AutoIntent: AutoML for Text Classification

Ilya Alekseev[1,2,4] Roman Solomatin[1,3] Darina Rustamova[3] Denis Kuznetsov[1]

1 DeepPavlov.ai 2 MSU 3 ITMO 4 DRESSCODE

30th ANNIVERSARY EMNLP 2025 Suzhou, China 中国苏州 November 4-9 11月4日-9日

Key Findings

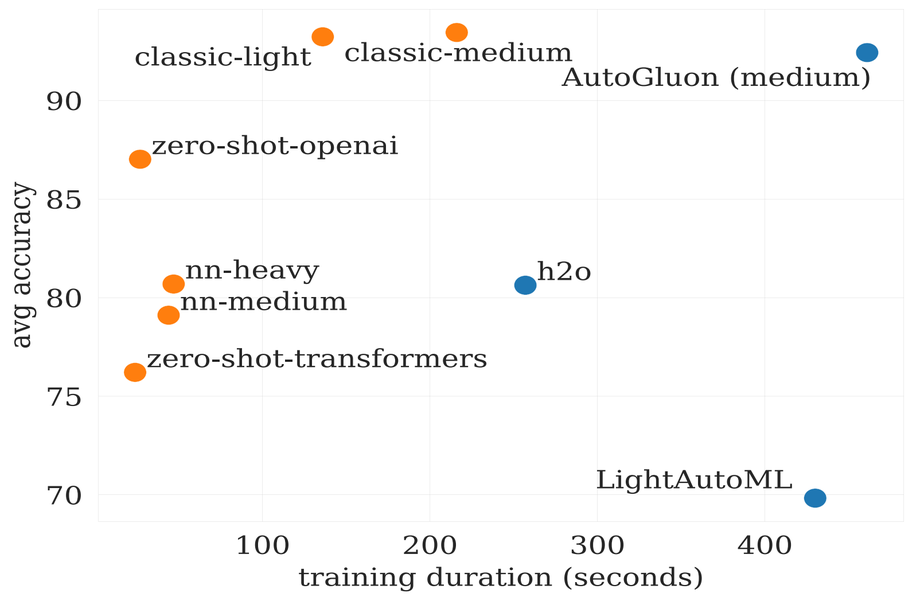
AutoIntent offers end-to-end automation with

- embedding model selection
- decision threshold tuning
- multi-label classification
- out-of-scope detection

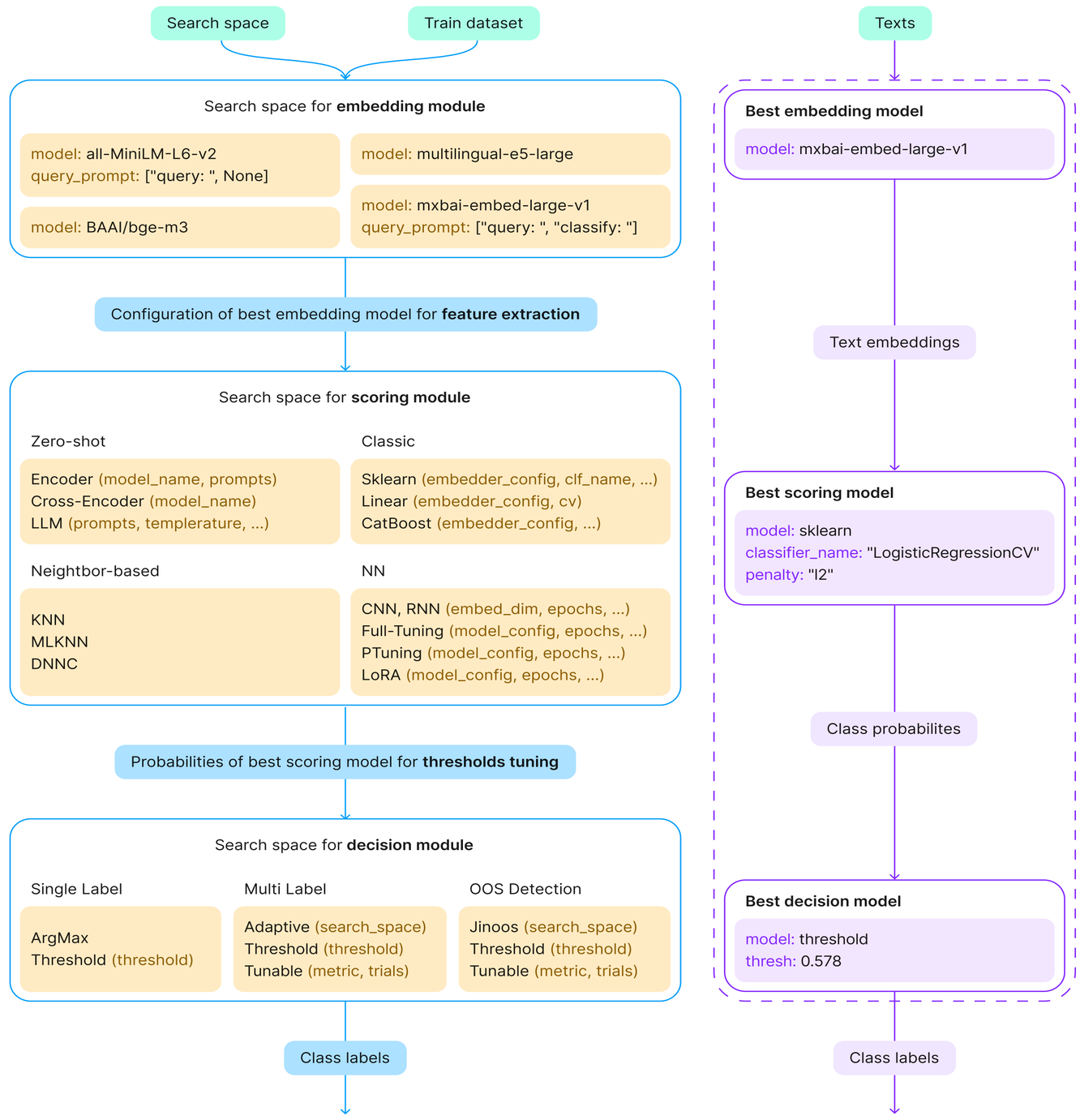
all within a modular design, sklearn-like interface and customizable presets.

```
from autointent import Pipeline, Dataset

dataset = Dataset.from_json(path_to_json)
pipeline = Pipeline.from_preset("classic-light")
pipeline.fit(dataset)
pipeline.predict(["show my transactions"])
```



