SALSA: Single-pass Autoregressive LLM Structured Classification

Anonymous ACL submission

Abstract

Despite their impressive generalization capabilities, instruction-tuned Large Language Models often underperform on text classification benchmarks. We introduce SALSA, a coherent pipeline that combines structured prompting, class-to-token mapping, and parameter-efficient fine-tuning, thereby avoiding cold-start training. Each class label is mapped to a distinct output token, and prompts are constructed to elicit a single-token response. During inference, the model's output is projected only onto the logits of the relevant class tokens, enabling efficient and accurate classification in a single forward pass. SALSA achieves state-of-the-art results across diverse benchmarks, demonstrating its robustness and scalability for LLM-based classification applications.

1 Introduction

Text classification is fundamental in natural language processing (NLP), underpinning applications such as spam detection, sentiment analysis, dialogue safety, and content moderation. Traditional methods involving handcrafted rules and features were limited by scalability and labor intensity. The emergence of deep learning transformed the field by enabling automated feature extraction through models such as word2vec (Mikolov et al., 2013), ELMo (Peters et al., 2018), and transformer-based architectures such as BERT (Devlin et al., 2019) and GPT (Brown et al., 2020), which deliver exceptional performance.

With the advent of Large Language Models (LLMs), particularly open-ended generative models, the capabilities of NLP systems have expanded significantly. These models, pre-trained on extensive corpora, encapsulate a wealth of transferable knowledge that can be leveraged for diverse downstream tasks, including text classification. Despite this, the effective adaptation of open-ended generative LLMs for classification still poses challenges, requiring efficient input representation and fine-tuning strategies.

Recent methods commonly utilize chain-of-thought (CoT) prompting (Wei et al., 2022), effective for reasoning, but computationally inefficient for classification. Such approaches also neglect valuable information in output distribution. In contrast, discriminative approaches (e.g., Pawar et al., 2024) remain underexplored.

In this paper, we introduce SALSA (Single-pass Autoregressive LLM Structured Classification), a method that adapts instruction-tuned, decoder-only LLMs for text classification. SALSA integrates structured prompt construction, targeted logit analysis, and fine-tuning into a unified pipeline. Its prompt-driven design enables strong zero-shot performance, providing a favorable initialization for subsequent tuning. Though built for generation, decoder-only LLMs can act as effective classifiers—matching or exceeding state-of-the-art results across benchmarks. By relying on a single forward pass, SALSA also offers a more efficient alternative to generation-based methods.

2 Background

Early NLP approaches used handcrafted features, deep learning then introduced RNNs and CNNs, improving classification (Kim, 2014). Transformer-based models, introduced by Vaswani et al. (Vaswani et al., 2017), revolutionized NLP by utilizing self-attention mechanisms for contextualized embeddings. Models like BERT represented a major leap forward by introducing bidirectional context understanding through unsupervised pre-training on large-scale corpora. Autoregressive transformer models like XLNet (Yang et al., 2019) demonstrated the benefits of autoregressive pre-training, outperforming traditional methods in classification tasks.

It has since been shown that large language models implicitly encode world knowledge, which can be extracted via their output logits (Petroni et al., 2019). Reformulating cloze-style tasks as multiple-choice classification has proven effective (Robinson et al., 2022), but the reliability of such approaches is highly sensitive to prompt structure (Cao et al., 2021).

Instruction tuning was a key breakthrough in demonstrating that language models can generalize across tasks when aligned with task-specific instructions (Wei et al., 2021). Building on this insight, decoder-only LLMs such as GPT (Brown et al., 2020), LLaMA (Touvron et al., 2023; Grattafiori et al., 2024), and Gemma (Team et al., 2024) have redefined the field, supporting zero-shot and in-context learning with strong generalization capabilities across a wide range of NLP tasks.

Parameter-efficient methods like BitFit (Ben Zaken et al., 2022) and Low-Rank Adaptation, LoRA (Hu et al., 2021), further limit overfitting by reducing the number of trainable parameters, ensuring stable fine-tuning especially in low-data scenarios. They also enable cost-effective deployment across tasks, requiring only minimal parameter swaps while leaving the base model intact.

