DeepPavlov Strikes Back: A Toolkit for Improving LLM Reliability and Trustworthiness

Evgeny Nikolaev[5,4*] , Timur Ionov[3,4*] , Anna Korzanova[1],
Vasily Konovalov[2,1], Maksim Savkin[2,1]

[1]MIRAI, [2]AXXX, [3]MWS AI, [4]ITMO University, [5]AI Talent Hub
Correspondence: savkin.max.k@gmail.com

Abstract

DeepPavlov 1.1 introduces new multilingual tools to enhance LLM reliability in production pipelines. It includes a span-level hallucination detector, an evergreen question classifier, and a toxicity classifier, all integrated into an easy-to-use open-source framework. These components address key LLM challenges: detecting factual inconsistencies against retrieved context, identifying static factual questions that bypass unnecessary retrieval, and flagging harmful content when alignment fails. Trained on PsiloQA, EverGreenQA, and TextDetox across 14+ languages, our encoder-based models outperform LLM baselines in accuracy and speed by orders of magnitude. Released under Apache 2.0 DeepPavlov 1.1 bridges traditional NLP and LLM-centric workflows for safer AI systems.

1 Introduction

Natural Language Processing (NLP) is a key component of many modern AI systems. It enables the automation of tasks that would otherwise require extensive manual labor, particularly those involving the processing of large volumes of raw text. As real-world applications of NLP continue to grow in complexity and scale, the demand for robust and easy-to-integrate tools grows as well.

At the core of our work is a long-term commitment to practical, user-focused development. As the field evolves, our tools continuously update in response to technological advances and shifting user needs.

The DeepPavlov library (Burtsev et al., 2018) was first introduced in the pre-BERT era, at a time when NLP systems were largely modular and relied heavily on explicit linguistic features. Early versions of the library focused on foundational components such as Part-of-Speech (POS) tagging and syntactic parsing. These tools acted as building blocks that extracted structured linguistic information from raw text and played a critical role in training downstream models that depended on hand-crafted features to understand language.

The release of DeepPavlov 1.0 (Savkin et al., 2024) marked our transition into the post-BERT era, reflecting a shift in paradigm towards pre-trained transformer-based models. These models moved beyond low-level syntactic analysis by using high-level language understanding capabilities of the BERT-family models. We introduced models for Named Entity Recognition (NER) (Chizhikova et al., 2023), Knowledge Base Question Answering (KBQA), Multi-task learning (MTL) (Karpov and Konovalov, 2023), Text classification tasks (intent, sentiment) (Savkin and Konovalov, 2024), and tasks from the SuperGLUE (Wang et al., 2019) benchmark (NLI, RTE, Paraphrasing).

The emergence of large language models (LLMs) marked yet another paradigm shift in NLP. Instead of training dedicated models for each task, LLMs demonstrated impressive few-shot and zero-shot capabilities across a wide range of tasks. To enhance their reasoning and task-solving abilities, they are often paired with auxiliary tools, such as information retrieval systems, code execution environments, or external APIs. However, these LLM-centered workflows introduce new challenges: both the tools and the outputs of the models need to be monitored and validated to avoid irrelevant, harmful, or fabricated content. In this new context, the role of the DeepPavlov library evolved once again – now focusing on supporting LLM-centric pipelines. A major concern in such systems is hallucination: the generation of plausible but untrue information (Rykov et al., 2025a). Although it remains an open problem, it is increasingly important to have mechanisms for de-

[*]Equal contribution. This work was done while the first two authors were affiliated with MIRAI.