Architecture



Performance of Automatic Embedder Selection

Model	Accuracy	NDCG
stella_en_400M_v5	94.28	93.83
multilingual-e5-[1]	93.65	92.97
GIST-large-[2]	93.51	93.32
UAE-Large-V1	92.89	93.25
bge-m3	92.69	92.49
multilingual-e5-large	91.41	92.45
LaBSE	90.47	89.51
KaLM-embedding-[3]	89.65	92.88
nomic-embed-[4]	87.24	89.63
deberta-v3-small	81.15	67.20
deberta-v3-large	75.39	59.59
deberta-v3-base	75.00	59.28

Table 5: Embedding models performance averaged over hwu64 (Liu et al., 2019), massive (FitzGerald et al., 2022), minds14 (Gerz et al., 2021), snips (Coucke et al., 2018). [1]large-unstruct, [2]Embedding-v0, [3]multilingual-mini-instruct-v1.5, [4]text-v1.5

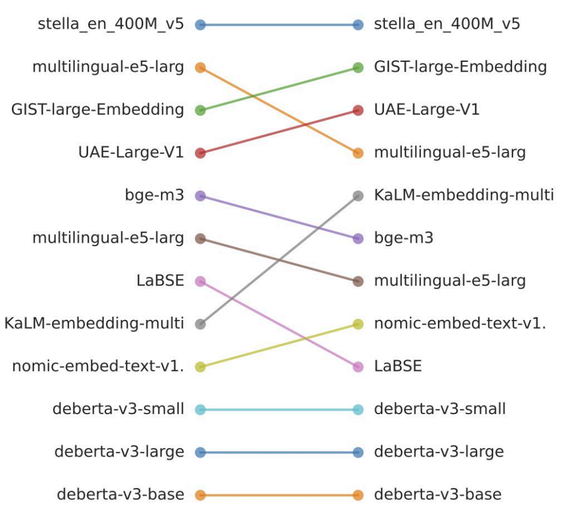


Figure 4: Encoders ranking: (**Left**) precise ranking obtained via training full AutoML pipeline with only this model, (**Right**) approximate ranking based on retrieval quality (NDCG).

We calculate retrieval quality on the fly (e.g. NDCG or hit rate) and use it as a proxy-metric for embedder performance as though it is used in a pipeline. It gives a close approximation to the actual embedders performance without the need to refit the whole pipeline

Out-of-Scope (OOS)

