Beyond Tokens: Concept-Level Training Objectives for LLMs

Anonymous ACL submission

Abstract

The next-token prediction (NTP) objective has been foundational in the development of modern large language models (LLMs), driving advances in fluency and generalization. Yet, NTP rewards only surface-level accuracy, motivating additional post-training methods such as reinforcement learning from human feedback (RLHF) and direct preference optimization (DPO). We propose a shift from token-level to concept-level prediction, where concepts group multiple surface forms of the same idea (e.g., "mom," "mommy," "mother" → *MOTHER*). We introduce methods for integrating conceptual supervision and show that concept-aware training is more robust to domain shifts in terms of perplexity, and is comparable and sometimes better than NTP on diverse downstream benchmarks. These results suggest concept-level supervision as a promising alternative training signal for building more human-aligned LLMs.

1 Introduction

Large language models (LLMs) have reshaped the landscape of natural language processing, achieving fluency and generalization once thought out of reach. At their core, however, today's LLMs are trained with a surprisingly narrow objective: predicting the next token in a sequence. This has been a powerful proxy for learning language, but it ties models to the surface level of text by rewarding them for producing the right strings, not for understanding the ideas those strings convey. This gap becomes especially pronounced as LLMs are increasingly expected to perform abstraction and reasoning rather than mere continuation.

Humans, by contrast, do not think or communicate in tokens. We reason in concepts: semantic units that unify different linguistic expressions under a shared meaning. For example, "mom," "mommy," and "mother" all point to the concept MOTHER. Concepts also stretch beyond literal synonymy: "father" may be understood as part of the broader concept PARENT, depending on context. Concepts are flexible, context-sensitive, and hierarchically structured, capturing meaning at a level that tokens cannot (Shani et al., 2023).

This gap matters because the expectations placed on LLMs are rapidly shifting. Beyond producing fluent continuations, we now ask them to explain, reason, and abstract, tasks that hinge on capturing meaning rather than string similarity. To address these shortcomings, researchers have layered post-training objectives such as Reinforcement Learning from Human Feedback (RLHF; Christiano et al. (2017)), Direct Preference Optimization (DPO; Rafailov et al. (2023)), Reinforcement Learning from AI Feedback (RLAIF; Lee et al. (2023)), Kahneman-Tversky Optimization (KTO; Ethayarajh et al. (2024)), and Identity Preference Optimization (IPO; Azar et al. (2024)) onto NTP-trained models. While effective for aligning outputs with human preferences, these methods leave the underlying training signal unchanged: models still learn primarily to predict tokens, not meanings.

In this paper, we explore a different foundation: **What if models were trained to predict concepts rather than tokens**, by recognizing that multiple forms can stand for the same idea, and to generalize across them? *Next-Concept-Prediction* (NCP) offers such a shift: instead of optimizing for exact surface matches, models are guided to capture the semantic structures underlying language.

We formalize concepts as clusters of synonymous and contextually interchangeable forms, and integrate them into training as units of supervision. We show that **NCP remains competitive with NTP on traditional metrics, exhibits better robustness to domain shifts, and shows small improvements on less saturated benchmarks**. These suggest that concept-aware training can provide a more human-centered foundation for LLM.

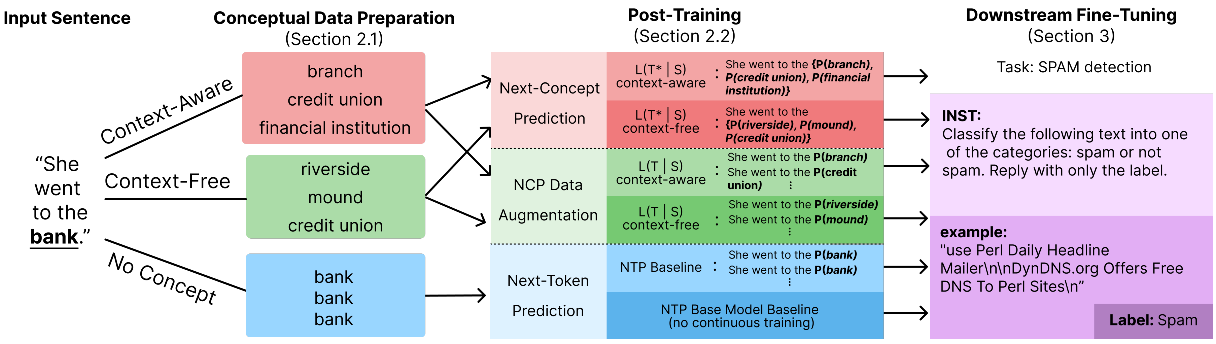

Figure 1: An outline of our Method. We extract context-dependent and independent synonyms and use them to train our NCP models. We upsample the original data for the NTP baselines. All models are fine-tuned on benchmarks.

