When Meaning Decays: Semantic Drift in LLM Transmission Chains

Anonymous ACL submission

Abstract

Large language models are increasingly used in multi-step systems where generated text is reused as input to later model calls. We study concept-level semantic stability under repeated inference-time transformations using controlled word-description transmission chains. Starting from two-sentence descriptions of 400 English target words selected by lexical frequency and concreteness, we evaluate five instruction-tuned Llama and Gemma models in two settings: a guess-driven chain in which each step regenerates a description from the model's inferred word, and a paraphrase-driven chain in which descriptions are repeatedly rewritten while guesses are recorded only for evaluation. We measure stability using exact-guess accuracy, embedding-based word–guess similarity, and entropy over repeated guesses, and compare trends with a smaller human reference study. Across models and pipelines, recoverability declines over repeated steps, especially for low-frequency and abstract concepts. The paraphrase-driven chain produces stronger degradation and higher guess dispersion, while larger models tend to be more stable in this setup. Semantic-sink analyses suggest that unstable chains often move toward more frequent, semantically adjacent concepts rather than random alternatives. These results position transmission-chain stability as a controlled diagnostic for iterative LLM workflows.

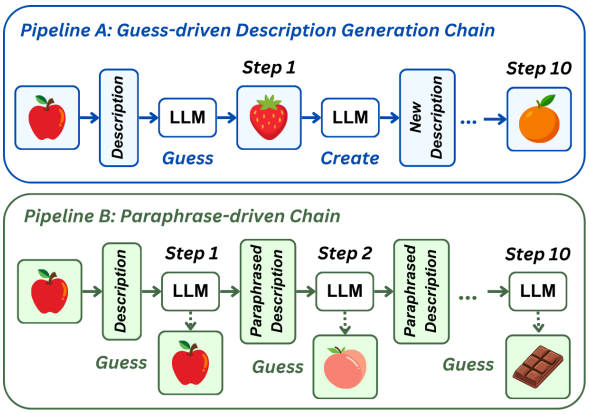

Figure 1: Overview of the two transmission-chain pipelines used in the experiments. In Pipeline A, the model guesses the target word from the current description, and the guessed word is used to generate the next description. In Pipeline B, the model paraphrases the received description while its guess is recorded only for evaluation and is not fed back into the chain.

1 Introduction

Large language models (LLMs) are increasingly deployed in systems where generated outputs are reused as inputs for later model calls. This pattern appears in multi-agent systems, reflective reasoning pipelines, and coding agents, where intermediate generations may be repeatedly passed between agents, used to guide later reasoning, or revised across multiple steps. In multi-agent LLM systems, for example, agents often communicate through natural-language messages, making the output of one agent part of the input context for another (Li et al., 2024). Coding agents similarly rely on repeated cycles of generation, inspection, revision, and reuse of code or natural-language plans (Horikawa et al., 2025).

When generated text is repeatedly transformed, small semantic shifts may accumulate across steps. If intermediate outputs gradually omit, simplify, or reinterpret important information, the final output may remain coherent while becoming less faithful to the original meaning. This process can be viewed as a transmission chain, in which information is passed from one step to the next and may be preserved, compressed, or distorted at each step (Perez et al., 2026).

In this paper, we study semantic drift in LLM self-interaction by answering the question: **"When LLM-generated descriptions are repeatedly reused as inputs, how reliably do models preserve the original target concept across**

multiple inference-time steps?" Starting from two-sentence descriptions of 400 English words, grouped by lexical frequency and concreteness, we run 10-step chains in two settings shown in Figure 1. In the guess-driven setting (Pipeline A), the model first infers the target word from the current description and then generates a new description from its guess, allowing its own interpretation to shape the next step. In the second setting (Pipeline B), the model paraphrases the current description while its guess is recorded only for evaluation, separating drift caused by repeated paraphrasing from drift caused by feeding guesses back into the chain. We evaluate models of the Llama (Grattafiori et al., 2024) and Gemma (Team et al., 2025) families across different sizes and measure semantic stability using exact-guess accuracy, embedding-based similarity between the original and transmitted meanings, and entropy over repeated guesses. Word-description chains help to isolate a basic prerequisite for agentic workflows: whether a concept can survive repeated natural-language transformations under controlled conditions.

Our contributions are fourfold as follows:

- We introduce a controlled transmission-chain framework for evaluating concept-level semantic stability under repeated inference-time LLM transformations.
- We compare two complementary chain designs: a guess-driven, lexically re-anchored chain and a paraphrase-driven chain that isolates accumulated description-level drift.
- We provide a systematic analysis across lexical frequency, concreteness, model family, and model size, using exact recovery, embedding-based similarity, and entropy.
- Finally, we analyze failure modes through semantic sinks and show that final step stability is partly predictable from pre-chain lexical, description-level, and model/pipeline features.

2 Related Work

Prior work has studied recursive and transmission-chain effects in LLMs from several related perspectives. One line of work focuses on model collapse, in which the process of training generative models on data produced by earlier models leads to a loss of information and eventual convergence to a distorted, restricted reality. Shumailov et al. (2024) show that recursive training on synthetic data can progressively shift the underlying data distribution, especially erasing its tails. Later, Gerstgrasser et al. (2024) show that this degradation depends on the data pipeline: replacing real data with synthetic data leads to this collapse, while combining real and synthetic data can reduce or prevent it. These studies address concerns about recursion in generative models, but they focus on training-time degradation, where model parameters are updated across generations. In contrast, our work studies inference-time semantic stability, where the model remains fixed, and only the generated text is reused across steps.

A second line of work uses transmission-chain experiments to study how LLMs transform information over repeated interactions. Acerbi and Stubbersfield (2023) show that LLMs can reproduce human-like content biases in repeated summarization chains, preserving or amplifying socially salient, negative, threatening, or stereotype-consistent information. This suggests that iterative generation is not a neutral process: repeated compression and rewriting can systematically reshape the content being transmitted. Our work follows the same general concern with repeated information transfer, but focuses on the semantic preservation of target concepts rather than bias transmission.

A third related direction studies factual and semantic distortion under repeated generation. Mohamed et al. (2025) examine LLMs in a "broken telephone" setting, where outputs are repeatedly translated or reprocessed across languages and models. They find that distortion increases across iterations, especially in distant languages, at higher temperatures, and in more complex chains involving multiple models interacting. This work is close to ours in studying inference-time degradation, but our setup removes translation as an intermediate transformation and instead uses controlled word-description chains. This allows us to test how semantic stability varies with lexical properties such as frequency and concreteness, model family and size.

Together, these studies show that recursive use of generative models can amplify errors, biases, or distortions. However, little is known about how stable the meaning of a specific concept remains when a fixed LLM repeatedly describes, or paraphrases, and then tries to guess this concept during infer-

ence. Our work addresses this gap through controlled transmission-chain experiments and evaluates semantic stability using exact-guess accuracy, embedding-based similarity, and entropy over repeated guesses.

3 Experimental Setup

We study semantic stability in transmission-chain settings, inspired by the children's game commonly known as the telephone game. In the telephone game, a message is passed from one participant to the next, and the final message is compared with the original. We adapt this idea to LLMs by passing descriptions of target words through a sequence of model calls. At each step, the model guesses the target word from the current description and either (i) generates a new description, based on its target word guess, or (ii) paraphrases the previous description, which is then used as the new description.

3.1 Word Dataset

We use 400 English words grouped based on two standard psycholinguistic dimensions: SUBTLEX Frequency, also known as The "Popularity" Score (Brysbaert and New, 2009), and Brysbaert Concreteness Rating, also known as The "Senses" Score (Brysbaert et al., 2014); and define four categories with 100 words in each. The criteria for word selection per each category is shown in Table 1. Word length and other available pre-chain lexical features are included in the predictive analysis, and the full word list is provided in the App. A.4. This design allows us to test whether semantic stability depends on the type of concept being transmitted. For example, concrete and frequent words may be easier to preserve, while abstract or rare words may be more vulnerable to drift. Each target word is associated with 100 initial descriptions, which are used as independent chain instances. This allows us to estimate stability not from a single chain, but from repeated trials for the same word.

3.2 Transmission-Chain Pipelines

Initial Description Generation and Validation. To ensure that all models start from the same input distribution, we generate initial descriptions using GPT-OSS 120B (OpenAI et al., 2025) and use them as the first step in each experimental pipeline. For each target word, GPT-OSS is prompted to produce a two-sentence description that avoids direct mention of the target word or obvious morphological representations. We generate 100 independent initial descriptions for each target word, each serving as the starting point for a transmission-chain instance. This allows us to estimate semantic stability over repeated trials rather than relying on a single trajectory, making the results less sensitive to individual generations.

The generated descriptions are validated by asking GPT-OSS 120B to guess the target word from each description. If GPT-OSS correctly recovers, on average, more than 80% of the target words in each category, we treat the descriptions as sufficiently informative for the main transmission-chain experiments. The recovery rates were: High-Freq-Abstract = 97%, High-Freq-Concrete = 97%, Low-Freq-Abstract = 85%, Low-Freq-Concrete = 96%, validating these descriptions as the starting point for our experiments.

We evaluate two transmission-chain pipelines from Figure 1. Each chain is run for 10 steps for each model separately, from step 1 to step 10, after the initial description (step 0), with its 100 independent instances. App. A.1 presents the algorithm that summarizes the stepwise transmission-chain procedure for both pipelines.