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 3: System Demonstrations, pages 21–32
March 24-29, 2026 ©2026 Association for Computational Linguistics

Tool / Framework	DeepPavlov 1.1	DeepPavlov 1.0	spaCy	Stanza	Flair	AllenNLP*	jiant*
Linguistic Features							
Embeddings	✓	✓	✓	✓	✓	✓	✗
POS Tagger	✓	✓	✓	✓	✓	✗	✗
Lemmatizer	✓	✓	✓	✓	✗	✗	✗
Dependency Parsing	✓	✓	✓	✓	✗	✗	✗
Morphotagger	✓	✓	✓	✗	✗	✗	✗
Syntax Parser	✓	✓	✓	✗	✗	✗	✗
Pretrained Encoders							
NER	✓	✓	✓	✓	✓	✓	✓
Sentiment Classification	✓	✓	✓	✓	✓	✗	✗
Entity Linking	✓	✓	✓	✗	✓	✗	✗
Intent Classification	✓	✓	✗	✗	✗	✗	✗
Context QA	✓	✓	✗	✗	✗	✓	✓
SuperGLUE Tasks	✓	✓	✗	✗	✗	✓	✓
LLM-centric Tools							
Hallucination Detection	✓	✗	✗	✗	✗	✗	✗
Evergreen QA Classification	✓	✗	✗	✗	✗	✗	✗
Toxicity Classification	✓	✗	✗	✗	✗	✗	✗

Table 1: Comparison of supported instruments across popular NLP frameworks. Frameworks marked with "*" are no longer supported.

tecting and flagging unreliable outputs, especially for high-stakes applications.

To address this need, we introduce DeepPavlov 1.1, an updated open-source NLP framework aimed at improving the reliability of LLM-based pipelines. This release introduces the following new multilingual components.

1. **Contextual Hallucination Detector** designed to estimate faithfulness – the factual consistency of model responses with retrieved content (Krayko et al., 2025).
2. **Evergreen Question Classifier** detects evergreen questions (factual questions whose correct answers are highly unlikely to change over extended periods of time). Usually evergreen question doesn't require RAG pipeline (Pletenev et al., 2025).
3. **Toxicity Detection** determines whether a given text contains toxic content, serving as a safeguard when a language model's alignment mechanisms fail to prevent harmful outputs (Dementieva et al., 2025).

2 Related Work

Before discussing directly comparable NLP frameworks, it is important to distinguish between related categories of tools that fall outside the scope of this work.

LLM orchestration frameworks such as **LangChain**[1] have gained popularity as platforms for building agent-based pipelines. However, they are better characterized as workflow managers: they focus on task chaining and integration rather than providing pre-trained NLP models for assessing content quality. Therefore, they are not considered further in this paper.

Similarly, low-level libraries such as **PyTorch** (Paszke et al., 2019) and **TensorFlow** (Abadi et al., 2016) lack ready-to-use, task-specific NLP models and supporting infrastructure, so again they have a different purpose than the higher-level frameworks discussed here.

While early NLP libraries focused on linguistic features and task-specific modeling, modern applications increasingly rely on LLMs augmented by tools for retrieval and content validation. Despite this shift, most NLP frameworks have not adapted to the demands of LLM-centric workflows. Table 1 provides a detailed comparison of tools supported by major open-source NLP frameworks.

Libraries such as **spaCy** (Honnibal, 2017), **Stanza** (Qi et al., 2020), and **Flair** (Akbik et al., 2019) offer robust components for traditional tasks like POS tagging, syntactic parsing, and named entity recognition, often leveraging pretrained transformer models. While effective in classical NLP pipelines, they do not address new challenges such as detecting hallucinations, minimizing un-

[1] https://langchain.com

necessary retrieval in RAG pipelines, or flagging unsafe content generated by LLMs.

In contrast, **DeepPavlov 1.1** is explicitly designed for the current LLM era. Maintains full support for traditional NLP tasks and pre-trained encoder-based models, while also introducing new tools aimed at improving the reliability and controllability of LLM outputs.

3 Design and Usage

DeepPavlov models are built and managed via modular configuration files that define all the components required for training, inference, and deployment. Each configuration file includes the following sections: (1) **dataset_reader/iterator** is responsible for loading data from file; (2) **chainer** is the core abstraction in DeepPavlov, used to construct processing pipelines from heterogeneous components (rule-based, ML, DL); (3) **train** specifies training hyperparameters; (4) **metadata** stores auxiliary variables referenced by other sections.

DeepPavlov emphasizes flexibility and ease of customization. Users can easily adjust hyperparameters, modify preprocessing steps, or swap models within the `chainer` block without breaking the input/output interface.

