Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Compact Open Encoder Model for Calibrated Typed Decisions on Unseen Tasks

Md. Minaruzzaman Shovon

minar.svn@gmail.com

Chittagong University of Engineering & Technology

Research Article

Keywords: decision models, zero-shot classification, text encoders, calibration, multitask learning, ModernBERT

Posted Date: October 20th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11163029/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

A Compact Open Encoder Model for Calibrated Typed Decisions on Unseen Tasks

Md. Minaruzzaman Shovon
Department of Water Resources Engineering
Chittagong University of Engineering and Technology (CUET)
Chattogram, Bangladesh
minar.svn@gmail.com

***Abstract*—Many software systems need quick, structured judgments: choose an option, estimate a yes/no probability, or assign an ordinal score. Decision models produce these answers in one forward pass, avoiding the cost and fragility of generating and parsing text. We present typic-bert, an open 396M-parameter decision model built on the Ettin encoder. The model places a `[MASK]` marker before each option, scores the marker states with a small head, and returns a calibrated probability distribution. We train it on 50,000 examples drawn from 25 public datasets and vary the question wording, labels, option count, and polarity to reduce surface-level shortcuts. Training takes 46 minutes on one free Kaggle T4 GPU. Across four tasks excluded from training (2,000 questions), typic-bert achieves 69.9% accuracy, compared with 56.7% for Laya (421M) on the same inputs. It nearly matches Laya on intent routing (81.2% versus 82.0%) and answers in 29 ms per question. It also retains 85-90% accuracy when the request follows up to 7,000 unrelated tokens, while both tested Laya checkpoints fall to 25-30%. We release the model openly and discuss important limitations, including task selection and single-seed training.**

***Index Terms*—decision models, zero-shot classification, text encoders, calibration, multitask learning, ModernBERT**

I. Introduction

Software often needs to make small but important judgments: route a ticket, check whether an agent contradicts a tool result, detect a prompt injection, or estimate urgency. A large language model can handle these tasks, but generating an answer and parsing it back into a fixed type adds latency and another point of failure. Decision models offer a simpler interface. Given a state and a typed question, they return probabilities over the allowed answers in one pass [1].

TypeSafe's Jev introduced this "System One" interface as a closed API [1]. Convai's Laya provides an open alternative built on ModernBERT-large [2], [3]. Its model card, however, reports that the base checkpoints are close to chance on the typed-decisions benchmark when used zero-shot (0.362 accuracy against a 0.318 random baseline). The stronger results require fine-tuning on that benchmark's training split [2]. We therefore ask a practical question: can a compact open model, trained cheaply, transfer to decision tasks it has never seen?

This report describes typic-bert, our attempt to answer that question with limited resources. Our contributions are:

- A simple option-marker architecture on an Ettin encoder that handles choice, yes/no, and ordinal questions with one scoring head (Section III).
- A training recipe that converts 25 public datasets into one format and adds augmentation designed to prevent surface shortcuts; full training fits on one free T4 GPU in under an hour (Section IV).
- A comparison with Laya on identical held-out inputs and a long-context stress test, reported together with their limitations (Sections VI and VII).
- An open release of the weights, loader code, and a demo application [4].

II. Related Work

A. Zero-Shot Text Classification

Yin *et al.* [5] framed zero-shot classification as textual entailment, letting a model trained on natural language inference score arbitrary labels. Instruction-tuned models such as FLAN [6] and T0 [7] showed that training on many tasks phrased in many ways improves generalization to unseen tasks. Our data design applies the same idea to an encoder with explicit option markers.

B. Bidirectional Encoders

BERT [8] established bidirectional encoders for classification. ModernBERT [3] modernized the architecture with rotary position embeddings, alternating local and global attention, and an 8,192-token context. Ettin [9] released paired encoders and decoders from 17M to 1B parameters trained with the same recipe; we use its 150M and 400M encoders.

C. Decision Models

Jev is described publicly as a non-autoregressive model with a parallel sampler, trained with Reinforcement Learning for Calibrated Decisions (RLCD) [1]. Laya [2] scores every option at its own `[MASK]` token, uses a two-layer transformer decision head, and is trained with RLCD against strictly proper scoring rules. typic-bert uses the same option-marker interface but a simpler head, plain cross-entropy training, and post-hoc temperature scaling.

