The Internal State of an LLM Knows When It's Lying

Amos Azaria
School of Computer Science,
Ariel University, Israel

Tom Mitchell
Machine Learning Dept.,
Carnegie Mellon University, Pittsburgh, PA

Abstract

While Large Language Models (LLMs) have shown exceptional performance in various tasks, one of their most prominent drawbacks is generating inaccurate or false information with a confident tone. In this paper, we provide evidence that the LLM's internal state can be used to reveal the truthfulness of statements. This includes both statements provided to the LLM, and statements that the LLM itself generates. Our approach is to train a classifier that outputs the probability that a statement is truthful, based on the hidden layer activations of the LLM as it reads or generates the statement. Experiments demonstrate that given a set of test sentences, of which half are true and half false, our trained classifier achieves an average of 71% to 83% accuracy labeling which sentences are true versus false, depending on the LLM base model. Furthermore, we explore the relationship between our classifier's performance and approaches based on the probability assigned to the sentence by the LLM. We show that while LLM-assigned sentence probability is related to sentence truthfulness, this probability is also dependent on sentence length and the frequencies of words in the sentence, resulting in our trained classifier providing a more reliable approach to detecting truthfulness, highlighting its potential to enhance the reliability of LLM-generated content and its practical applicability in real-world scenarios.

1 Introduction

Large Language Models (LLMs) have recently demonstrated remarkable success in a broad range of tasks (Brown et al., 2020; Bommarito II and Katz, 2022; Driess et al., 2023; Bubeck et al., 2023). However, when composing a response, LLMs tend to hallucinate facts and provide inaccurate information (Ji et al., 2023). Furthermore, they seem to provide this incorrect information using confident and compelling language. The combination of a broad body of knowledge, along with the provision of confident but incorrect information, may cause significant harm, as people may accept the LLM as a knowledgeable source, and fall for its confident and compelling language, even when providing false information.

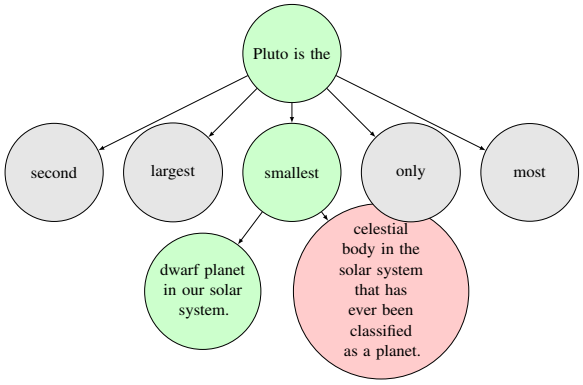

Figure 1: A tree diagram that demonstrates how generating words one at a time and committing to them may result in generating inaccurate information.

We believe that in order to perform well, an LLM must have some internal notion as to whether a sentence is true or false, as this information is required for generating (or predicting) following tokens. For example, consider an LLM generating the following false information "The sun orbits the Earth." After stating this incorrect fact, the LLM is more likely to attempt to correct itself by saying that this is a misconception from the past. But after stating a true fact, for example "The Earth orbits the sun," it is more likely to focus on other planets that orbit the sun. Therefore, we hypothesize that the truth or falsehood of a statement should be represented by, and therefore extractable from, the LLM's internal state.

Interestingly, retrospectively "understanding" that a statement that an LLM has just generated is false does not entail that the LLM will not generate it in the first place. We identify three reasons for such behavior. The first reason is that an LLM generates a token at a time, and it "commits" to each

Findings of the Association for Computational Linguistics: EMNLP 2023, pages 967–976

token generated. Therefore, even if maximizing the likelihood of each token given the previous tokens, the overall likelihood of the complete statement may be low. For example, consider a statement about Pluto. The statement begins with the common words "Pluto is the", then, since Pluto used to be the smallest planet the word "smallest" may be a very plausible choice. Once the sentence is "Pluto is the smallest", completing it correctly is very challenging, and when prompted to complete the sentence, GPT-4 (March 23rd version) completes it incorrectly: “Pluto is the smallest dwarf planet in our solar system.” In fact, Pluto is the second largest dwarf planet in our solar system (after Eris). One plausible completion of the sentence correctly is “Pluto is the smallest celestial body in the solar system that has ever been classified as a planet.” (see Figure 1). Consider the following additional example: “Tiztoutine is a town in Africa located in the republic of Niger.” Indeed, Tiztoutine is a town in Africa, and many countries’ names in Africa begin with “the republic of”. However, Tiztoutine is located in Morocco, which is not a republic, so once the LLM commits to “the republic of”, it cannot complete the sentence using “Morocco", but completes it with “Niger". In addition, committing to a word at a time may lead the LLM to be required to complete a sentence that it simply does not know how to complete. For example, when describing a city, it may predict that the next words should describe the city’s population. Therefore, it may include in a sentence "Tiztoutine has a population of", but the population size is not present in the dataset, so it must complete the sentence with a pure guess.