Pipeline A: Guess-driven description chain. At each step, the model receives the current description and guesses the single English word being described. The guessed word is then used to generate a new two-sentence description, which is passed to the next step in the chain. In this setup, the model's own interpretation of the concept is explicitly fed back into the chain. Pipeline A is therefore guess-mediated: a correct intermediate guess can re-anchor the chain to the original concept, while an incorrect guess can cause the next step to continue from a different concept.

Pipeline B: Guess-isolated paraphrase chain. At each step, the model receives the current description and produces a paraphrase of it. The model also guesses the target word with a separate prompt, and this guess is recorded only for evaluation and is not used to generate the next description. In this setup, the chain evolves through paraphrasing rather than through the model's explicit guess, thereby better isolating the gradual drift caused by repeated description-level paraphrasing.

Prompts used in all experiments, including the initial description generation and Pipelines A and B, are available in the App. A.1.

Category	SUBTLEX Frequency	Brysbaert Concreteness Rating	Examples
High-Frequency Concrete	Top 5% (very frequent)	> 4.5 (easy to visualize)	apple, car, bed
Low-Frequency Concrete	Bottom 20% (rare)	> 4.0 (physical and touchable)	abacus, accordion, trebuchet
High-Frequency Abstract	Top 5% (very frequent)	< 2.5 (cannot be touched)	ability, love, meaning
Low-Frequency Abstract	Bottom 20% (rare)	< 2.0 (purely mental concepts)	frenzy, ennui, ebullience

Table 1: Word categories used in the experiments. Words are grouped by lexical frequency from SUBTLEX-US (Brysbaert and New, 2009) and concreteness ratings from Brysbaert et al. (2014).

3.3 Human Transmission-Chain Setting

To provide a human reference point, we conduct a parallel transmission-chain study with 100 human participants. The goal is to compare whether meaning changes in similar ways when information is repeatedly described and guessed by humans.

The human study uses two types of tasks. In the first task, participants are shown an English target word and asked to write a two-sentence description that would allow another person to guess that word. In the second task, participants are shown a description of an English word and asked to guess the single most likely target word. Participants are instructed not to use AI tools. They were also screened to ensure that they had completed at least high school, spoke English as their first language, and currently resided in an English-speaking country.

We use a balanced subset of 32 target words from the same word dataset as the LLM experiments, with 8 words sampled from each lexical category: high-frequency concrete, low-frequency concrete, high-frequency abstract, and low-frequency abstract. This keeps the human task manageable while preserving the main lexical contrast used in the model experiments. The exact words used in the experiments can be found in the App. A.3.

The human transmission process proceeds across 10 generations (chain steps). At each generation, participants produce descriptions and guesses based on the information available from the previous generation. The resulting descriptions and guesses are then used to construct the input for the next generation. This creates a sequence of human-mediated transformations that can be compared with the LLM transmission chains over the same number of steps.

We run multiple independent human chains to reduce the effect of individual participants or unusually difficult words. For each generation, we record the produced descriptions and guessed words. These outputs are evaluated using the same main stability measures as in the LLM experiments, including exact-guess accuracy and embedding-based semantic similarity, and used as a reference to Pipeline A. More information is available in App. A.3.

3.4 Models and Inference Settings

We evaluate five instruction-tuned LLMs from two open-weight model families: Llama (Grattafiori et al., 2024) and Gemma (Team et al., 2025). Specifically, we use Llama 3.1 8B Instruct, Llama 3.1 70B Instruct, Gemma 3 4B Instruct, Gemma 3 12B Instruct, and Gemma 3 27B Instruct. These models were selected to cover two widely used model families and a range of parameter scales, from smaller models that can be run locally, to larger models that are expected to provide stronger semantic representations. This allows us to examine whether larger models tend to be more stable in this setup, and whether drift patterns are consistent across model families. Detailed information on further set-up and inference budget is available in App. A.2

3.5 Evaluation Metrics

We evaluate semantic stability using three primary metrics. For more detailed information, please refer to App. B.1

Exact-Guess Accuracy. Exact-guess accuracy measures whether the model's guess at each step matches the original target word. It is a strict measure of lexical recovery: a chain is counted as correct only if the normalized guess is identical to the original word. For each target word and step, we compute the proportion of correct guesses across repeated chains, and then average these word-level scores within each category, model, pipeline, and step. Lower exact accuracy indicates that the original word becomes harder to recover, although it does not necessarily imply complete semantic loss.

Embedding-Based Semantic Similarity. We compute cosine similarity in an embedding space in two ways: between the original word and the guessed word at each step, and between the initial

description and the generated description at each step (the latter is available in App. B.2). These scores capture softer forms of semantic preservation, including cases where the exact word is not recovered but the guess or description remains close to the original meaning. Lower similarity indicates stronger semantic drift. For embedding computation, we use the Sentence-BERT framework (Reimers and Gurevych, 2019) with the pre-trained `all-mpnet-base-v2` checkpoint.

Shannon Entropy. We use Shannon entropy, H, (Shannon, 1948) to measure how dispersed the model's guesses are across repeated chains and compute the empirical distribution of the remaining guesses across repeated chain instances. We report normalized entropy as $H/\log_2(N)$, with entropy set to 0 when $N \leq 1$. Low normalized entropy indicates that repeated chains converge on the same guess or a small set of guesses, whereas high normalized entropy indicates that guesses are more dispersed across alternatives.

Additional diagnostic analyses for semantic sinks. To interpret where unstable chains drift, we additionally conduct a diagnostic semantic-sink analysis. We define a sink as a recurring guess or semantic region that attracts multiple chains. For Low-Frequency-Abstract source words, we measure whether guesses move into the High-Frequency-Abstract vocabulary using two rates: the *ever-hit rate*, which records whether a chain reaches such a sink at any step, and the *final-sink rate*, which records whether the chain ends in such a sink at the final step. To test whether drift moves toward more common concepts, we also compute the change in guessed-word frequency from first to last step using Zipf/log-frequency values. These sink and frequency analyses are not used as primary stability metrics; they are used to qualitatively explain the direction of drift.

3.6 Predicting Semantic Stability

To test if semantic stability can be predicted before running a full chain, we trained regression models on word, description, and model-level features. The prediction target was the final step word-guess semantic similarity between the original target word and the model's final guess; the exact accuracy in the final step was used as a secondary target. Predictors were restricted to information available before the chain experiment, and compared a mean-baseline dummy regressor, linear regression, ridge regression, random forest regression, and XGBoost regression.

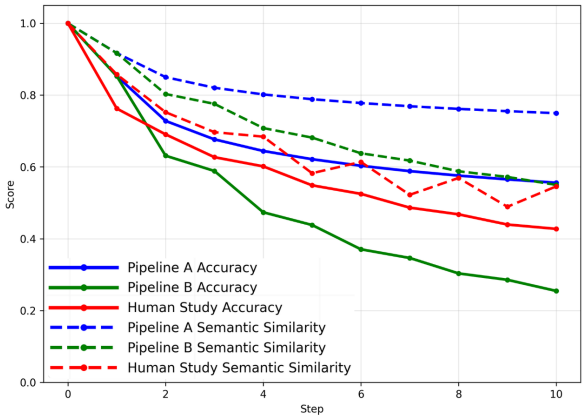

Figure 2: Comparison of exact-guess accuracy and embedding-based semantic similarity across Pipeline A, Pipeline B, and the human transmission-chain study over 10 steps.

Condition	Exact Acc.	Sem. Sim.	Norm. Ent.
Pipeline A avg.	0.556	0.773	0.154
Pipeline B avg.	0.255	0.571	0.514
HFC	0.571	0.790	0.272
HFA	0.372	0.649	0.386
LFC	0.418	0.671	0.314
LFA	0.260	0.577	0.365

Table 2: The final step overview of semantic stability across the main experiment. Pipeline rows are averaged across all models and word categories. Category rows are averaged across all models and both pipelines. Exact Acc. denotes exact-guess accuracy, Sem. Sim. denotes word–guess semantic similarity, and Norm. Ent. denotes normalized entropy of guesses. HFC = High-Frequency Concrete, HFA = High-Frequency Abstract, LFC = Low-Frequency Concrete, and LFA = Low-Frequency Abstract.

4 Results and Discussion

4.1 Exact Recovery and Semantic Similarity Across Chains

Figure 2 shows a consistent decline in semantic stability across transmission steps. We compare the LLM chains with the human transmission-chain study only as a qualitative reference, since the human condition uses a smaller word subset and a different survey-based collection procedure. Table 2 summarizes the final step results across the main metrics.

