advanced prompt

- **Prompt Structuring Frameworks**
  Prompt Structuring Frameworks Understanding the role of CO STAR in structured prompting   How CRISPE enhances clarity in AI generated outputs   SPEC as a guiding model for consistent prompts   Using SCQA framing to align prompts with user intent   Adapting BRIEF for instructional content design   When to combine CO STAR and CRISPE for complex tasks   Framework selection for multi step reasoning prompts   Practical uses of SPEC in technical documentation   How SCQA improves logical flow in AI conversations   Evaluating framework fit for different content goals   Framework based prompting for collaborative writing   Mapping prompt frameworks to industry applications
- **Reasoning and Problem-Solving Techniques**
  Reasoning and Problem-Solving Techniques Exploring chain of thought for stepwise reasoning   Tree of thought as a method for decision exploration   Applying ReAct to combine reasoning with actions   How self ask prompts support Socratic style inquiry   Critic and editor prompting for iterative refinement   Plan and solve prompting for structured solutions   Self consistency sampling to stabilize reasoning outputs   Using scratchpad memory to extend logical processes   Multi pass reasoning for deeper content generation   Combining few shot examples with reasoning prompts   Exploring debate style multi agent reasoning   Adaptive reasoning strategies for complex AI tasks
- **About Us**

Exploring debate style multi agent reasoning

Multi-Stage Prompt Design

In the ever-evolving realm of artificial intelligence, the interplay between multi-agent reasoning and prompt engineering has emerged as a fascinating area of study. This landscape is particularly intriguing when applied to the context of debate-style interactions among multiple agents.

At its core, multi-agent reasoning involves the coordination and decision-making processes among several autonomous entities, each with its own goals and strategies. When these agents engage in a debate, they not only need to articulate their own positions effectively but also to understand, challenge, and respond to the arguments of others. This dynamic requires a sophisticated level of reasoning, where agents must navigate complex social and logical landscapes.

Prompt engineering plays a crucial role in this scenario. Context and token management in prompts allows for longer and more coherent conversations **advanced prompt engineering techniques** Web page. It involves crafting inputs or prompts that guide the behavior and responses of these agents. Effective prompts can elicit more nuanced and relevant arguments, encourage critical thinking, and foster a more engaging and productive debate. The art of prompt engineering lies in balancing specificity with openness, allowing agents enough room to explore various angles of an argument while still steering them towards the debates core issues.

In exploring debate-style multi-agent reasoning, one must consider the ethical implications and the potential for bias. Agents might be inclined to favor certain arguments or perspectives, influenced by their programming or the data theyve been trained on. Therefore, its essential to design prompts and reasoning frameworks that promote fairness, inclusivity, and a diversity of viewpoints.

Moreover, the landscape of multi-agent reasoning in debates is not static. It evolves with advancements in AI, changes in societal norms, and the emergence of new topics of debate. This necessitates a continuous refinement of both the agents reasoning capabilities and the prompts that guide them.

In conclusion, the landscape of multi-agent reasoning and prompt engineering in the context of debate-style interactions is a rich and complex field. It offers a window into the future of AI-driven discussions, where machines not only assist in human debates but also engage in meaningful, autonomous discourse. As this field matures, it holds the promise of enhancing our understanding of both artificial and human intelligence.

Debate-Style Reasoning: A Framework for Enhanced Agent Collaboration offers a fascinating approach to multi-agent systems, particularly in the context of decision-making and problem-solving. This method draws inspiration from human debate, where multiple viewpoints are presented, scrutinized, and refined through discussion. In the realm of artificial intelligence, this translates into a process where agents engage in a structured form of argumentation, aiming to reach a consensus or the most optimal solution by leveraging the strengths of each participants perspective.

In a debate-style multi-agent reasoning system, each agent functions like a debater, equipped with its unique dataset, algorithms, and objectives. Imagine a scenario where agents are tasked with deciding on the best environmental policy. One agent might advocate for renewable energy based on economic data, while another might focus on the ecological benefits, and yet another might bring in social impact assessments. The debate begins with each agent presenting its initial stance, supported by data and reasoning.

The beauty of this framework lies in its dynamic interaction. Agents do not simply present static viewpoints; they engage in a back-and-forth, challenging each others assumptions, refining arguments, and sometimes even altering their positions based on compelling evidence or logic from peers. This interaction mirrors human debate clubs where the goal isnt just to win but to arrive at a deeper understanding or a better solution. For instance, the agent focusing on economic data might be persuaded to consider long-term ecological costs after a robust counter-argument from the ecological-focused agent.

This collaborative reasoning process not only enhances the decision-making quality but also fosters learning among the agents. Each agent learns from the strengths and weaknesses of others arguments, potentially leading to the evolution of more sophisticated reasoning capabilities over time. Moreover, this method encourages transparency since the reasoning process is laid bare, allowing for better traceability and accountability in AI decision-making.

In practical applications, this could mean more effective policy-making in governance, where AI systems could simulate various stakeholder perspectives before human decision-makers finalize policies. In business, it could lead to more nuanced market strategies by considering diverse consumer behaviors and market conditions from different angles. The key advantage here is the reduction of bias, as the debate-style ensures that multiple facets are considered, reducing the likelihood of overlooking critical aspects due to a single-minded approach.