2 Methods

We post-trained Llama-3-8B (Grattafiori et al., 2024) using both the standard NTP and our Next-Concept-Prediction (NCP) loss. To avoid data contamination, we gathered new data not used for training the original LLM. We now detail the data preparation and training processes (see Figure 1).[1]

2.1 Conceptual Data Preparation

To avoid training Llama-3-8B on examples from its pretraining corpus, we collected data produced after its public release date (April 18, 2024). Our dataset draws from three distinct sources: YouTube comments, arXiv abstracts, and New York Times abstracts, chosen to provide diversity across informal, scientific, and journalistic domains.

From this corpus, we extracted nouns from each sentence, treating them as core conceptual units, since nouns typically carry substantial semantic content.[2] We operationalize concepts as interchangeable lexical realizations of a shared referent. Thus, we wish to extract for each noun in our data a set of interchangeable nouns, or concepts.

As illustrated in Figure 1, we generated conceptual training data using three complementary methods: (1) **Context-Free** extracts context-independent, dictionary-based synonyms from WordNet, (2) **Context-Aware**, which extracts contextual synonyms by prompting Llama-3 with the full sentence and the target noun to produce contextually appropriate alternatives (Appendix B), and (3) **No Concept**, which inflates the data reusing the original word in the sentence, multiple times.[3]

[1] We will release data and code upon acceptance.
[2] Verbs can also be meaningful, left for future research.
[3] These methods are imperfect; see Limitations Section.

2.2 Post-Training

We post-trained on each dataset split $\in$ {YouTube, arXiv, New York Times} as well as on a combined dataset, downsampling where necessary for consistency across splits. This allows us to assess whether data variation across domains leads to different levels of conceptual awareness. All NTP and NCP models were post-trained on the same datapoints (different target nouns depending on the concept handling) and the name *number* of datapoints.

2.2.1 NTP Baselines

We used two NTP-based baselines:
Base Model. Used without any post-training to evaluate the model's baseline performance.
NTP Baseline. Trained using the standard NTP on the same datapoints as the NCT models:

$$L(T \mid S) = \log\left(p(T \mid S, \Theta)\right)$$

This ensures that the model is exposed to the same datapoints but does not learn any new semantic relationships or synonyms.

2.2.2 Next Concept Prediction (NCP)

To shift training from token- to concept-level, we enrich our data with conceptual signals (see Section 2.1). As a result, each target noun T in the corpus was paired with a set of synonym nouns T^*, extracted either with or without context. Using these, we implemented two NCP training procedures:
Data Augmentation. We inflated the data using the extracted synonyms. Meaning, if a sentence had five possible noun completions, we duplicated it to five datapoints, each one with a different target noun to predict. We then trained the model using its standard NTP objective. By rewarding these variations, we effectively flatten the probability distribution of the next likely token, reducing the dominance of any single lexical choice.

NCP Loss Function. A more straightforward method is to modify the NTP loss: Let the original target noun be T, and the set of interchangeable completions be T^*. The new objective is to predict all completions in T^*, conditioned on the input sentence S and the model parameters Θ:

$$L(T^* \mid S) = \frac{1}{|T^*|} \sum_{n=1}^{N} \log \left(p(t_n \in T^* \mid S, \Theta)\right)$$

We name the four NCP models *NCP Context-Aware Data Aug.*, *NCP Context-Free Data Aug.*, *NCP Context-Aware*, and *NCP Context-Free*, based on the NCP implementation used (data augmentation or customized loss function) and the synonym extraction method used (context-aware using an LLM or context-free using a dictionary).

3 Benchmarks for Fine-Tuning

After post-training the NTP baseline and all NCP variants, we fine-tuned them on nine diverse benchmarks (parameters and details in Appendix B):

SNLI. (Bowman et al., 2015) Stanford Natural Language Inference evaluates a model's ability to determine entailment, contradiction, or neutrality between a premise and a hypothesis.

GLUE. (Wang et al., 2018) GLUE aggregates several tasks, such as sentiment analysis, paraphrase detection, and linguistic acceptability, making it a robust testbed for general NLU capabilities.

