Introducing a Reputation System into the NARS Environment

CIS 5590 – Topics in Computer Science
Temple University

by

Phillip Shebel
Darlin Kola

Instructor: Dr. Pei Wang

May 2, 2025

Introduction:
This project explores introducing a reputation system into the nars environment. This began when thinking about how a nars system would learn in practice, and through discussion with Dr.Wang we agreed that it would probably look more like teaching a child rather than training a machine. This led us to think about how the nars system would behave when exposed to many different teachers. Something that we all do growing up, but also something heavily explored in systems, is reputation. You may, for example, trust your teacher more than you trust your older brother, or a primary source more than a secondary, etc. We attempt to introduce this notion into nars to compare the accuracy of the system with and without a reputation system.

Design of the Reputation System:
Design of the Reputation System: The goal of the system is to use the previous data a user has submitted to get a sense of how often they are correct and how confident they are in their answers. The system will then be used only to update the confidence values of the users' submissions. This would be useful in a situation where a user is often correct but not sure of themselves. The system would add a positive weight to their confidence values to compensate for this user behavior. Another illustrative example would be the troll, someone who submits many false statements with high confidence. This system would ideally identify such users and apply a negative weight to their confidence values.

Our system was written in Python, and maps users to a reputation score. This score starts at zero, and increases or decreases by some learning rate based on the user's responses compared to the consensus. We then apply the reputation score to users' confidence values. A user who often agrees with consensus would end with a positive weight, and vice versa.

Reputation Systems in Other Contexts: Before jumping into our own system, we thought it would be helpful to take a look at how reputation systems work in other environments. These systems are everywhere—from online marketplaces to community forums—and they all try to answer the same basic question: *can I trust this person?*

One classic example is eBay, where buyers and sellers rate each other after every transaction. Over time, users build up a reputation score based on this feedback. If someone scams you, you can leave a negative review, and their reputation takes a hit. The idea is that the more positive reviews you have, the more trustworthy you are. This system works decently well, but it has some well-known issues. For one, most people don't leave negative reviews, even if something goes wrong. And sometimes users inflate each other's scores on purpose. So while it helps, it's not perfect.

Then there's Stack Overflow, where users earn reputation points by asking good questions and giving helpful answers. People can upvote or downvote your posts, and the more upvotes you get, the more reputation you earn. As your reputation grows, you unlock privileges like editing other people's posts or closing questions. Unlike eBay, this system isn't really about trust in a social sense—it's more about recognizing helpful contributions. But it's still a reputation system, and a pretty effective one.

Reddit also uses reputation through upvotes and downvotes, and users collect something called karma. The more karma you have, the more active and (in theory) respected you are in the community. But this system is more volatile, and it can easily be swayed by groupthink or ideology. Sometimes the most popular answer isn't the most accurate—it's just the one that fits the mood of the crowd.

Even in machine learning, there are reputation-like ideas. In boosting, a technique where you combine multiple weak models to make a strong one, the system increases the influence of models that perform well. Over time, models that are consistently wrong get less say, and the ones that are right more often have a bigger impact. It's a form of reputation, just applied to algorithms instead of people.

Comparing to Our System: Our reputation system in nars is similar to these in spirit, but works a little differently in practice. We don't have user ratings or upvotes—instead, we try to figure out how often a user is right by comparing their answers to the consensus. If someone is usually in agreement with everyone else, we assume they're probably more reliable, and we adjust their confidence scores accordingly. If someone is often wrong or intentionally misleading, like our troll user, their confidence gets weighted down.

Unlike Reddit or Stack Overflow, we're not measuring popularity or perceived helpfulness—we're using actual agreement with a shared belief system to guide reputation. And unlike boosting, we're not adjusting weights to minimize prediction error, but rather trying to help nars make better judgments by highlighting which sources are likely to be trustworthy.

Of course, our system assumes that consensus is usually right, which isn't always true. That's one of the risks of relying on group agreement as a proxy for truth. But in practice, especially when you have a lot of users and a mix of knowledge levels, it's a decent starting point—and much better than treating every input as equal, especially when trolls or noisy data are involved.

Experiment One:

Control: First we generate a set of boolean statements in nars, all either p implies q or p does not imply q, that will act as the ground truth. We then generated many users who are assigned a pseudo iq based on a normal distribution and a confidence value. We generate user responses by giving the users the statements, and having them respond correctly based on their iq. Users with higher iq answer correctly more often than those with lower. The same confidence score was given to each response corresponding to the user's confidence value. Users' responses are then given directly to nars, followed by a test to determine accuracy. The accuracy for each statement is the absolute value of the difference of the ground truth and what nars believes the truth to be.