In conclusion, the debate-style reasoning framework provides a robust method for enhancing collaboration among AI agents. By mimicking the human debate process, it ensures that decisions are not only well-rounded but also robustly tested through a form of intellectual competition. This approach holds significant promise for advancing AI applications in complex, multi-dimensional problem spaces where traditional, linear decision-making models fall short.

Dynamic Prompt Adaptation Strategies

In the realm of artificial intelligence and natural language processing, the concept of Advanced Prompt Engineering Techniques for Eliciting Debate has emerged as a fascinating and intricate approach to exploring debate style multi-agent reasoning. This technique involves the careful crafting of prompts that not only stimulate conversation but also encourage the development of complex, multi-faceted arguments among multiple agents.

At its core, this method is about more than just asking questions. Its about creating a dynamic environment where agents, whether they are AI systems or human participants, are prompted to engage in a structured yet fluid debate. The goal is to foster a rich exchange of ideas, where each agent builds upon or challenges the arguments of others, leading to a deeper exploration of the topic at hand.

One of the key aspects of this technique is the design of prompts that are open-ended yet specific enough to guide the conversation in a meaningful direction. These prompts should be crafted to provoke thought, encourage critical thinking, and invite diverse perspectives. For instance, a prompt might ask agents to debate the ethical implications of a particular technology, requiring them to consider various viewpoints and articulate their reasoning clearly.

Moreover, the effectiveness of these prompts lies in their ability to adapt and evolve as the debate progresses. As agents present their arguments and counterarguments, the prompts can be adjusted to delve deeper into specific points, challenge assumptions, or introduce new angles to the discussion. This dynamic nature of the prompts ensures that the debate remains engaging and thought-provoking.

In the context of multi-agent reasoning, this technique is particularly powerful. It allows for the simulation of real-world debates, where multiple stakeholders with differing interests and perspectives come together to discuss complex issues. By engaging in such debates, agents can develop a more nuanced understanding of the topic, learn to articulate their thoughts more effectively, and appreciate the value of diverse viewpoints.

In conclusion, Advanced Prompt Engineering Techniques for Eliciting Debate represent a sophisticated approach to exploring debate style multi-agent reasoning. By carefully crafting prompts that stimulate thought, encourage critical engagement, and adapt to the evolving nature of the debate, this technique not only enhances the quality of the discussion but also fosters a deeper, more comprehensive understanding of complex issues. As we continue to explore the potential of AI in facilitating meaningful conversations, this approach stands out as a valuable tool in the arsenal of natural language processing techniques.

Evaluation Metrics for Prompt Effectiveness

When exploring the intricacies of debate style multi-agent reasoning, the experimental setup for evaluating various debate strategies becomes crucial. This setup involves creating a controlled environment where multiple AI agents engage in debates, each equipped with different strategies prompted by diverse topics. The aim is to understand how variations in prompts can influence the effectiveness of debate strategies, thereby shedding light on the dynamics of multi-agent reasoning in a competitive discourse.

In our experimental setup, we begin by defining a broad range of debate topics that cover various domains such as ethics, technology, economics, and social issues. Each topic is designed to challenge the agents in unique ways, promoting a rich diversity in the debates. For instance, a topic might revolve around the ethical implications of AI in healthcare, while another might delve into the economic impacts of climate change policies. This diversity ensures that the strategies are tested under different conditions, revealing their robustness and adaptability.

Next, we assign each agent a distinct debate strategy. Some agents might employ a logical, step-by-step approach, focusing on building a coherent argument from premises to conclusion. Others might use emotional appeals or employ rhetorical questions to sway the debate in their favor. There are also strategies that involve aggressive rebuttals or a more defensive stance, aiming to protect ones position rather than attack the opponents. By having this variety, we can observe how different strategies fare when faced with prompts that might favor one approach over another.

The debate process itself is structured in rounds, where each agent presents its case, responds to opponents, and adapts its strategy based on the flow of the debate. Here, the role of prompts becomes evident. A prompt that is highly technical might benefit an agent with a strategy tailored towards detailed analysis and data-driven arguments. Conversely, a prompt that touches on societal values might see an agent with an emotional or value-based strategy perform better.

To evaluate these debates, we use both qualitative and quantitative metrics. Qualitatively, we assess the coherence of arguments, the creativity in approach, and the ability to adapt to new information or counterarguments. Quantitatively, we measure success through win rates, the persuasiveness of arguments as judged by human evaluators or machine learning models trained on human preferences, and the efficiency of strategy implementation, like how quickly an agent can adapt its tactics.

This experimental setup not only helps in understanding the effectiveness of various debate strategies but also provides insights into how AI agents can simulate human-like reasoning in debates. It highlights the importance of flexibility, context awareness, and strategic depth in AI-driven discussions. By continually refining our prompts and strategies, we aim to push the boundaries of what AI can achieve in the realm of competitive, reasoned discourse, offering valuable lessons for both AI development and our understanding of human debate dynamics.

In the realm of artificial intelligence, the exploration of debate-style multi-agent reasoning has emerged as a fascinating approach to enhance the performance and decision-making capabilities of AI systems. This method, inspired by human debate, involves multiple agents presenting arguments, counterarguments, and evidence to reach a conclusion. The results and analysis of implementing such a system reveal significant performance gains, underscoring the potential of this approach in various applications.

