Decision 1.0

Towards Open Decision Foundation Models

Xunzhuo Liu, Haichen Zhang, Andy Luo,
Huamin Chen, Xue Liu, Bowei He

Abstract. Software often needs a choice, a Boolean judgment, or a score rather than a generated explanation. We present Decision 1.0, an open-weight family that maps textual state, instructions, and runtime-defined candidates directly to typed probability distributions. Six models span bidirectional encoders, causal language-model backbones, and workflow specialization, with deployed sizes from 0.57B to 7.94B parameters. Training combines verifiable decision tasks, natural-language supervision, and adaptation with replay or retention objectives. A common English/Chinese evaluation covers 54 task slices. Lux-9B reaches 84.10% on explicit decisions and 91.46% on inference, improving over its untuned Qwen3.5 reference by 10.18 and 11.88 percentage points. It ranks second among 15 systems by an equal-weight mean of five evaluation panels, while composition and transfer remain below Jev. We examine probability quality, candidate-order sensitivity, and the cost of answering multiple questions. The released weights and inference code provide an inspectable basis for research on reusable decision models.

Date: September 2026

1 Introduction

A routing service chooses a backend. An invoice pipeline decides whether a discrepancy requires review. An evaluator assigns a grade using a supplied rubric. Each operation requires language understanding, but its result belongs to a small, application-defined space. The model must interpret both the evidence and the meaning of the available outcomes.

Language models can perform these operations through prompting and structured generation. Constrained decoding can enforce syntax, but the resulting interface still expresses a decision as output tokens. Fixed-label classifiers return numbers directly, yet usually require retraining when their label set changes. Candidate-conditioned prediction combines a direct numerical output with a task definition supplied at inference time.

Jev popularized this formulation through the System One interface, which pairs unstructured state with typed questions and returns probabilistic decisions [1]. Its public service demonstrates the appeal of the interface, while its weights and full training method remain unavailable. Decision 1.0 studies an open implementation of this paradigm, building on both pretrained language models and open decision-model designs.

Our contribution is a family of six released models under three common decision types, together with a training formulation and an evaluation of their strengths and limits. Kai-0.6B is a multilingual encoder; Lex-0.6B adapts an earlier Kai checkpoint to operational workflows. Eos-0.8B, Sol-2B, Nox-4B, and Lux-9B use causal Qwen3.5 text backbones. Every model returns a decision distribution in one forward pass per question. We use *decision foundation model* for the intended role of a reusable, task-conditioned model that can be adapted to multiple decision problems; the term describes an objective rather than universal competence.

Table 1 Decision 1.0 model family. Parameter counts include the deployed decision readout; context covers the complete packed question. Lex is evaluated separately as a workflow specialist.

Model	Initialization	Parameters	Context	Architecture / scope
Kai-0.6B	Vela / mmBERT	0.57B	1K	Bidirectional; general
Lex-0.6B	Earlier Kai	0.57B	1K	Bidirectional; workflow
Eos-0.8B	Qwen3.5-0.8B	0.75B	16K	Causal; general
Sol-2B	Qwen3.5-2B	1.88B	16K	Causal; general
Nox-4B	Qwen3.5-4B	4.21B	16K	Causal; general
Lux-9B	Qwen3.5-9B	7.94B	16K	Causal; general

2 Task Formulation

A request contains textual state x and questions $\{q_j\}_{j=1}^{Q}$. A question specifies a type t, instructions u, and candidate descriptions $C = (c_1, \ldots, c_K)$. The model predicts

$$p_\theta(i \mid x, u, C, t) = \frac{\exp(z_i/T)}{\sum_{k=1}^{K} \exp(z_k/T)}, \tag{1}$$

where z_i is a candidate score and $T > 0$ is an inference temperature. Padded candidates are excluded from normalization.

Choice returns this distribution and the identifier of its maximum. **Noul**, the interface name for a Boolean judgment, returns $p(\text{true})$ over the alternatives false and true. **Score** returns a distribution over ordered rubric levels and its expected index, $\sum_i (i-1)p_i$. The native encoder interface also supports expectations over supplied numeric values.

For example, the same support record can be classified among billing, technical support, and escalation, then evaluated for whether sufficient evidence exists to issue a refund. Candidate descriptions define each task at runtime. If abstention or an unknown outcome is needed, the application must include it in the task definition: a closed candidate set cannot express an omitted answer.