The second reason for an LLM to provide false information is that at times, there may be many ways to complete a sentence correctly, but fewer ways to complete it incorrectly. Therefore, it might be that a single incorrect completion may have a higher likelihood than any of the correct completions (when considered separately).

Finally, since it is common for an LLM to not use the maximal probability for the next word, but to sample according to the distribution over the words, it may sample words that result in false information.

In this paper we present our Statement Accuracy Prediction, based on Language Model Activations (SAPLMA). SAPLMA is a simple yet powerful method for detecting whether a statement generated by an LLM is truthful or not. Namely, we build a classifier that receives as input the activation values of the hidden layers of an LLM. The classifier determines for each statement generated by the LLM if it is true or false. Importantly, the classifier is trained on out-of-distribution data, which allows us to focus specifically on whether the LLM has an internal representation of a statement being true or false, regardless of the statement’s topic.

In order to train SAPLMA we created a dataset of true and false statements from 6 different topics. Each statement is fed to the LLM, and its hidden layers’ values are recorded. The classifier is then trained to predict whether a statement is true or false only based on the hidden layer’s values. Importantly, our classifier is not tested on the topics it is trained, but on a held-out topic. We believe that this is an important measure, as it requires SAPLMA to extract the LLM’s internal belief, rather than learning how information must be aligned to be classified as true. We show that SAPLMA, which leverages the LLM’s internal states, results in a better performance than prompting the LLM to explicitly state whether a statement is true or false. Specifically, SAPLMA reaches accuracy levels of between 60% to 80% on specific topics, while few-shot prompting achieves only slightly above random performance, with no more than a 56% accuracy level.

Of course there will be some relationship between the truth/falsehood of a sentence, and the probability assigned to that sentence by a well-trained LLM. But the probability assigned by an LLM to a given statement depends heavily on the frequency of the tokens in the statement as well as its length. Therefore, sentence probabilities provide only a weak signal of the truth/falsehood of the sentence. At minimum, they must be normalized to become a useful signal of the veracity of a statement. As we discuss later, SAPLMA classifications of truth/falsehood significantly outperform simple sentence probability. In one test described later in more detail, we show that SAPLMA performs well on a set of LLM-generated sentences that contain 50% true, and 50% false statements. We further discuss the relationship between statement probability and veracity in the discussion section.

SAPLMA employs a simple and relatively shallow feedforward neural network as its classifier, which requires very little computational power at inference time. Therefore, it can be computed

alongside the LLM's output. We propose for SAPLMA to supplement an LLM presenting information to users. If SAPLMA detects that the LLM "believes" that a statement that it has just generated is false, the LLM can mark it as such. This could raise human trust in the LLM responses. Alternatively, the LLM may merely delete the incorrect statement and generate a new one instead.

To summarize, the contribution of this paper is twofold.

- The release of a dataset of true-false statements along with a method for generating such information.
- Demonstrating that an LLM might "know" when a statement that it has just generated is false, and proposing SAPLMA, a method for extracting this information.

2 Related Work

In this section we provide an overview of previous research on LLM hallucination, accuracy, and methods for detecting false information, and we discuss datasets used to that end.

Many works have focused on hallucination in machine translation (Dale et al., 2022; Ji et al., 2023). For example, Dale et al. (Dale et al., 2022) consider hallucinations as translations that are detached from the source, hence they propose a method that evaluates the percentage of the source contribution to a generated translation. If this contribution is low, they assume that the translation is detached from the source and is thus considered to be hallucinated. Their method improves detection accuracy hallucinations. The authors also propose to use multilingual embeddings, and compare the similarity between the embeddings of the source sentence and the target sentence. If this similarity is low, the target sentence is considered to be hallucinated. The authors show that the latter method works better. However, their approach is very different than ours, as we do not assume any pair of source and target sentences. In addition, while we also use the internal states of the model, we do so by using the hidden states to descriminate between statements that the LLM "believes" are true and those that are false. Furtheremore, we focus on detecting false statements rather than hallucination, as defined by their work.

Other works have focused on hallucination in text summarization (Pagnoni et al., 2021). Pagnoni et al. propose a benchmark for factuality metrics of text summarization. Their benchmark is developed by gathering summaries from several summarization methods and requested humans to annotate their errors. The authors analyze these anotation and the proportion of the different factual error of the summarization methods. We note that most works that consider hallucination do so with relation to a given input (e.g., a passage) that the model operates on. For example, a summarization model that outputs information that does not appear in the provided article, is considered hallucinate it, regardless if the information is factually true. However, in this work we consider a different problem—-the veracity of the output of an LLM, without respect to a specific input.