When comparing results, fine-tuned encoder-based LLMs have achieved better performance in classification tasks, such as in the GLUE benchmark (Wang et al., 2018). Surprisingly, the much larger instruction decoder-only LLMs, which often outperform encoder-based LLMs in several tasks, generally fail to achieve competitive classification results (Bucher and Martini, 2024).

Our work aims to bridge the gap between the potential of instruction-decoder-only LLMs and the performance of classification tasks, both in terms of quality and efficiency.

3 Method

SALSA leverages the internal knowledge of LLMs by using their output estimated distribution to perform classification in a single forward pass per query. Our method employs LoRA for efficient parameter updates and knowledge exposure, allowing SALSA to deliver competitive performance.

Prompt Construction. We design a structured instruction prompt that encapsulates the task. The prompt first provides a clear task description, then maps each class to a unique single-token representation, and finally specifies the expected answer format, including fixed prefix and suffix elements. A structured response containing a placeholder token is appended to complete the prompt. This process is illustrated in Figure 1. A detailed example is provided in A.4

Forward Pass, Filtering, and Classification. We perform a single forward pass through the LLM to extract the logits for the placeholder token, which represent the model's predictions. These logits are then filtered based on the prompt's mapping and normalized via softmax to yield an estimated probability distribution over the classes. The final prediction corresponds to the class with the highest probability.

Training. We optimize our model using a backpropagation-based procedure (see A.2). In particular, we employ LoRA in conjunction with a cross-entropy loss function. The loss is defined as follows:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log(\hat{P}_{i,c}) \qquad (1)$$

where N is the number of samples, C is the number of classes, $y_{i,c}$ represents the ground truth labels, and $\hat{P}_{i,c}$ denotes the predicted probabilities. See A.3 for more details.

Controlling the Precision–Recall Trade-off. Adjusting decision threshold values offers precise control over the trade-off between precision and recall. This flexibility allows the model to be tailored to specific application needs, enabling tuning to optimize performance based on the desired balance.

Efficient Single-Pass Inference. SALSA eliminates autoregressive overhead by computing all logits in a single forward pass, reducing latency, resource use and cost. Mapping classification to a single-token output ensures only valid class tokens are considered, enhancing efficiency and correctness.

4 Experiments and Results

4.1 Datasets

We evaluated SALSA on multiple text classification datasets, including a subset of GLUE (Wang et al., 2018), covering SST-2 (Socher et al., 2013), MRPC (Dolan and Brockett, 2005), QQP (Iyer et al., 2017), MNLI (Bowman et al., 2015), QNLI (Rajpurkar et al., 2016), and RTE (Dagan et al., 2005). Additional datasets included AG's News (Zhang et al., 2015) for topic classification, IMDb (Maas et al., 2011) for binary sentiment analysis, and Yelp-5

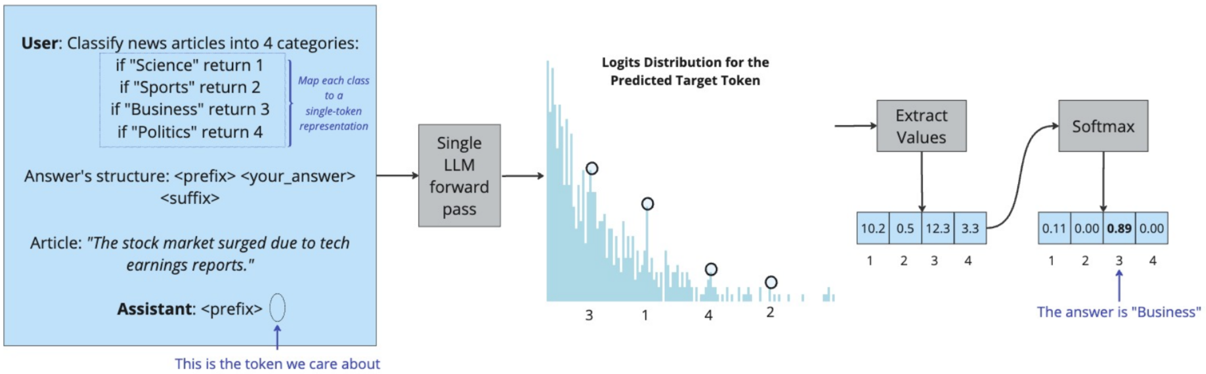

Figure 1: SALSA single-token classification pipeline: each category is mapped to a distinct token, and the LLM's logits determine the predicted label in one forward pass.