The framework uses PyTorch as its underlying ML engine, with support for multi-GPU training. DeepPavlov integrates HuggingFace's `transformers` library, enabling direct use of any `AutoModel`-compatible pretrained model from the HuggingFace Hub.

Models can be used and managed via multiple interfaces: Command-Line Interface (CLI), REST API, or Python. Installation is straightforward via `pip install deeppavlov`, and CLI usage examples, code, and documentation are available on the GitHub[2].

4 Reference Models

This section introduces the new models included in the latest release of the DeepPavlov library. All models were trained and evaluated using the library configuration files and are publicly accessible through our demo platform[3]. All training hyperparameters are detailed in the Appendix B.

[2] https://github.com/deeppavlov/DeepPavlov

[3] https://demo.deeppavlov.ai

4.1 Hallucination Detection

Task Formulation. We formulate hallucination detection as a span level sequence labeling task: given a question, one or more supporting passages, and a generated answer, the model predicts for each answer token whether it is grounded in the provided context or constitutes a hallucination. This span level formulation allows us to capture fine grained hallucinations within otherwise correct answers and supports direct interventions, such as masking or rewriting only the hallucinated fragments. Figure 1 illustrates an example where a hallucinated entity is identified within an otherwise plausible answer.

PsiloQA Sample

QUESTION: *For which company did Albert Herman mainly work as a director?*

PASSAGE: *Albert Herman (1887–1958) was an American film director who worked mainly for Producers Releasing Corporation.*

RESPONSE: *Albert Herman mainly worked as a director for Albert Herman & Cie.*

Figure 1: Example from PsiloQA with span level hallucination annotation. The correct entity from the passage is highlighted in green, while the hallucinated company name in the model answer is marked in red.

Datasets and Metric. We evaluate our hallucination detector on the PsiloQA benchmark, a multilingual span-level hallucination detection suite covering 14 languages. Each example consists of a question, one or more supporting passages, and an LLM-generated answer, with character-level annotations marking hallucinated spans in the answer text. We follow the original PsiloQA setup and use its predefined train, development, and test splits, without any additional filtering or relabeling.

For evaluation, we adopt the character-level Intersection over Union (IoU) metric proposed in PsiloQA. Given the predicted hallucination mask and the gold hallucination mask over answer characters, IoU is defined as the size of their intersection divided by the size of their union, computed per example and then macro-averaged over all instances in a language.

Model	Params	Mode	ar	ca	cs	de	en	es	eu	fa	fi	fr	hi	it	sv	zh	Avg.
Open-source language-models																	
Qwen2.5-7B-Instruct	7B	3-shot	33.18	39.13	29.78	28.13	37.68	36.38	36.86	28.43	58.70	29.64	53.22	39.25	35.91	31.25	36.97
Qwen2.5-32B-Instruct	32B	3-shot	30.06	50.60	44.67	25.11	59.01	58.99	38.15	38.52	60.97	43.71	52.71	61.44	37.81	53.00	46.77
Qwen2.5-72B-Instruct	72B	3-shot	45.16	56.12	<u>48.17</u>	32.12	49.82	56.02	42.59	49.35	58.96	49.06	50.37	40.60	45.01	54.02	48.38
gpt-oss-120B	120B	3-shot	38.92	46.56	40.44	27.13	58.75	48.84	39.78	25.25	55.64	38.70	47.16	36.87	43.34	44.72	42.29
Proprietary language-models																	
FActScore (GPT-4o)	—	—	20.75	28.99	10.44	26.68	25.84	28.54	19.68	26.62	28.16	10.21	21.03	43.92	19.25	25.18	23.95
Transformer encoder models																	
lettuce-detect-base	395M	SFT	37.81	44.37	30.08	30.31	43.28	40.08	33.35	32.45	56.44	35.60	16.95	34.97	49.11	35.94	37.20
ModernBERT-base	395M	SFT	<u>55.27</u>	<u>65.70</u>	44.73	<u>46.27</u>	<u>68.23</u>	<u>61.69</u>	**50.43**	<u>68.63</u>	<u>64.68</u>	<u>53.90</u>	54.15	<u>62.75</u>	**67.09**	<u>56.95</u>	<u>58.61</u>
mmBERT-base (our)	110M	SFT	**58.10**	**67.01**	**48.81**	**54.97**	**70.67**	**66.18**	<u>50.27</u>	**76.61**	**68.16**	**56.38**	**61.19**	**66.57**	<u>66.24</u>	**61.58**	**62.34**