One of the primary benefits observed is the improvement in decision accuracy. When agents engage in debate, they are compelled to thoroughly examine different perspectives and evidence. This rigorous scrutiny leads to more robust and well-rounded conclusions. For instance, in a scenario where an AI system is tasked with making a complex decision, such as diagnosing a medical condition, the debate-style reasoning allows for a comprehensive evaluation of symptoms, patient history, and potential diagnoses. This results in a more accurate and reliable decision compared to traditional single-agent reasoning methods.

Moreover, the debate-style approach fosters a dynamic learning environment. As agents present and defend their arguments, they are exposed to new information and viewpoints. This exposure not only enhances their individual knowledge but also contributes to the collective intelligence of the system. Over time, the agents become more adept at identifying relevant information, evaluating evidence, and constructing persuasive arguments. This continuous learning process leads to gradual improvements in the systems performance, making it more effective in handling diverse and complex tasks.

Another notable gain is the increased transparency and explainability of the decision-making process. In traditional AI systems, the reasoning behind a decision can often be opaque, making it difficult to understand and trust. However, debate-style reasoning provides a clear trail of the arguments and evidence considered. This transparency not only enhances user trust but also facilitates easier debugging and refinement of the system. Stakeholders can trace the decision-making process, identify any biases or errors, and make necessary adjustments to improve the systems performance.

Additionally, the debate-style approach encourages collaboration and synergy among agents. Unlike competitive or adversarial methods, where agents may work at cross-purposes, debate-style reasoning promotes a cooperative environment. Agents learn to build upon each others arguments, leading to more innovative and comprehensive solutions. This collaborative dynamic is particularly beneficial in scenarios requiring multidisciplinary expertise, such as urban planning or environmental policy-making, where diverse perspectives and knowledge domains must be integrated.

In conclusion, the implementation of debate-style multi-agent reasoning yields substantial performance gains across various dimensions. It enhances decision accuracy, fosters dynamic learning, increases transparency, and promotes collaboration. As this approach continues to evolve, it holds the promise of revolutionizing how AI systems reason and make decisions, paving the way for more intelligent, reliable, and transparent artificial intelligence.

Implementing debate-style multi-agent systems presents a fascinating yet complex endeavor. These systems, which aim to simulate human-like debate and reasoning among multiple agents, hold great promise for advancing artificial intelligence and decision-making processes. However, several challenges and limitations must be carefully considered to ensure their effective implementation.

One of the primary challenges is the development of robust argumentation frameworks. Agents must be capable of constructing, evaluating, and countering arguments in a coherent and logical manner. This requires sophisticated natural language processing capabilities and an understanding of rhetorical strategies, which are still areas of active research. Moreover, ensuring that agents can adapt their arguments based on the context and the opposing viewpoints adds another layer of complexity.

Another significant challenge lies in the design of effective dialogue protocols. Debate-style interactions necessitate clear rules and norms to govern the exchange of arguments. These protocols must balance the need for structured debate with the flexibility required for dynamic and unpredictable discussions. Additionally, they must account for potential adversarial behaviors, such as deception or manipulation, which can undermine the integrity of the debate.

The issue of scalability is also a critical concern. As the number of agents increases, managing the complexity of interactions becomes more daunting. Ensuring that debates remain productive and that all agents have an opportunity to contribute requires careful orchestration. Furthermore, the computational resources needed to simulate large-scale debates can be substantial, posing practical limitations on the size and scope of such systems.

Ethical considerations cannot be overlooked. Debate-style multi-agent systems must be designed with safeguards to prevent the propagation of misinformation or biased arguments. Agents should be programmed to prioritize factual accuracy and logical consistency, and mechanisms must be in place to detect and correct errors or misleading information.

Lastly, the challenge of evaluating the performance of these systems is non-trivial. Traditional metrics may not suffice for assessing the quality of debate or the effectiveness of argumentation. New evaluation frameworks that can capture the nuances of debate-style interactions are needed to ensure that these systems meet their intended goals.

In conclusion, while debate-style multi-agent systems offer exciting possibilities for enhancing reasoning and decision-making, their implementation is fraught with challenges. Addressing these issues requires a multidisciplinary approach, combining insights from artificial intelligence, linguistics, ethics, and social sciences to create systems that are both effective and ethically sound.

The exploration of debate-style multi-agent reasoning presents a fascinating frontier in artificial intelligence, where the convergence of diverse perspectives can lead to more robust decision-making processes. As we delve into this domain, the concept of "Future Directions: Enhancing Debate through Advanced Prompting Methods" becomes crucial. This approach not only promises to refine the quality of debates among AI agents but also aims to mirror the complexity and nuance of human discourse.

Advanced prompting methods in this context involve the strategic formulation of questions or statements that guide AI agents towards deeper, more critical analysis. By enhancing prompts, we can steer agents to consider a broader range of viewpoints, anticipate counterarguments, and develop more sophisticated reasoning pathways. This methodology is akin to training a skilled debater, where the quality of preparation through prompts can significantly influence the outcome of the debate.

One key aspect of enhancing debate through advanced prompting is the introduction of dynamic prompts that evolve based on the conversations progression. Imagine an AI debate where each agents response triggers a new set of prompts tailored to delve deeper into unresolved issues or to challenge emerging consensus. This dynamic interaction not only keeps the debate lively but also ensures that no stone is left unturned in the quest for a well-rounded conclusion.