(Zhang et al., 2015) for multi-class sentiment analysis. We further included MedNLI (Romanov and Shivade, 2018) for clinical natural language inference, MedMCQA (Pal et al., 2022) for multiple-choice medical question answering, and HateXplain (Mathew et al., 2020) for hate speech and offensive language detection. For more details see section A.1.

4.2 Analysis

In this section, we delve into a comprehensive analysis of SALSA by examining performance metrics, convergence efficiency, and other key aspects across various benchmarks.

State-of-the-Art Results. SALSA demonstrates state-of-the-art performance across multiple text classification benchmarks, as outlined in Table 4.2 (and Table 2).

The method consistently outperforms existing models, including T5-11B (Raffel et al., 2020), XLNet (Yang et al., 2019), RoBERTa$_{\text{LARGE}}$ (Liu et al., 2019), and ALBERT (Lan et al., 2019). Furthermore, we compared SALSA against the top three performers on the GLUE benchmark, Turing ULR v6 (Team, 2022), Vega v1 (Zhong et al., 2023), and Turing ULR v5 (Tiwary and Zhou, 2021), and SALSA outperforms them all in 3 of 7 tasks.

Furthermore, we evaluated SALSA on three domain-specific benchmarks: MedNLI, MedMCQA, and HateXplain. As shown in Table 3, SALSA outperforms previous SOTA, demonstrating strong generalization in diverse NLP tasks.

For each validation set experiment, we train the model five times with different random seeds and report the average performance on the validation set. For test set experiments, we evaluate the model that achieves the highest results on the validation set using the GLUE test set evaluation server. These findings validate the efficiency and robustness of SALSA in leveraging generative LLMs for classification tasks.

Zero-Shot and Few-Shot Classification. To further evaluate SALSA, we conducted zero-shot and few-shot classification experiments using Meta's Instruct LLaMA 3.3 70B model.

In the zero-shot setting, the model received structured prompts containing task instructions and class options, without labeled examples. For few-shot classification, we added ten balanced random examples as in-context demonstrations. In both cases, the model generated open-ended responses that we parsed to extract the predicted classes.

We also applied SALSA in both settings, without fine-tuning. The zero-shot variant included only task instructions and class labels; the few-shot variant appended a few formatted examples. As shown in Table 4.2, SALSA clearly outperforms standard prompting approaches.

Efficient Optimization and Convergence. To evaluate optimization efficiency, we compared SALSA to standard fine-tuning, where a linear classification head is added atop the base LLM's final token output. Both methods were trained with identical hyperparameters to minimize cross-entropy loss. As shown in Figure 2, SALSA achieves faster convergence and consistently higher training and validation accuracy across steps. These results underscore its ability to reduce training time while improving generalization, making it well-suited for resource-constrained settings.

5 Conclusion

SALSA exhibits consistent performance across its pipeline. As shown in Tables 4.2, it achieves strong

		QQP	SST-2	RTE	MRPC	QNLI	MNLI$_{\text{M}}$	MNLI$_{\text{MM}}$
(V)	Zero Shot	81.4	94.9	86.3	77.0	90.7	81.9	80.9
(V)	Few Shot	81.5	96.1	85.2	77.2	91.4	80.1	80.2
(V)	RoBERTa$_{\text{LARGE}}$	92.2	96.4	86.6	90.9	94.7	90.2	90.2
(V)	ALBERT	92.2	96.9	89.2	90.9	95.3	90.8	90.8
(V)	XLNet	92.3	97.0	85.9	90.8	94.9	90.8	90.8
(V)	SALSA Zero Shot †	82.1	95.0	90.6	76.4	92.7	84.1	83.1
(V)	SALSA Few Shot †	83.3	95.4	92.0	80.1	92.9	86.7	86.3
(V)	SALSA	**92.4±0.2**	**97.1±0.2**	**94.2±0.4**	**91.7±0.5**	**96.7±0.2**	**92.8±0.3**	**92.6±0.2**
(T)	BERT$_{\text{LARGE}}$	89.3	94.9	70.1	85.4	92.7	86.7	85.9
(T)	T5-11B	90.6	97.5	92.8	90.4	96.9	92.2	91.9
(T)	Turing ULR v6	90.9	97.5	93.6	92.3	96.7	92.5	92.1
(T)	Vega v1	**91.1**	**97.9**	92.4	**92.6**	96.7	92.2	91.9
(T)	Turing ULR v5	91.1	97.6	94.1	91.7	**97.9**	92.6	**92.4**
(T)	SALSA	90.9	**97.9**	**94.8**	91.1	97.1	**92.7**	92.0

