SALSA: Single-pass Autoregressive LLM Structured Classification

Anonymous ACL submission

Abstract

We propose SALSA (Single-pass Autoregressive LLM Structured Classification), a method that harnesses the transferred knowledge of open-ended generative Large Language Models (LLMs) for text classification. By structuring task prompts and response formats while analyzing only the relevant target logits, SALSA enables computationally efficient classification with the generation of a single token only. We demonstrate that fine-tuning LLMs using Low-Rank Adaptation (LoRA) using SALSA's approach, achieves state-of-the-art results on selected classification benchmarks. Not only does SALSA improve results, but it also achieves top-rated results faster than existing methods.

1 Introduction

Text classification is a fundamental task in natural language processing (NLP). Its applications include spam detection, sentiment analysis, dialogue safety, and content moderation. Traditional methods relied on rule-based systems and early machine learning models using hand-crafted features, which were limited by labor-intensive processes and scalability issues. 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.

In this paper, we present SALSA (Single-pass Autoregressive LLM Structured Classification), a novel approach to harness the potential of open-ended generative LLMs for text classification tasks. SALSA leverages structured prompts and tailored response formats, combined with targeted logits analysis, to fully exploit the generative capacities of these models. SALSA can be applied to any model that provides logit outputs. Given such model, we employ Low-Rank Adaptation (Hu et al., 2021) to fine-tune the models with a focus on optimizing the cross-entropy loss over relevant logits, from the classification-related tokens only. This approach results in state-of-the-art performance on benchmark datasets faster than existing methods, requiring fewer training steps. Thanks to SALSA's design, tuning begins with zero-shot performance, giving it an advantageous position on the optimization surface. To the best of our knowledge, this is the first work to show that generative decoder-only LLMs outperform conventional methods for classification tasks.

2 Background

Text classification is a core NLP task, categorizing text into predefined labels. It includes (1) multi-class classification, assigning one label per instance; (2) multi-label classification, allowing multiple labels per instance; and (3) multi-task classification, where models handle multiple tasks simultaneously.

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 (Devlin et al.,

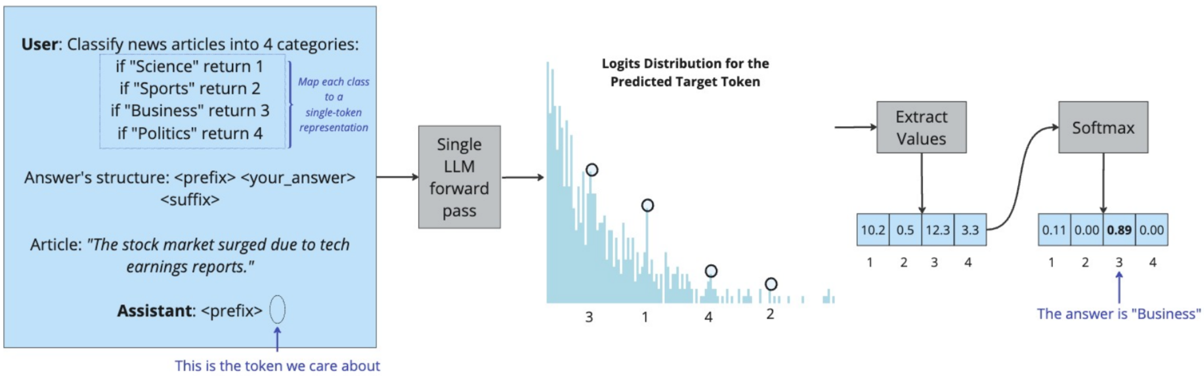

Figure 1: SALSA single-token classification pipeline: each category is mapped to a distinct token, and the LLM's logits determine the predicted label in one forward pass.

2019) represented a major leap forward by introducing bidirectional context understanding through unsupervised pretraining on large-scale corpora. Autoregressive transformer models like XLNet (Yang et al., 2019) demonstrated the benefits of autoregressive pretraining, outperforming traditional methods in classification tasks.