Empathetic dialogs. (Rashkin et al., 2019) contains thousands of short conversations grounded in emotional situations, requiring the model to exhibit nuanced understanding and empathetic reasoning.

Hate speech. (Davidson et al., 2017) composed of tweets annotated for hate speech and offensive language from `hatebase.org` and challenges models to distinguish between harmful and benign content.

Spam. (Talby, 2020) includes real-world email messages labeled as spam or ham.

Suicidal Ideation. (Mafi and Alam, 2023) contains Reddit posts annotated as suicidal/not.

Fake News. (Cartinoe5930, 2025) is built from PolitiFact fact-checks (PolitiFact, 2007–2025). It provides news/claims with fake versus real labels on contemporary U.S. political content.

Logical Fallacy. (Jin et al., 2022) reasoning patterns detection dataset spanning ad hominem, ad populum, circular reasoning, false causality, etc.

Amazon Polarity. (Zhang et al., 2015) contains reviews for Amazon products assigned to positive/negative polarity (4-5 vs. 1-2 stars).

4 Results

We now compare standard NTP with NCP training for both post-training and fine-tuning procedures.

4.1 Post-Training

We computed NTP and NCP perplexity scores on held-out sets from all four domains (YouTube comments, arXiv abstracts, and New York Times abstracts). We report the perplexity scores of the NTP baseline and the four NCP variants in Table 3.

The NCP models and the NTP baseline yield comparable validation perplexity scores. This is an important validation, as any pre-/post-training method that leads to collapsed performance at the token level shall be deemed unusable. Interestingly, when models trained on one domain are evaluated on others, NCP models consistently achieve lower perplexity, indicating that **NCP training is more robust to domain shifts** (Table 1; metric explanation and full PPL scores in Appendix G).

Table 1: **[NCP models show superior cross domain robustness.]** For each eval domain, we compute the NTP/NCP perplexity score of all models *not* trained on the domain and divide them by the corresponding score of the corresponding model that was trained on the eval domain. This captures the robustness to domain shifts.

Train	Eval	Best Model
YouTube	News	NCP Context-Free Data Aug.
	ArXiv	NTP Baseline
	Combined	NCP Context-Aware & NCP Context-Free
News	YouTube	NCP Context-Aware & NCP Context-Free
	ArXiv	NCP Context-Aware Data Aug.
	Combined	NCP Context-Aware & NCP Context-Free
ArXiv	YouTube	NCP Context-Aware
	News	NCP Context-Free Data Aug.
	Combined	NCP Context-Aware & NCP Context-Free

To illustrate the qualitative difference between NTP and NCP, consider the following sentence from our data: "This word has appeared in 53 ."

The NTP's top five predictions are different variations of 'articles' (singular versus plural, with and without capitalization and spaces). In contrast, the

Table 2: **[All Fine-Tuned Models Exhibit Similar Performance; NTP Models are a Bit Better on Popular Benchmarks and NCP Models are a Bit Better on Less Popular Datasets.]** Downstream fine-tuned accuracy scores across six benchmarks: EMPATHETIC DIALOGUES (EMO), GLUE, HATE SPEECH (HATE), SNLI, SUICIDAL IDEATION REDDIT DATASET (SI-R), and SPAMASSASSIN (SPAM), FAKE NEWS (FAKE), LOGICAL FALLACY (LOG), AMAZON POLARITY (POL). Best accuracy for each dataset within a domain is in bold. A double horizontal line separates the NCP models and the NCP baselines. NTP baseline is slightly better on GLUE and SNLI, which are very oversaturated benchmarks. NCP models are slightly better on less popular datasets.