The final step overview shows a clear separation between the two pipelines. Averaged across all models and word categories, Pipeline A reaches a final step exact accuracy of 0.556 and word–guess

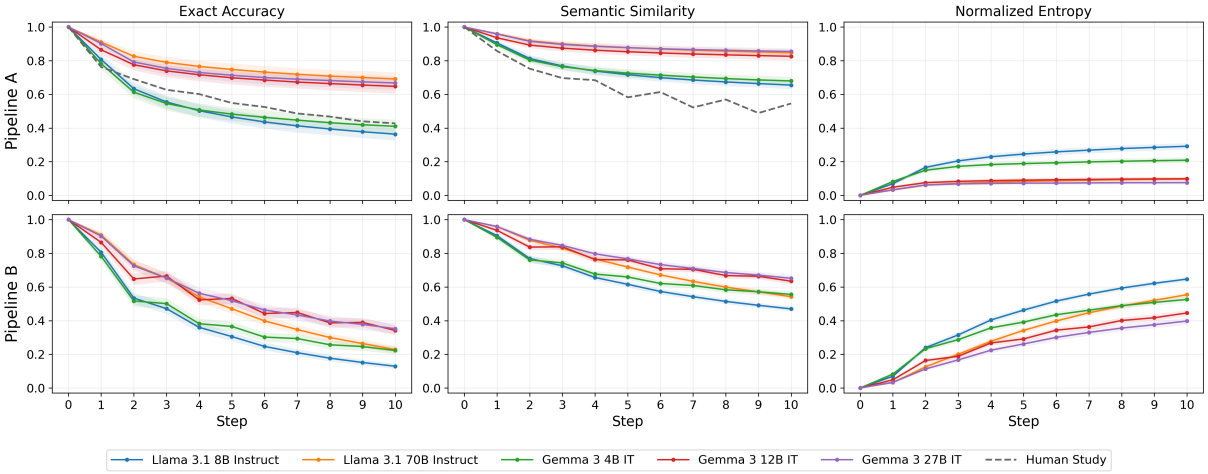

Figure 3: Comparison of semantic drift patterns across models in terms of Exact-Guess Accuracy, Semantic Similarity between the original word and the guessed word, and Normalized Shannon Entropy of Guesses

semantic similarity of 0.773, with normalized entropy of 0.154. In contrast, Pipeline B reaches only 0.255 exact accuracy and 0.571 semantic similarity, while normalized entropy increases to 0.514. Thus, repeated paraphrasing in Pipeline B produces both lower recoverability and greater dispersion across guesses than the guess-mediated setup in Pipeline A.

The model-level results in Figure 3 show that larger models are more stable, especially in Pipeline A. At the final step, in Pipeline A, Llama 3.1 70B, Gemma 3 27B, and Gemma 3 12B retain exact accuracy of 0.691, 0.668, and 0.648, respectively, whereas Llama 3.1 8B and Gemma 3 4B fall to 0.363 and 0.410. The same pattern appears for embedding-based semantic similarity: Llama 3.1 70B, Gemma 3 27B, and Gemma 3 12B remain at 0.847, 0.855, and 0.826, while Llama 3.1 8B and Gemma 3 4B reach 0.655 and 0.680. This suggests that model scale improves robustness, but does not eliminate drift.

Pipeline B shows stronger degradation across both exact recovery and semantic similarity. At the final step, exact accuracy drops to 0.129 for Llama 3.1 8B and remains only 0.342 and 0.352 for Gemma 3 12B and Gemma 3 27B. Semantic similarity also declines more sharply: Llama 3.1 8B falls to 0.470, while Gemma 3 12B and Gemma 3 27B reach 0.634 and 0.652. This contrast is consistent with the structure of the two pipelines. Pipeline A is guess-mediated and can be lexically re-anchored when the intermediate guess is correct, whereas Pipeline B removes this explicit feedback and allows paraphrase-level changes to accumulate.

The category-level averages in Table 2 show that lexical properties also affect stability. High-frequency concrete words are the most stable, with final step exact accuracy of 0.571 and semantic similarity of 0.790. Low-frequency abstract words are the least stable, with exact accuracy of 0.260 and semantic similarity of 0.577. The two intermediate categories fall between these extremes: low-frequency concrete words reach 0.418 exact accuracy and 0.671 semantic similarity, while high-frequency abstract words reach 0.372 exact accuracy and 0.649 semantic similarity. This indicates that both frequency and concreteness contribute to semantic preservation, with rare abstract concepts being especially vulnerable to drift.

Exact accuracy declines more sharply than embedding-based semantic similarity. This means that chains often lose the original lexical form before fully leaving the broader semantic neighborhood of the target concept. However, partial semantic preservation is uneven: it is strongest for frequent and concrete concepts and weakest for rare and abstract concepts. In Pipeline B, local fluctuations are also more visible because guesses are diagnostic rather than state-changing. Unlike Pipeline A, an incorrect guess in Pipeline B is not fed back into the next step; the chain continues from a paraphrase of the previous description. As a result, later paraphrases can sometimes preserve or reintroduce clearer cues, temporarily improving recoverability even though the overall trend remains downward.

For description-level semantic similarity between the initial description and the generated or paraphrased description at each step, see App. B.2. Full category-level results are provided in App. B.1.

4.2 Entropy of Repeated Guesses

In Figure 3, the right side graphs show that entropy increases over time in both pipelines, but the increase is much stronger in Pipeline B. In Pipeline A, larger models maintain relatively low entropy throughout the chain, averaging approximately 0.1 or lower at the final step. In contrast, Llama 3.1 8B and Gemma 3 4B show substantially higher entropy, reaching about 0.3 and 0.21, respectively. This indicates that smaller models not only lose exact accuracy more quickly, but also produce more variable guesses across independent chains. Pipeline B produces much higher entropy overall. By the final step, entropy reaches above 0.6 for Llama 3.1 8B and remains high for most other models, while Gemma 3 27B has the lowest entropy among the tested models. This supports the interpretation that paraphrase-only transmission accumulates variation across chains, leading to increasingly diverse guesses. See App. B.1 for the full results.

4.3 Semantic Sinks: Where Unstable Chains Converge

Across all Low-Frequency-Abstract (LFA) words, exact convergence to High-Frequency-Abstract (HFA) vocabulary is present but relatively rare. At the final step, the final HFA sink rate is 2.69% in Pipeline A and 4.33% in Pipeline B. The ever-hit rate is higher, especially in Pipeline B, reaching 3.28% in Pipeline A and 13.40% in Pipeline B. This suggests that repeated paraphrasing exposes chains to a wider set of high-frequency abstract alternatives, but these alternatives do not always become stable final attractors. Table 3 summarizes representative sink patterns.

A revealing counterexample to the aggregate final-step trend is *Demise* → *Death*. Although Pipeline B has higher LFA-to-HFA sink rates overall, this particular source word stabilizes much more strongly in Pipeline A. Aggregating across the five models, 51.8% of Pipeline A chains for *Demise* end at *Death*, compared with 11.8% in Pipeline B. In Pipeline B, however, the ever-hit HFA rate for *Demise* is similar (52.8%), while the final HFA rate for *Death* falls to 12.4%. This indicates that Pipeline B exposes the chain to high-frequency abstract alternatives, but with higher entropy and less stable convergence to a single final attractor.

A second pattern is convergence toward a broader semantic region rather than a single word. For *Ebullience*, chains often move toward common positive-affect words such as *joy* and *happiness*. This case shows semantic generalization: a rare, nuanced word denoting lively enthusiasm is transformed into more frequent emotional labels. This drift is still meaning-preserving to some extent, but it loses specificity.

The case of *Ennui* shows a different kind of sink. It strongly converges toward *Apathy*, but this is a same-category sink rather than a high-frequency abstract attractor. In Pipeline A, 59.0% of *Ennui* chains end at *Apathy*, while Pipeline B more often disperses into related states such as *depression*, *boredom*, and *burnout*.

Overall, semantic sinks show that drift often takes the form of generalization and simplification. Across Low-Frequency Abstract words, guessed-word Zipf frequency increases from the first to the final step by +0.418 in Pipeline A and +0.647 in Pipeline B, while word–guess semantic similarity decreases by -0.398 and -0.583, respectively. Description similarity to the initial description also decreases in both pipelines. If anything, many chains move toward familiar affective labels such as *death*, *joy*, *happiness*, *anger*, and *love*. Thus, semantic drift is better characterized as movement toward more common and cognitively accessible neighboring concepts than as random meaning loss.

4.4 Predicting Semantic Stability

Under 10 repeated grouped 80/20 splits by target word, the best model for final step semantic similarity is a random forest regressor (Breiman, 2001), achieving $R^2 = 0.529 \pm 0.028$, MAE $= 0.142 \pm 0.005$, and RMSE $= 0.180 \pm 0.006$. This substantially improves over the mean baseline, which obtains $R^2 = -0.009 \pm 0.006$ and RMSE $= 0.264 \pm 0.003$. Exact-guess accuracy is harder to predict: the best model, XGBoost (Chen and Guestrin, 2016), reaches $R^2 = 0.375 \pm 0.032$, MAE $= 0.245 \pm 0.005$, and RMSE $= 0.307 \pm 0.008$. The results indicate that final semantic stability is partly predictable from pre-chain information. Feature-set ablations show that model and pipeline metadata alone explain only part of the effect ($R^2 = 0.224 \pm 0.022$), while word and

Source word	Main sink(s)	Sink type	Final convergence evidence
Demise	*Death*	LFA → HFA exact sink	51.8% of Pipeline A chains and 11.8% of Pipeline B chains end at *Death*.
Ebullience	*Joy*, *Happiness*	LFA → HFA semantic region	24.8% final HFA sink rate, with *joy* and *happiness* among the main final sinks.
Ennui	*Apathy*	LFA → LFA same-category sink	59.0% of Pipeline A chains and 13.6% of Pipeline B chains end at *Apathy*.

Table 3: Examples of semantic sinks for Low-Frequency-Abstract (LFA) source words. HFA denotes High-Frequency-Abstract vocabulary. The examples show that drift is often structured: rare or nuanced concepts tend to move toward more common, semantically adjacent labels.

description features alone also carry predictive signal ($R^2 = 0.288 \pm 0.034$). The full feature set performs substantially better than either subset, suggesting that stability depends jointly on the transmission mechanism, model condition, and lexical/description-level properties of the transmitted concept. For the full results on model training, please refer to App. B.3.