Table 2: Character level Intersection over Union (IoU, in %) of span level hallucination detection methods on the PsiloQA test set across 14 languages. Encoder models are supervised fine tuned on the full PsiloQA train split. Language model results are averaged over 5 independent runs; see Table 10 for variance. The rightmost column reports macro averaged IoU across all languages.

Baselines. We reuse the PsiloQA evaluation suite and focus on two types of baselines. Encoder-based detectors fine-tuned on PsiloQA: the English only `lettuce-detect-base` model built on ModernBERT (Kovács and Recski, 2025), `ModernBERT-base`[4] trained on PsiloQA, and `mmBERT-base`[5], a multilingual ModernBERT with 307M parameters covering all 14 languages (Marone et al., 2025; Rykov et al., 2025b). Since PsiloQA and the present work share authorship, we directly reuse the mmBERT checkpoint released by Rykov et al. (2025b) instead of retraining it from scratch. All encoder models take the concatenation of passage, question, and answer and output token level hallucination scores in a single forward pass. LLM-based detectors, treat large generative models as span level judges. We consider *FActScore* with GPT-4o (Min et al., 2023), which decomposes answers into atomic claims and verifies them against retrieved context, the original 3-shot `Qwen2.5-32B-Instruct` baseline, and three additional judges evaluated in this work: `gpt-oss-120b`, `Qwen2.5-(72B|7B)-Instruct`. All LLMs are prompted to insert `[HAL]` tags around hallucinated spans with default sampling parameters; the prompt template is provided in Figure 4 in Appendix D.Qwen models were evaluated at a temperature of 0.3; gpt-oss-120B at 1.0. LLM results (except Qwen2.5-72B, single run) are averaged over 5 independent runs.

Experimental Setup and Results. ModernBERT is fine-tuned on the multilingual PsiloQA train split with a token level cross entropy loss over answer tokens. The lettuce-detect and mmBERT checkpoints are taken directly from Rykov et al. (2025b) and evaluated without modification. LLM baselines are used in a purely prompted regime with 3-shot examples drawn from the PsiloQA training data.

Table 2 reports character level IoU per language. Encoder-based detectors clearly outperform LLM judges: mmBERT-base achieves the best overall performance with 70.7 IoU on English and 62.3 macro-averaged IoU across 14 languages, consistently improving over ModernBERT. FActScore with GPT-4o attains much lower IoU, while `Qwen2.5-32B-Instruct` and our additional `gpt-oss-120B` and `Qwen2.5-72B-Instruct` judges close part of the gap but remain below the encoder models, especially in low resource languages. The smallest model, `Qwen2.5-7B-Instruct`, performs competitively only on a subset of languages and reaches 37.68 IoU on English and 36.97 IoU on average. Overall, encoder-based detectors trained on PsiloQA, and mmBERT in particular, provide the most accurate and efficient span level hallucination detection for our multilingual setting.

4.2 Evergreen Questions Classification

Task Formulation. Evergreen Question Classification is the task of identifying factual questions whose correct answers are highly unlikely to change over extended periods of time. We formulate this as a binary classification problem: Given a question, predict whether it is evergreen or non-evergreen (see examples in Table 3).