Recent years have witnessed significant advances in the development of decoder-based LLMs, generating text autoregressively. They perform classification via zero-shot and few-shot learning, enabling generalization with minimal data and in context learning. Breakthrough models like LLaMA (Touvron et al., 2023; Grattafiori et al., 2024), Gemma (Team et al., 2024), and GPT (Brown et al., 2020) have redefined text-based tasks. Their exceptional capabilities, as highlighted in (Brown et al., 2020), enable high performance across diverse tasks, including text classification.

Another leap in the field came from new and enhanced training methods for LLMs: Instruction aware training has been shown to transform language models into robust zero-shot learners (Wei et al., 2021). Parameter-efficient methods like BitFit (Ben Zaken et al., 2022) and LoRA (Hu et al., 2021) further limit overfitting by reducing the number of trainable parameters, ensuring stable finetuning especially in low-data scenarios. They also enable cost-effective deployment across tasks, requiring only minimal parameter swaps while leaving the base model intact.

A common method for autoregressive LLM-based classification is prompting the model to generate a label, which introduces variability unsuited for categorical tasks. Techniques like Chain-of-Thought (CoT) prompting (Wei et al., 2022) enhance performance by structuring reasoning steps but require generating many tokens for each classification query, making it expensive and inefficient.

When comparing results, finetuned encoder-based large language models have achieved state-of-the-art (SOTA) performance in classification tasks, such as those in the GLUE benchmark (Wang et al., 2018). Surprisingly, the much bigger generative decoder-only LLMs, which often outperforms 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 generative decoder-only LLMs and the performance for classification tasks, both in terms of quality and efficiency.

3 Method

SALSA (Single-pass Autoregressive LLM Structured Classification) is a novel approach for addressing classification tasks with large language models (LLMs). It leverages the internal knowledge of LLMs by using their output logits 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.

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 Algorithm 1 in the Appendix). In particular, we employ LoRA in conjunction with a cross-entropy loss function. The loss is defined as follows:

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

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.1 for more details.

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 (Zhang et al., 2015) for multi-class sentiment analysis. For more details see section A.2.

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 1 (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.

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 assess SALSA, we compared it with zero-shot and few-shot classification experiments using Meta's Instruct LLama 3.3 70B model.

In the zero-shot setting, we used structured prompts without any labeled examples. The model generated open-ended responses that we parsed to determine the predicted classes.

For a few-shot classification, we randomly selected ten balanced examples to include in the prompt as contextual cues for the model. As in zero-shot classification, we parsed the model output to identify the classes.

Efficient Optimization and Convergence. To assess SALSA, we implemented traditional (Vanilla) fine-tuning by passing input text through the same base LLM and adding a linear layer to the final token's output, matching the number of classes. Fine-tuning used identical LoRA parameters to minimize cross-entropy loss. Figure 2 compares SALSA's convergence to traditional fine-tuning (Vanilla). SALSA demonstrates consistently higher training and validation accuracy across training steps, achieving faster convergence and superior performance. This efficiency highlights SALSA's effectiveness in structured classification tasks, reducing training time while enhancing generalization, making it highly practical for resource-constrained scenarios.

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 dynamic 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 and resource use. Mapping classification to a single-token output ensures only valid class tokens are

		QQP	SST-2	RTE	MRPC	QNLI	MNLI_{M}	MNLI_{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)	$\text{RoBERTa}_{\text{LARGE}}$	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	**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)	$\text{BERT}_{\text{LARGE}}$	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	90.7	**97.9**	**94.8**	91.2	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.

	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 of SALSA, XLNet, and Zero-Shot on AGNews, IMDb, and Yelp-5 test datasets.