Variant	Domain	EMO	GLUE	HATE	SNLI	SI-R	SPAM	FAKE	LOG	POL
NCP Context-Aware	ArXiv	0.8757	0.8504	0.8905	0.8908	0.9849	0.9376	0.7224	0.4209	0.9659
	News	0.8780	0.8428	0.9155	0.8765	0.9844	0.9438	0.7106	0.5252	**0.9682**
	YouTube	0.8735	0.8504	0.9124	0.8844	0.9860	**0.9548**	0.7265	0.4784	0.9659
	Combined	0.8661	0.8589	**0.9166**	0.8998	0.9801	0.9501	0.7247	0.482	**0.9682**
NCP Context-Free	ArXiv	0.8690	0.8393	0.9047	0.8881	0.9822	0.9470	0.6824	0.4353	0.9635
	News	0.8774	0.7849	0.8782	0.8674	0.9806	0.9454	0.7106	0.5252	0.9682
	YouTube	0.8735	0.8529	0.9120	0.8844	**0.9871**	**0.9548**	0.7265	0.4784	0.9659
	Combined	0.8661	0.8615	0.9170	0.8998	0.9795	0.9454	**0.7306**	0.4892	0.9665
NCP Context-Aware Data Aug.	ArXiv	0.8785	0.8207	0.9078	0.8887	0.9833	0.9376	0.6894	0.4748	0.9641
	News	0.8690	0.4448	0.8874	0.8404	0.9855	0.9438	0.7071	0.5252	0.9635
	YouTube	0.8718	0.8358	0.9143	0.8828	0.9828	0.9282	0.7294	**0.5288**	0.9653
	Combined	0.8746	0.8297	0.9001	0.8796	0.9785	0.9438	0.7006	0.4604	0.9594
NCP Context-Free Data Aug.	ArXiv	0.8735	0.8212	0.9109	0.886	0.9812	0.9532	0.6729	0.5036	0.9659
	News	0.8763	0.8322	0.9124	0.877	0.9812	0.9516	0.7159	0.5252	0.9653
	YouTube	0.8661	0.8338	0.9059	0.8892	0.9822	0.9485	0.7141	0.4640	0.9647
	Combined	0.8774	0.8569	0.9143	0.8876	0.9774	0.9298	0.7159	0.5000	0.9676
NTP Baseline Fine-tuned	ArXiv	0.8735	0.8292	0.9028	0.8855	0.9828	0.9407	0.6906	0.4748	0.9635
	News	0.8796	0.8574	0.9159	0.8892	0.9849	0.9438	0.7235	0.5108	0.9641
	YouTube	0.8706	**0.8700**	0.9136	**0.9008**	0.9860	0.9423	0.7229	0.4784	0.9647
	Combined	**0.8802**	0.8610	0.9109	0.8950	0.9769	0.9485	0.7224	0.5036	0.9653
Base Model Fine-tuned	-	0.8791	0.8660	0.9120	0.8913	0.9806	0.9282	0.7224	0.4748	0.9665
Base Model	-	0.5681	0.3204	0.7933	0.3144	0.6615	0.6505	0.4424	0.0071	0.7406

NCP models distribute the probability mass across semantically related completions: 'searches,' 'articles,' 'episodes,' and 'cases,' reflecting a broader conceptual understanding. This example highlights that while NTP rewards reproducing surface strings, NCP encourages models to capture underlying semantic relationships, producing outputs that are more meaning-equivalent and less lexically rigid.

4.2 Downstream Benchmark Fine-Tuning

We now report the accuracy scores of all models after fine-tuning on the nine benchmarks presented in Section 3 (See Table 2). Across all datasets, all models and baselines perform better than the non-fine-tuned variant, as expected. In addition, the NCP models, the NTP baseline, and the fine-tuned variant that was not post-trained perform similarly. **NTP baselines show minor improvement on oversturated benchmarks such as GLUE and SNLI, while NCP models show slightly better performance on the less popular datasets**. We note that saturated GLUE and SNLI may disproportionately reward surface-level optimization.

5 Conclusions & Future Work

Here, we rethink the standard NTP approach by incorporating more human-inspired supervision signals. We introduce NCP, which unifies synonymous forms into shared semantic units, enabling models to capture meaning beyond surface text. **NCP proves more robust to domain shifts and matches or exceeds NTP performance, marking it as a promising foundation for LLM training**. Moreover, existing alignment techniques often operate post-hoc, shaping model behavior after training. In contrast, NCT is flexible, supporting both pre- and post-training applications.

Future work can explore: NCP pre-training; varying levels of concept granularity; hierarchical concept representations (e.g., *MOTHER* $\rightarrow$ *PARENT* $\rightarrow$ *FAMILY*); and cross-linguistic extensions.

We hope to encourage viewing NTP as one of many possible training signals, whereas **NCP opens the door to foundations that are not only statistically effective but also more aligned with how humans communicate and think**.

6 Limitations

While our findings highlight the promise of concept-aware training, several limitations remain. First, we explore only one paradigm to incorporate concept supervision. Other formulations, such as hierarchical concepts, cross-lingual mappings, or integration with generative objectives, may provide richer signals.

Second, our evaluation is limited to fine-tuning on classification tasks. These benchmarks already achieve high baseline accuracy, leaving little room to demonstrate the full potential of concept-level prediction. Extending evaluation to tasks that require more abstraction, such as generation, reasoning, or transfer learning, would offer a clearer picture of its benefits. Broader evaluations and larger-scale experiments are essential to fully establish its effectiveness.