By serving as a real-time preprocessing filter, we can determine whether to rely solely on the internal knowledge of the LLM (in the case of ev-

[4] `https://hf.co/answerdotai/ModernBERT-base`
[5] `https://hf.co/jhu-clsp/mmBERT-base`

Evergreen Questions	Non-Evergreen Questions
Who painted the 'Mona Lisa'?	*What time is it?*
Which is lighter: a kilogram of feathers or a kilogram of iron?	*Who won the last football World Cup?*
What is the ultimate question of life, the universe, and everything?	*The last time a bright comet was visible to the naked eye?*

Table 3: Examples of Evergreen and Non-Evergreen questions. Among the non-evergreen examples, a darker red background highlights answers that change more rapidly, and a lighter red background indicates answers that change relatively slow.

Model	Params	Mode	Russian	English	French	German	Hebrew	Arabic	Chinese	Average
Open-source language-models										
Qwen2.5-7B-Instruct	7B	10-shot	78.2	78.9	78.6	79.4	69.2	71.1	77.4	76.1
Qwen2.5-32B-Instruct	32B	10-shot	88.2	88.5	87.5	88.3	86.2	86.2	87.2	87.4
Qwen 2.5-72B-Instruct	72B	10-shot	80.6	81.5	80.2	80.5	78.1	75.8	76.8	79.1
gpt-oss-120b	120B	10-shot	**92.5**	**95.2**	**94.3**	**92.4**	**93.1**	**93.6**	**92.1**	**93.3**
Proprietary language-models										
GPT-4.1	–	10-shot	80.6	79.4	81.6	81.3	80.3	81.1	80.9	80.7
Transformer encoder models										
EG-BERT-base	110M	SFT	89.3	90.0	88.9	88.4	88.9	88.3	90.2	89.1
BERT-base (our)	110M	SFT	87.4	88.1	87.2	86.5	85.5	87.3	87.2	87.0
ModernBERT-base (our)	395M	SFT	75.9	88.0	82.8	81.3	78.4	77.5	85.7	81.4
EG-E5-large	560M	SFT	91.0	91.3	90.9	91.0	90.4	90.0	89.7	90.6

Table 4: EvergreenQA classifier F1-weighted scores comparison across different languages.

ergreen questions) or to apply the RAG pipeline. Relying on the RAG pipeline for every query is not always the best solution because: (1) retrieval in RAG adds additional latency; (2) noisy context returned by retrieval can deteriorate the quality of generation (Fang et al., 2024).

Therefore, it is best practice to include an adaptive component that decides whether the LLM alone can answer the question or whether the RAG component should be used (Moskvoretskii et al., 2025).

Datasets. To train and evaluate our Evergreen classifier we leverage an **EverGreenQA** (Pletenev et al., 2025) dataset comprising of 4,757 examples and covering seven languages. We evaluated our models in both the EverGreenQA test set and the multilingual version of the FreshQA data set (Vu et al., 2024), which had been translated into all target languages in Pletenev et al. (2025).

Baselines. To contextualize our results, we compare them with zero-shot LLM baselines of various sizes, as well as with BERT-family models fine-tuned on the binary classification task from EverGreenQA (Pletenev et al., 2025). The prompt used for LLM-based classification is provided in Figure 5 in Appendix D. All LLMs were evaluated at a temperature of 0.0

Experimental Setup and Results. As a primary evaluation metric, we follow EverGreenQA and report the weighted F1 score, using the same train/test split.

Our experimental results presented in Table 4 show that our models do not outperform larger baselines on the EverGreenQA test set, multilingual BERT-base achieves modest improvements over the original BERT-base on FreshQA, particularly for some languages. This indicates competitive performance in specific cases, although the gap compared to `EG-E5-large` highlights the challenges of generalizing temporal sensitivity classification to unseen multilingual data. ModernBERT's results are mixed: overall weaker on FreshQA, but with strong performance for certain languages, matching `EG-E5-large` for Russian and ranking second for French. This suggests limited cross-lingual generalization but potential in certain language-specific contexts. Our retrained multilingual BERT-base demonstrates consistent improvements over the original, offering a practical lightweight alternative.