Moreover, incorporating elements of emotional intelligence into prompts could revolutionize how AI agents engage in debates. By prompting agents to consider emotional responses or the psychological impact of their arguments, we could move towards AI systems that not only understand logical constructs but also appreciate the human elements of persuasion and empathy. This could lead to debates that are not only intellectually stimulating but also resonate on a more human level, making AI interactions more relatable and acceptable to human users.

Another promising direction is the use of meta-prompts, which are prompts about how to prompt. This meta-level strategy encourages AI agents to self-reflect on their prompting techniques, potentially leading to self-improvement in how they frame and engage with debates. Such an approach could foster a learning environment where AI agents continuously refine their debate strategies, much like human debaters who evolve through experience.

In practice, implementing these advanced prompting methods would require sophisticated AI architectures capable of real-time analysis and adaptation. The integration of machine learning models that can predict the flow of a debate, understand the nuances of language, and adjust prompts accordingly would be essential. Additionally, ethical considerations must be at the forefront, ensuring that the debates remain fair, unbiased, and beneficial to all stakeholders involved.

In conclusion, by pushing the boundaries of how we prompt AI agents in debate-style interactions, we are not just enhancing the capabilities of these systems but also enriching the way AI can contribute to complex decision-making processes. As we continue to explore and refine these methods, the potential for AI to assist in nuanced, multi-faceted discussions grows, promising a future where AI debate is as insightful and impactful as human discourse.

About Training, validation, and test data sets

- v
- t
- e

Part of a series on

Machine learning and data mining

Paradigms

- Supervised learning
- Unsupervised learning
- Semi-supervised learning
- Self-supervised learning
- Reinforcement learning
- Meta-learning
- Online learning
- Batch learning
- Curriculum learning
- Rule-based learning
- Neuro-symbolic AI
- Neuromorphic engineering
- Quantum machine learning

Problems

- Classification
- Generative modeling
- Regression
- Clustering
- Dimensionality reduction
- Density estimation
- Anomaly detection
- Data cleaning
- AutoML
- Association rules
- Semantic analysis
- Structured prediction
- Feature engineering
- Feature learning
- Learning to rank
- Grammar induction
- Ontology learning
- Multimodal learning

Supervised learning
(**classification • regression**)

- Apprenticeship learning
- Decision trees
- Ensembles
  - Bagging
  - Boosting
  - Random forest
- *k*-NN
- Linear regression
- Naive Bayes
- Artificial neural networks
- Logistic regression
- Perceptron
- Relevance vector machine (RVM)
- Support vector machine (SVM)

Clustering

- BIRCH
- CURE
- Hierarchical
- *k*-means
- Fuzzy
- Expectation–maximization (EM)
- DBSCAN
- OPTICS
- Mean shift

Dimensionality reduction

- Factor analysis
- CCA
- ICA
- LDA
- NMF
- PCA
- PGD
- t-SNE
- SDL

Structured prediction

- Graphical models
  - Bayes net
  - Conditional random field
  - Hidden Markov

Anomaly detection

- RANSAC
- *k*-NN
- Local outlier factor
- Isolation forest

Neural networks

- Autoencoder
- Deep learning
- Feedforward neural network
- Recurrent neural network
  - LSTM
  - GRU
  - ESN
  - reservoir computing
- Boltzmann machine
  - Restricted
- GAN
- Diffusion model
- SOM
- Convolutional neural network
  - U-Net
  - LeNet
  - AlexNet
  - DeepDream
- Neural field
  - Neural radiance field
  - Physics-informed neural networks
- Transformer
  - Vision
- Mamba
- Spiking neural network
- Memtransistor
- Electrochemical RAM (ECRAM)

Reinforcement learning

- Q-learning
- Policy gradient
- SARSA
- Temporal difference (TD)
- Multi-agent
  - Self-play

Learning with humans

- Active learning
- Crowdsourcing
- Human-in-the-loop
- Mechanistic interpretability
- RLHF

Model diagnostics

- Coefficient of determination
- Confusion matrix
- Learning curve
- ROC curve

Mathematical foundations

- Kernel machines
- Bias–variance tradeoff
- Computational learning theory
- Empirical risk minimization
- Occam learning
- PAC learning
- Statistical learning
- VC theory
- Topological deep learning

Journals and conferences

- AAAI
- ECML PKDD
- NeurIPS
- ICML
- ICLR
- IJCAI
- ML
- JMLR

Related articles

- Glossary of artificial intelligence
- List of datasets for machine-learning research
    - List of datasets in computer vision and image processing
- Outline of machine learning

In machine learning, a common task is the study and construction of algorithms that can learn from and make predictions on data.[1] Such algorithms function by making data-driven predictions or decisions,[2] through building a mathematical model from input data. These input data used to build the model are usually divided into multiple data sets. In particular, three data sets are commonly used in different stages of the creation of the model: training, validation, and test sets.

The model is initially fit on a **training data set**,[3] which is a set of examples used to fit the parameters (e.g. weights of connections between neurons in artificial neural networks) of the model.[4] The model (e.g. a naive Bayes classifier) is trained on the training data set using a supervised learning method, for example using optimization methods such as gradient descent or stochastic gradient descent. In practice, the training data set often consists of pairs of an input vector (or scalar) and the corresponding output vector (or scalar), where the answer key is commonly denoted as the *target* (or *label*). The current model is run with the training data set and produces a result, which is then compared with the *target*, for each input vector in the training data set. Based on the result of the comparison and the specific learning algorithm being used, the parameters of the model are adjusted. The model fitting can include both variable selection and parameter estimation.