D. Calibration

Modern neural networks are often overconfident; temperature scaling [10] fixes much of this with one parameter fitted on held-out data. We report expected calibration error (ECE) [11].

III. Method

A. Problem Formulation

An input consists of a context c, a question q, and a set of options $O = \{o_1, \ldots, o_n\}$, with $2 \le n \le 20$. The model returns a probability p_i for each option. Three question types share this form: a choice question uses arbitrary labels; a yes/no question uses the options "no" and "yes" in random order, and p_{yes} is the probability that the statement is true; a score question uses ordered levels, and the expected level is $\sum_k k p_k$.

B. Input Encoding

The input sequence is

$$\begin{aligned}&\texttt{[CLS]}\ q\ \texttt{[SEP]}\ c\ \texttt{[SEP]}\\ &\quad\texttt{[MASK]}\ o_1\ \texttt{[MASK]}\ o_2\ \cdots\ \texttt{[MASK]}\ o_n.\end{aligned} \tag{1}$$

The question is limited to 64 tokens and each option to 32 tokens, shortened further if many options would not fit. Only the context is truncated so that the full sequence fits the training length of 384 tokens; the options are never cut. Because the options are part of the input, new labels and new tasks require no retraining.

C. Scoring Head

Let H be the final hidden states of the encoder and m_i the position of the marker before option i. The head scores each marker state and normalizes over the options:

$$h_i = H[m_i], \tag{2}$$

$$s_i = \boldsymbol{w}_2^{\mathsf{T}}\,\mathrm{GELU}(W_1 h_i + b_1) + b_2, \tag{3}$$

$$p_i = \frac{\exp(s_i/T)}{\sum_j \exp(s_j/T)}. \tag{4}$$

During training, $T = 1$ and the loss is the cross-entropy $-\log p_y$ of the correct option y; padded option slots are masked out. The encoder and head are trained jointly.

D. Calibration

After training, a single temperature T is fitted on the validation set by minimizing negative log-likelihood over a grid from 0.5 to 6.0, then applied to all outputs [10]. Inference is one forward pass with no sampling, so the same input produces the same output on the same hardware.

IV. Training Data

A. Sources and Unified Format

We converted 25 public datasets into the (c, q, O, y) format (Table I). Classification labels become options; multiple-choice answers become options; binary labels become yes/no questions; and graded labels become ordered levels (STS-B similarity rounded to 0-5, Yelp ratings as 1-5 stars, and HelpSteer2 helpfulness, correctness, or coherence as 0-4). Civil Comments was balanced, with toxicity ≥ 0.5 as positive and < 0.1 as negative. Most sources contributed 2,000 examples; MNLI, ANLI, and CLINC150 contributed 3,000, and prompt-injections contributed all 546 available examples. The 51,546 examples were split into 50,000 for training and 1,546 (3%) for validation.

B. Augmentation

Without variation, a model can learn shortcuts such as reacting to one fixed question wording or one fixed set of label strings. We therefore applied the following augmentations to the training data:

- *Question wording:* two to four phrasings per task, chosen at random.
- *Label wording:* synonyms for common labels (e.g., "positive" and "favorable") in 50% of cases and random casing changes in 30%.
- *Option sets:* for large label spaces (CLINC150, DBpedia, and Yahoo Answers), the correct label plus random distractors for a total of 4 to 20 options; option order is shuffled for choice and yes/no questions, while score levels keep their order.
- *None of these:* in 8% of label questions, the correct label is removed and "none of these" becomes the answer; in another 10%, it is added as a distractor.
- *Polarity:* 30% of yes/no questions are asked in negated form with the label flipped (e.g., "Is this a legitimate message?" instead of "Is this spam?").

C. Held-Out Evaluation Tasks

Four tasks were excluded from training entirely: Banking77 [32] (intent routing; 5 to 20 of the 77 intents shown per question), CommonsenseQA [33], COPA [34], and OpenBookQA [35], with 500 questions each and no augmentation. Two further sources (SocialIQA for training and TREC for evaluation) failed to load and were dropped.