Third, our approach to extracting concept signals is imperfect. The context-aware method relies on LLMs, which may introduce or amplify existing biases and inconsistencies in their understanding of concepts. The context-free method neglects the crucial role of context in shaping meaning. More robust methods are needed to induce concept representations.

7 Ethical Considerations

In terms of the potential risks of our work, we realize that concepts could lead to the risk of overgeneralizing, overextending concept boundaries, and amplifying spurious associations or stereotypes. Care should be taken when defining concept clusters, especially for sensitive or demographic-related content, to avoid reinforcing biases present in the training data.

We also note that, similar to all NTP LLMs, NCP might lead to hallucinations and other types of undesired model behaviors. These were not explored in this work, and thus we recommend practitioners, as usual, to validate their artifacts before releasing them to the public.

Finally, the introduction of concept-level reasoning may shift the interpretability of model outputs: while grouping tokens into concepts can improve semantic coherence, it may obscure the model's reasoning at the token level, potentially making errors harder to detect. We encourage transparency in reporting both concept definitions and model behaviors to support responsible use.

Disclosure: LLMs were used to refine the text and tables.

References

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. ACL.

Cartinoe5930. 2025. Politifact fake news. Hugging Face Datasets. Accessed: 2025-10-06.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of ICWSM*.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

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. *Preprint*, arXiv:2407.21783.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Zhijing Jin, Abhinav Lalwani, Tejas Vaidhya, Xiaoyu Shen, Yiwen Ding, Zhiheng Lyu, Mrinmaya Sachan, Rada Mihalcea, and Bernhard Schölkopf. 2022. Logical fallacy detection. *arXiv preprint arXiv:2202.13758*.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and 1 others. 2023. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

Md Mafiul Hasan Matin Mafi and Md. Sabbir Alam. 2023. Suicidal ideation detection reddit dataset.

PolitiFact. 2007–2025. Politifact: Fact-checks and truth-o-meter. Accessed: 2025-10-06.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381. ACL.

Chen Shani, Jilles Vreeken, and Dafna Shahaf. 2023. Towards concept-aware large language models. *Preprint*, arXiv:2311.01866.

David Talby. 2020. Spamassassin public corpus dataset. `https://huggingface.co/datasets/talby/spamassassin`.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP*, pages 353–355. ACL.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NeurIPS*.

A Prompt to obtain contextual synonym

System Prompt `Answer the question using a comma-separated list and remove any extraneous information. An example output for a sentence will be [item1, item2, item3]. If no synonyms are found, return an empty array. Do not repeat this prompt in your output.`

Message Provided a **sentence** and a **noun** of interest, the message reads: `"Generate contextual synonyms for the word` **`noun`** `in the sentence` **`sentence`**`."`

B Fine-Tuning Implementation

For each of the following tasks, we fine-tuned all models using LoRA (Hu et al., 2021) with parameters r=16 and α=16, targeting the attention and feed-forward modules (q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj) for efficient adaptation. Models were trained using 4-bit quantization with the AdamW 8-bit optimizer, a learning rate of 2e-4 with linear scheduling, and gradient accumulation over four steps. Each model was trained for 100 steps with a batch size of 2, employing the Alpaca instruction format for consistent prompt structuring across tasks. Training incorporated validation-based checkpointing every 20 steps to monitor convergence. This resulted in a total of 189 fine-tuned models across 9 downstream tasks, enabling us to systematically evaluate the transferability and robustness of conceptual understanding across domains.

We evaluated each fine-tuned model's ability to classify instances for the task for which it was trained. The evaluation process involved matching each model to its input template and generating predictions using the Alpaca prompt format. We computed match accuracy by comparing lowercased, stripped predictions against ground truth labels. The evaluation focused on accuracy as the primary metric for comparing the concept-aware training paradigm against baseline approaches, with results stored in JSON format, including sample predictions for qualitative analysis. This systematic evaluation enabled direct comparison of how different post-training strategies transferred to downstream classification tasks.

C Post-Training

Table 3: **[NCP Models are Show Comparable Token token-based Performance, Even Though Not Optimized to Predict the Next Token Per Se.]** Perplexity and Accuracy on the evaluation held-out dataset after performing post-training. Bold numbers are per domain, where each variant was tested on three domains, and the combined which is a downsampling from all three domains. Lower perplexity (PPL) and higher accuracy (Acc) are better. A double horizontal line separates the concept-aware models and the token-based baselines.