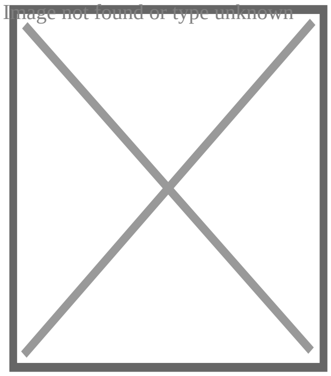
Successively, the fitted model is used to predict the responses for the observations in a second data set called the **validation data set**.[3] The validation data set provides an unbiased evaluation of a model fit on the training data set while tuning the model's hyperparameters[5] (e.g. the number of hidden units—layers and layer widths—in a neural network[4]). Validation data sets can be used for regularization by early stopping (stopping training when the error on the validation data set increases, as this is a sign of over-fitting to the training data set).[6] This simple procedure is complicated in practice by the fact that the validation data set's error may fluctuate during training, producing multiple local minima. This complication has led to the creation of many ad-hoc rules for deciding when over-fitting has truly begun.[6]

Finally, the **test data set** is a data set used to provide an unbiased evaluation of a *final* model fit on the training data set.[5] If the data in the test data set has never been used in training (for example in cross-validation), the test data set is also called a **holdout data set**. The term "validation set" is sometimes used instead of "test set" in some literature (e.g., if the original data set was partitioned into only two subsets, the test set might be referred to as the validation set).[5]

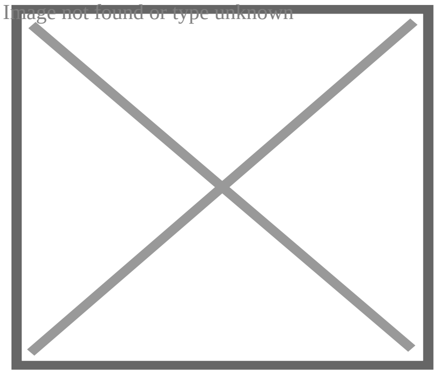
Deciding the sizes and strategies for data set division in training, test and validation sets is very dependent on the problem and data available.[7]

Training data set

[edit]



Simplified example of training a neural network in object detection: The network is trained by multiple images that are known to depict starfish and sea urchins, which are correlated with "nodes" that represent visual features. The starfish match with a ringed texture and a star outline, whereas most sea urchins match with a striped texture and oval shape. However, the instance of a ring textured sea urchin creates a weakly weighted association between them.



Subsequent run of the network on an input image (left):[8] The network correctly detects the starfish. However, the weakly weighted association between ringed texture and sea urchin also confers a weak signal to the latter from one of two intermediate nodes. In addition, a shell that was not included in the training gives a weak signal for the oval shape, also resulting in a weak signal for the sea urchin output. These weak signals may result in a false positive result for sea urchin.
In reality, textures and outlines would not be represented by single nodes, but rather by associated weight patterns of multiple nodes.

A training data set is a data set of examples used during the learning process and is used to fit the parameters (e.g., weights) of, for example, a classifier.[9][10]

For classification tasks, a supervised learning algorithm looks at the training data set to determine, or learn, the optimal combinations of variables that will generate a good predictive model.[11] The goal is to produce a trained (fitted) model that generalizes well to new, unknown data.[12] The fitted model is evaluated using “new” examples from the held-out data sets (validation and test data sets) to estimate the model’s accuracy in classifying new data.[5] To reduce the risk of issues such as over-fitting, the examples in the validation and test data sets should not be used to train the model.[5]

Most approaches that search through training data for empirical relationships tend to overfit the data, meaning that they can identify and exploit apparent relationships in the training data that do not hold in general.

When a training set is continuously expanded with new data, then this is incremental learning.

Validation data set

[edit]

A validation data set is a data set of examples used to tune the hyperparameters (i.e. the architecture) of a model. It is sometimes also called the development set or the "dev set".[13] An example of a hyperparameter for artificial neural networks includes the number of hidden units in each layer.[9][10] It, as well as the testing set (as mentioned below), should follow the same probability distribution as the training data set.

In order to avoid overfitting, when any classification parameter needs to be adjusted, it is necessary to have a validation data set in addition to the training and test data sets. For example, if the most suitable classifier for the problem is sought, the training data set is used to train the different candidate classifiers, the validation data set is used to compare their performances and decide which one to take and, finally, the test data set is used to obtain the performance characteristics such as accuracy, sensitivity, specificity, F-measure, and so on. The validation data set functions as a hybrid: it is training data used for testing, but neither as part of the low-level training nor as part of the final testing.

The basic process of using a validation data set for model selection (as part of training data set, validation data set, and test data set) is:[10][14]

> Since our goal is to find the network having the best performance on new data, the simplest approach to the comparison of different networks is to evaluate the error function using data which is independent of that used for training. Various networks are trained by minimization of an appropriate error function defined with respect to a training data set. The performance of the networks is then compared by evaluating the error function using an independent validation set, and the network having the smallest error with respect to the validation set is selected. This approach is called the *hold out* method. Since this procedure can itself lead to some overfitting to the validation set, the performance of the selected network should be confirmed by measuring its performance on a third independent set of data called a test set.