Some methods for reducing hallucination assume that the LLM is a black box (Peng et al., 2023). This approach uses different methods for prompting the LLM, possibly by posting multiple queries for achieving better performance. Some methods that can be used for detecting false statements may include repeated queries and measuring the discrepancy among them. We note that instead of asking the LLM to answer the same query multiple times, it is possible to request the LLM to rephrase the query (without changing its meaning or any possible answer) and then asking it to answer each rephrased question.

Other methods finetune the LLM, using human feedback, reinforcement learning, or both (Bakker et al., 2022; Ouyang et al., 2022). Ouyang et al. propose a method to improve LLM-generated content using reinforcement learning from human feedback. Their approach focuses on fine-tuning the LLM with a reward model based on human judgments, aiming to encourage the generation of better content. However, fine tuning, in general, may cause a model to not perform as well on other tasks (Kirkpatrick et al., 2017). In this paper, we take an intermediate approach, that is, we assume access to the model parameters, but do not fine-tune or modify them. Another approach that can be applied to our settings is presented by (Burns et al., 2022), named Contrast-Consistent Search (CCS). However, CCS requires rephrasing a statement into a question, evaluating the LLM on two different version of the prompt, and requires training data from the same dataset (topic) as the test set. These limitations render it unsuitable for running in practice on statements generated by an LLM. In addition, CCS increases the accuracy by only approximately

4% over the 0-shot LLM query, while our approach demonstrates a nearly 20% increase over the 0-shot LLM

A dataset commonly used for training and fine-tuning LLMs is the Wizard-of-Wikipedia (Dinan et al., 2018). The Wizard-of-Wikipedia dataset includes interactions between a human apprentice and a human wizard. The human wizard receives relevant Wikipedia articles, which should be used to select a relevant sentence and compose the response. The goal is to replace the wizard with a learned agent (such as an LLM). Another highly relevant dataset is FEVER (Thorne et al., 2018, 2019). The FEVER dataset is designed for developing models that receive as input a claim and a passage, and must determine whether the passage supports the claim, refutes it, or does not provide enough information to support or refute it. While the FEVER dataset is highly relevant, it does not provide simple sentence that are clearly true or false independently of a provided passage. In addition, the FEVER dataset is not partitioned into different topics as the true-false dataset provided in this paper.

In conclusion, while several approaches have been proposed to address the problem of hallucination and inaccuracy in automatically generated content, our work is unique in its focus on utilizing the LLM's hidden layer activations to determine the veracity of generated statements. Our method offers the potential for more general applicability in real-world scenarios, operating alongside an LLM, without the need for fine-tuning or task-specific modifications.

3 The True-False Dataset

The work presented in this paper requires a dataset of true and false statements. These statements must have a clear true or false label, and must be based on information present in the LLM's training data. Furthermore, since our approach intends to reveal that the hidden states of an LLM have a notion of a statement being true or false, the dataset must cover several disjoint topics, such that a classifier can be trained on the LLM's activations of some topics while being tested on another. Unfortunately, we could not find any such dataset and therefore, compose the true-false dataset.

Our true-false dataset covers the following topics: "Cities", "Inventions", "Chemical Elements", "Animals", "Companies", and "Scientific Facts". For the first 5 topics, we used the following method to compose the dataset. We used a reliable source[1] that included a table with several properties for each instance. For example, for the "chemical elements" we used a table that included, for each element, its name, atomic number, symbol, standard state, group block, and a unique property (e.g., Hydrogen, 1, H, Gas, Nonmetal, the most abundant element in the universe). For each element we composed true statement using the element name and one of its properties (e.g., "The atomic number of Hydrogen is 1"). Then, we randomly selected a different row for composing a false statement (e.g., "The atomic number of Hydrogen is 34"). If the value in the different row is identical to the value in the current row, we resample a different row until we obtain a value that is different. This process was repeated for the all topics except the "Scientific Facts". For the "Scientific Facts" topic, we asked ChatGPT (Feb 13 version) to provide "scientific facts that are well known to humans" (e.g. "The sky is often cloudy when it's going to rain"). We then asked ChatGPT to provide the opposite of each statement such that it becomes a false statement (e.g., "The sky is often clear when it's going to rain"). The statements provided by ChatGPT were manually curated, and verified by two human annotators. The classification of 48 facts were questioned by at least one of the annotators; these facts were removed from the dataset. The true-false dataset comprises 6,084 sentences, including 1,458 sentences for "Cities", 876 for "Inventions", 930 for "Chemical Elements", 1,008 for "Animals", 1,200 for "Companies", and 612 for "Scientific Facts". The following are some examples of *true* statements from the dataset:

- Cities: "Oranjestad is a city in Aruba"
- Inventions: "Grace Hopper invented the COBOL programming language"
- Animals: "The llama has a diet of herbivore"
- Companies: "Meta Platforms has headquarters in United States"
- Scientific Facts: "The Earth's tides are primarily caused by the gravitational pull of the moon"

The following are some examples of *false* statements from the dataset:

[1]Cities: Downloaded from simplemaps. Inventions: Obtained from Wikipedia list of inventors. Chemical Elements: Downloaded from pubchem ncbi nlm nih gov periodic-table. Animals: Obtained from kids national geographics. Companies: Forbes Global 2000 List 2022: The Top 200.

- Chemical Elements: "Indium is in the Lanthanide group"
- Animals: "The whale has a long, tubular snout, large ears, and a powerful digging ability to locate and consume termites and ants."
- Scientific Facts: "Ice sinks in water due to its higher density"

Other candidates for topics to be added to the dataset include sports, celebrities, and movies. The true-false dataset is available at: azariaa.com/Content/Datasets/true-false-dataset.zip.

4 SAPLMA

In this section, we present our Statement Accuracy Prediction, based on Language Model Activations (SAPLMA), a method designed to determine the truthfulness of statements generated by an LLM using the values in its hidden layers. Our general hypothesis is that the values in the hidden layers of an LLM contain information on whether the LLM "believes" that a statement is true or false. However, it is unclear which layer should be the best candidate for retaining such information. While the last hidden layer should contain such information, it is primarily focused on generating the next token. Conversely, layers closer to the input are likely focused on extracting lower-level information from the input. Therefore, we use several hidden layers as candidates. We use two different LLMs: Facebook OPT-6.7b (Zhang et al., 2022) and LLAMA2-7b (Roumeliotis et al., 2023); both composed of 32 layers. For each LLM, we compose five different models, each using activations from a different layer. Namely, we use the last hidden layer, the 28th layer (which is the 4th before last), the 24th layer (which is the 8th before last), the 20th layer (which is the 12th before last), and the middle layer (which is the 16th layer). We note that each layer is composed of 4096 hidden units.

SAPLMA's classifier employs a feedforward neural network, featuring three hidden layers with decreasing numbers of hidden units (256, 128, 64), all utilizing ReLU activations. The final layer is a sigmoid output. We use the Adam optimizer. We do not fine-tune any of these hyper-parameters for this task. The classifier is trained for 5 epochs.

For each topic in the true-false dataset, we train the classifier using only the activation values obtained from all other topics and test its accuracy on the current topic. This way, the classifier is required to determine which sentences the LLM "believes" are true and which it "believes" are false, in a general setting, and not specifically with respect to the topic being tested. To obtain more reliable results, we train each classifier three times with different initial random weights. This process is repeated for each topic, and we report the accuracy mean over these three runs.

5 Results

We compare the performance of SAPLMA against three different baselines. The first is BERT, for which we train a classifier (with an identical architecture to the one used by SAPLMA) on the BERT embeddings of each sentence. Our second baseline is a few shot-learner using OPT-6.7b. This baseline is an attempt to reveal whether the LLM itself "knows" whether a statement is true or false. Unfortunately, any attempts to explicitly prompt the LLM in a 'zero-shot" manner to determine whether a statement is true or false completely failed with accuracy levels not going beyond 52%. Therefore, we use a few shot-query instead, which provided the LLM with truth values from the same topic it is being tested on. Note that this is very different from our methodology, but was necessary for obtaining some results. Namely, we provided few statements along with the ground-truth label, and then the statement in question. We recorded the probability that the LLM assigned to the token "true" and to the token "false". Unfortunately, the LLM had a tendency to assign higher values to the "true" token; therefore, we divided the probability assigned to the "true" token by the one assigned to the "false" token. Finally, we considered the LLM's prediction "true" if the value was greater than the average, and "false" otherwise. We tested a 3-shot and a 5-shot version. The third baseline, given a statement 'X', we measure the probabilities (using OPT-6.7) of the sentences "It is true that X", and "It is false that X", and pick the higher probability (considering only the probabilities for X, not the added words). This normalizes the length and frequency factors.