Table 1: Performance metrics of SALSA compared to baseline models across multiple GLUE Benchmark datasets. Results are reported separately for the validation (V) and test (T) sets, with accuracy as the key evaluation metric. SALSA achieves state-of-the-art performance on all validation tasks and outperforms competitors on 3 out of 7 test tasks. Test set results are benchmarked against the top 3 GLUE leaderboard models as of January 27, 2025. †No fine-tuning applied.

	AG News	IMDb	Yelp-5
Zero Shot	88.8	95.2	62.7
XLNet	95.5	96.8	72.9
SALSA	**95.9±0.1**	**97.6±0.1**	**74.2 ±0.2**

Table 2: Accuracy on AGNews, IMDb, and Yelp-5 test datasets.

	MedNLI	MedMCQA	HateXplain
Zero-Shot	83.4	70.3	51.5
SOTA	90.2†	73.6†	70.4†
SALSA	**91.3±0.4**	**74.1±0.3**	**71.8±0.4**

Table 3: Accuracy on MedNLI, MedMCQA, and HateXplain test datasets. †SOTA sources: GatorTron-large for MedNLI (Yang et al., 2022), GPT-4 for MedMCQA (Nori et al., 2023), and BERT-MRP for HateXplain (Kim et al., 2022).

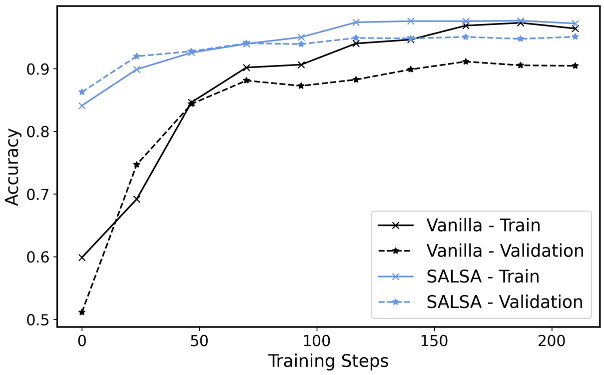

Figure 2: Convergence comparison between SALSA and Vanilla fine-tuning on RTE (Dagan et al., 2005). SALSA achieves faster convergence with higher accuracy on both training and validation sets, indicating better generalization and training efficiency.

zero-shot results even without tuning. With fine-tuning, SALSA improves further without the instability often seen in cold-start training (Figure 2). It also reaches state-of-the-art accuracy across diverse tasks—sentiment analysis, medical QA, and hate speech detection—demonstrating broad applicability and strong generalization (Tables 2).

By reducing classification to a single forward pass, SALSA enables high-throughput use of large models, offering a more efficient alternative to generation-based approaches. Its use of LoRA adapters also preserves the base model's capabilities for other LLM tasks.

While prompt design remains partly empirical, our ablation study (Section A.5) shows that fine-tuning mitigates label-token sensitivity. SALSA further supports regression tasks via discrete class ensembles (Section A.6), extending its scope.

Future directions include systematic prompt optimization, adaptive thresholding, and unified extensions for multi-label and multi-task settings (Section A.7). Overall, SALSA offers a flexible and efficient framework for robust, general-purpose classification with generative LLMs.

6 Limitations

One key limitation of SALSA is its reliance on accessing the internal logit distribution of large language models (LLMs), which restricts its use to models or third-party services that expose such information. Additionally, the structured prompt design used to map classes to single tokens may not be applicable in all scenarios, particularly in tasks with more complex or nuanced label representations. Another concern is model contamination. Since we have no control over the data used to train the underlying LLM there is the possibility that some test examples may have been inadvertently incorporated during unsupervised training. Finally, SALSA inherits the biases and ethical concerns of its underlying LLM. As these models are trained on large-scale web corpora, they may encode and propagate societal biases, necessitating responsible use in real-world applications.