The complete input includes state, instructions, candidates, and special tokens. Encoders accept up to 1,024 tokens; causal models accept up to 16,384. Overlength requests are rejected. Choice supports 2–255 candidates and the common Score interface supports 2–10 levels. These are interface bounds, not quality guarantees. Questions are computed independently and may be batched. Candidate order can affect predictions; causal serialization also includes candidate keys, whereas encoder IDs remain outside the token stream.

3 Model Architectures

Table 1 summarizes the family. Both architectures contextualize the supplied candidates and apply a shared scoring function, allowing the output cardinality to vary without replacing the task head.

3.1 Bidirectional decision encoders

Kai is initialized from Vela Encoder [15], a derivative of mmBERT [3] using the ModernBERT architecture [4]. Shared token embeddings and embedding normalization feed three independently parameterized, 22-layer encoder paths for Choice, Noul, and Score. Each path has width 768 and combines local and global attention. A type embedding is added after final normalization. Following Laya's candidate-marker and interaction-head design [11], two additional global-attention layers combine state, instructions, and candidate descriptions.

For the contextualized marker h_i of candidate i, the selected type head computes

$$z_i = w_t^\top \operatorname{GELU}(W_t \operatorname{LN}(h_i) + b_t) + a_t. \tag{2}$$

The scorer is shared across candidates within a type. Joint bidirectional encoding lets each candidate interact with every other candidate. Lex retains this architecture. The three paths increase stored parameters relative to a single encoder, while a question executes only its selected type path.

3.2 Causal decision models

The causal models adapt Qwen3.5 text backbones [5]. Eos and Sol have 24 layers; Nox and Lux have 32. In each group of four layers, three use Gated DeltaNet and one uses full causal attention. Pretrained embeddings, normalization, and feed-forward blocks are retained. Neither the vision tower nor the vocabulary-generation readout is used for decision inference.

Inputs end with a decision suffix after all candidate blocks. Let h_i be the hidden state at candidate i's endpoint and h_q the final input position. Separate LayerNorms yield $c_i = \operatorname{LN}_c(h_i)$ and $q = \operatorname{LN}_q(h_q)$. A shared head combines bilinear compatibility with a nonlinear interaction:

$$z_i = \frac{(W_k c_i)^\top (W_q q)}{\sqrt{d_h}} + w^\top \operatorname{GELU}(A_c c_i + A_q q + b), \qquad d_h = 256. \tag{3}$$

The head runs in FP32 above a BF16 backbone. An early candidate endpoint cannot attend to later candidates, but the final query sees the complete candidate set and conditions every score. This supports variable candidate counts without imposing permutation invariance.

3.3 From token generation to direct decisions

Figure 1 contrasts the two readouts on the same backbone topology. An autoregressive model projects its final hidden state onto a vocabulary and feeds the selected token back into the model. For an answer of M tokens, prompt prefill supplies the first prediction and $M - 1$ subsequent decode steps produce the remainder. A decision model instead gathers candidate representations and returns all K probabilities together. Its output cost is independent of the length of a textual answer.

This changes the readout and inference schedule, not the causal mask. The entire packed question still passes through the backbone. A one-token generative classifier also avoids a long decode loop, so the architectural distinction alone does not establish a latency advantage. Increasing Q introduces more contextual computation in the current implementation; batching can amortize device overhead, but shared request text is not a shared activation cache. Longer states and larger candidate menus also increase input work. We leave controlled latency comparisons to future evaluation.