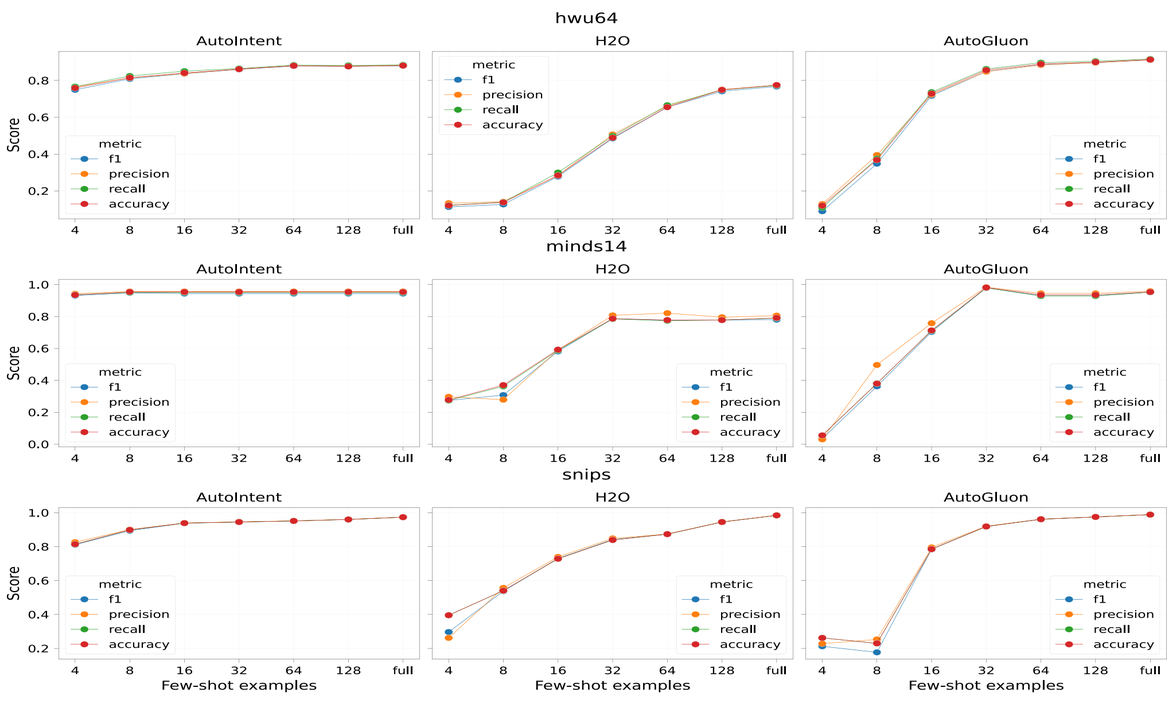
framework	in domain accuracy	out-of-scope F1-measure
AutoIntent	**96.13**	**76.79**
AutoGluon (Tang et al., 2024)	95.76	48.53
H2O (LeDell and Poirier, 2020)	85.22	40.69

Table 3: Performance comparison on out-of-scope detection task on CLINC150 (Larson et al., 2019).

preset	duration	banking77	hwu64	massive	minds14	snips	avg
Baselines							
AutoGluon (best)	–	6.98	12.64	21.39	85.19	96.00	44.44
AutoGluon (high)	–	**92.60**	90.80	**89.22**	95.37	98.86	93.37
AutoGluon (medium)	461	92.40	**91.17**	87.13	92.59	98.86	92.43
LightAutoML	430	53.31	77.85	47.41	72.22	98.38	69.83
h2o	257	75.32	77.32	75.30	76.85	98.36	80.63
AutoIntent Presets							
zero-shot-transformers	**24**	69.51	71.47	63.58	87.04	89.43	76.21
nn-medium	44	79.95	70.79	72.75	75.31	96.74	79.11
nn-heavy	47	78.84	72.96	73.39	80.86	97.40	80.69
zero-shot-openai	27	76.43	85.04	80.49	96.30	96.86	87.02
classic-light	136	92.23	90.83	87.11	97.53	98.43	93.23
classic-medium	216	92.34	90.92	87.19	**97.84**	**98.98**	**93.45**

Table 2: Performance comparison across different presets averaged from three runs (except H2O and AutoGluon which were launched once). **Column 1**: Baseline AutoML frameworks: AutoGluon (Tang et al., 2024) with non-HPO presets `best_quality`, `high_quality`, `medium_quality`, H2O (LeDell and Poirier, 2020) with their word2vec, LightAutoML (Vakhrushev et al., 2022); and AutoIntent presets: `nn` (CNN (Kim, 2014), RNN), `zero-shot` (description-based bi- and cross-encoder, LLM prompting), `classic` (knn, logreg, random forest, catboost (Prokhorenkova et al., 2018)). **Column 2**: Duration in seconds evaluated on minds14 (Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz, single Tesla P100-SXM2-16GB). **Columns 3–7**: Accuracy on test sets.

Few Shot



Superior robustness to scarce data scenario thanks to classic ML models

Future work

- MCP
- integration with ReAct agents
- convenient deployment and Inference
- better augmentations
- any of your ideas!

Documentation Code Paper