An application of this process is in early stopping, where the candidate models are successive iterations of the same network, and training stops when the error on the validation set grows, choosing the previous model (the one with minimum error).

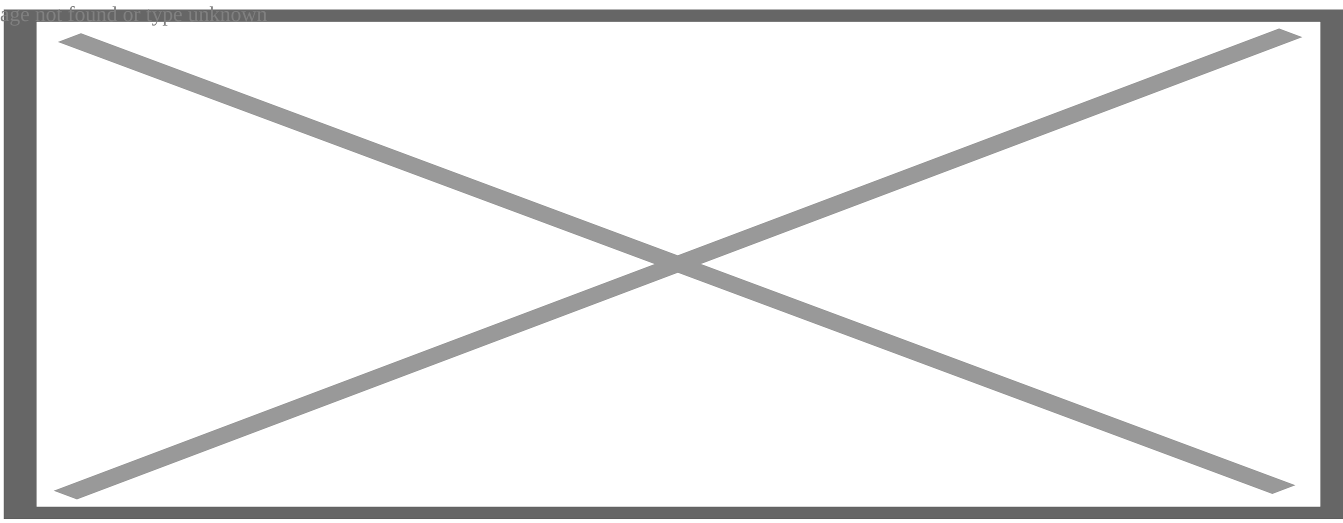
Test data set

[edit]

A test data set is a data set that is independent of the training data set, but that follows the same probability distribution as the training data set. If a model fit to the training data set also fits the test data set well, minimal overfitting has taken place (see figure below). A better fitting of the training data set as opposed to the test data set usually points to over-fitting.

A test set is therefore a set of examples used only to assess the performance (i.e. generalization) of a fully specified classifier.[9][10] To do this, the final model is used to predict classifications of examples in the test set. Those predictions are compared to the examples' true classifications to assess the model's accuracy.[11]

In a scenario where both validation and test data sets are used, the test data set is typically used to assess the final model that is selected during the validation process. In the case where the original data set is partitioned into two subsets (training and test data sets), the test data set might assess the model only once (e.g., in the holdout method).[15] Note that some sources advise against such a method.[12] However, when using a method such as cross-validation, two partitions can be sufficient and effective since results are averaged after repeated rounds of model training and testing to help reduce bias and variability.[5][12]



A training set (left) and a test set (right) from the same statistical population are shown as blue points. Two predictive models are fit to the training data. Both fitted models are plotted with both the training and test sets. In the training set, the MSE of the fit shown in orange is 4 whereas the MSE for the fit shown in green is 9. In the test set, the MSE for the fit shown in orange is 15 and the MSE for the fit shown in green is 13. The orange curve severely overfits the training data, since its MSE increases by almost a factor of four when comparing the test set to the training set. The green curve overfits the training data much less, as its MSE increases by less than a factor of 2.

Confusion in terminology

[edit]

Testing is trying something to find out about it ("To put to the proof; to prove the truth, genuineness, or quality of by experiment" according to the Collaborative International Dictionary of English) and to validate is to prove that something is valid ("To confirm; to render valid" Collaborative International Dictionary of English). With this perspective, the most common use of the terms **test set** and **validation set** is the one here described. However, in both industry and academia, they are sometimes used interchanged, by considering that the internal process is testing different models to improve (test set as a development set) and the final model is the one that needs to be validated before real use with an unseen data (validation set). "The literature on machine learning often reverses the meaning of 'validation' and 'test' sets. This is the most blatant example of the terminological confusion that pervades artificial intelligence research."[16] Nevertheless, the important concept that must be kept is that the final set, whether called test or validation, should only be used in the final experiment.

Cross-validation

[edit]

In order to get more stable results and use all valuable data for training, a data set can be repeatedly split into several training and a validation data sets. This is known as cross-validation. To confirm the model's performance, an additional test data set held out from cross-validation is normally used.

It is possible to use cross-validation on training and validation sets, and *within* each training set have further cross-validation for a test set for hyperparameter tuning. This is known as nested cross-validation.