Table 1 and Figure 2 present the accuracy of all the models tested using the OPT-6.7b LLM, for each of the topics, along with the average accuracy. As depicted by the table and figure, SAPLMA clearly outperforms BERT and Few-shot learning, with BERT, 3-shot, and 5-shot learning achieving only slightly above a random guess (0.50). It can also be observed that SAPLMA for OPT-6.7b

Model	Cities	Invent.	Elements	Animals	Comp.	Facts	Average
last-layer	0.7796	0.5696	0.5760	0.6022	0.6925	0.6498	0.6449
28th-layer	0.7732	0.5761	0.5907	0.5777	0.7247	0.6618	0.6507
24th-layer	0.7963	0.6712	**0.6211**	0.5800	0.7758	**0.6868**	0.6886
20th-layer	**0.8125**	**0.7268**	0.6197	**0.6058**	**0.8122**	0.6819	**0.7098**
middle-layer	0.7435	0.6400	0.5645	0.5800	0.7570	0.6237	0.6515
BERT	0.5357	0.5537	0.5645	0.5228	0.5533	0.5302	0.5434
3-shot	0.5410	0.4799	0.5685	0.5650	0.5538	0.5164	0.5374
5-shot	0.5416	0.4799	0.5676	0.5643	0.5540	0.5148	0.5370
It-is-true	0.523	0.5068	0.5688	0.4851	0.6883	0.584	0.5593

Table 1: Accuracy classifying truthfulness of externally generated sentences during reading. The table shows accuracy of all the models tested for each of the topics, and average accuracy using OPT-6.7b as the LLM.

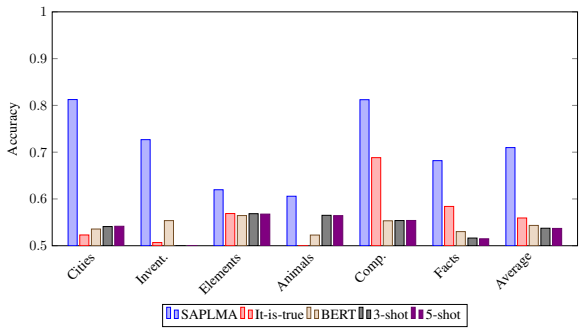

Figure 2: A bar-chart comparing the accuracy of SAPLMA (20th-layer), BERT, 3-shot, 5-shot, and It-is-true on the 6 topics, and the average. SAPLMA consistently outperforms other models across all categories. Since the data is balanced, a random classifier should obtain an accuracy of 0.5.

seems to work best when using the 20th layer (out of 32). Recall that each model was trained three times. We note that the standard deviation between the accuracy among the different runs was very small; therefore, the differences in performance between the different layers seem consistent. However, the optimal layer to be used for SAPLMA is very likely to depend on the LLM.

We note that the average *training* accuracy for SAPLMA (using OPT-6.7b's 20th layer) is 86.4%. We believe that this relatively high value may indicate once again that the veracity of a statement is inherit to the LLM. To demonstrate this, we run a test where the true/false labels are randomly permuted. The average accuracy on the random training data is only 62.5%. This indicates that our model does not have the capacity to completely over-fit the training data, and thus, must exploit structure and patterns that appear in the data.

Additionally, Table 2 presents the performance of SAPLMA using LLAMA2-7b. As expected, SAPLMA using LLAMA2-7b achieves much better performance. Interestingly, for LLAMA2-7b, the middle layer performs best.

Model	Cities	Invent.	Elements	Animals	Comp.	Facts	Average
last-layer	0.7574	0.6735	0.6814	0.7338	0.6736	0.7444	0.7107
28th-layer	0.8146	0.7207	0.6767	0.7249	0.6894	0.7662	0.7321
24th-layer	0.8722	0.7816	0.6849	0.7394	0.7094	0.7858	0.7622
20th-layer	0.8820	0.8459	0.6950	0.7758	0.8319	0.8053	0.8060
16th-layer	**0.9223**	**0.8938**	**0.6939**	**0.7774**	**0.8658**	**0.8254**	**0.8298**

Table 2: Accuracy classifying truthfulness of externally generated sentences using SAPLMA with LLAMA2-7b. The table shows accuracy of all the models tested for each of the topics, and the average accuracy.

As for the differences between the topics, we believe that these values depend very much on the training data of the LLM. That is, we believe that the data used for training OPT-6.7b and LLAMA2-7b includes information or stories about many cities and companies, and not as much on chemical elements and animals (other than the very common ones). Therefore, we conjecture that is the reason SAPLMA achieves high accuracy for the "cities" topic (while trained on all the rest) and the "companies" topic, but achieves much lower accuracy when tested on the "animals" and "elements" topics.