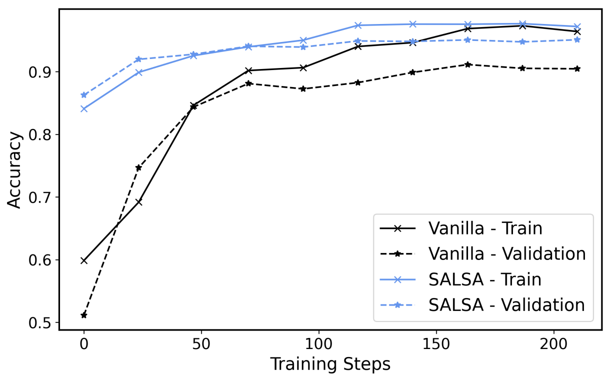

Figure 2: Convergence comparison between SALSA and Vanilla fine-tuning on RTE (Dagan et al., 2005). SALSA achieves faster convergence with higher accuracy on both training and validation sets, indicating better generalization and training efficiency.

considered, enhancing efficiency and correctness.

Possible Extensions. 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 (see Figure 3 in the Appendix).

5 Discussion

SALSA demonstrates that structured prompts and targeted logit extraction can effectively harness the generative capacity of large language models for text classification. By condensing classification into a single forward pass, SALSA achieves stronger performance than baselines on diverse benchmarks while also converging more rapidly. This efficiency is particularly valuable in resource-constrained scenarios, where fine-tuning large models can be computationally demanding. Furthermore, SALSA's design readily extends to multi-label and multi-task settings, indicating its potential as a flexible framework for real-world NLP pipelines.

However, prompt engineering remains partly empirical, highlighting the need for systematic strategies to optimize prompt formats. Future work could investigate adaptive thresholding for multi-label tasks and comprehensive evaluations across multi-task and multi-label datasets. In general, SALSA offers a practical and extensible framework that uses pre-trained generative models for robust text classification.

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*.

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.

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.

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*.

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.

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.

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.

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 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 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.

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.2 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.

For the train:validation:test size split and the number of samples in each dataset used for the evaluation, see Table 3.

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	–

Table 3: Dataset Sizes

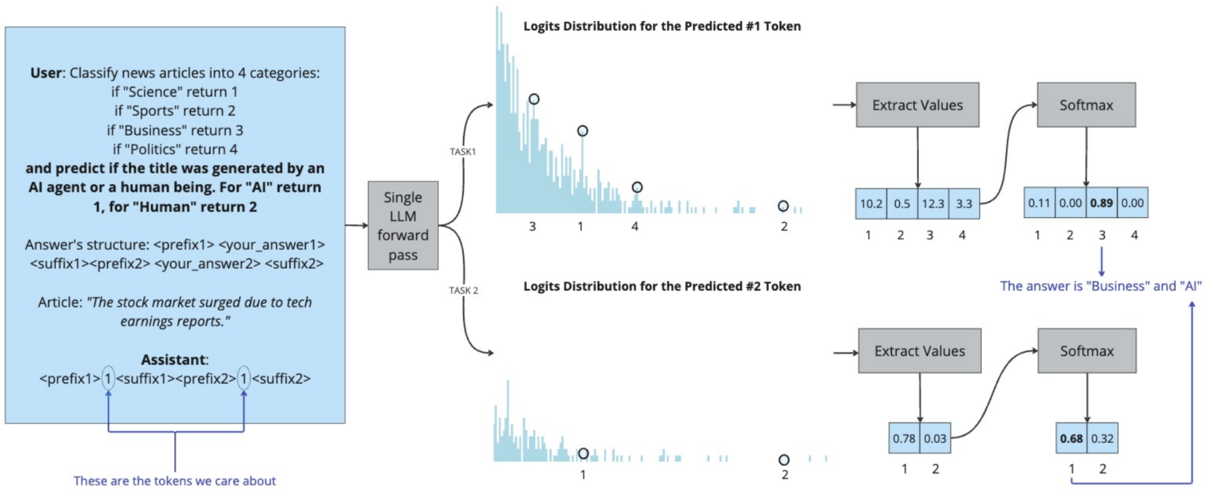

Figure 3: SALSA two-token classification pipeline: the LLM's logits are used in a single pass to predict both the article's topic (1–4) and its source (AI=1 or Human=2).

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_{prob})
end for

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