Causes of error

[edit]

Image not found or type unknown

Comic strip demonstrating a fictional erroneous computer output (making a coffee 5 million degrees, from a previous definition of "extra hot"). This can be classified as both a failure in logic and a failure to include various relevant environmental conditions.[17]

Omissions in the training of algorithms are a major cause of erroneous outputs.[17] Types of such omissions include:[17]

- Particular circumstances or variations were not included.
- Obsolete data
- Ambiguous input information
- Inability to change to new environments
- Inability to request help from a human or another AI system when needed

An example of an omission of particular circumstances is a case where a boy was able to unlock the phone because his mother registered her face under indoor, nighttime lighting, a condition which was not appropriately included in the training of the system.[17][18]

Usage of relatively irrelevant input can include situations where algorithms use the background rather than the object of interest for object detection, such as being trained by pictures of sheep on grasslands, leading to a risk that a different object will be interpreted as a sheep if located on a grassland.[17]

See also

[edit]

- Statistical classification
- List of datasets for machine learning research
- Hierarchical classification

References

[edit]

1. **^** *Ron Kohavi; Foster Provost (1998). "Glossary of terms". Machine Learning. **30**: 271–274. doi:10.1023/A:1007411609915.*
2. **^** *Bishop, Christopher M. (2006). Pattern Recognition and Machine Learning. New York: Springer. p. vii. ISBN 0-387-31073-8. "Pattern recognition has its origins in engineering, whereas machine learning grew out of computer science. However, these activities can be viewed as two facets of the same field, and together they have undergone substantial development over the past ten years."*
3. ^ ***a b*** *James, Gareth (2013). An Introduction to Statistical Learning: with Applications in R. Springer. p. 176. ISBN 978-1461471370.*
4. ^ ***a b*** *Ripley, Brian (1996). Pattern Recognition and Neural Networks. Cambridge University Press. p. 354. ISBN 978-0521717700.*
5. ^ ***a b c d e f*** *Brownlee, Jason (2017-07-13). "What is the Difference Between Test and Validation Datasets?". Retrieved 2017-10-12.*
6. ^ ***a b*** *Prechelt, Lutz; Geneviève B. Orr (2012-01-01). "Early Stopping — But When?". In Grégoire Montavon; Klaus-Robert Müller (eds.). Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science. Springer Berlin Heidelberg. pp. 53–67. doi:10.1007/978-3-642-35289-8_5. ISBN 978-3-642-35289-8.*
7. **^** *"Machine learning - Is there a rule-of-thumb for how to divide a dataset into training and validation sets?". Stack Overflow. Retrieved 2021-08-12.*
8. **^** *Ferrie, C., & Kaiser, S. (2019). Neural Networks for Babies. Sourcebooks. ISBN 978-1492671206.*cite book: CS1 maint: multiple names: authors list (link)
9. ^ ***a b c*** Ripley, B.D. (1996) *Pattern Recognition and Neural Networks*, Cambridge: Cambridge University Press, p. 354
10. ^ ***a b c d*** "Subject: What are the population, sample, training set, design set, validation set, and test set?", Neural Network FAQ, part 1 of 7: Introduction (txt), comp.ai.neural-nets, Sarle, W.S., ed. (1997, last modified 2002-05-17)
11. ^ ***a b*** *Larose, D. T.; Larose, C. D. (2014). Discovering knowledge in data : an introduction to data mining. Hoboken: Wiley. doi:10.1002/9781118874059. ISBN 978-0-470-90874-7. OCLC 869460667.*
12. ^ ***a b c*** *Xu, Yun; Goodacre, Royston (2018). "On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning". Journal of Analysis and Testing. **2** (3). Springer Science and Business Media LLC: 249–262. doi:10.1007/s41664-018-0068-2. ISSN 2096-241X. PMC 6373628. PMID 30842888.*
13. **^** *"Deep Learning". Coursera. Retrieved 2021-05-18.*
14. **^** Bishop, C.M. (1995), *Neural Networks for Pattern Recognition*, Oxford: Oxford University Press, p. 372
15. **^** *Kohavi, Ron (2001-03-03). "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection". **14**.* cite journal: Cite journal requires

|journal= (help)

16. **^** *Ripley, Brian D. (2008-01-10). "Glossary". Pattern recognition and neural networks. Cambridge University Press. ISBN 9780521717700. OCLC 601063414.*
17. ^ ***a b c d e*** *Chanda SS, Banerjee DN (2022). "Omission and commission errors underlying AI failures". AI Soc.* ***39*** *(3): 1–24. doi:10.1007/s00146-022-01585-x. PMC 9669536. PMID 36415822.*
18. **^** *Greenberg A (2017-11-14). "Watch a 10-Year-Old's Face Unlock His Mom's iPhone X". Wired.*

- v
- t
- e

Artificial intelligence (AI)

- History
    - timeline
- Companies
- Projects

Concepts

- Parameter
  - Hyperparameter
- Loss functions
- Regression
  - Bias–variance tradeoff
  - Double descent
  - Overfitting
- Clustering
- Gradient descent
  - SGD
  - Quasi-Newton method
  - Conjugate gradient method
- Backpropagation
- Attention
- Convolution
- Normalization
  - Batchnorm
- Activation
  - Softmax
  - Sigmoid
  - Rectifier
- Gating
- Weight initialization
- Regularization
- Datasets
  - Augmentation
- Prompt engineering
- Reinforcement learning
  - Q-learning
  - SARSA
  - Imitation
  - Policy gradient
- Diffusion
- Latent diffusion model
- Autoregression
- Adversary
- RAG
- Uncanny valley
- RLHF
- Self-supervised learning
- Reflection
- Recursive self-improvement
- Hallucination
- Word embedding
- Vibe coding

