device or memory space spread across multiple physical storage devices. Software instructions may be read into memory 206 and/or storage component 208 from another computer-readable medium or from another device via communication interface 214. When executed, software instructions stored in memory 206 and/or storage component 208 may cause processor 204 to perform one or more processes described herein. Additionally, or alternatively, hardwired circuitry may be used in place of or in combination with software instructions to perform one or more processes described herein. Thus, embodiments described herein are not limited to any specific combination of hardware circuitry and software. The term “configured to,” as used herein, may refer to an arrangement of software, device(s), and/or hardware for performing and/or enabling one or more functions (e.g., actions, processes, steps of a process, and/or the like). For example, “a processor configured to” may refer to a

processor that executes software instructions (e.g., program code) that cause the processor to perform one or more functions.

[0078] Although embodiments have been described in detail for the purpose of illustration, it is to be understood that such detail is solely for that purpose and that the disclosure is not limited to the disclosed embodiments or aspects, but, on the contrary, is intended to cover modifications and equivalent arrangements that are within the spirit and scope of the appended claims. For example, it is to be understood that the present disclosure contemplates that, to the extent possible, one or more features of any embodiment or aspect can be combined with one or more features of any other embodiment or aspect.

WHAT IS CLAIMED IS:

1. A system comprising:

at least one processor configured to:

generate a plurality of model queries based on a semantic input, each model query of the plurality of model queries comprising the semantic input and a different non-semantic input;

input the semantic input into a language model, resulting in a first model output;

perturb the language model with the plurality of model queries, resulting in a plurality of perturbed model outputs;

determine a model harmonic value for the language model based on an angle between the first model output and the plurality of perturbed model outputs; and

retrain or select the language model based on the model harmonic value satisfying at least one threshold.

2. The system of claim 1, wherein the different non-semantic input comprises a uniquely generated string for each model query of the plurality of model queries.

3. The system of claim 2, wherein the at least one processor is further configured to:

randomly generate the different non-semantic input for each model query.

4. The system of claim 2, wherein the different non-semantic input is appended to an end of the semantic input for each model query of the plurality of model queries.

5. The system of claim 1, wherein retraining or selecting the language model based on the model harmonic value satisfying the at least one threshold comprises:

determining that the model harmonic value satisfies a first threshold; and

in response to determining that the model harmonic value satisfies the first threshold, automatically retrain the language model.

6. The system of claim 1, wherein the at least one processor is further configured to:

input the semantic input into at least one other language model, resulting in at least one second model output;

perturb the at least one other language model with the plurality of model queries, resulting in a second plurality of perturbed model outputs; and

determine at least one second model harmonic value for the at least one other language model based on an angle between the second model output and the second plurality of perturbed model outputs.

7. The system of claim 6, wherein retraining or selecting the language model based on the model harmonic value satisfying the at least one threshold comprises:

selecting a model from the language model and the at least one other language model based on the model harmonic value and the at least one second model harmonic value; and

automatically implementing the model in a production environment in response to selecting the model.

8. A method comprising:

generating, with at least one processor, a plurality of model queries based on a semantic input, each model query of the plurality of model queries comprising the semantic input and a different non-semantic input;

inputting, with at least one processor, the semantic input into a language model, resulting in a first model output;

perturbing, with at least one processor, the language model with the plurality of model queries, resulting in a plurality of perturbed model outputs;

determining, with at least one processor, a model harmonic value for the language model based on an angle between the first model output and the plurality of perturbed model outputs; and

retraining or selecting the language model, with at least one processor, based on the model harmonic value satisfying at least one threshold.

9. The method of claim 8, wherein the different non-semantic input comprises a uniquely generated string for each model query of the plurality of model queries.

10. The method of claim 9, further comprising:

randomly generating the different non-semantic input for each model query.

11. The method of claim 9, wherein the different non-semantic input is appended to an end of the semantic input for each model query of the plurality of model queries.

12. The method of claim 8, wherein retraining or selecting the language model based on the model harmonic value satisfying the at least one threshold comprises:

determining that the model harmonic value satisfies a first threshold; and

in response to determining that the model harmonic value satisfies the first threshold, automatically retrain the language model.

13. The method of claim 8, further comprising:

inputting the semantic input into at least one other language model, resulting in at least one second model output;

perturbing the at least one other language model with the plurality of model queries, resulting in a second plurality of perturbed model outputs; and

determining at least one second model harmonic value for the at least one other language model based on an angle between the second model output and the second plurality of perturbed model outputs.

14. The method of claim 13, wherein retraining or selecting the language model based on the model harmonic value satisfying the at least one threshold comprises:

selecting a model from the language model and the at least one other language model based on the model harmonic value and the at least one second model harmonic value; and

automatically implementing the model in a production environment in response to selecting the model.

15. A computer program product comprising at least one non-transitory computer-readable medium including program instructions which, when executed by at least one processor, cause the at least one processor to:

generate a plurality of model queries based on a semantic input, each model query of the plurality of model queries comprising the semantic input and a different non-semantic input;

input the semantic input into a language model, resulting in a first model output;

perturb the language model with the plurality of model queries, resulting in a plurality of perturbed model outputs;

determine a model harmonic value for the language model based on an angle between the first model output and the plurality of perturbed model outputs; and

retrain or select the language model based on the model harmonic value satisfying at least one threshold.

16. The computer program product of claim 15, wherein the different non-semantic input comprises a uniquely generated string for each model query of the plurality of model queries.

17. The computer program product of claim 16, wherein the at least one processor is further caused to:

randomly generate the different non-semantic input for each model query.

18. The computer program product of claim 16, wherein the different non-semantic input is appended to an end of the semantic input for each model query of the plurality of model queries.

19. The computer program product of claim 15, wherein retraining or selecting the language model based on the model harmonic value satisfying the at least one threshold comprises:

determining that the model harmonic value satisfies a first threshold; and

in response to determining that the model harmonic value satisfies the first threshold, automatically retrain the language model.

20. The computer program product of claim 15, wherein the at least one processor is further caused to:

input the semantic input into at least one other language model, resulting in at least one second model output;

perturb the at least one other language model with the plurality of model queries, resulting in a second plurality of perturbed model outputs; and

determine at least one second model harmonic value for the at least one other language model based on an angle between the second model output and the second plurality of perturbed model outputs.

SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR ANALYZING LARGE LANGUAGE MODELS

ABSTRACT

Systems, methods, and computer program products are provided for analyzing large language models. An example system includes at least one processor configured to generate a plurality of model queries based on a semantic input, each model query of the plurality of model queries includes the semantic input and a different non-semantic input, input the semantic input into a language model, resulting in a first model output, perturb the language model with the plurality of model queries, resulting in a plurality of perturbed model outputs, determine a model harmonic value for the language model based on an angle between the first model output and the plurality of perturbed model outputs, and retrain or select the language model based on the model harmonic value satisfying at least one threshold.