V. Experimental Setup

A. Models and Training

We trained on the Ettin 150M and 400M encoders [9] with the Hugging Face Transformers Trainer [36]. We used AdamW with weight decay 0.01, linear decay with 6% warmup, and fp16 mixed precision. The 400M run also used gradient checkpointing. All runs used a single NVIDIA T4 (16 GB) on Kaggle's free tier. Table II lists the settings.

B. Baseline

We compared against Laya's English checkpoint (ModernBERT-large, 421M) using the official `laya` package, version 0.3.20, at default settings [2]. Each held-out question was converted into Laya's input format: the context as the state and a choice question whose criteria are the same options. Laya warned that its shipped temperature for questions with 11 or more options was invalid, so its confidence on those Banking77 questions is uncalibrated; this does not affect accuracy.

TABLE I
TRAINING SOURCES

Type	Datasets	Examples
Choice	MNLI [12], ANLI [13], RACE [14], ARC [15], HellaSwag [16], WinoGrande [17], CLINC150 [18], AG News, DBpedia, Yahoo Answers [19], Emotion [20], SST-2 [21], TweetEval sentiment [22]	29,000
Yes/no	BoolQ [23], SciTail [24], QQP [25], PAWS [26], Civil Comments [27], prompt-injections [28], SMS Spam [29], TweetEval hate and irony [22]	16,546
Score	HelpSteer2 [30], STS-B [31], Yelp reviews [19]	6,000
Total	25 datasets	51,546

TABLE II
TRAINING SETTINGS

Setting	150M	400M (final)
Parameters	149.6M	395.8M
Learning rate	5×10^{-5}	3×10^{-5}
Epochs	2	1.5
Effective batch size	32	32 (8×4)
Max. input length	384	384
Optimizer steps	3,126	2,345
Training time (T4)	17.9 min	45.8 min
Fitted temperature T	1.55	1.55

TABLE III
EFFECT OF ENCODER SIZE

Metric	150M	400M
Validation accuracy (seen)	75.9%	82.3%
Held-out accuracy (unseen)	58.3%	69.9%
Banking77	74.8%	81.2%
CommonsenseQA	45.8%	59.8%
COPA	68.0%	83.0%
OpenBookQA	44.6%	55.4%
Held-out ECE, raw	0.203	0.147
Held-out ECE, calibrated	0.116	0.072
Latency per question	28 ms	29 ms

C. Metrics

We report accuracy, ECE with 15 bins [11], and median latency per question at batch size 1 on the T4. Validation accuracy on the 1,546 held-back examples measures seen tasks; held-out accuracy measures generalization to unseen tasks.

VI. RESULTS

A. Effect of Model Size

Table III compares the two encoder sizes trained on the same data. Moving from 150M to 400M parameters raised held-out accuracy by 11.6 points, with the largest gains on the reasoning tasks, while latency stayed at about 29 ms. In earlier pilot runs with a smaller 12-task dataset, a 68M encoder reached only 32.7% on CommonsenseQA; these pilots are not directly comparable but point in the same direction.

For the 400M model, held-out loss was lowest around step 1,500 (0.82) and rose to 1.01 by step 2,000 while held-out accuracy stayed near 70%, a sign that the model became more confident without becoming more accurate. Stopping near one epoch would likely be slightly better; we report the final model.

TABLE IV
HELD-OUT ACCURACY ON IDENTICAL INPUTS

Task	Random	typic-bert (396M)	Laya (421M)
Banking77	9%	81.2%	82.0%
CommonsenseQA	20%	59.8%	41.6%
COPA	50%	83.0%	70.2%
OpenBookQA	25%	55.4%	32.8%
Overall	26%	69.9%	56.7%
ECE	-	0.072 (0.147 raw)	0.069 (raw)
Latency	-	29 ms	37 ms

Random is the expected accuracy of uniform guessing given each question's number of options. Laya was run with `laya` 0.3.20 at default settings.