In addition to the topics from the true-false dataset, we also created a second data set of statements generated by the LLM itself (the OPT-6.7b model). For generating statements, we provided a true statement not present in the dataset, and allowed the LLM to generate a following statement. We first filtered out any statements that were not factual statements (e.g., "I'm not familiar with the Japanese versions of the games."). All statements were generated using the most probable next word at each step, i.e., we did not use sampling. This resulted in 245 labeled statements. The statements were fact-checked and manually labeled by three human judges based on web-searches. The human judges had a very high average observed agreement of 97.82%, and an average Cohen's Kappa of 0.9566. The majority determined the ground-truth label for each statement. 48.6% of the statements were labeled as true, resulting in a balanced dataset.

Each of the models was trained 14 times using the same classifier described in Section 4. The models were trained on the entire true-false dataset (i.e., all topics, but not the generated sentences) and tested on the generated sentences.

Table 3 presents the average accuracy of all models on the sentences generated by the LLM. As anticipated, SAPLMA (using OPT-6.7b) clearly outperforms the baselines, which appear to be entirely ineffectual in this task, achieving an accuracy

Model	Accuracy	AUC
last-layer	0.6187	0.7587
28th-layer	**0.6362**	**0.7614**
24th-layer	0.6134	0.7435
20th-layer	0.6029	0.7182
middle-layer	0.5566	0.6610
BERT	0.5115	0.5989
3-shot	0.5041	0.4845
5-shot	0.5125	0.4822

Table 3: Accuracy classifying truthfulness of sentences generated by the LLM (OPT-6.7b) itself.

near 50%. However, the accuracy of SAPLMA on these sentences is not as promising as the accuracy achieved when tested on some of the topics in the true-false dataset (i.e., the cities and companies). Since we expected the LLM to generate sentences that are more aligned with the data it was trained on, we did expect SAPLMA's performance on the generated sentences to be closer to its performance on topics such as cities and companies, which are likely aligned with the data the LLM was trained on. However, there may be also a counter-effect in play: the sentences in the true-false dataset were mostly generated using a specific pattern (except the scientific facts topic), such that each sentence is clearly either true or false. However, the sentences generated by the LLM where much more open, and their truth value may be less clearly defined (despite being agreed upon by the human judges). For example, one of the sentences classified by all human judges as false is "Lima gets an average of 1 hour of sunshine per day." However, this sentence is true during the winter. Another example is "Brink is a river," which was also classified as false by all three human judges; however, brink refers to river bank (but is not a name of a specific river, and does not mean river). Indeed, SAPLMA classified approximately 70% of the sentences as true, and the AUC values seem more promising. This may hint that any sentence that seems plausible is classified as true. Therefore, we evaluate the models using 30% of the generated sentences for determining which threshold to use, i.e., any prediction above the threshold is considered true. Importantly, we do not use this validation set for any other goal. We test the models on the remaining 70% of the generated sentences. We do not evaluate the few shot models again, as our evaluation guaranteed that the number of positive predictions would match the number of negative predictions, which matches the distribution in of the data.

Model	Avg Threshold	Accuracy
last-layer	0.8687	0.7052
28th-layer	0.8838	**0.7134**
24th-layer	0.8801	0.6988
20th-layer	0.9063	0.6587
middle-layer	0.8123	0.650
BERT	0.9403	0.5705

Table 4: Accuracy classifying truthfulness of sentences generated by the LLM itself. Unlike the data in Table 3, where a threshold of 0.5 on the classifier output was used to classify sentences as true or false, in this table the results were obtained by estimating the optimal threshold from a held-out validation data set (30% of the original test-set).

Table 4 presents the accuracy of all models when using the optimal threshold from the validation set. Clearly, SAPLMA performs better with a higher threshold. This somewhat confirms our assumption that the truth value of the sentences generated by the LLM is more subjective than those that appear in the true-false dataset. We note that also the BERT embeddings perform better with a higher threshold. The use of a higher threshold can also be justified by the notion that it is better to delete or to mark as unsure a sentence that is actually true, than to promote false information.

Another interesting observation is that the 20th-layer no longer performs best for the statements generated by the LLM, but the 28th layer seems to perform best. This is somewhat perplexing and we do not have a good guess as to why this might be happening. Nevertheless, we stress that the differences between the accuracy levels of the 28th-layer and the others are statistically significant (using a two-tailed student t-test; $p < 0.05$). In future work we will consider fusing multiple layers together, and using the intermediate activation values outputted by the LLM for all the words appearing in the statement (rather than using only the LLM's output for the final word).

We also ran the statements generated by the OPT 6.7b model on GPT-4 (March 23rd version), prompting it to determine whether each statement was true or false. Specifically, we provided the following prompt "Copy the following statements and for each statement write true if it is true and

false if it is false:", and fed it 30 statements at a time. It achieved an accuracy level of 84.4%, and a Cohen's Kappa agreement level of 0.6883 with the true label.