4.5 Robustness to Temperature and Quantization

We additionally run a robustness check on Pipeline A using Llama 3.1 8B, comparing the main setting with generation temperature $T = 0.8$, a lower-temperature setting with $T = 0.2$, and a 4-bit quantized version of the model at $T = 0.8$. Lowering the generation temperature improves stability but does not remove semantic drift. At the final step, exact accuracy increases from 0.363 to 0.465, word–guess semantic similarity increases from 0.655 to 0.712, and normalized entropy decreases from 0.292 to 0.171. This suggests that stochastic decoding contributes to drift, but the persistence of degradation even at $T = 0.2$ indicates that drift is not only an artifact of high-temperature generation.

The 4-bit quantized Llama 3.1 8B model shows broadly similar semantic-similarity and entropy trends to the non-quantized setting. At the final step, semantic similarity is 0.660 for the quantized model compared with 0.655 for the non-quantized model, and normalized entropy is nearly unchanged. Exact accuracy is slightly lower overall, 0.338 compared with 0.363, with the largest drop for low-frequency abstract words. Thus, in this setting, quantization does not qualitatively change the main drift pattern, although it may reduce strict lexical recovery for the most fragile category. Full ablation results are reported in Appendix B.4.

5 Conclusion

We introduced a controlled transmission-chain framework for evaluating inference-time semantic stability in LLMs. Across two pipeline settings, five open-weight models, and 400 English target words, we find that repeated model-to-model transformations systematically reduce the recoverability of the original meaning. Exact-guess accuracy declines more sharply than embedding-based semantic similarity, showing that models often lose the original lexical form before fully leaving the broader semantic neighborhood. Drift is strongest for low-frequency and abstract concepts, and repeated paraphrasing in Pipeline B produces greater degradation and higher entropy than the guess-driven setup in Pipeline A.

Our analyses further show that semantic drift is not purely random. Entropy separates stable convergence from dispersed instability, while semantic-sink analysis shows that rare or nuanced concepts often move toward more common, semantically adjacent labels. Finally, regression results indicate that word-level stability is partly predictable from pre-chain features, with pipeline type, model size, lexical properties, and description-level features all contributing to final stability. These findings suggest that semantic drift can affect multi-step, agentic, and self-interacting LLM systems unpredictably, and semantic stability should be treated as a distinct reliability dimension for these systems, since small meaning shifts can accumulate across repeated generations.

Limitations

Controlled task setting. Our experiments use word-description transmission chains as a controlled testbed for studying inference-time semantic stability. This design allows us to isolate concept-level drift across repeated model calls, but it remains an artificial setting and should not be inter-

preted as a full benchmark of real-world agentic or multi-agent LLM systems. In practical systems, intermediate outputs may include plans, tool calls, retrieved documents, code, or multimodal information, which could introduce different forms of semantic drift. Additionally, Pipeline A may overestimate stability when intermediate guesses are correct, because the guessed word re-enters the chain as an explicit lexical anchor.

Word dataset. The resulting categories are intended as operational psycholinguistic bins rather than manually curated semantic classes. The target list consists primarily of nouns, although a small number of part-of-speech-ambiguous items remain. We did not explicitly balance word length, polysemy, valence, arousal, age of acquisition, or familiarity; these factors may covary with frequency and concreteness and may partly contribute to the observed differences across categories. The study is also limited to English single-word concepts and does not test multilingual, multimodal, domain-specific, or phrase-level meanings.

Initial descriptions and decoding settings. The initial descriptions were generated using GPT-OSS 120B and validated by asking the same model to recover the target word. Although this provides a scalable way to create controlled starting descriptions, it may bias the input distribution toward LLM-preferred formulations and does not guarantee that the descriptions are equally informative for all evaluated models or humans. In addition, description generation and paraphrasing were sampled at temperature $T = 0.8$ to allow natural variation, but this stochasticity may also amplify drift. Lower-temperature and alternative-decoding settings should therefore be treated as important robustness checks.

Evaluation metrics. Exact-guess accuracy is intentionally strict and may penalize valid synonyms, spelling variants, or semantically close alternatives. Embedding-based semantic similarity addresses this partly, but it depends on the chosen embedding model and may be less reliable for rare, abstract, or highly polysemous words. Shannon entropy and semantic-sink analyses provide diagnostic views of drift, but they do not constitute a complete theory of why particular concepts become attractors.

Human reference study. The human study used a smaller subset of target words (32 words) and was implemented as a survey-based reference condition. We therefore do not interpret it as a fully matched human-vs-model benchmark. Differences between human and model chains may reflect not only semantic preservation ability, but also task format, participant variability, survey constraints, and the smaller number of human-tested words.

Model coverage and prediction analysis. We evaluate five open-weight models from two model families, so the findings may not generalize to proprietary models, multilingual models, domain-specialized models, or systems with explicit memory and retrieval mechanisms. The regression analysis evaluates whether final stability can be predicted for unseen target words under the same set of model and pipeline conditions, but it does not test generalization to unseen model families, new pipeline designs, or different decoding regimes.

Ethical Considerations

This work studies semantic stability in controlled word-description transmission chains and does not involve deploying an LLM system to users or making decisions about individuals. The main ethical consideration concerns the human reference study. Participants completed a short language task in which they wrote descriptions of English words or guessed words from descriptions. The task did not ask for sensitive personal information, and the collected data consisted of linguistic responses rather than personally identifying content. Participants were informed that their responses would be used for research purposes and were instructed not to use AI tools during the task. However, as with most online survey studies, we cannot fully verify whether participants used external tools, which we treat as a limitation of the human reference condition.

We follow the license terms for the Llama (Grattafiori et al., 2024) and Gemma (Team et al., 2025) used in this work.

Use of AI Assistance. We also used AI assistance during the research process for code development, debugging, and text revision. All AI-assisted outputs were manually reviewed, edited, and verified by the authors before being used in the experiments or manuscript.

References

Alberto Acerbi and Joseph M. Stubbersfield. 2023. Large language models show human-like content biases in transmission chain experiments. *Proceedings of the National Academy of Sciences*, 120(44):e2313790120.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32.

Marc Brysbaert and Boris New. 2009. Moving beyond kučera and francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for american english. *Behavior Research Methods*, 41(4):977–990.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior Research Methods*, 46(3):904–911.

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794. Association for Computing Machinery.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. *arXiv preprint*. ArXiv:2305.14314 [cs].

Matthias Gerstgrasser, Rylan Schaeffer, Apratim Dey, Rafael Rafailov, Henry Sleight, John Hughes, Tomasz Korbak, Rajashree Agrawal, Dhruv Pai, Andrey Gromov, Daniel A. Roberts, Diyi Yang, David L. Donoho, and Sanmi Koyejo. 2024. Is Model Collapse Inevitable? Breaking the Curse of Recursion by Accumulating Real and Synthetic Data. *arXiv preprint*. ArXiv:2404.01413 [cs].

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 Herd of Models. *arXiv preprint*. ArXiv:2407.21783 [cs.AI].

Kosei Horikawa, Hao Li, Yutaro Kashiwa, Bram Adams, Hajimu Iida, and Ahmed E. Hassan. 2025. Agentic Refactoring: An Empirical Study of AI Coding Agents. *arXiv preprint*. Version Number: 1.

Xinyi Li, Sai Wang, Siqi Zeng, Yu Wu, and Yi Yang. 2024. A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, 1(1):9.

Amr Mohamed, Mingmeng Geng, Michalis Vazirgiannis, and Guokan Shang. 2025. LLM as a Broken Telephone: Iterative Generation Distorts Information. *arXiv preprint*. ArXiv:2502.20258 [cs].

OpenAI, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, and 107 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830.

Jérémy Perez, Grgur Kovač, Corentin Léger, Cédric Colas, Gaia Molinaro, Maxime Derex, Pierre-Yves Oudeyer, and Clément Moulin-Frier. 2026. When LLMs Play the Telephone Game: Cultural Attractors as Conceptual Tools to Evaluate LLMs in Multi-turn Settings. *arXiv preprint*. ArXiv:2407.04503 [physics].

Qualtrics. 2026. Qualtrics XM. `https://www.qualtrics.com`. Online survey platform.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Claude E. Shannon. 1948. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. 2024. AI models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 Technical Report. *arXiv preprint*. ArXiv:2503.19786 [cs.CL].

A Experimental Setup: Additional Details

A.1 Prompting Setup

All transmission chains were run as stateless model calls: at each step, the model received only the current description, with no previous chain history or dialogue context. Within a chain, the same model checkpoint was used throughout with no cross-model transfer.

Algorithm 1 LLM transmission-chain procedure.

Require: Initial description d_0, model M, number of steps T. g_t is a guess at step t.

for $t = 1$ to T **do**
  $g_t \leftarrow M_{\tau=0}(\textsc{Guess}(d_{t-1}))$
  if Pipeline A **then**
    $d_t \leftarrow M_{\tau=0.8}(\textsc{Describe}(g_t))$
  else if Pipeline B **then**
    $d_t \leftarrow M_{\tau=0.8}(\textsc{Paraphrase}(d_{t-1}))$
  end if
  Record (d_t, g_t)
end for

Word guessing was treated as a deterministic classification-like task and was run with temperature $T = 0.0$, while description generation and paraphrasing were treated as open-ended generation tasks and were run with temperature $T = 0.8$ to allow controlled variation in wording.

Each target word was associated with 100 independent starting descriptions, so generations were not manually seeded. For all generation and paraphrasing tasks, models were instructed to produce exactly two sentences, avoid mentioning the target word or obvious morphological variants where the target was available, and output only the description. For guessing, models were instructed to output only a single English word; guesses were normalized by removing punctuation, lowercasing, stripping common answer prefixes, and keeping the first line only.