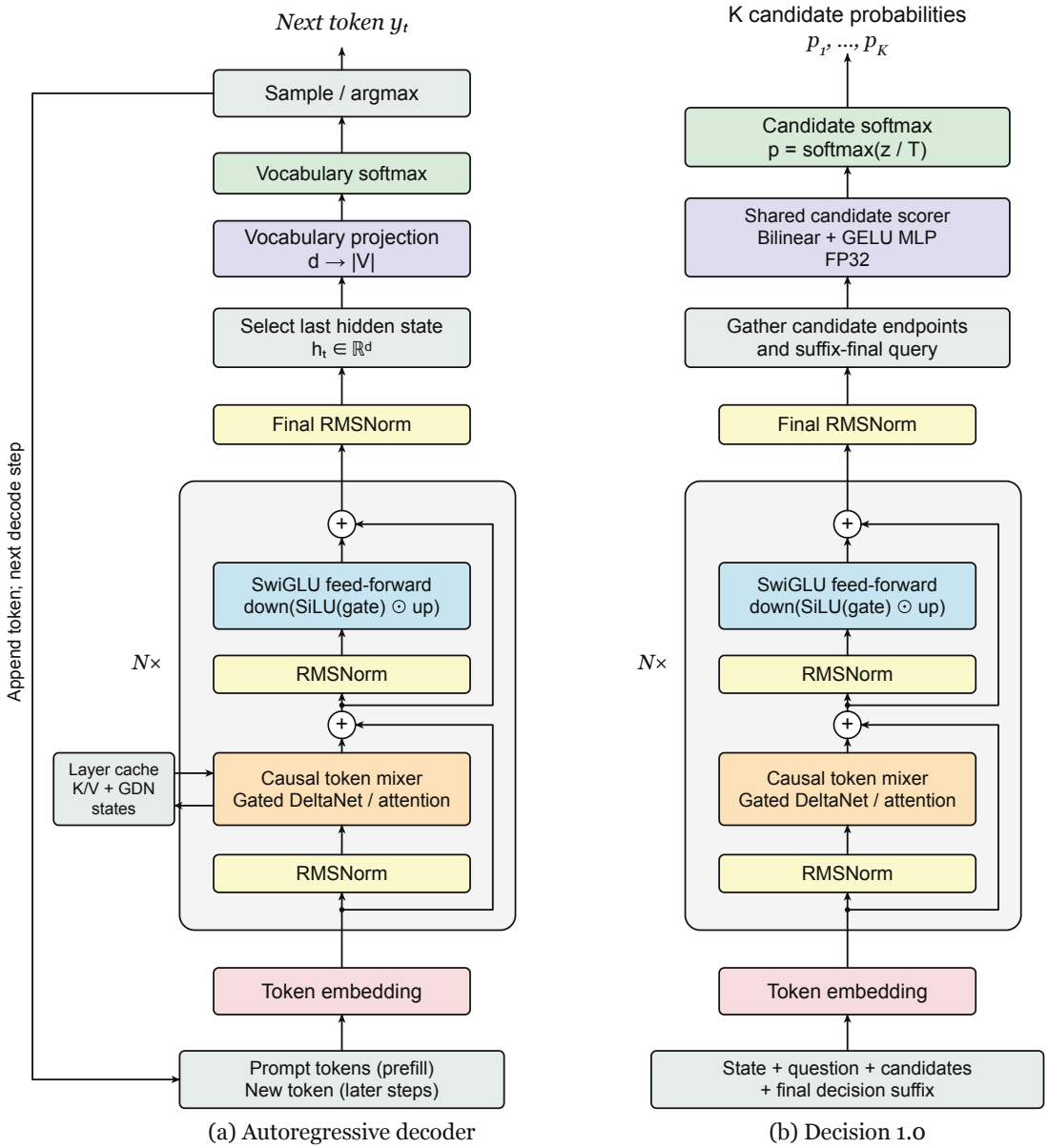

Figure 1 Autoregressive generation (left) and causal Decision inference (right), using the Qwen3.5 text-backbone topology. Residual paths, pre-normalization, token mixing, and feed-forward blocks remain. Decision replaces the vocabulary projection and token-feedback loop with candidate gathering and the shared head in Equation 3. The two branches represent different trained weights. One Decision forward pass answers one question; it does not jointly answer every question sharing the same state.

4 Data and Training

The central training problem is to bind a textual task definition to its allowed outcomes while preserving useful language understanding. Procedural supervision supplies explicit rules, controlled state changes, and verifiable answers. Natural-language annotations provide reading, intent, inference, and rubric judgments that templates alone do not cover. The documented Sol, Nox, and Lux mixtures do not use official Jev predictions as training labels.

4.1 Distribution supervision

In the documented Lex, Sol, Nox, and Lux distribution-supervised stages, a target $y \in \Delta^{K-1}$ gives the objective

$$\mathcal{L}_{\mathrm{CE}} = -\sum_{i=1}^{K} y_i \log p_\theta(i \mid x, u, C, t), \tag{4}$$

with $T = 1$ during training. Hard labels become one-hot distributions; source soft labels retain their recorded distribution. Human ordinal means can be represented by interpolation between adjacent levels, preserving the annotated mean. Candidate reordering must also reorder its targets, and ordered rubric levels cannot be shuffled without preserving their semantics.