To illustrate, here are some sample values for each step of the process:
Statement: <UHF --> JHS>. is true
User: correctness score of 106.3788, confidence score of 123.5197
User Response to statement: <UHF --> JHS>. %0.9999;0.7613%

Test question: <UHF --> JHS>?
Test Response: Answer <UHF --> JHS>. %0.76;0.99%
Accuracy: |0.9999 - 0.76| = 0.23

Reputation: In the reputation system, we take the user's responses but attempt to generate a score for each user that indicates how often the user is agreeing with consensus. This score is then applied to the confidence score that is submitted to nars. The motivation behind this is that a user with a high intelligence but low confidence should have their confidence raised, while a user with low intelligence but high confidence should have their confidence lowered.

In practice this would be an online algorithm, where reputation is changing over time rather than computed from the data all at once. We attempted to recreate this by gathering consensus for statements one at a time, and updating the users response and then the reputation after each round of statements.

Experiment Two:
This is essentially the same, but with one additional user we refer to as a troll. In real life this might be an older cousin who tells you to bury a $20 bill in order to grow a money tree, but in any system with user submitted data there will be people who act in bad faith for one reason or another. This experiment attempts to identify so-called trolls, and adjust their confidence score accordingly. To represent a troll, we add a single user who always answers incorrectly and with a maximum confidence value.

Results: These experiments were done with 50 users and 50 true false questions. Each experiment was repeated 5 times. The values in the table below are the sum of the difference between nars' belief and the ground truth. A score of 0 would indicate that nars has a perfect understanding of the statements, and the larger the score the worse the accuracy.

	Control	Reputation	Troll Control	Troll Reputation
Run 1	15.89	14.33	34.41	16.93
Run 2	16.49	14.88	37.33	18.08
Run 3	15.14	12.78	35.93	15.78
Run 4	15.37	13.92	37.62	17.19
Run 5	17.72	16.79	33.87	18.97
Average	16.122	14.54	35.832	17.39

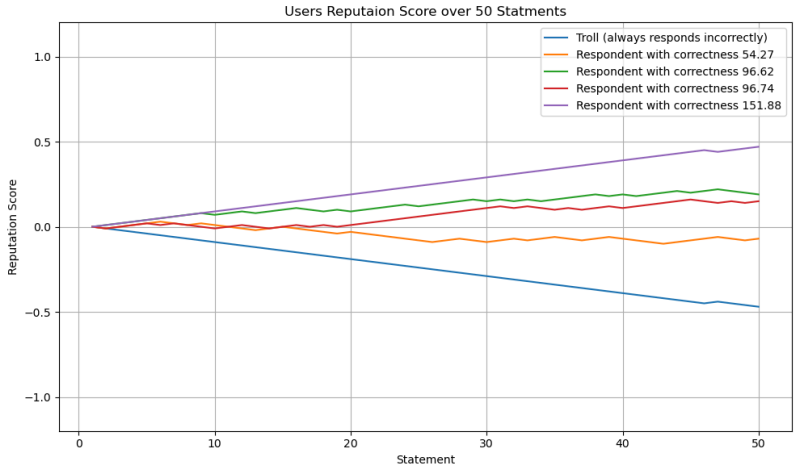
Analysis:
From our experiments we see a 9.8% improvement in accuracy when using the reputation system compared to the control, and a 51.4% improvement when a troll is added to the users. We take this as an indication that a reputation system may only provide marginal utility to the nars system when testing for accuracy, but a promising result for identifying malicious users.

While designing this experiment, we did consider the ethical implications of essentially removing a user from the system entirely. It felt a bit like doing AI eugenics. So perhaps it is for the best we got the result we did, and lead us to a more focused use of the reputation system. And indeed it is one of nars great strengths that it is able to change its beliefs based on new input, and not simply use consensus as it's value system.

Discussion:
We would like to acknowledge that these experiments are not particularly rigorous. We make many assumptions that would require greater research into sociology and ethics that were outside the scope of this project. The choice for correctness was based on a cursory look at human IQ ranges, while the confidence value was chosen based on the assumption it was also normal and not correlated to iq.

We would also accept criticism for the methodology, and invite others to tinker and argue for changes. For example, the way in which users select their responses is by selecting a random number between 0 and 145, and then if their IQ is greater than the number, they answer correctly, otherwise they answer incorrectly. To give some insight as to why we chose this, we imagined a great deal of topics being general knowledge, such as the sky being blue, while fewer being specialized knowledge, such as subset-sum being np-complete. This does introduce a bias in the results as a large percentage would be answered correctly.

Another interesting choice was how to apply and update the reputation of users. We experimented with other approaches such as gradient descent, but opted for something less dramatic. Our thinking was that each response tells us very little about a user, and so reputation should grow slowly rather than having a scaled approach that can take big steps. We include a graph below to help illustrate how a user's reputation changes in our system.



Ultimately, this experiment should be done with humans providing responses to real questions. The purpose of this paper is to motivate the research into creating a reputation system for nars. While we accept the above critiques, we believe we have demonstrated evidence for how such a system could interact with nars, and how it could benefit the system.

Conclusion:

In this paper we have constructed a sample reputation system for nars. Through our experiments we show the potential benefits of such a system, including an accuracy improvement of 51% in the case where a user is providing malicious responses. Future work should involve tuning the reputation system as well as begin testing on human respondents.