References

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

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, Lisbon, Portugal. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Martin Juan José Bucher and Marco Martini. 2024. Fine-tuned 'small' llms (still) significantly outperform zero-shot generative ai models in text classification. *arXiv preprint arXiv:2406.08660*.

Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. 2021. Knowledgeable or educated guess? revisiting language models as knowledge bases. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1860–1874, Online. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

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, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde,

Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve

Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 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. *arXiv preprint arXiv:2106.09685*.

Shankar Iyer, Nikhil Dandekar, and Kornel Csernai. 2017. First quora dataset release: Question pairs.

Jiyun Kim, Byounghan Lee, and Kyung-Ah Sohn. 2022. Why is it hate speech? masked rationale prediction for explainable hate speech detection. *Preprint*, arXiv:2211.00243.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2020. Hatexplain: A benchmark dataset for explainable hate speech detection. *CoRR*, abs/2012.10289.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. 2023. Capabilities of gpt-4 on medical challenge problems. *Preprint*, arXiv:2303.13375.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa : A large-scale multi-subject multi-choice dataset for medical domain question answering. *Preprint*, arXiv:2203.14371.

Sachin Pawar, Nitin Ramrakhiyani, Anubhav Sinha, Manoj Apte, and Girish Palshikar. 2024. Why generate when you can discriminate? a novel technique for text classification using language models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1099–1114, St. Julian's, Malta. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? 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)*, pages 2463–2473. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(1):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Joshua Robinson, Christopher Michael Rytting, and David Wingate. 2022. Leveraging large language models for multiple choice question answering. *arXiv preprint arXiv:2210.12353*.

Alexey Romanov and Chaitanya Shivade. 2018. Lessons from natural language inference in the clinical domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1586–1596, Brussels, Belgium. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

Microsoft Turing Team. 2022. Microsoft turing universal language representation model (t-ulrv6).

Saurabh Tiwary and Lidong Zhou. 2021. Microsoft turing universal language representation model, t-ulrv5, tops xtreme leaderboard and trains 100x faster. *Microsoft Research Blog*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Xi Yang, Aokun Chen, Nima PourNejatian, Hoo Chang Shin, Kaleb E Smith, Christopher Parisien, Colin Compas, Cheryl Martin, Mona G Flores, Ying Zhang, Tanja Magoc, Christopher A Harle, Gloria Lipori, Duane A Mitchell, William R Hogan, Elizabeth A Shenkman, Jiang Bian, and Yonghui Wu. 2022. Gatortron: A large clinical language model to unlock patient information from unstructured electronic health records. *Preprint*, arXiv:2203.03540.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS)*, Red Hook, NY, USA. Curran Associates Inc.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Qihuang Zhong, Liang Ding, Keqin Peng, Juhua Liu, Bo Du, Li Shen, Yibing Zhan, and Dacheng Tao. 2023. Bag of tricks for effective language model pretraining and downstream adaptation: A case study on glue. *Preprint*, arXiv:2302.09268.

A Appendices

A.1 Datasets

We used multiple datasets to evaluate SALSA, focusing on text classification tasks.

GLUE Benchmark. We evaluated SALSA on a subset of tasks from the GLUE benchmark (Wang et al., 2018) and report both the task details and evaluation metrics. Specifically, we tested on the following tasks: the Stanford Sentiment Treebank (SST-2; Socher et al. (2013)), the Microsoft Research Paraphrase Corpus (MRPC; Dolan and Brockett (2005)), the Quora Question Pairs (QQP; Iyer et al. (2017)), the Multi-Genre Natural Language Inference Corpus (MNLI; Bowman et al. (2015)), the Stanford Question Answering Dataset (QNLI; Rajpurkar et al. (2016)), and Recognizing Textual Entailment (RTE; Dagan et al. (2005)).

AG's News. The AG's News dataset (Zhang et al., 2015) includes 120,000+ news articles across four categories (World, Sports, Business, Science/Technology), testing LLM robustness with diverse topics and journalistic tones.