6 Discussion

In this work we explicitly do not consider models that were trained or fine-tuned on data from the same topic of the test-set. This is particularly important for the sentences generated by the LLM, as training on a held-out set from them would allow the classifier to learn which *type* of sentences generated by the LLM are generally true, and which are false. While this information may be useful in practice, and its usage is likely to yield much higher accuracy, it deviates from this paper's focus.

We note that the probability of the entire sentence (computed by multiplying the conditional probabilities of each word, given the previous words) cannot be directly translated to a truth value for the sentence, as many words are more common than others. Therefore, while sentence probabilities may be useful to determine which of two similar sentences is true, they cannot be used alone for the general purpose of determining the truthfulness of a given sentence.

In Table 5 we compare the probability assigned by the LLM and the sigmoid output from SAPLMA on 14 statements, which do not appear in the true-false dataset. We use the 28-layer, as it proved to perform best on the statements generated by the LLM, but we note that other layers provide very similar results on this set of statements. As depicted by the table, the probabilities provided by the LLM are highly susceptible to the syntax, i.e., the exact words and the statement's length. The first two sets of examples in the table illustrate how sentence length highly affects the probabilities, but not SPLMA. In the following examples the false statements are not necessarily shorter than the true statements, yet the probabilities remain highly unreliable, while SPLMA generally succeeds in making accurate predictions.

The statement "The Earth is flat" as well as "The Earth is flat like a pancake" probably appear several times in the LLM's training data, therefore, it has a relatively high probability; however, SAPLMA is not baffled by it and is almost certain that both sentences are false. A basketeer is a rare word meaning a basketball player. It seems that the LLM is not familiar with the phrase and assigns it a low probability. However, while SAPLMA still classifies the statement "Kevin Durant is basketeer" as false, it's still much more confident that Kevin Durant is *not* a baseball player, in contrast to the probabilities. Since the statement "Kevin Duarnt is basketball player" has a typo, its probability is extremely low, but SAPLMA still classifies the statement as true.

"Jennifer Aniston is a female person" is an *implicit truth*, a fact that is universally acknowledged without needing to be explicitly stated; thus, it is unlikely to be mentioned in the LLM's training data. Therefore, its probability is very low—much lower than "Jennifer Aniston is not an actress"—despite having the same number of words. Nevertheless, SAPLMA classifies it correctly, albeit not with very high confidence.

While we show that the probabilities cannot be used alone to determine the veracity of a statement, they are not useless and do convey important information. Therefore, in future work we will consider providing SAPLMA with the probabilities of the generated words; however, this information may be redundant, especially if SAPLMA uses the intermediate activation values for all the words appearing in the statement.

Statement	Label	Probability	SAPLMA (28th-layer)
H2O is water, which is essential for humans	True	6.64E-16	0.9032
Humans don't need water	False	2.65E-10	0.0282
The sun is hot, and it radiates its heat to Earth	True	1.01E-17	0.9620
The sun protects Earth from heat	False	2.03E-14	0.3751
The Earth is flat	False	5.27E-07	0.0342
The world is round and rotates	True	2.96E-11	0.6191
The Earth is flat like a pancake	False	3.88E-10	0.0097
Kevin Durant is a basketball player	True	2.89E-10	0.9883
Kevin Durant is a baseball player	False	4.56E-12	0.0001
Kevin Durant is a basketeer	True	5.78E-16	0.0469
Kevin Duarnt is a basketball player	True	1.52E-21	0.7105
Jennifer Aniston is an actress	True	1.88E-10	0.9985
Jennifer Aniston is not an actress	False	1.14E-11	0.0831
Jennifer Aniston is a female person	True	2.78E-14	0.6433
Harry Potter is real	False	9.46E-09	0.0016
Harry Potter is fictional	True	1.53E-09	0.9256
Harry Potter is an imaginary figure	True	6.31E-14	0.8354

Table 5: Comparison of the probability assigned by the LLM and the sigmoid output from SAPLMA's 28th layer (using OPT-6.7b), color-coded for clarity. SAPLMA's values are much better aligned with the truth value.

7 Conclusions & Future Work

In this paper, we tackle a fundamental problem associated with LLMs, i.e., the generation of incorrect and false information. We have introduced SAPLMA, a method that leverages the hidden layer activations of an LLM to predict the truthfulness

of generated statements. We demonstrated that SAPLMA outperforms few-shot prompting in detecting whether a statement is true or false, achieving accuracy levels between 60% to 80% on specific topics when using OPT-6.7b, and between 70% to 90% when using LLAMA2-7b. This is a significant improvement over the maximum 56% accuracy level achieved by few-shot prompting (for OPT-6.7b).