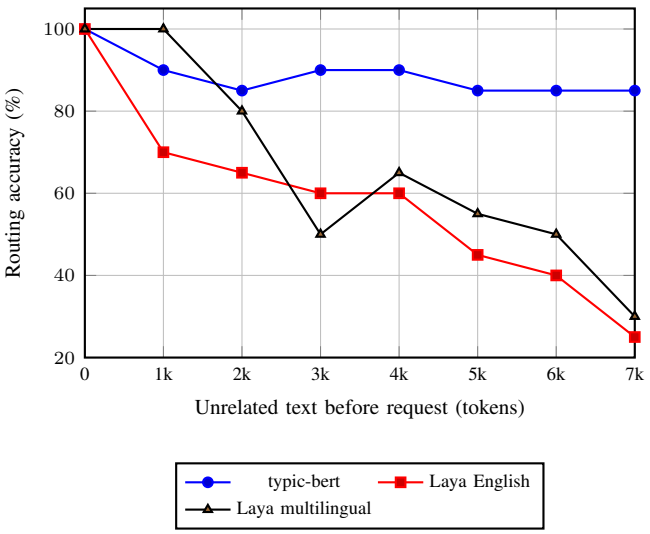


Fig. 1. Routing accuracy as unrelated text is added before the request. Each point contains 20 English requests; all models use a maximum length of 8,192 tokens.

B. Comparison with Laya

Table IV shows both models on the same 2,000 held-out questions. typic-bert is 13.2 points more accurate overall. With 2,000 questions, the standard error of each overall accuracy is about 1 point, so this gap is well beyond sampling noise. Laya is 0.8 points ahead on Banking77, which is within noise. After temperature scaling, the two models are equally well calibrated, although Laya is better calibrated without it.

TABLE V
REPRESENTATIVE VALIDATION RESULTS (400M)

Source	Type	n	Accuracy
DBpedia	choice	64	100.0%
SMS Spam	yes/no	60	100.0%
CLINC150	choice	100	99.0%
BoolQ	yes/no	67	94.0%
WinoGrande	choice	71	90.1%
RACE	choice	70	67.1%
STS-B	score	63	65.1%
Yelp reviews	score	56	64.3%
HelpSteer2	score	45	60.0%
TweetEval sentiment	choice	54	59.3%

Selected strongest and weakest sources. Samples contain 44 to 100 questions; score tasks use exact-level accuracy.

C. Seen Tasks

Table V shows representative validation results for the 400M model, including the strongest and weakest sources. Topic, intent, and spam classification are nearly solved, while tweet sentiment, HelpSteer2 quality ratings, and RACE reading comprehension remain the weakest. RACE likely suffers from truncating long passages to 384 tokens. For score tasks, ranking quality is more informative than exact accuracy: Spearman correlation between expected and true level is 0.81 for STS-B and 0.87 for Yelp but only 0.29 for HelpSteer2.

D. Long-Context Robustness

To test whether a model can still find a request inside a long input, we placed 20 support requests (four each for billing, technical support, account access, shipping, and cancellation) after 0 to 7,000 tokens of unrelated text. The filler covered topics such as geology, astronomy, and cooking. Although typic-bert was trained only on 384-token inputs, it keeps 17 or 18 of 20 requests correct across the full range. Both Laya checkpoints decline much more sharply, reaching only 5 or 6 correct at 7,000 tokens (Fig. 1). Median latency at 7,000 tokens was 1.76 s for typic-bert, 1.86 s for Laya English, and 1.09 s for the smaller multilingual Laya encoder.

VII. LIMITATIONS

These results should be read with the following limitations in mind:

- *Evaluation-set selection:* We chose both test sets. Our training data contains reasoning tasks (ARC, HellaSwag, and WinoGrande) similar in style to CommonsenseQA, COPA, and OpenBookQA, which likely favors typic-bert on those tasks. Laya was designed for business workflows such as invoice processing and security incidents, and its typed-decisions benchmark has not yet been run on typic-bert.
- *Contamination:* We did not check for textual overlap between the training and held-out sets.
- *Single seed and small samples:* Each configuration was trained once. Per-task held-out results carry about ± 2 points of sampling error, and each long-context row about ± 10 points.
- *Restricted label spaces:* Banking77 questions show 5 to 20 of the 77 intents rather than all of them, which is easier than the full task.
- *Calibration:* Laya is better calibrated out of the box; typic-bert matches it only after temperature scaling with one global temperature.
- *Scope:* The model is English only, weak at rating answer quality, sensitive to option order like other decision models, and not a substitute for human review in high-stakes decisions.