4.3 Toxicity Detection

Task Formulation. We frame the task of toxicity classification as a binary classification problem. The goal is to determine whether a given text contains toxic content, such as vulgar, obscene, or profane language. This component serves as

Model	Params	Mode	EN	RU	UK	DE	ES	AR	AM	HI	ZH	IT	FR	HI-EN	HE	JA	TT	Avg.
Open-source language-models																		
Qwen2.5-7B-Instruct	7B	3-shot	93.4	93.6	83.4	77.1	86.4	76.4	62.9	75.2	66.8	74.7	95.4	67.2	67.2	65.3	72.2	77.7
Qwen2.5-32B-Instruct	32B	3-shot	**97.5**	95.7	84.4	75.3	86.6	78.6	53.7	76.0	71.9	75.9	**97.4**	73.7	73.3	75.0	74.1	79.7
Qwen2.5-72B-Instruct	72B	3-shot	95.8	93.4	87.0	80.8	**90.9**	79.1	64.6	85.0	76.8	72.0	94.8	**78.4**	71.1	77.9	75.9	81.6
gpt-oss-120B	120B	3-shot	96.4	96.0	86.0	80.7	88.9	**81.9**	56.3	73.2	67.1	71.5	96.2	69.2	75.0	66.1	82.5	80.1
Transformer encoder models																		
TextDetox-2024-RoBERTa-large	355M	SFT	96.5	**97.9**	92.5	87.6	87.0	77.8	**77.8**	93.6	73.2	–	–	–	–	–	–	**87.1**
TextDetox-2025-RoBERTa-large	355M	SFT	92.3	95.3	**96.0**	73.3	71.3	66.3	55.8	**97.3**	**91.8**	58.6	92.4	61.0	**87.8**	**87.7**	57.4	78.9
TextDetox-BERT-base	110M	SFT	90.4	92.2	94.6	51.8	72.9	51.4	63.2	72.7	67.0	64.9	91.3	68.5	86.9	86.4	61.7	74.4
BERT-base (our)	110M	SFT	97.0	91.0	87.9	**87.8**	81.6	67.8	58.6	89.2	60.1	77.8	95.2	71.9	67.5	73.9	**87.4**	81.6

Table 5: Toxicity classification F1 scores across different languages. Language model results are averaged over 5 independent runs; see Table 11 for variance.

a safeguard when LLM's alignment mechanisms fail to prevent harmful outputs (Moskovskiy et al., 2024).

Dataset. We train and evaluate our model on the multilingual `TextDetox` (Dementieva et al., 2024) dataset, which includes $5,000$ examples for each of 15 languages.

Baselines. We compare our model against several encoder-based baselines, including `BERT-base` and `XLM-RoBERTa-large`, using checkpoints from the official TextDetox hub[6]. The prompt used for LLM-based classification is provided in Figure 3 in Appendix D. All LLMs were evaluated at a temperature of 0.0

Experimental Setup and Results. The `BERT-base` model is trained as a binary example-level classifier. Since the original dataset split is unavailable, we divide the TextDetox dataset into 70% training, 10% validation, and 20% test sets, preserving the language distribution.

Table 5 shows that our multilingual `BERT-base` model ranks second overall, although it is significantly smaller than `XLM-RoBERTa-large` model. It achieves the highest F1-score in English (97.00) and outperforms baselines in several other languages, demonstrating its strong cross-lingual generalization.

5 Perfomance

Experimental Setup. All experiments were conducted on single Nvidia H200 GPU with 140 Gb VRAM. The vLLM framework was used for LLM inference, while the DeepPavlov and Transformers libraries were used to run encoder models. Encoder models were executed with a batch size of 256 to maximize throughput.

Performance Analysis. Figure 2 and Table 6 provide a detailed comparison of model performance. Our models consistently achieve accuracy near state-of-the-art levels while exhibiting inference speeds that are two orders of magnitude faster compared to LLMs. The results reveal a consistent pattern: larger models yield higher accuracy but require more inference time.

6 Conclusion and Future Work

DeepPavlov 1.1 is an updated tool that makes LLMs more reliable and safe. In the future, we plan to continue improving DeepPavlov by adding more features and making it even easier for researchers and developers to build safe and trustworthy AI systems.