4.2 Encoder adaptation and workflow specialization

Kai inherits multilingual supervision spanning inference, intent, evidence, and human ratings. Representative sources include SNLI, MASSIVE, BoolQ, TyDi QA, HelpSteer2, and OpenAssistant [9, 16, 17, 19–21]. Their annotations are converted into candidate decisions with supporting context where required. Lineage datasets and later replay examples overlap, so stage sizes are not additive corpus counts.

The released Kai incorporates adaptation on approximately 21,000 complete-input examples with frozen shared embeddings. It combines the adapted Choice and Score paths with an earlier Noul path. This is a fixed released model, not task-dependent checkpoint selection. Lex specializes an earlier Kai checkpoint on the 6,000 training decisions in `typed-decisions` [6]: customer service, invoices, security incidents, and agent traces. These are generated cases with teacher-derived labels. Lex combines soft and hard targets equally, adapts all three paths with frozen embeddings, and uses a fixed refit schedule before evaluation on the 2,000-decision test set.

4.3 Causal adaptation and retention

Sol and Nox use a progressive English/Chinese curriculum covering rules, evidence, arithmetic, ordinal judgments, candidate binding, and state transitions. Intent and inference annotations complement procedural tasks [22–24]. Replay preserves earlier task coverage as the curriculum broadens. Natural-language adaptation uses a 24,000-example mixture of retained decisions, Cosmos QA, SQuAD 2.0 answerability, and SNLI [7–9]. Source-parent and detected near-duplicate groups are kept together when splitting training, selection, and calibration, and known evaluation matches are excluded from custom training construction.

Sol continues candidate cross-entropy training. Nox adds supervised retention on natural-language examples using frozen vocabulary rows for legal answer letters; this auxiliary readout is removed at deployment. Lux starts independently from Qwen3.5-9B, combines natural-language adaptation with composition replay, and uses KL retention against the initial model of its final stage on existing examples. Added composition examples use hard-label supervision. These objectives address the tension between acquiring decision behavior and retaining pretrained capabilities.

Eos uses head-only warmup followed by full-model adaptation over 24,000 training rows. The family shares an interface, but not one controlled training recipe. Different initialization and data histories mean its results should not be interpreted as a scaling law or an ablation of attention direction.

4.4 Selection and calibration

The causal releases separate checkpoint selection from temperature calibration. Selection emphasizes decision accuracy, with probability quality used in tie-breaking where documented. A positive scalar temperature is fitted using negative log likelihood [10]. This changes distribution sharpness without changing the argmax. Lex uses a fixed final schedule and no test-based temperature fitting. Source transformations and disclosed training settings are documented with the released models; not every final optimizer setting or assembled training corpus is distributed.

5 Evaluation

We evaluate five general models on a common English/Chinese suite and Lex separately on its workflow test. The common suite contains 3,766 scored decisions across 54 task slices. Results refer to the fixed public revisions in Appendix A.

5.1 Protocol and comparators

The five panels measure complementary behavior. *Decisions* covers classification, Boolean constraints, evidence placement, state tracking, and rubric judgments. *Composition* tests assignment constraints, record identity, transaction recovery, and rule closure. These panels each contain 880 examples and macro-average ten families. *Reading* contains 480 BoolQ and English/Chinese Belebele examples [16, 18], weighting BoolQ by one half and each Belebele language by one quarter. *Inference* averages four 120-example sources: Cosmos QA, SQuAD answerability, SNLI, and supplied-fact QASC. *Transfer* uses micro accuracy on the 1,046 answerable examples of the external Kev suite [13]. Score questions use categorical accuracy here, not error in the returned expected ordinal score.

We compare Jev 1.13.0 through its official service, Laya's English and multilingual models, Decider, Kev at three sizes, and untuned Qwen3.5 references [1, 5, 11–13]. Untuned Qwen uses fixed answer-letter logits without generated-text parsing. Systems retain their native templates and temperatures. Kev runs with a BF16 backbone and FP32 head, with date injection disabled; this differs from its authors' FP32 configuration. Laya truncates some transfer inputs, while Jev's internal input retention cannot be inspected. Failures count as incorrect when a gold accuracy label exists. These are comparisons of released systems rather than isolated backbone effects.

Repeated evaluation during development has made much of the suite an observed regression benchmark. The external suite is excluded from custom training, checkpoint selection, and temperature fitting, but its observed results may inform later research. We therefore do not characterize the entire evaluation as a fresh blind test. Reported confidence intervals use paired bootstrap resampling over source groups and measure example-sampling uncertainty, not variation across training seeds.