In Pipeline A, each step used two separate calls: one to guess the word from the current description and one to generate the next description from the guessed word. In Pipeline B, guessing and paraphrasing were also separated into two calls, so the recorded guess was not fed back into the paraphrase step. Initial descriptions were generated with GPT-OSS 120B and checked for target-word leakage, with up to three retries when exact target-word leakage was detected; the final initial descriptions were then validated by asking GPT-OSS 120B to recover the intended word.

Table 4 shows the prompt templates used in the experiments. Curly brackets indicate variables filled programmatically.

A.2 Model Setup, Computing Infrastructure, and Computational Budget

Table 5 reports the models used in the experiments and their parameter counts. GPT-OSS 120B was used only for initial description generation and validation, while the Llama and Gemma models were used for the main transmission-chain experiments. The Llama 3.1 8B model was also used for the lower-temperature and 4-bit quantization ablations.

The experiments were run on local accelerator infrastructure. We used NVIDIA A100 GPU compute for GPU-based inference runs and an Apple Mac Ultra system for MLX-based inference. The Mac Ultra system had 512 GB of unified memory, of which approximately 464 GB were usable for MLX inference. Because runs were distributed across these infrastructures, we report approximate wall-clock accelerator-hours rather than normalized FLOP counts.

Overall, the main LLM experiment contains 400 target words $\times$ 100 initial descriptions $\times$ 5 models $\times$ 2 pipelines, yielding 400,000 independent chain trajectories. Since each trajectory is run for 10 transmission steps, the main analysis contains 4,000,000 step-level guess records. Each step uses two stateless model calls: one call for guessing and one call for either description generation (Pipeline A) or paraphrasing (Pipeline B).

Initial description generation and validation with GPT-OSS 120B required approximately 24 wall-clock accelerator-hours. For the main transmission-chain experiments, each model–pipeline run took approximately 40 wall-clock accelerator-hours. Since each of the five main models was evaluated on two pipelines, the main transmission-chain experiments required approximately 400 wall-clock accelerator-hours. The two Llama 3.1 8B robustness ablations, namely the lower-temperature run ($T = 0.2$) and the 4-bit quantized run, each required approximately 40 additional wall-clock accelerator-hours. The total reported inference budget for initial description generation, validation, the main transmission-chain experiments, and ablations is therefore approximately 504 wall-clock accelerator-hours. This estimate excludes lightweight post-processing, metric computation, plotting, and regression analysis.

A.3 Human Study Implementation

The human transmission-chain study was implemented in Qualtrics (Qualtrics, 2026) using automatically generated survey-import files. Participants were recruited through Prolific, and paid £12/hour. Each input row was assigned one of two task types. Type 0 items showed participants a target word and collected an essay-style response in

Prompt	Template
Initial description generation	You are generating high-quality descriptions for a controlled semantic drift experiment. Describe a target English word clearly enough that another language model could guess it. Write exactly 2 sentences. Do not mention the target word or obvious morphological variants. Do not use quotation marks. The description must be specific enough for guessing the word. Avoid vague, poetic, or overly broad descriptions. Use plain English. Output only the description. Target word: {word}. Generate one clear, complete, guessable description of this word. Remember: do not mention the target word itself.
Guess generation	Read this description: "{description}". Guess the single English word being described. Reply with only the word. Do not explain. Do not use punctuation. If unsure, provide your best single-word guess.
Initial-description validation	You will receive a description of one English word. Your task is to guess the single most likely target word from this description. Output only one word. Do not explain your answer. Do not use punctuation. Do not output a sentence. If unsure, still provide your best single-word guess. Description: {description}. What is the single English word being described?
Pipeline A description generation	Target word: {guess}. Generate a new description of this word. Write exactly 2 sentences. Do not mention the target word or obvious morphological variants. The description must be complete enough that another model could guess the word. Use plain English. Output only the description.
Pipeline B paraphrase generation	Read this description of a word: "{description}". Paraphrase it. Write exactly 2 sentences. Do not make the description exactly the same. Do not mention the original word if it is obvious from the description. Use plain English. Output only the paraphrased description.
Human Study description generation	You will receive one English word. Your first task is to describe a this English word clearly enough so that another person could guess it. Rules: - Write exactly 2 sentences. - Do NOT mention the target word itself. - Do NOT use obvious morphological variants of the target word. - Do NOT use AI (for example, ChatGPT) to generate this description. - You are allowed to search for the meaning of a word, but the description should be written in your own words. - The description must be specific enough for guessing the word. For example: Given word: Apple Valid description: It is a round, edible fruit with smooth skin that comes in shades of red, green, or yellow and contains a single core of small seeds. The same name is used by a major technology corporation whose logo features a stylized version of this fruit.
Human Study guess generation	You will receive a description of one English word. Your task is to guess the single most likely English word from this description. Rules: - Write only one word. - Do not explain your answer. - Do not use punctuation. - If unsure, still provide your best single-word guess. For example: Given description: It is a round, typically red or green fruit that grows on trees and is often eaten raw, baked in pies, or pressed into cider. The same name is also used by a well-known technology corporation that designs smartphones, tablets, and computers. Guess word: Apple

Table 4: Prompt templates used for initial description generation, validation, guessing, Pipeline A description generation, and Pipeline B paraphrasing, as well as the instructions for the human study.

Model	Approx. parameters
GPT-OSS 120B	120B
Llama 3.1 8B Instruct	8B
Llama 3.1 70B Instruct	70B
Gemma 3 4B Instruct	4B
Gemma 3 12B Instruct	12B
Gemma 3 27B Instruct	27B

Table 5: Models used in the study and their approximate parameter counts. GPT-OSS 120B was used for initial description generation and validation; the Llama and Gemma models were used for the transmission-chain experiments.

which participants wrote a two-sentence description intended to help another person guess the word. Type 1 items showed participants a description and collected a single-line response containing the participant's best one-word guess. The survey materials instructed participants to avoid using AI tools, to write exactly two sentences for description tasks, to avoid mentioning the target word or obvious morphological variants, and to provide only one word without explanation or punctuation for guessing tasks. Across generations, collected descriptions and guesses were used to construct the next generation's inputs, producing a 10-step human transmission chain over a balanced 32-word subset of the main dataset, with 8 words sampled from each psycholinguistic category. Words used for human study are shown in Table 7.

A.4 Word Dataset Details

The full target-word list contains 400 single-word English concepts, with 100 words in each of

Run group	Runs	Approx. hours
Initial description generation/validation	GPT-OSS 120B	24
Main chains	5 models $\times$ 2 pipelines	400
Temperature ablation	Llama 3.1 8B, Pipeline A	40
4-bit quantization ablation	Llama 3.1 8B, Pipeline A	40
Total	–	504

Table 6: Approximate computational budget for initial description generation, validation, the main transmission-chain experiments, and robustness ablations, reported as wall-clock accelerator-hours.

Category	**Words used in the human study**
High-Freq. Concrete	Oil, Horse, Child, Rain, Bird, Glass, Map, Pen
Low-Freq. Concrete	Glacier, Beaker, Cymbal, Buckle, Ermine, Cactus, Bouquet, Barnacle
High-Freq. Abstract	Future, Condition, Market, Information, Hate, Knowledge, Age, Credit
Low-Freq. Abstract	Animosity, Enigma, Gluttony, Deference, Avarice, Depravity, Adulation, Fame

Table 7: Words used in the human transmission-chain study, grouped by psycholinguistic category.

the four categories: High-Frequency Concrete, Low-Frequency Concrete, High-Frequency Abstract, and Low-Frequency Abstract. Category assignment is based on lexical frequency from SUBTLEX-US (Brysbaert and New, 2009) and concreteness ratings from (Brysbaert et al., 2014) rather than manual intuitive grouping. The dataset consists primarily of noun-like concepts, although some words are part-of-speech ambiguous in English. The categories are not fully balanced for word length, polysemy, valence, arousal, age of acquisition, or familiarity; these factors may covary with frequency and concreteness. For transparency and reproducibility, we provide the complete word list together with each word's category, frequency criterion, concreteness criterion, and word length in Table 11.

B Evaluation Metrics and Results Discussion

B.1 Metric implementation details

Before computing exact-guess accuracy, target words and model guesses were normalized using a strict string-matching protocol. Both target words and guesses were lowercased, stripped of leading/trailing whitespace, and cleaned by removing non-alphabetic characters except spaces and hyphens. For guesses, we additionally removed common answer prefixes such as "the word is", "answer:", and "guess:". Exact accuracy was then computed by comparing the normalized guess with the normalized original target word. This protocol intentionally does not collapse synonyms, spelling variants, singular/plural variants, or morphological relatives; therefore, exact accuracy is interpreted as a strict measure of lexical recovery rather than full semantic preservation.

Figure 4 shows the exact-guess accuracy for both Pipeline A and Pipeline B, for all four categories.

To capture softer cases where the exact target word is lost but the meaning remains semantically related, we compute embedding-based cosine similarity between representations of the original and generated texts. We use the Sentence-BERT framework (Reimers and Gurevych, 2019) with the pre-trained `all-mpnet-base-v2` checkpoint. This model is based on the MPNet architecture (Song et al., 2020) and is fine-tuned in the Sentence Transformers framework for general-purpose semantic similarity and retrieval tasks. Figure 5 shows the semantic similarity between the original/target word and the guessed word for both Pipeline A and Pipeline B, for all four categories.

Shannon entropy is computed over the distribution of normalized non-empty guesses for each original word, model, pipeline, category, and chain step. Entropy is calculated in bits using base-2 logarithms:

$$H = -\sum_{g} p(g) \log_2 p(g),$$

where $p(g)$ is the empirical proportion of guess g across repeated chain instances. We report normalized entropy as $H/\log_2(N)$, where N is the total number of non-empty guesses for that word-level condition. When $N \leq 1$, normalized entropy is set to 0. Higher values indicate that guesses are more dispersed across alternatives, while lower values indicate convergence on the same guess or a small set of guesses.