Limitations

Framework-Level Limitations. DeepPavlov 1.1 does not yet offer native integration with modern LLM orchestration pipelines, which limits its out-of-the-box applicability in production workflows. Although the framework offers API and Python integration, we did not conduct rigorous latency, throughput, or scalability testing. Users should verify performance under production constraints such as inference time or memory footprint. The framework depends on external libraries such as PyTorch and HuggingFace Transformers. API changes, deprecations, or version incompatibilities could break core functionality or degrade performance.

Experimental Setup. Our evaluation protocol does not include multiple training runs with different random seeds. Similarly, LLMs are only evaluated once per experiment. We also performed only minimal exploratory data analysis (EDA) and

[6] `https://hf.co/textdetox`

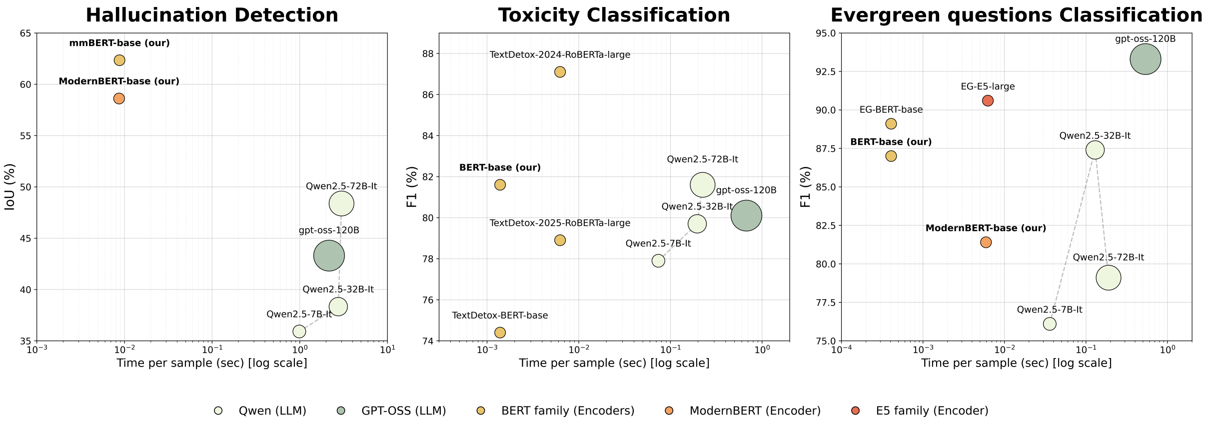

Figure 2: Trade-off plot illustrating how different models perform on three tasks supported in the updated DP library. Each subplot shows the relationship between *inference latency* and *task performance*.

ablation studies. This limits our understanding of how design choices affect model behavior.

Hallucination Detection. The current hallucination detection model is still in an early stage, its span-level performance remains low, especially for summarization tasks. The detector is restricted to contextual factual hallucinations and does not address other types such as logical or commonsense errors. Its effectiveness also hinges on the retriever's ability to supply consistent and relevant context. While the model offers a basic safeguard, it is not yet suitable for high-stakes applications and requires substantial future development.

Toxicity Detection. The toxicity classifier was trained on the TextDetox dataset, which contains a non-negligible amount of label noise and inconsistent annotations across languages. This can cause instability in predictions, especially for borderline or multilingual cases.

Model Coverage and Scope. DeepPavlov 1.1 includes only three reliability-oriented models: a hallucination detector, an evergreen classifier, and a toxicity classifier. While these were prioritized based on user demand, other crucial capabilities, such as uncertainty-based detectors, adversarial input filters are absent and should be considered for future releases.

Ethics Statement

All models and experiments described in this paper were developed and evaluated exclusively using publicly available datasets. No proprietary, private, or personally identifiable data were used at any stage of model training, testing, or deployment. We are committed to transparency and reproducibility: all code, configuration files, and pretrained models are released under an open-source license to facilitate independent verification and responsible reuse by the research community.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Gregory S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian J. Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Józefowicz, Lukasz Kaiser, Manjunath Kudlur, and 21 others. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *CoRR*, abs/1603.04467.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: an easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations*, pages 54–59. Association for Computational Linguistics.