Variant	Domain	NTP PPL	NCP PPL
NCP Context-Aware	ArXiv	1002.6701	**4.6216**
	News	1680.7794	**3.6551**
	YouTube	1281.9174	**3.3604**
	Combined	2717.3075	**269.349**
NCP Context-Free	ArXiv	1142.1247	**4.8344**
	News	1680.7794	**3.6551**
	YouTube	1281.9174	**3.3604**
	Combined	2717.3075	**269.349**
Context-Aware Data Aug.	ArXiv	**316.4381**	449111.7608
	News	**171.6298**	216192.4155
	YouTube	257.4089	260072.7351
	Combined	141.0141	255268.2881
Context-Free Data Aug.	ArXiv	372.3505	287483.5986
	News	205.1227	470666.8797
	YouTube	258.3046	278252.2883
	Combined	130.5401	930607.8148
NTP Baseline	ArXiv	554.2414	254618.3941
	News	250.4959	67641981.74
	YouTube	**184.3536**	663891.0267
	Combined	**122.677**	1673866.06

D Multi-Token Completions

Our loss function supports *multi-token* words and completions. We precompute a map word $\rightarrow$ token IDs for all targets and their completions to avoid repeated tokenization and keep training steps fast/deterministic. Using this dictionary from words (nouns of interest) to their tokenized IDs we replace the entire token span of the target word with the completion's span (which may be longer or shorter) during training. The model is then evaluated using the completions and the NCP loss function.

E Model Size and Budget

In this paper, we use LLaMA-3-8B as our main model, which is an 8-billion-parameter model. Computational resources and GPUs were provided by the authors' research institute.

F Dataset Descriptive Statistics

The following is additional details about each of our fine-tuning datasets:

SNLI (Bowman et al., 2015) (`stanfordnlp/snli`)

- **Task:** 3-way NLI (entailment/contradiction/neutral).
- **Size/Splits/Labels:** ~570k pairs; train/dev/test; labels: *entailment, contradiction, neutral.*
- **Examples from the dataset:**
  (1) `Premise: A man inspects the uniform of a figure in an East Asian country. Hypothesis: The man is sleeping`
  Label: Contradiction

  (2) `Premise: Two men on a roof with snow shovels. Hypothesis: They are clearing snow.`
  Label: Entailment

GLUE (Wang et al., 2018) (`nyu-mll/glue`)

- **Task:** Aggregated NLU suite (acceptability, sentiment, paraphrase, NLI, STS).
- **Size/Splits/Labels:** 13.2k pairs; train/dev/test; labels: entailment, contradiction, neutral.
- **Examples from the dataset:**
  (1) `Premise: but see but you're going there and you know what you're getting into. Hypothesis: By getting involved, you understand what is in store.`
  Label: Entailment

  (2) `Premise: it is in Texas too. Hypothesis: It's not in Texas`
  Label: Contradiction

Empathetic Dialogues (Rashkin et al., 2019) (`facebook/empathetic_dialogues`)

- **Task:** Emotion-grounded open-domain dialogue.

- **Size/Splits/Labels:** 76.7k/12.0k/10.9k (train/dev/test); emotion in context.

- **Examples from the dataset:**
(1) `Utterance: I remember going to see the fireworks with my best friend.`
Label: Sentimental

(2) `Utterance: I finally finished my last exam today!`
Label: Proud

Hate Speech (Davidson et al., 2017) (`tdavidson/hate_speech_offensive`)

- **Task:** Tweet toxicity (hate_speech/offensive/neither)

- **Size/Splits/Labels:** train/dev/test; 3 labels.

- **Examples from the dataset:**
(1) `Input: @user we gotta find this h**.`
Label: Offensive

(2) `Input: Burritos are trash`
Label: Neither

- **Content note:** Contains offensive language.

Spam Assassin (Talby, 2020) (`talby/spamassassin`)

- **Task:** Spam vs. ham email classification.

- **Size/Splits/Labels:** ∼21.5k messages; labels: *spam, ham*.

- **Examples from the dataset:**
(1) `Input: Free trial for . . .`
Label: Spam

(2) `Input: Meeting moved to 3pm . . .`
Label: Ham

Suicidal Ideation (Reddit) (Mafi and Alam, 2023) (Mendeley Data (DOI))

- **Task:** Spam vs. ham email classification.

- **Size/Splits/Labels:** 15,477 posts (paper).