Applications

- Machine learning
    - In-context learning
- Artificial neural network
    - Deep learning
- Language model
    - Large language model
    - NMT
- Reasoning language model
- Model Context Protocol
- Intelligent agent
- Artificial human companion
- Humanity's Last Exam
- Artificial general intelligence (AGI)

Implementations

Audio–visual

- AlexNet
- WaveNet
- Human image synthesis
- HWR
- OCR
- Computer vision
- Speech synthesis
  - 15.ai
  - ElevenLabs
- Speech recognition
  - Whisper
- Facial recognition
- AlphaFold
- Text-to-image models
  - Aurora
  - DALL-E
  - Firefly
  - Flux
  - Ideogram
  - Imagen
  - Midjourney
  - Recraft
  - Stable Diffusion
- Text-to-video models
  - Dream Machine
  - Runway Gen
  - Hailuo AI
  - Kling
  - Sora
  - Veo
- Music generation
  - Riffusion
  - Suno AI
  - Udio
- Word2vec
- Seq2seq
- GloVe
- BERT
- T5
- Llama
- Chinchilla AI
- PaLM
- GPT
  - 1
  - 2
  - 3
  - J
  - ChatGPT

People

- Alan Turing
- Warren Sturgis McCulloch
- Walter Pitts
- John von Neumann
- Claude Shannon
- Shun'ichi Amari
- Kunihiko Fukushima
- Takeo Kanade
- Marvin Minsky
- John McCarthy
- Nathaniel Rochester
- Allen Newell
- Cliff Shaw
- Herbert A. Simon
- Oliver Selfridge
- Frank Rosenblatt
- Bernard Widrow
- Joseph Weizenbaum
- Seymour Papert
- Seppo Linnainmaa
- Paul Werbos
- Geoffrey Hinton
- John Hopfield
- Jürgen Schmidhuber
- Yann LeCun
- Yoshua Bengio
- Lotfi A. Zadeh
- Stephen Grossberg
- Alex Graves
- James Goodnight
- Andrew Ng
- Fei-Fei Li
- Alex Krizhevsky
- Ilya Sutskever
- Oriol Vinyals
- Quoc V. Le
- Ian Goodfellow
- Demis Hassabis
- David Silver
- Andrej Karpathy
- Ashish Vaswani
- Noam Shazeer
- Aidan Gomez
- John Schulman
- Mustafa Suleyman
- Jan Leike
- Daniel Kokotajlo
- François Chollet

Architectures

- Neural Turing machine
- Differentiable neural computer
- Transformer
    - Vision transformer (ViT)
- Recurrent neural network (RNN)
- Long short-term memory (LSTM)
- Gated recurrent unit (GRU)
- Echo state network
- Multilayer perceptron (MLP)
- Convolutional neural network (CNN)
- Residual neural network (RNN)
- Highway network
- Mamba
- Autoencoder
- Variational autoencoder (VAE)
- Generative adversarial network (GAN)
- Graph neural network (GNN)

- Image not found or type unknown Category

About Search engine optimization

Seo (SEARCH ENGINE OPTIMIZATION) is the process of boosting the high quality and quantity of web site web traffic to a website or a websites from internet search engine. SEO targets unpaid search web traffic (generally referred to as "organic" outcomes) as opposed to straight website traffic, referral web traffic, social media sites web traffic, or paid traffic. Organic online search engine traffic originates from a range of type of searches, consisting of image search, video search, scholastic search, information search, industry-specific vertical online search engine, and huge language versions. As an Internet marketing strategy, SEO takes into consideration just how online search engine work, the formulas that determine online search engine results, what individuals look for, the actual search queries or search phrases keyed in right into search engines, and which internet search engine are favored by a target market. SEO assists web sites attract more visitors from a search engine and rank higher within a search engine results web page (SERP), aiming to either convert the visitors or build brand name awareness.
.

About Prompt engineering

Trigger design is the process of structuring or crafting a direction in order to create far better results from a generative expert system (AI) model. A prompt is all-natural language message explaining the job that an AI need to carry out. A timely for a text-to-text language version can be a query, a command, or a much longer statement consisting of context, instructions, and discussion background. Motivate engineering might entail phrasing a query, defining a design, selection of words and grammar, offering appropriate context, or defining a personality for the AI to resemble. When connecting with a text-to-image or a text-to-audio design, a typical timely is a description of a preferred output such as "a top notch photo of an astronaut riding a horse" or "Lo-fi slow BPM electro chill with natural samples". Motivating a text-to-image version may include including, eliminating, or stressing words to attain a desired topic, style, format, lights, and visual.

.

Frequently Asked Questions

How can I prompt agents in a debate format, to effectively integrate new information or adapt to unforeseen circumstances encountered during the debate?

Design prompts that allow agents to update their knowledge base based on new evidence presented by their opponents. Provide mechanisms for self-reflection and revision of arguments. Allow access to external knowledge sources or APIs during the debate. Prompt the agents to summarize and analyze the current state of the debate regularly, to identify gaps in reasoning or new directions for exploration.

Sitemap

Privacy Policy

About Us