5.2 General decisions and transfer

Lux ranks second by the five-panel mean (78.87%), behind Jev (83.40%). Its strongest results are 84.10% on Decisions and 91.46% on Inference. Relative to untuned Qwen3.5-9B, Lux gains 10.18 points on Decisions, 7.13 on Composition, 11.88 on Inference, and 4.11 on Transfer, with Reading 0.16 points lower. Sol and Nox improve Decisions by 16.62 and 13.11 points over their corresponding references. These comparisons support the usefulness of adaptation, while leaving the separate contributions of the head and training unresolved.

Table 2 General decision evaluation (accuracy, %). **Decision models are shaded and marked (ours).** Rank orders the unrounded equal-weight mean of five panels, a descriptive summary rather than pooled accuracy or a significance ranking. Panel aggregation follows Section 5.1; Lex is evaluated separately.

Rank	Model	Mean	Decisions	Composition	Reading	Inference	Transfer
1	Jev 1.13.0	83.40	79.10	66.38	94.53	89.79	87.19
2	**Lux-9B (ours)**	78.87	84.10	51.75	89.69	91.46	77.34
3	Kev 9B	74.40	76.75	45.75	86.72	83.54	79.25
4	**Nox-4B (ours)**	73.94	83.00	51.79	79.06	86.25	69.60
5	Kev 4B	72.60	71.90	48.54	81.88	84.58	76.10
6	Qwen3.5-9B	72.24	73.91	44.63	89.84	79.58	73.23
7	Decider 2B	71.26	64.01	46.58	92.03	84.38	69.31
8	Qwen3.5-4B	69.96	69.89	43.33	87.97	79.79	68.83
9	**Sol-2B (ours)**	67.53	73.75	46.08	76.56	84.17	57.07
10	**Eos-0.8B (ours)**	63.19	65.94	46.04	70.31	81.67	52.01
11	Kev 0.8B	60.04	60.14	42.29	67.81	68.75	61.19
12	Qwen3.5-2B	59.70	57.12	39.00	73.75	72.29	56.31
13	**Kai-0.6B (ours)**	54.33	57.96	40.83	54.69	69.79	48.37
14	Laya English	52.02	56.54	35.33	51.41	63.75	53.06
15	Laya Multilingual	48.27	47.25	38.92	50.78	57.29	47.13

The remaining gaps are substantial. Lux trails Jev by 14.63 points on Composition and 9.85 on Transfer; Kev 9B also retains higher transfer accuracy. Nox's gains accompany an 8.91-point reading decrease. Eos exceeds Kev 0.8B on the first four panels, but trails it on Transfer (52.01% versus 61.19%). Kai improves over both general Laya models on the first four panels, while its transfer accuracy remains below Laya English. The family offers different capacity and specialization choices rather than a uniform improvement on every task.

5.3 Workflow specialization

On the 2,000 original typed-decisions test judgments from 400 state groups, Lex reaches 78.15% accuracy versus 76.60% for the official Laya typed-decisions specialist. The paired group-bootstrap 95% interval for the 1.55-point gain is $[-0.20, +3.15]$ and includes zero. Lex improves Score accuracy, while Noul accuracy and probability-quality measures favor the baseline. This provides evidence of a specialization trade-off, not a statistically established overall advantage.

5.4 Probability quality and robustness

On answerable transfer examples, Lux has Brier score 0.300, negative log likelihood 0.609, and expected calibration error 5.89%. Jev obtains 0.191, 0.574, and 3.35%, respectively; Kev 9B obtains 0.295, 0.603, and 4.45%. The smaller general Decision models have larger errors on this set. Temperature fitting does not eliminate the transfer gap.

Across 36 paired candidate permutations, semantic choices change in 13.9% of Kai pairs, 11.1% for Eos, 38.9% for Sol, 16.7% for Nox, and 8.3% for Lux. Lux answers both variants correctly in 77.8% of pairs, illustrating why stability must be assessed alongside accuracy. On 110 unknowable requests, mean maximum probability remains 63.7% for Lux and 78.7% for Nox. These variants mix evidence removal, candidate deletion, and presentation changes, so they do not isolate a pure abstention effect. Applications need explicit abstention outcomes and validation on their own distributions.

6 Related Work

Candidate-conditioned classification connects to entailment-based zero-shot classification, where labels are supplied as descriptions rather than fixed output indices [14]. ModernBERT and mmBERT provide bidirectional representations; Qwen3.5 provides the causal initialization used here [3–5]. The decision interface is compatible with either attention direction.