IMDb. The IMDb data set (Maas et al., 2011) is a benchmark for binary sentiment analysis with positive or negative movie reviews, testing classification models on diverse styles of writing, topics, and sentiment intensities.

Yelp-5. The Yelp-5 dataset (Zhang et al., 2015), used for multi-class sentiment analysis, contains customer reviews rated 1-5 stars, challenging models with varied review lengths, tones, and topics.

HateXplain. HateXplain (Mathew et al., 2020) is a benchmark dataset for explainable hate speech detection, sourced from social media platforms. Each post in the dataset is annotated from three perspectives: a three-class classification (hate, offensive, or normal), the targeted community, and rationales highlighting the specific text spans that justify the annotations.

MedNLI. MedNLI (Romanov and Shivade, 2018) is a specialized natural language inference (NLI) dataset tailored for the clinical domain. It comprises sentence pairs extracted from the Past Medical History sections of MIMIC-III clinical notes, annotated by physicians to determine whether a given hypothesis can be inferred from a premise.

MedMCQA. MedMCQA (Pal et al., 2022) is a comprehensive multiple-choice question answering dataset designed to emulate real-world medical entrance examinations. Each question is accompanied by multiple answer options and detailed explanations.

For the train:validation:test size split and the number of samples in each dataset used for the evaluation, see Table 4.

Dataset	Train Size	Val. Size	Test Size
SST-2	67.3k	0.8k	1.8k
MRPC	3.6k	0.4k	1.7k
QQP	363.8k	40.4k	390.9k
MNLI$_m$	392.7k	9.8k	9.8k
MNLI$_{mm}$	392.7k	9.8k	9.8k
QNLI	104.7k	5.4k	5.4k
RTE	2.4k	0.3k	3.0k
AG News	120.0k	7.6k	–
IMDb	25.0k	25.0k	–
Yelp-5	650.0k	50.0k	–
MedNLI	11.2k	1.4k	1.4k
MedMCQA	182.8k	4.2k	6.2k
HateXplain	16.0k	2.0k	2.0k

Table 4: Dataset Sizes

A.2 Detailed Inference and Tuning Algorithm

The algorithm 1 outlines the explicit steps of SALSA's approach, covering both the training and inference flows. While using LLMs for classification at inference time is not a novel concept, steps 5 and 6 distinguish SALSA by showing how it leverages LLMs not just for auto-generation, but also for their underlying statistical properties - resulting in a richer and more informative output representation, and consequently, better performance. During training, SALSA goes beyond the generic objective of predicting the correct next token for every position. Instead, it focuses specifically on the task-related tokens and updates the model weights based solely on the loss computed from these tokens, making the training process more efficient and better aligned with the classification objectives.

A.3 Training Details

The base model was Meta's Instruct LLama 3.3 70b (Meta's license). It was tuned for a total of 6 epochs, and gradient accumulation steps set to

Algorithm 1 SALSA's Training and Inference for Single-Task, Single-Label, Multi-Class Classification

Require: instructions, answer template, answer's start ▷ Input parameters

Definition: Let N be the vocabulary size.
for each s in samples-to-classify **do**
  $x \leftarrow$ wrap in the method's notation and tokenize$(s, input_parameters)$
  logits $\leftarrow$ model's forward_pass(x) ▷ logits' size $=$ |input| $\times N$
  $y_{placeholder} \leftarrow$ logits[placeholder] ▷ $y_{placeholder}$'s size $= N$
  $y_{relevant} \leftarrow y_{placeholder}$[categories] ▷ $y_{relevant}$'s size =|categories|
  $y_{\text{prob}} \leftarrow$ softmax$(y_{relevant})$
  $y_{true} \leftarrow$ one_hot(true_label, |categories|)
  loss $\leftarrow$ cross_entropy$(y_{\text{prob}}, y_{true})$
  model.backward_pass(loss)
  update_parameters()
  report $\arg\max(y_{\text{prob}})$
end for

Note: The blue-colored lines correspond to training-specific steps.

50 with batch size 1 to effectively handle large batch sizes in limited memory environment. To ensure reproducibility, a fixed random seed was used throughout the experiments.