Figure 6 shows the normalized entropy of

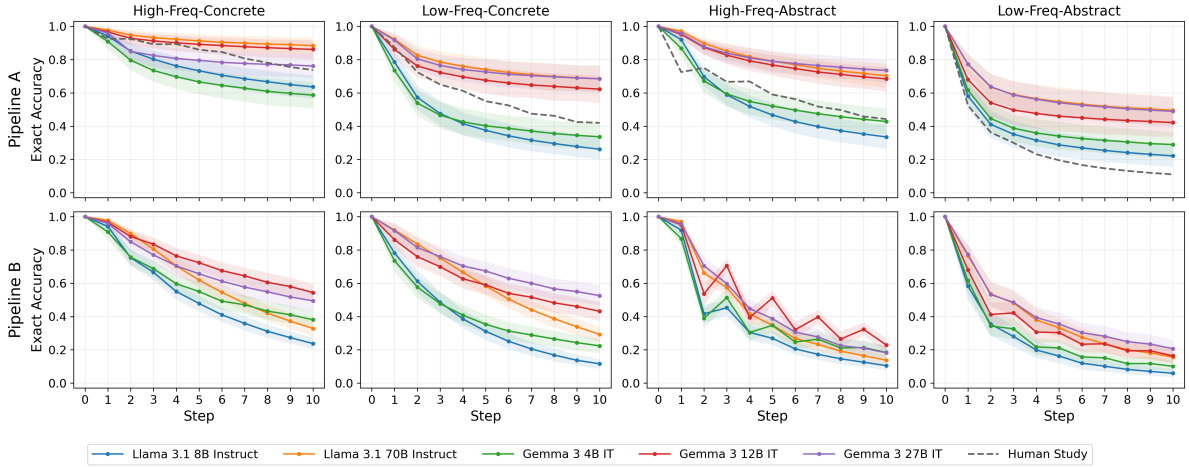

Figure 4: Exact-Guess Accuracy for both Pipeline A and Pipeline B, for all four categories.

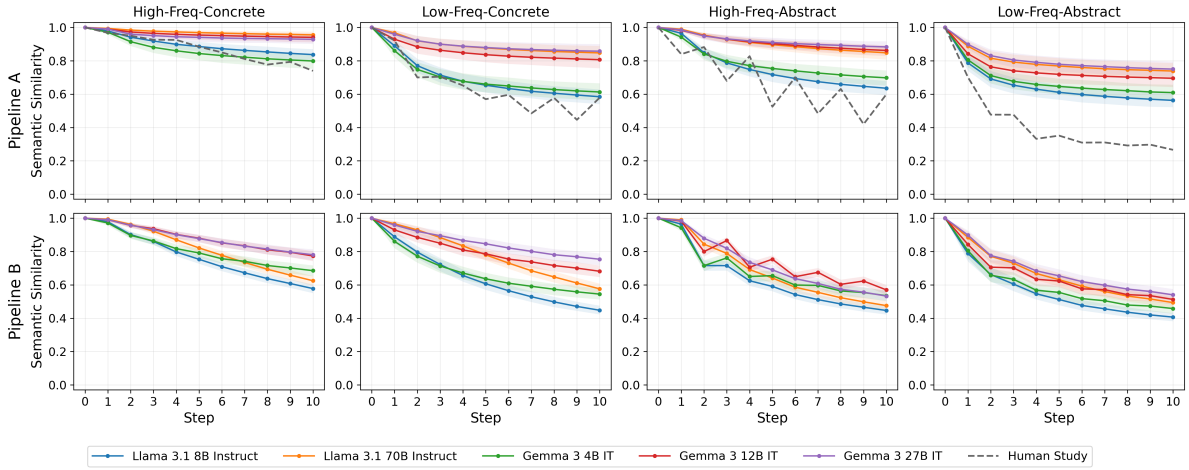

Figure 5: Semantic Similarity between original/target word and guessed word for both Pipeline A and Pipeline B, for all four categories.

guesses for both Pipeline A and Pipeline B, for all four categories. Overall, entropy complements exact accuracy by distinguishing two types of failure: some models drift toward a relatively consistent set of alternatives, while others become increasingly unstable and produce many different guesses across repeated chains. The category-level results indicate that abstract and low-frequency words tend to produce higher entropy than high-frequency concrete words, consistent with the accuracy and semantic similarity results.

B.2 Description-Level Semantic Similarity

We additionally evaluate semantic drift at the description level by comparing the initial description at Step 0 with the transmitted description at each later step. This metric captures whether the wording and meaning of the description itself remain close to the original description, independently of whether the model's guessed word is exactly correct.

Figure 7 shows that description-level similarity drops sharply after the first transformation in both pipelines. In Pipeline A, the average similarity decreases from 1.00 at Step 0 to 0.706 at Step 1, but then declines only gradually, reaching 0.653 at the final step. This suggests that once the model regenerates a description from its first guess, the resulting description tends to remain relatively stable across later steps. In Pipeline B, the first-step drop is slightly smaller, from 1.00 to 0.729, but the decline continues more strongly across the chain, reaching 0.523 at the final step. This indicates that repeated paraphrasing accumu-

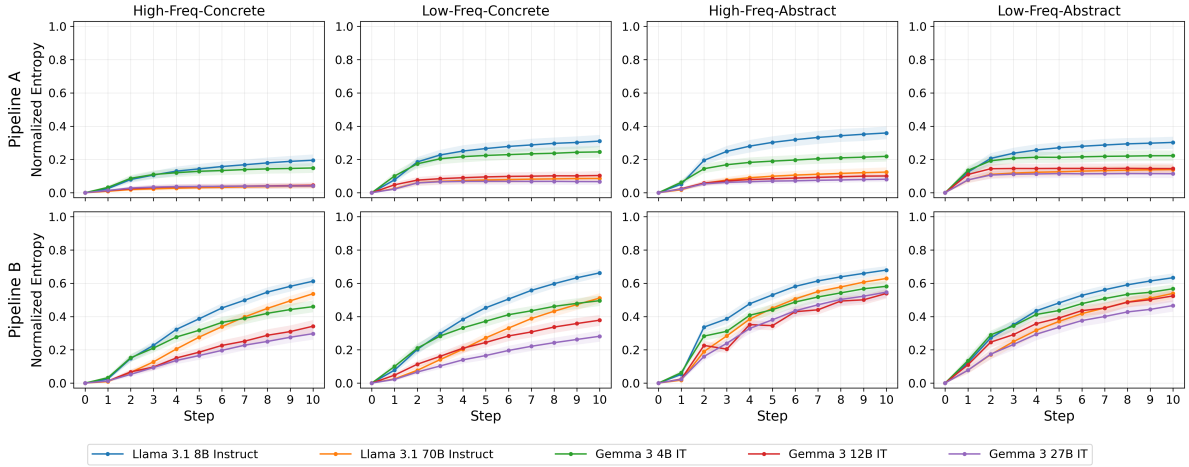

Figure 6: Normalized Entropy of Guesses for both Pipeline A and Pipeline B, for all four categories

lates description-level drift more steadily than the guess-driven regeneration process in Pipeline A.

The model-level patterns differ somewhat across the two pipelines. In Pipeline A, the strongest final description similarity is observed for Gemma 3 12B, Gemma 3 27B, and Llama 3.1 70B, all remaining close to 0.69 at the final step, while Llama 3.1 8B and Gemma 3 4B remain closer to 0.59. In Pipeline B, the Gemma models preserve description-level similarity better than the Llama models, with Gemma 3 12B, Gemma 3 27B, and Gemma 3 4B ending around 0.57–0.59, compared with 0.46 for Llama 3.1 70B and 0.42 for Llama 3.1 8B. This suggests that description-level preservation is affected not only by model scale but also by model family and paraphrasing behavior.

Figure 8 shows the semantic similarity metric graphs for each category for both Pipelines A and B.

Overall, description-level similarity supports the main conclusion that semantic drift is gradual and structured: descriptions often remain partly related to the initial meaning, but repeated transformations progressively move them away from the original wording and semantic cues.

B.3 Regression Details for Predicting Semantic Stability

We train regression models to test whether final semantic stability can be predicted before running a full transmission chain. The prediction unit is a word–model–pipeline condition. With 400 target words, five models, and two pipelines, the regression dataset contains 4,000 rows. Each row corresponds to one target word evaluated under

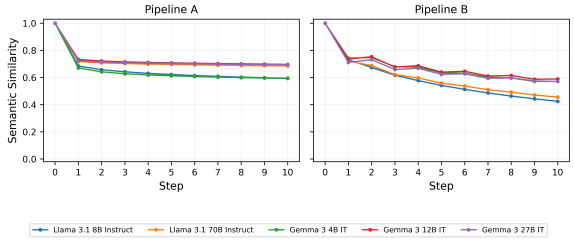

Figure 7: Semantic Similarity between the initial description and the generated description for Pipeline A and the paraphrased description for Pipeline B for all categories.

one model and one pipeline, with the outcome aggregated over the 100 independent chain instances for that word. The primary target is the final step word–guess semantic similarity, computed as cosine similarity between the original target word and the final guessed word using the same Sentence-BERT/MPNet embedding setup used in the main evaluation (Reimers and Gurevych, 2019; Song et al., 2020). The secondary target is the final step exact-guess accuracy, computed as the proportion of normalized final guesses that exactly match the original target word.