Jev, Laya, Decider, and Kev explore closely related typed probabilistic interfaces and candidate readouts [1, 11–13]. We build on this work rather than claim candidate scoring or one-pass classification as new primitives. Decision's contribution is a released encoder and causal family, multilingual and workflow adaptation, and a common empirical account of accuracy, transfer, and probability quality. Calibration is a separate concern from accuracy: temperature scaling can improve probability calibration without changing a prediction's class [10].

7 Future Directions

The architecture suggests several ways to reduce the cost of multi-question requests. These are research directions, not measured speedups in the present release.

Reuse state computation and improve batching. An identical causal prefix could be computed once, then branched for each question. Qwen3.5 requires reuse of both attention key/value caches and Gated DeltaNet recurrent and convolutional states, with identical prefix tokens and positions. Length-aware batching, device-side candidate gathering, and fewer host synchronizations could reduce additional overhead. Work such as FlashInfer [25] motivates attention scheduling for variable workloads, but the gains require validation on this hybrid backbone and the target serving stack. Evaluation should vary state length, question count, candidate count, and batch size, and report both throughput and tail latency.

Compress the model while preserving decisions. Quantization methods such as SmoothQuant [26] motivate testing lower-precision backbone execution while retaining sensitive readout operations at higher precision. Distillation could transfer a larger model's candidate distribution to a smaller decision model. Decoupled knowledge distillation [27] offers a way to separate target confidence from relations among alternatives; for binary questions, the non-target term vanishes, leaving binary distribution matching. Accuracy, calibration, and candidate-order robustness should constrain compression, not just output agreement.

Learn a reusable state representation. Perceiver IO and in-context autoencoding [28, 29] suggest encoding a long state into a compact representation that multiple question-specific readers can reuse. This would change the present architecture and require training that preserves evidence needed by future questions. The main risk is an information bottleneck: a state summary useful for one decision may erase details required by another.

Allocate computation to difficult questions. Routing and cascading can select model size or escalate uncertain cases under an explicit cost–quality objective [30]. Within-model early exits are another possibility, but would require trained intermediate heads and validation of exit confidence. For grouped requests, policies should consider both per-question quality and the slowest branch's contribution to response latency. Fresh task-held-out evaluation is needed to establish whether these policies generalize beyond the observed development suite.

8 Limitations and Availability

Decision 1.0 is text-only and evaluates supplied evidence without live retrieval. Quality varies across tasks and languages; the general evaluation covers English and Chinese rather than uniform multilingual competence. Input budgets, omitted candidates, and evidence quality constrain every model. Training and evaluation reuse public task families, upstream pretraining exposure is unknown, and differences in recipes and native input handling limit causal comparisons.

We release weights, inference implementations, configurations, and supporting evaluation documentation [2]. Encoder releases also include continued fine-tuning code. The bundles do not redistribute all assembled training corpora, construction scripts, or final optimization settings, so they support inspection and adaptation more fully than exact reconstruction of every training stage. Project artifacts use Apache 2.0 with retained upstream notices; source datasets retain their own terms.

9 Conclusion

Decision 1.0 maps runtime-defined alternatives directly to typed probabilities through bidirectional or causal models. The family improves explicit decision performance and supports workflow adaptation, while composition, transfer, order sensitivity, and uncertainty remain open problems. Released weights and inference code provide a basis for further research.

Acknowledgments

We thank the authors and maintainers of the pretrained models, datasets, and open decision implementations on which this work builds.

References

[1] Diogo Almeida. Introducing System One Models & Jev. TypeSafe AI, September 2026. https://typesafe.ai/blog/introducing-system-one-models-and-jev.

[2] vLLM Semantic Router Team. Decision 1.0: model weights, implementations, and evaluation documentation. 2026. https://huggingface.co/collections/llm-semantic-router/decision-10-6ab12177bd0002394d8409f9.

[3] Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn Lawrie, and Benjamin Van Durme. mmBERT: A Modern Multilingual Encoder with Annealed Language Learning. arXiv:2509.06888, 2025.

[4] Benjamin Warner et al. Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference. arXiv:2412.13663, 2024.

[5] Qwen Team. Qwen3.5 model releases. 2026. https://huggingface.co/Qwen/Qwen3.5-9B. Companion 0.8B, 2B, and 4B checkpoints initialize Eos-0.8B, Sol-2B, and Nox-4B.