- **Examples from the dataset:**
(1) `Input: "I can't see a way out . . . I'm so tired."`
Label: Suicidal

(2) `Input: "Having a rough day but trying to stay positive."`
Label: Non-suicidal

- Content note: Sensitive mental-health content.

Fake News (PolitiFact-based) (Cartinoe5930, 2025) (Cartinoe5930/Politifact_fake_news)

- **Task:** Short political claim with fact-check label (e.g., *true*, *false*)

- **Size/Splits/Labels:** train 17.1k, text 4.23k; labels: true or false.

- **Examples from the dataset:**
(1) `Input: PayPal has reinstated its policy to fine users $2,500 directly from their accounts if they spread 'misinformation.`
Label: False

(2) `Input: Kids are resistant to COVID as opposed to older people.`
Label: True

Logical Fallacy (Jin et al., 2022) (tasksource/logical-fallacy)

- **Task:** Multi-class fallacy detection (e.g., ad hominem, ad populum, circular reasoning, false causality)

- **Size/Splits/Labels:** train 2.68k, test 500; labels: ad hominem, ad populum, appeal to emotion, circular reasoning, equivocation, fallacy of credibility, fallacy of extension, fallacy of logic, fallacy of relevance, false causality, false dilemma, faulty generalization, intentional.

- **Examples from the dataset:**
(1) `Input: Don't listen to Senator Bob's opinion. He is a crook, and a spiteful loony man.`
Label: ad hominem

(2) `Input: Did your misleading claims result in you getting promoted?`
Label: intentional

Amazon Polarity (Zhang et al., 2015) (amazon_polarity)

- **Task:** Binary review sentiment.

- **Size/Splits/Labels:** train 3.6M, test 400k; labels: positive, negative.
- **Examples from the dataset:**
  (1) `Input: A complete waste of time. Typographical errors, poor grammar, and a totally pathetic plot add up to absolutely nothing. I'm embarrassed for this author and very disappointed I actually paid for this book.`
  Label: negative

  (2) `Input: got this for my daughter in NC, she is now making prefect bread. Wish she lived closer to make me some`
  Label: positive

G Cross-Domain Table

Post-training perplexity scores using both the standard NTP and our NCP for calculating these perplexity scores. The rightmost column depicts the domain-shift robustness and is calculated as follows: the perplexity score (using the relevant N_P; NTP for baselines and NCP for all others) trained on a different domain than the evaluation is divided by the perplexity score of the corresponding model that was trained on the domain. This allows us to normalize the perplexity in a way that only preserves robustness to domain shifts. For example, the NTP perplexity score of the NTP baseline trained on *news* and evaluated on *YouTube* is 244.5672. We divide it by the NTP perplexity score of the NTP baseline trained on *YouTube* (184.3536), resulting in 1.32662015. Similar to perplexity scores, lower numbers indicate better robustness to domain shifts.

Table 4: Post-training perplexity (PPL) with NTP and NCP objectives. The last column (N_P/Domain N_P) shows cross-domain transfer: PPL of a model trained on a source domain and evaluated on a target, normalized by the model trained (and evaluated) on that target with the same objective.