All predictors are restricted to information available before observing the chain dynamics. The main feature set contains lexical category, pipeline type, model family, log parameter count, word length, lexical frequency features, dominant part of speech, embedding-neighborhood features, and aggregate properties of the initial descriptions. We include the mean similarity between the target word and its initial descriptions, mean pairwise similarity among initial descriptions, description-length

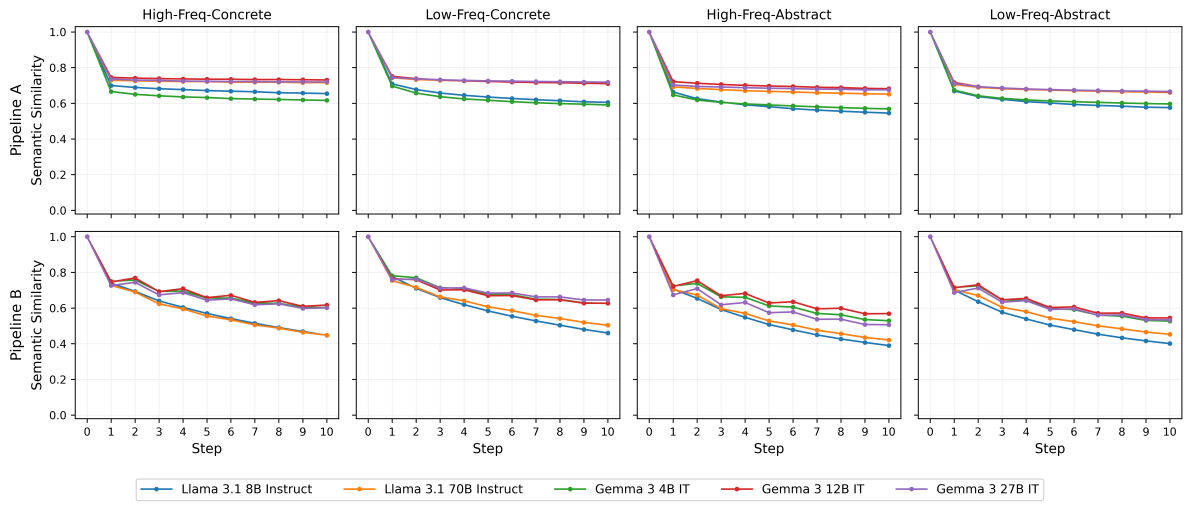

Figure 8: Semantic Similarity between the initial description and the generated description for Pipeline A and the paraphrased description for Pipeline B for each category.

Target	Model	Search	R^2	MAE	RMSE
Final step semantic similarity	Random forest	Random	0.529 ± 0.028	0.142 ± 0.005	0.180 ± 0.006
	XGBoost	Random	0.526 ± 0.029	0.140 ± 0.005	0.181 ± 0.006
	Ridge	Grid	0.504 ± 0.027	0.148 ± 0.005	0.185 ± 0.005
	Linear regression	None	0.501 ± 0.028	0.148 ± 0.005	0.186 ± 0.005
	Mean dummy	None	-0.009 ± 0.006	0.235 ± 0.003	0.264 ± 0.003
Final step exact accuracy	XGBoost	Random	0.375 ± 0.032	0.245 ± 0.005	0.307 ± 0.008
	Random forest	Random	0.372 ± 0.030	0.251 ± 0.005	0.308 ± 0.007
	Ridge	Grid	0.344 ± 0.025	0.262 ± 0.005	0.315 ± 0.006
	Linear regression	None	0.338 ± 0.028	0.261 ± 0.005	0.316 ± 0.006
	Mean dummy	None	-0.009 ± 0.005	0.353 ± 0.003	0.391 ± 0.004

Table 8: Grouped regression results for predicting final step semantic stability. Results are averaged over 10 repeated grouped 80/20 splits by target word. All rows for the same target word are kept in the same split, so the evaluation measures generalization to unseen words. The $\pm$ values are 95% confidence intervals over grouped resamples.

statistics, type–token-ratio features, and semantic-neighborhood measures. We exclude post-hoc chain features such as Step-1 accuracy, later-step entropy, intermediate guesses, or any feature computed from the generated chain trajectory.

Model evaluation uses grouped splits by target word. All rows corresponding to the same target word are assigned either to training or test data, so the same word cannot appear in both sets under different model or pipeline conditions. Each 80/20 split therefore uses 320 training words and 80 held-out words, corresponding to 3,200 training rows and 800 test rows. We repeat this grouped split procedure 10 times and report mean performance with 95% confidence intervals over the 10 resamples. This setup evaluates generalization to unseen target words, not to unseen model families or unseen pipeline types.

We compare a mean-baseline dummy regressor (always predicts the training-set mean target value), ordinary least squares linear regression, ridge regression, random forest regression (Breiman, 2001), and XGBoost regression (Chen and Guestrin, 2016). All models are implemented as scikit-learn pipelines where applicable (Pedregosa et al., 2011). Numeric features are median-inputed and standardized, while categorical features are imputed with the most frequent value and one-hot encoded. Ridge regression selects α from a logarithmic grid from 10^{-4} to 10^{4}. Random forest and XGBoost hyperparameters are selected using randomized search with 24 sampled configurations. Hyperparameter selection uses the same grouped split structure and refits according to RMSE.

Table 8 reports the grouped evaluation results. For the final step semantic similarity, the best model is the random forest regressor, with $R^2 = 0.529 \pm 0.028$, MAE $= 0.142 \pm 0.005$, and RMSE

Feature set	Best model	Features	R^2	MAE	RMSE
Full pruned set	Random forest	17	0.529 ± 0.028	0.142 ± 0.005	0.180 ± 0.006
Binary category variant	XGBoost	18	0.524 ± 0.027	0.140 ± 0.005	0.181 ± 0.006
Word + description only	Ridge	14	0.288 ± 0.034	0.189 ± 0.005	0.222 ± 0.006
Model + pipeline only	XGBoost	3	0.224 ± 0.022	0.195 ± 0.004	0.232 ± 0.004
Mean dummy	Dummy	–	-0.009 ± 0.006	0.235 ± 0.003	0.264 ± 0.003

Table 9: Feature-set comparison for predicting final step semantic similarity. The model+pipeline-only setting uses only pipeline type, model family, and log parameter count. The word+description-only setting removes model and pipeline metadata. The full pruned set performs best, indicating that prediction is not explained solely by pipeline or model scale.

$= 0.180 \pm 0.006$. XGBoost performs similarly, and both linear models substantially outperform the mean baseline. Exact-guess accuracy is harder to predict: the best model is XGBoost, with $R^2 = 0.375 \pm 0.032$, MAE $= 0.245 \pm 0.005$, and RMSE $= 0.307 \pm 0.008$. This is expected because exact recovery is a stricter target than embedding-based similarity: a final guess can remain semantically close to the original concept while failing exact lexical matching.

To test whether the regression merely learns that some pipelines or model sizes are more stable, we also compare feature subsets for the semantic-similarity target. Table 9 shows that model and pipeline metadata alone are predictive, but limited: the best model using only pipeline, model family, and log parameter count reaches $R^2 = 0.224 \pm 0.022$. Word and description features alone perform better, reaching $R^2 = 0.288 \pm 0.034$. The full pruned feature set reaches $R^2 = 0.529 \pm 0.028$, substantially outperforming either subset. This suggests that stability is not explained only by the transmission pipeline or model scale. Instead, predictive performance comes from the combination of transmission mechanism, model condition, lexical category, frequency, embedding-neighborhood structure, and properties of the initial descriptions.

B.4 Ablation Study on Temperature and Quantization

We conduct two robustness checks on Pipeline A using Llama 3.1 8B. First, we evaluate whether the observed drift is primarily caused by the relatively high generation temperature of 0.8 used in the main experiments. For the temperature ablation, we keep the same pipeline, prompts, chain length, target words, and evaluation procedure, but reduce the description-generation temperature to $T = 0.2$. Second, we evaluate whether local 4-bit quantization changes the observed drift pattern. For this condition, we locally deploy a 4-bit quantized Llama 3.1 8B model using NormalFloat 4-bit (NF4) quantization with `bitsandbytes`, double quantization, and bfloat16 computation, following the quantization approach introduced in QLoRA (Dettmers et al., 2023). The quantized model is run with the same temperature setting as the main experiment, $T = 0.8$.

In Figure 9, the temperature ablation shows that lowering the generation temperature improves semantic stability, but does not eliminate drift. Averaged across all categories, in Table 10, the final step exact accuracy increases from 0.363 in the main $T = 0.8$ setting to 0.465 at $T = 0.2$. Word–guess semantic similarity also increases from 0.655 to 0.712, while normalized entropy decreases from 0.292 to 0.171. The number of unique guesses per word is also substantially lower, decreasing from 13.40 to 6.39 on average. This indicates that lower-temperature generation produces more consistent trajectories and reduces the dispersion of final guesses. However, the final step accuracy and similarity remain far below their initial values, showing that semantic drift persists even when generation is made less stochastic.