[6] LocalLLaMA. Typed Decisions. Dataset release, 2026. https://huggingface.co/datasets/LocalLLaMA/typed-decisions.

[7] Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Cosmos QA: Machine Reading Comprehension with Contextual Commonsense Reasoning. EMNLP-IJCNLP, 2019. https://aclanthology.org/D19-1243/.

[8] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know What You Don't Know: Unanswerable Questions for SQuAD. ACL, 2018. https://aclanthology.org/P18-2124/.

[9] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A Large Annotated Corpus for Learning Natural Language Inference. EMNLP, 2015. https://aclanthology.org/D15-1075/.

[10] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. ICML, 2017. https://proceedings.mlr.press/v70/guo17a.html.

[11] ConvAI Innovations. Laya: model and implementation release. 2026. https://huggingface.co/convaiinnovations/laya.

[12] Mapika. Decider: System One-style decision models. 2026. https://github.com/Mapika/decider.

[13] Jared Palmer. Kev: decision models, training code, and evaluation suites. 2026. https://github.com/jaredpalmer/kev.

[14] Wenpeng Yin, Jamaal Hay, and Dan Roth. Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach. EMNLP-IJCNLP, 2019. arXiv:1909.00161.

[15] vLLM Semantic Router Team. Vela-1.0-Encoder-307M. Model release, 2026. https://huggingface.co/llm-semantic-router/Vela-1.0-Encoder-307M.

[16] Christopher Clark et al. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. NAACL, 2019. arXiv:1905.10044.

[17] Jonathan H. Clark et al. TyDi QA: A Benchmark for Information-Seeking Question Answering in Typologically Diverse Languages. TACL, 2020. arXiv:2003.05002.

[18] Lucas Bandarkar et al. The Belebele Benchmark: a Parallel Reading Comprehension Dataset in 122 Language Variants. arXiv:2308.16884, 2023.

[19] Zhilin Wang et al. HelpSteer2: Open-source Dataset for Training Top-performing Reward Models. arXiv:2406.08673, 2024.

[20] Andreas Köpf et al. OpenAssistant Conversations—Democratizing Large Language Model Alignment. arXiv:2304.07327, 2023.

[21] Jack FitzGerald et al. MASSIVE: A 1M-Example Multilingual Natural Language Understanding Dataset with 51 Typologically-Diverse Languages. arXiv:2204.08582, 2022.

[22] Adina Williams, Nikita Nangia, and Samuel R. Bowman. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. NAACL, 2018. https://aclanthology.org/N18-1101/.

[23] Iñigo Casanueva et al. Efficient Intent Detection with Dual Sentence Encoders. 2020. https://github.com/PolyAI-LDN/task-specific-datasets.

[24] Stefan Larson et al. An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction. EMNLP-IJCNLP, 2019. https://github.com/clinc/oos-eval.

[25] Zihao Ye et al. FlashInfer: Efficient and Customizable Attention Engine for LLM Inference Serving. MLSys, 2025. https://proceedings.mlsys.org/paper_files/paper/2025/hash/dbf02b21d77409a2db30e56866a8ab3a-Abstract-Conference.html.

[26] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models. ICML, 2023. https://proceedings.mlr.press/v202/xiao23c.html.

[27] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled Knowledge Distillation. CVPR, 2022. https://openaccess.thecvf.com/content/CVPR2022/html/Zhao_Decoupled_Knowledge_Distillation_CVPR_2022_paper.html.

[28] Andrew Jaegle et al. Perceiver IO: A General Architecture for Structured Inputs & Outputs. ICLR, 2022. https://openreview.net/forum?id=fILj7WpI-g.

[29] Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context Autoencoder for Context Compression in a Large Language Model. ICLR, 2024. arXiv:2307.06945.

[30] Jasper Dekoninck, Maximilian Baader, and Martin Vechev. A Unified Approach to Routing and Cascading for LLMs. ICML, 2025. https://proceedings.mlr.press/v267/dekoninck25a.html.

A Released Artifacts

The results describe fixed public model snapshots. The following links resolve to the exact revisions used in this report: Kai-0.6B, Lex-0.6B, Eos-0.8B, Sol-2B, Nox-4B, and Lux-9B. The linked repositories provide inference code, configurations, and source attributions. Panel scores can be checked against `metrics/benchmark.json` in the linked Lux release.