VIII. CONCLUSION AND FUTURE WORK

Our results suggest that useful zero-shot decision models do not require a large training budget. typic-bert trains in under an hour on one free GPU, yet it outperforms a similarly sized Laya checkpoint across our held-out tasks, nearly matches it on intent routing, and remains reliable on much longer inputs than those seen during training. These findings are promising but not final. The next steps are to test on Laya's benchmark and other neutral evaluations, repeat training with several seeds, measure the contribution of each augmentation, check for contamination, add realistic agent-decision data, and calibrate each question type separately.

ACKNOWLEDGMENT

We thank Kaggle for free GPU access, the Ettin authors for their open encoders, the Laya authors for releasing open weights that made a direct comparison possible, and the authors of all datasets used in this work.

DECLARATIONS

Data Availability

This study used only publicly available secondary datasets; no new data were collected. The training data were drawn from MNLI, ANLI, RACE, ARC, HellaSwag, WinoGrande, CLINC150, AG News, DBpedia, Yahoo Answers, Emotion, SST-2, TweetEval, BoolQ, SciTail, QQP and STS-B from the GLUE benchmark, PAWS, Civil Comments, prompt-injections, SMS Spam, HelpSteer2, and Yelp reviews.

The held-out evaluation data were drawn from Banking77, CommonsenseQA, COPA from the SuperGLUE benchmark, and OpenBookQA. All datasets are available from these public repositories under their respective licenses. The model weights, loader code, and demonstration application are available at the typic-bert model repository.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Consent to Publish

Consent to Publish declaration: not applicable.

Consent to Participate

Consent to Participate declaration: not applicable.

Ethics

Ethics declaration: not applicable.

Competing Interests

The author declares that there are no competing interests.

Author Contributions

Md. Minaruzzaman Shovon conceived the study, developed the methodology, conducted the analysis and experiments, and wrote and revised the manuscript. The author read and approved the final manuscript.

REFERENCES

[1] TypeSafe AI, "Jev," Product website, 2026. [Online]. Available: https://typesafe.ai/

[2] Convai Innovations, "Laya: Multilingual, non-autoregressive system 1 decision model," Hugging Face model card, 2026. [Online]. Available: https://huggingface.co/convaiinnovations/laya

[3] B. Warner, A. Chaffin, B. Clavié, O. Weller, O. Hallström, S. Taghadouini, A. Gallagher, R. Biswas, F. Ladhak, T. Aarsen, N. Cooper, G. Adams, J. Howard, and I. Poli, "Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference," *arXiv preprint arXiv:2412.13663*, 2024.

[4] M. M. Shovon, "typic-bert," Hugging Face model repository, 2026. [Online]. Available: https://huggingface.co/minar-svn/typic-bert

[5] W. Yin, J. Hay, and D. Roth, "Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach," in *Proc. EMNLP-IJCNLP*, 2019.

[6] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned language models are zero-shot learners," in *Proc. ICLR*, 2022.

[7] V. Sanh, A. Webson, C. Raffel, S. H. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, T. L. Scao, A. Raja, M. Dey, M. S. Bari, C. Xu, U. Thakker, S. Sharma, E. Szczechla, T. Kim, G. Chhablani, N. Nayak, D. Datta, J. Chang, M. T.-J. Jiang, H. Wang, M. Manica, S. Shen, Z. X. Yong, H. Pandey, R. Bawden, T. Wang, T. Neeraj, J. Rozen, A. Sharma, A. Santilli, T. Fevry, J. A. Fries, R. Teehan, S. Biderman, L. Gao, T. Wolf, and A. M. Rush, "Multitask prompted training enables zero-shot task generalization," in *Proc. ICLR*, 2022.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019.

[9] O. Weller *et al.*, "Seq vs seq: An open suite of paired encoders and decoders," *arXiv preprint arXiv:2507.11412*, 2025.

[10] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. ICML*, 2017.

[11] M. P. Naeini, G. F. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using bayesian binning," in *Proc. AAAI*, 2015.