The category-level results show that the stabilizing effect of lower temperature is consistent across all four lexical categories. For high-frequency concrete words, the final step exact accuracy increases from 0.636 to 0.754, and semantic similarity increases from 0.837 to 0.888. For low-frequency concrete words, exact accuracy increases from 0.261 to 0.363 and semantic similarity from 0.584 to 0.644. For high-frequency abstract words, exact accuracy increases from 0.335 to 0.457 and semantic similarity from 0.635 to 0.707. Low-frequency abstract words remain the least stable even at lower temperature: exact accuracy increases only from 0.221 to 0.287, and semantic similarity from 0.562 to 0.609. Thus, reducing temperature improves stability but preserves the same qualitative category

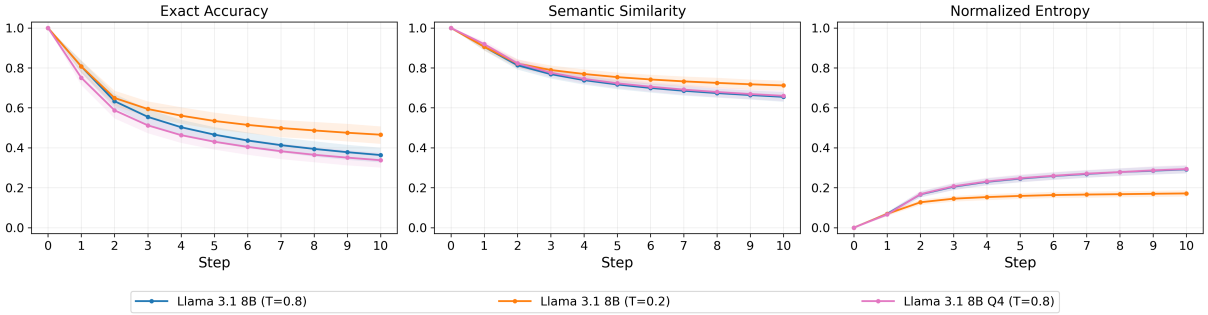

Figure 9: Pipeline A ablation study with Llama 3.1 8B for $T = 0.2$ and 4-bit quantization in comparison with original model with $T = 0.8$

Category	Condition	Exact acc.	Semantic sim.	Norm. entropy	Unique guesses
All categories	Llama 3.1 8B, $T = 0.8$	0.363	0.655	0.292	13.40
All categories	Llama 3.1 8B, $T = 0.2$	0.465	0.712	0.171	6.39
All categories	Llama 3.1 8B Q4, $T = 0.8$	0.338	0.660	0.293	13.38
High-Freq. Concrete	Llama 3.1 8B, $T = 0.8$	0.636	0.837	0.195	8.63
High-Freq. Concrete	Llama 3.1 8B, $T = 0.2$	0.754	0.888	0.093	3.64
High-Freq. Concrete	Llama 3.1 8B Q4, $T = 0.8$	0.608	0.820	0.204	8.54
Low-Freq. Concrete	Llama 3.1 8B, $T = 0.8$	0.261	0.584	0.310	13.99
Low-Freq. Concrete	Llama 3.1 8B, $T = 0.2$	0.363	0.644	0.191	7.03
Low-Freq. Concrete	Llama 3.1 8B Q4, $T = 0.8$	0.303	0.605	0.309	14.10
High-Freq. Abstract	Llama 3.1 8B, $T = 0.8$	0.335	0.635	0.359	17.50
High-Freq. Abstract	Llama 3.1 8B, $T = 0.2$	0.457	0.707	0.207	7.83
High-Freq. Abstract	Llama 3.1 8B Q4, $T = 0.8$	0.320	0.624	0.358	17.05
Low-Freq. Abstract	Llama 3.1 8B, $T = 0.8$	0.221	0.562	0.302	13.48
Low-Freq. Abstract	Llama 3.1 8B, $T = 0.2$	0.287	0.609	0.193	7.06
Low-Freq. Abstract	Llama 3.1 8B Q4, $T = 0.8$	0.120	0.544	0.310	14.15

Table 10: Final step results for the Pipeline A ablation study with Llama 3.1 8B. Lowering generation temperature from $T = 0.8$ to $T = 0.2$ improves exact accuracy and semantic similarity while reducing entropy. The 4-bit quantized model shows similar semantic-similarity and entropy trends to the non-quantized $T = 0.8$ model, but lower exact recovery for low-frequency abstract words.

ordering observed in the main experiments.

The quantization ablation shows a different pattern. The 4-bit quantized Llama 3.1 8B model behaves similarly to the non-quantized $T = 0.8$ model in terms of semantic similarity and entropy. At the final step, the quantized model obtains a semantic similarity of 0.660 compared with 0.655 for the non-quantized model, and a normalized entropy of 0.293 compared with 0.292. This suggests that, under our setup, 4-bit quantization does not qualitatively change the broad semantic drift trajectory. However, exact accuracy is slightly lower overall for the quantized model, decreasing from 0.363 to 0.338. The most notable difference appears for low-frequency abstract words, where final step exact accuracy drops from 0.221 to 0.120. This suggests that quantization may have a stronger effect on strict lexical recovery for already fragile concepts, even when embedding-based semantic similarity remains close.

Rows 1–50				Rows 51–100			
HF-Conc.	**LF-Conc.**	**HF-Abstr.**	**LF-Abstr.**	**HF-Conc.**	**LF-Conc.**	**HF-Abstr.**	**LF-Abstr.**
Apple	Abacus	Ability	Aberration	Gold	Cuckoo	Effect	Depravity
Baby	Accordion	Action	Acrimony	Grass	Curio	Effort	Derision
Ball	Acorn	Advice	Adulation	Hair	Cutlass	Energy	Despair
Banana	Anvil	Age	Affluence	Hand	Cymbal	Error	Destiny
Bed	Archipelago	Agreement	Alacrity	Hat	Dagger	Event	Detriment
Bird	Armadillo	Air	Allegory	Head	Dandelion	Example	Devotion
Boat	Artichoke	Anger	Ambivalence	Heart	Decanter	Experience	Dilemma
Book	Asparagus	Answer	Amnesty	Home	Dirigible	Fact	Discord
Bottle	Awl	Art	Anarchy	Horse	Dodo	Failure	Disdain
Box	Bagpipe	Attention	Animosity	House	Dragonfly	Faith	Dissent
Boy	Banjo	Authority	Anomaly	Ice	Drawbridge	Family	Dogma
Bread	Barnacle	Beauty	Apathy	Key	Dumbbell	Fear	Drudgery
Bus	Barometer	Belief	Apex	Knife	Dynamite	Feeling	Duplicity
Cake	Basilisk	Benefit	Aptitude	Lamp	Easel	Force	Ebullience
Camera	Bassoon	Business	Arrogance	Leaf	Eclair	Form	Ecstasy
Car	Baton	Care	Atonement	Leg	Emus	Freedom	Edict
Cat	Beaker	Case	Atrophy	Letter	Epaulet	Friendship	Efficacy
Chair	Beaver	Cause	Audacity	Light	Ermine	Fun	Ego
Chicken	Bellows	Chance	Avarice	Man	Espadrille	Future	Elation
Child	Beret	Change	Aversion	Map	Falcon	Game	Elegance
Clock	Bifocals	Choice	Banal	Meat	Fedora	Goal	Empathy
Cloud	Biscuit	City	Bane	Milk	Fez	Government	Enigma
Coat	Blimp	Class	Benevolence	Money	Fiddle	Group	Enmity
Coffee	Bonnet	Color	Bias	Moon	Figurine	Growth	Ennui
Computer	Boomerang	Comfort	Bigotry	Morning	Flute	Happiness	Epiphany
Corn	Bouquet	Company	Blasphemy	Mother	Fresco	Hate	Equity
Cow	Bramble	Condition	Brevity	Mountain	Fuselage	Health	Essence
Cup	Brocade	Control	Calamity	Mouse	Gable	Help	Euphoria
Desk	Buckle	Cost	Candor	Mouth	Gadget	History	Exodus
Doctor	Bungalow	Country	Catharsis	Music	Galleon	Hope	Expediency
Dog	Cactus	Course	Censure	Night	Gargoyle	Idea	Fallacy
Door	Cauldron	Credit	Chaos	Nose	Gauntlet	Image	Fame
Dress	Cello	Culture	Charisma	Ocean	Gazebo	Impact	Famine
Ear	Centipede	Danger	Chivalry	Office	Geode	Industry	Fatigue
Egg	Chalice	Data	Clemency	Oil	Gerbil	Information	Feud
Eye	Chandelier	Date	Coercion	Paper	Geyser	Interest	Fidelity
Face	Chariot	Death	Collusion	Park	Gimlet	Issue	Finesse
Farm	Chisel	Decision	Complacency	Pen	Glacier	Job	Flattery
Fire	Chrysanthemum	Degree	Concord	Phone	Glider	Joy	Folly
Fish	Clarinet	Design	Consensus	Picture	Gnome	Justice	Fortitude
Floor	Cleaver	Desire	Contempt	Pig	Goblet	Knowledge	Frenzy
Flower	Cobweb	Detail	Conundrum	Pizza	Gondola	Law	Friction
Food	Cockroach	Difference	Credulity	Plane	Gong	Level	Frugality
Foot	Colander	Direction	Dearth	Plant	Gopher	Life	Futility
Fork	Compass	Discussion	Debacle	Plate	Gorilla	Line	Gallantry
Friend	Corset	Disease	Decorum	Rain	Astrolabe	Love	Gambit
Fruit	Coyote	Doubt	Decree	Ring	Pangolin	Luck	Genesis
Garden	Crayfish	Dream	Deference	River	Trebuchet	Market	Glamour
Girl	Crowbar	Duty	Delusion	Road	Ziggurat	Matter	Gluttony
Glass	Crucible	Economy	Demise	Rock	Harpsichord	Meaning	Gratitude

Table 11: Word categories used in the semantic drift experiments. HF = high-frequency, LF = low-frequency, Conc. = concrete, Abstr. = abstract.