LoRA(Hu et al., 2021) was used for fine-tuning, the rank was set to 8, the alpha parameter to 16, and a dropout rate of 0.05. It is 103M trainable parameters.

Optimization was carried out using the Adam optimizer (Kingma, 2014) with default parameter settings, where beta1=0.9, beta2=0.999, and epsilon=1E-8. A linear learning rate scheduler was employed, incorporating 100 warmup steps to progressively increase the learning rate at the beginning of training to 1E-4. After warmup the learning rate was reduced linearly to 0. For each experiment, the best-performing validation epoch was identified, and the experiment was repeated five times with different data shuffling seeds to ensure robustness of results.

Empirical observations revealed that optimal validation performance was typically achieved within the first 2 to 3 epochs. Training beyond this point, particularly when each sample was seen more than three times, often resulted in overfitting for small size datasets. The hardware used for this work was the Nvidia DGX system with eight H100 80GB GPU blades, and each model training run lasted between 1 and 36 hours. In this work, no hyperparameter optimization was conducted.

A.4 Prompt Construction Example

Figure 3 shows a sample prompt compiled from the RTE dataset. The prompt follows the default chat template of Instruct LLaMA 3.3, beginning with a default system prompt, followed by a user prompt containing task-specific instructions and data, and ending with the assistant response template. The compiled prompt is tokenized and processed in a single forward pass through the LLM to produce a classification output.

A.5 Ablation Study on Label Mapping Strategies

We performed an ablation study on the RTE dataset to evaluate how different label mapping schemes affect classification performance. Specifically, we tested six mappings: numerical ('0', '1'), reverse numerical ('1', '0'), alphabetical ('A', 'B'), reverse alphabetical ('B', 'A'), semantic ('Y', 'N'), and reverse semantic ('N', 'Y').

The inclusion of "Y" and "N" label tokens was motivated by their implicit alignment with natural language concepts of affirmation and negation ("Yes"/"No"). We hypothesized that when the token aligns semantically with the correct label—e.g., "Y" for entailment—the model may perform better in a zero-shot setting. Conversely, using misleading or contradictory mappings, such as assigning "N" to entailment, could degrade performance due to interference with prior token associations.

Table 5 summarizes the average accuracy (mean ± standard deviation over 5 runs) for each mapping strategy, evaluated in both zero-shot and fine-tuned conditions.

In the zero-shot setting, we observe substantial variation in performance across mappings. Reversing the labels ("N/Y") led to the poorest perfor-

System prompt

Task + input

Class mapping

Answer format with masked token

```
<|begin_of_text|>
<|start_header_id|>system <|end_header_id|>
Cutting Knowledge Date: December 2023 Today Date: 26 Jul 2024 <|eot_id|>
<|start_header_id|>user <|end_header_id|>
Given the premise:
<PREMISE> Mangla was summoned after Madhumita’s sister Nidhi Shukla,
who was the first witness in the case. </ PREMISE>
and hypothesis:
<HYPOTHESIS> Shukla is related to Mangla. </ HYPOTHESIS>
Is the hypothesis entailed by the premise?
Provide answer in format: <ANSWER>#Number</ANSWER>
where the number is one of the following:
0 - entailment
1 - not entailment
<|eot_id|>
<|start_header_id|>assistant <|end_header_id|>
<ANSWER>X </ANSWER>
<|eot_id|>
```

Figure 3: A compiled prompt from RTE dataset before applying a forward pass.

Mapping Strategy	Zero-Shot(%)	Finetuned (%)
Numerical (0/1)	90.6	95.1 ± 0.4
Reverse Numerical (1/0)	89.1	94.4 ± 0.2
Alphabetical (A/B)	91.3	94.5 ± 0.3
Reverse Alphabetical (B/A)	90.2	94.2 ± 0.3
Semantic (Y/N)	85.5	95.4 ± 0.7
Reverse Semantic (N/Y)	44.7	94.0 ± 0.2
Mean	**81.9 ± 18**	**94.9 ± 0.5**

Table 5: Ablation study on different label mapping strategies for the RTE dataset. Accuracy is reported as mean ± std over 5 runs.

mance, suggesting that mismatches between token semantics and label intent can confuse the model. After finetuning the differences between mappings diminish considerably, with all variants converging to similar accuracy levels. These findings confirm that the finetuning process effectively suppresses sensitivity to the mapping choices and enables the model to adapt even in the presence of initially misleading token associations.