[12] A. Williams, N. Nangia, and S. R. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *Proc. NAACL-HLT*, 2018.

[13] Y. Nie, A. Williams, E. Dinan, M. Bansal, J. Weston, and D. Kiela, "Adversarial NLI: A new benchmark for natural language understanding," in *Proc. ACL*, 2020.

[14] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, "RACE: Large-scale ReAding comprehension dataset from examinations," in *Proc. EMNLP*, 2017.

[15] P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord, "Think you have solved question answering? try ARC, the AI2 reasoning challenge," *arXiv preprint arXiv:1803.05457*, 2018.

[16] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi, "HellaSwag: Can a machine really finish your sentence?" in *Proc. ACL*, 2019.

[17] K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi, "WinoGrande: An adversarial winograd schema challenge at scale," in *Proc. AAAI*, 2020.

[18] S. Larson, A. Mahendran, J. J. Peper, C. Clarke, A. Lee, P. Hill, J. K. Kummerfeld, K. Leach, M. A. Laurenzano, L. Tang, and J. Mars, "An evaluation dataset for intent classification and out-of-scope prediction," in *Proc. EMNLP-IJCNLP*, 2019.

[19] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. NeurIPS*, 2015.

[20] E. Saravia, H.-C. T. Liu, Y.-H. Huang, J. Wu, and Y.-S. Chen, "CARER: Contextualized affect representations for emotion recognition," in *Proc. EMNLP*, 2018.

[21] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. EMNLP*, 2013.

[22] F. Barbieri, J. Camacho-Collados, L. E. Anke, and L. Neves, "TweetEval: Unified benchmark and comparative evaluation for tweet classification," in *Findings of EMNLP*, 2020.

[23] C. Clark, K. Lee, M.-W. Chang, T. Kwiatkowski, M. Collins, and K. Toutanova, "BoolQ: Exploring the surprising difficulty of natural yes/no questions," in *Proc. NAACL-HLT*, 2019.

[24] T. Khot, A. Sabharwal, and P. Clark, "SciTail: A textual entailment dataset from science question answering," in *Proc. AAAI*, 2018.

[25] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *Proc. ICLR*, 2019.

[26] Y. Zhang, J. Baldridge, and L. He, "PAWS: Paraphrase adversaries from word scrambling," in *Proc. NAACL-HLT*, 2019.

[27] D. Borkan, L. Dixon, J. Sorensen, N. Thain, and L. Vasserman, "Nuanced metrics for measuring unintended bias with real data for text classification," in *Companion Proc. WWW*, 2019.

[28] deepset, "prompt-injections," Hugging Face dataset. [Online]. Available: https://huggingface.co/datasets/deepset/prompt-injections

[29] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of SMS spam filtering: New collection and results," in *Proc. ACM DocEng*, 2011.

[30] Z. Wang *et al.*, "HelpSteer2: Open-source dataset for training top-performing reward models," *arXiv preprint arXiv:2406.08673*, 2024.

[31] D. Cer, M. Diab, E. Agirre, I. nigo Lopez-Gazpio, and L. Specia, "SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation," in *Proc. SemEval*, 2017.

[32] I. nigo Casanueva, T. Temčinas, D. Gerz, M. Henderson, and I. Vulić, "Efficient intent detection with dual sentence encoders," in *Proc. 2nd Workshop on NLP for Conversational AI*, 2020.

[33] A. Talmor, J. Herzig, N. Lourie, and J. Berant, "CommonsenseQA: A question answering challenge targeting commonsense knowledge," in *Proc. NAACL-HLT*, 2019.

[34] M. Roemmele, C. A. Bejan, and A. S. Gordon, "Choice of plausible alternatives: An evaluation of commonsense causal reasoning," in *Proc. AAAI Spring Symp. on Logical Formalizations of Commonsense Reasoning*, 2011.

[35] T. Mihaylov, P. Clark, T. Khot, and A. Sabharwal, "Can a suit of armor conduct electricity? a new dataset for open book question answering," in *Proc. EMNLP*, 2018.

[36] T. Wolf *et al.*, "Transformers: State-of-the-art natural language processing," in *Proc. EMNLP: System Demonstrations*, 2020.