Evaluated	**Domain**	**Variant**	**NTP PPL**	**NCP PPL**	**N_P/Domain N_P**
YouTube	youtube	context-loss	1281.9174	**3.3604**	1
YouTube	youtube	dict-loss	1281.9174	**3.3604**	1
YouTube	youtube	context	257.4089	260 072.7351	1
YouTube	youtube	dict	258.3046	278 252.2883	1
YouTube	youtube	vanilla	184.3536	663 891.0267	1
YouTube	news	context-loss	5021.3823	3.8017	**1.1313**
YouTube	news	dict-loss	5021.3823	3.8017	**1.1313**
YouTube	news	context	494.5056	321 449.3919	1.2359
YouTube	news	dict	553.3879	751 897.6226	2.7022
YouTube	news	vanilla	244.5672	88 239 614.03	1.3266
YouTube	arxiv	context-loss	3433.2579	3.991	**1.1876**
YouTube	arxiv	dict-loss	2731.1732	4.0168	1.1953
YouTube	arxiv	context	448.1877	948 690.9478	3.6477
YouTube	arxiv	dict	523.8087	504 478.2195	1.8130
YouTube	arxiv	vanilla	841.5366	508 670.5981	4.5647
YouTube	combined	context-loss	1364.8438	315.4954	93.8862
YouTube	combined	dict-loss	9910.6903	315.4954	93.8862
YouTube	combined	context	194.8153	367 184.1088	1.4118
YouTube	combined	dict	192.9121	1 415 625.616	5.087 56
YouTube	combined	vanilla	**162.7477**	2 243 779.964	**0.8828**
Combined	youtube	context-loss	1675.9424	**4.123**	**0.0153**
Combined	youtube	dict-loss	1675.9424	**4.123**	**0.0153**
Combined	youtube	context	297.2277	198 108.5509	0.7760
Combined	youtube	dict	307.2453	222 821.4406	0.2394
Combined	youtube	vanilla	209.5685	516 784.9742	1.7082
Combined	news	context-loss	3704.0467	4.1497	0.0154
Combined	news	dict-loss	3704.0467	4.1497	0.0154
Combined	news	context	289.5167	228 742.4686	0.8961
Combined	news	dict	317.2183	499 541.8261	0.5368
Combined	news	vanilla	159.1297	62 285 244.35	3.7888
Combined	arxiv	context-loss	1771.6198	4.176	**0.01550**
Combined	arxiv	dict-loss	1785.0295	4.2526	**0.01579**
Combined	arxiv	context	349.571	639 379.0527	2.5047
Combined	arxiv	dict	435.6399	380 467.2669	0.4088
Combined	arxiv	vanilla	471.326	358 696.7699	3.8419
Combined	combined	context-loss	2717.3075	269.349	1
Combined	combined	dict-loss	2717.3075	269.349	1
Combined	combined	context	141.0141	255 268.2881	1
Combined	combined	dict	130.5401	930 607.8148	1
Combined	combined	vanilla	**122.6778**	1 673 866.06	1
ArXiv	youtube	context-loss	1776.955	5.5599	1.2030
ArXiv	youtube	dict-loss	1776.955	5.5599	1.1501
ArXiv	youtube	context	248.606	143 742.8271	0.3201
ArXiv	youtube	dict	308.6261	172 417.5462	0.5997
ArXiv	youtube	vanilla	170.67	371 409.9797	**0.3079**
ArXiv	news	context-loss	4027.9929	5.5573	1.2025
ArXiv	news	dict-loss	4027.9929	5.5573	1.1495
ArXiv	news	context	267.5049	168 677.4707	**0.3756**
ArXiv	news	dict	298.6867	344 438.8796	1.1981
ArXiv	news	vanilla	112.5678	36 702 868.79	1.9408
ArXiv	arxiv	context-loss	1002.6701	**4.6216**	1
ArXiv	arxiv	dict-loss	1142.1247	4.8344	1
ArXiv	arxiv	context	316.4381	449 111.7608	1
ArXiv	arxiv	dict	372.3505	287 483.5986	1
ArXiv	arxiv	vanilla	554.2414	254 618.3941	1
ArXiv	combined	context-loss	683.2708	227.0362	49.1250
ArXiv	combined	dict-loss	683.2708	227.0362	46.9626
ArXiv	combined	context	147.8072	185 486.1319	0.4130

(continued)

Evaluated	Domain	Variant	NTP PPL	NCP PPL	N_P/Domain N_P
ArXiv	combined	dict	**104.4113**	579 093.5982	2.014 35
ArXiv	combined	vanilla	121.6163	1 201 784.535	**0.2194**
News	youtube	context-loss	1543.8018	4.0254	1.1013
News	youtube	dict-loss	1543.8018	4.0254	1.1013
News	youtube	context	255.102	200 136.836	0.9257
News	youtube	dict	246.7716	222 645.3484	**0.4730**
News	youtube	vanilla	163.8084	525 066.3462	0.6539
News	news	context-loss	1680.7794	**3.6551**	1
News	news	dict-loss	1680.7794	**3.6551**	1
News	news	context	171.6298	216 192.4155	1
News	news	dict	205.1227	470 666.8797	1
News	news	vanilla	250.4959	67 641 981.74	1
News	arxiv	context-loss	1870.5691	4.2061	1.1507
News	arxiv	dict-loss	1602.5725	4.2067	1.1509
News	arxiv	context	213.5528	606 818.904	2.8068
News	arxiv	dict	299.128	366 432.3553	**0.7785**
News	arxiv	vanilla	371.2635	343 182.1046	1.4821
News	combined	context-loss	1905.4865	275.443	75.3585
News	combined	dict-loss	1905.4865	275.443	**0.0013**
News	combined	context	102.0031	236 488.3831	0.5025
News	combined	dict	**79.9511**	957 988.102	0.0142
News	combined	vanilla	85.6417	1 600 376.441	0.0458