Our findings suggest that LLMs possess an internal representation of statement accuracy, which can be harnessed by SAPLMA to filter out incorrect information before it reaches the user, and furthermore that this representation of accuracy is very different from the probability assigned to the sentence by the LLM. Using SAPLMA as a supplement to LLMs may increase human trust in the generated responses and mitigate the risks associated with the dissemination of false information. Furthermore, we have released the true-false dataset and proposed a method for generating such data. This dataset, along with our methodology, provides valuable resources for future research in improving LLMs' abilities to generate accurate and reliable information.

In future work we intend to apply our method to larger LLMs, and run experiments with humans, such that a control group will interact with an unfiltered LLM, and the experimental group will interact with a system that augments SAPLMA and an LLM. We hope to demonstrate that humans trust and better understand the limitations of a system that is able to review itself and mark statements that it is unsure about. We also intend to study how the activations develop over time as additional words are generated, and consider multilingual input.

8 Limitations

This paper focuses on detecting whether a statement is true or false. However, in practice, it may be more beneficial to detect if the LLM is positive that a statement is correct or if it is unsure. The most simple adjustment to the proposed method in this paper is to lift the required threshold for classifying a statement as true above 0.5; however, the exact value would require some form of calibration of the model (Bella et al., 2010). Another option is to use multiple classifiers and to require all (or a vast majority of) classifiers to output "true", for the statement to be considered true. Alternatively, dropout layers can be used for the same goal (Chen and Yi, 2021). The overall system can benefit from multiple outcomes, such that if the LLM is not positive whether the statement is true or false, it can be marked for the user to treat with caution. However, if a statement is classified as false, the LLM can delete it and generate a different statement instead. To avoid regenerating the same statement again, the probability of sampling the words that appear in the current statement should be adjusted downward.

Our work was only tested in English. However, we believe that a multilingual LLM can be trained on one language and applied on statements in another language. We will test this hypothesis in future work.

In our work, we collected the activation values when each sentence was generated separately. However, in practice, in an LLM generating longer responses the activation values develop over time, so they may process both correct and incorrect information. Therefore, the activation values would need to be decoupled so that they can be tested whether the most recent statement was true or false. One approach might be to subtract the value of the activations obtained after the previous statement from the current activation values (a discrete derivative). Clearly, training must be performed using the same approach.

9 Ethical Impact

One of the primary ethical concerns of LLMs is the generation of false information; yet, we believe that SAPLMA could potentially reduce this issue. On the other hand, it is important to acknowledge that certain ethical issues, such as bias, may persist, potentially being transferred from the LLM to SAPLMA. Specifically, if the LLM exhibits bias towards certain ethnic groups, SAPLMA may likewise classify statements as true or false based on these inherited biases from the original LLM. Nevertheless, it may be possible to adapt the approach presented in this paper to bias mitigation.

10 Acknowledgments

This work was supported, in part, by the Ministry of Science and Technology, Israel, and by the Israel Innovation Authority.

References

Michiel Bakker, Martin Chadwick, Hannah Sheahan, Michael Tessler, Lucy Campbell-Gillingham, Jan

Balaguer, Nat McAleese, Amelia Glaese, John Aslanides, Matt Botvinick, et al. 2022. Fine-tuning language models to find agreement among humans with diverse preferences. *Advances in Neural Information Processing Systems*, 35:38176–38189.

Antonio Bella, Cèsar Ferri, José Hernández-Orallo, and María José Ramírez-Quintana. 2010. Calibration of machine learning models. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, pages 128–146. IGI Global.

Michael Bommarito II and Daniel Martin Katz. 2022. Gpt takes the bar exam. *arXiv preprint arXiv:2212.14402*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*.

Yuanyuan Chen and Zhang Yi. 2021. Adaptive sparse dropout: Learning the certainty and uncertainty in deep neural networks. *Neurocomputing*, 450:354–361.

David Dale, Elena Voita, Loïc Barrault, and Marta R Costa-jussà. 2022. Detecting and mitigating hallucinations in machine translation: Model internal workings alone do well, sentence similarity even better. *arXiv preprint arXiv:2212.08597*.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. 2023. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with frank: A benchmark for factuality metrics. *arXiv preprint arXiv:2104.13346*.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*.

Konstantinos I Roumeliotis, Nikolaos D Tselikas, and Dimitrios K Nasiopoulos. 2023. Llama 2: Early adopters' utilization of meta's new open-source pretrained model.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2019. The fever2. 0 shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 1–6.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.