A.6 Discrete to Continuous Extension

For tasks involving continuous value estimation, such as the STS-B benchmark (Cer et al., 2017) from the GLUE, we adapt our method to produce scalar outputs through a discretization-based approach.

We represent the predicted score as the expected value over a fixed set of discrete scalar values. Each value corresponds to a predefined class token and is associated with a probability derived from the model's output distribution. Formally, let $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$ denote the set of discrete values (e.g., $[0.0, 0.2, \dots, 5.0]$), and let $P(s_i \mid x)$ be the probability assigned to state s_i given input x. The model prediction $\hat{y}$ is computed as:

$$\hat{y} = \sum_{i=1}^{n} P(s_i \mid x) \cdot s_i \qquad (2)$$

During training on STS-B, we perform the inverse operation. Given a scalar label $y \in [0, 5]$, we identify the two discrete values s_i and s_{i+1} such that $s_i \leq y \leq s_{i+1}$, and assign probabilities:

$$y = \alpha \cdot s_i + (1 - \alpha) \cdot s_{i+1}, \qquad (3)$$

$$P(s_i \mid x) = \alpha, \qquad (4)$$

$$P(s_{i+1} \mid x) = 1 - \alpha \qquad (5)$$

This construction ensures that the expected value of the predicted distribution matches the ground truth during supervision, while keeping the label space discrete and aligned with our logit-based framework.

Our method achieves a Pearson/Spearman correlation of **93.8/93.6** on the STS-B test set, compared to Turing v5's **93.7/93.3**, representing a new state-of-the-art result.

A.7 Possible extention

SALSA's framework can be naturally extended to more complex scenarios. For multi-label classification, one can replace the softmax layer with a sigmoid function and apply a probability threshold to select all relevant classes. For multi-task classification, a prompt with placeholders for each task enables the extraction of separate logits distributions, allowing simultaneous classification across multiple tasks in a single forward pass (see Figure 4).

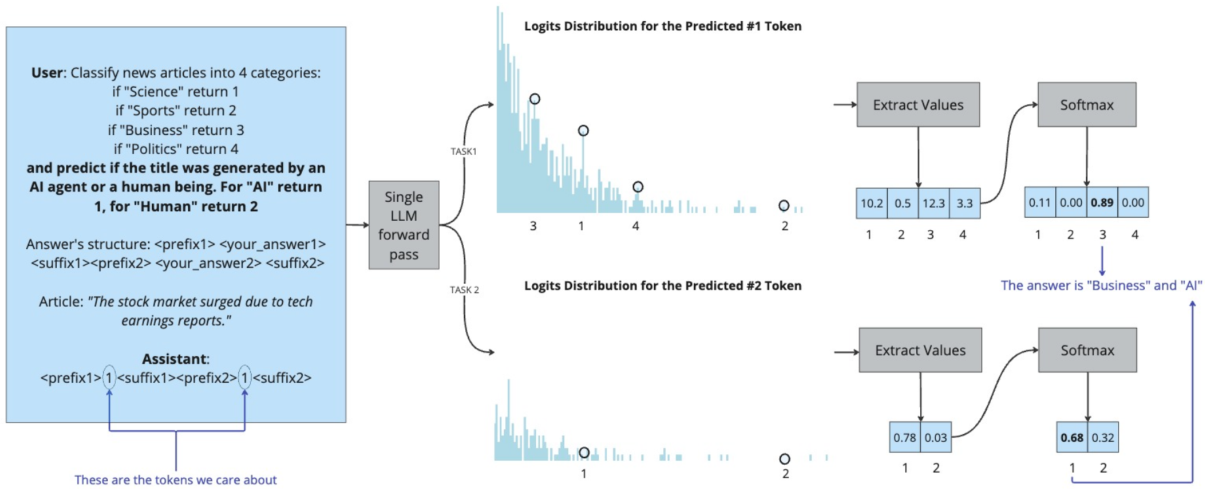

Figure 4: SALSA two-token classification pipeline: the LLM's logits are used in a single pass to predict both the article's topic (1–4) and its source (AI=1 or Human=2).