Mikhail Burtsev, Alexander Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina, Nickolay Bushkov, Olga Gureenkova, Taras Khakhulin, Yuri Kuratov, Denis Kuznetsov, Alexey Litinsky, Varvara Logacheva, Alexey Lymar, Valentin Malykh, Maxim Petrov, Vadim Polulyakh, Leonid Pugachev, Alexey Sorokin, Maria Vikhreva, and Marat Zaynutdinov. 2018. DeepPavlov: Open-source library for dialogue systems. In *Proceedings of ACL 2018, System Demonstrations*, pages 122–127, Melbourne, Australia. Association for Computational Linguistics.

Anastasia Chizhikova, Vasily Konovalov, and Mikhail Burtsev. 2023. Multilingual case-insensitive named entity recognition. In *Advances in Neural Computation, Machine Learning, and Cognitive Research VI*, pages 448–454, Cham. Springer International Publishing.

Daryna Dementieva, Daniil Moskovskiy, Nikolay Babakov, Abinew Ali Ayele, Naquee Rizwan, Frolian Schneider, Xintog Wang, Seid Muhie Yimam, Dmitry Ustalov, Elisei Stakovskii, Alisa Smirnova, Ashraf Elnagar, Animesh Mukherjee, and Alexander Panchenko. 2024. Overview of the multilingual text detoxification task at pan 2024. In *Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum*. CEUR-WS.org.

Daryna Dementieva, Vitaly Protasov, Nikolay Babakov, Naquee Rizwan, Ilseyar Alimova, Caroline Brun, Vasily Konovalov, Arianna Muti, Chaya Liebeskind, Marina Litvak, Debora Nozza, Shehryaar Shah Khan, Sotaro Takeshita, Natalia Vanetik, Abinew Ali Ayele, Florian Schneider, Xintong Wang, Seid Muhie Yimam, Ashraf Elnagar, and 2 others. 2025. Overview of the multilingual text detoxification task at PAN 2025. In *Working Notes of the Conference and Labs of the Evaluation Forum, CLEF 2025, Madrid, Spain, 9-12 September 2025*, volume 4038 of *CEUR Workshop Proceedings*, pages 3535–3567. CEUR-WS.org.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Feiteng Fang, Yuelin Bai, Shiwen Ni, Min Yang, Xiaojun Chen, and Ruifeng Xu. 2024. Enhancing noise robustness of retrieval-augmented language models with adaptive adversarial training. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10028–10039, Bangkok, Thailand. Association for Computational Linguistics.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Matthew Honnibal. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *(No Title)*.

Dmitry Karpov and Vasily Konovalov. 2023. Knowledge transfer between tasks and languages in the multi-task encoder-agnostic transformer-based models. In *Computational Linguistics and Intellectual Technologies*, volume 2023.

Ádám Kovács and Gábor Recski. 2025. Lettucedetect: A hallucination detection framework for RAG applications. *CoRR*, abs/2502.17125.

Nikita Krayko, Ivan Sidorov, Fedor Laputin, Alexander Panchenko, Daria Galimzianova, and Vasily Konovalov. 2025. Rurage: Robust universal rag evaluator for fast and affordable qa performance testing. In *Advances in Information Retrieval*, pages 135–145, Cham. Springer Nature Switzerland.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn J. Lawrie, and Benjamin Van Durme. 2025. mmbert: A modern multilingual encoder with annealed language learning. *CoRR*, abs/2509.06888.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 12076–12100. Association for Computational Linguistics.

Daniil Moskovskiy, Sergey Pletenev, and Alexander Panchenko. 2024. LLMs to replace crowdsourcing for parallel data creation? the case of text detoxification. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14361–14373, Miami, Florida, USA. Association for Computational Linguistics.

Viktor Moskvoretskii, Maria Marina, Mikhail Salnikov, Nikolay Ivanov, Sergey Pletenev, Daria Galimzianova, Nikita Krayko, Vasily Konovalov, Irina

Nikishina, and Alexander Panchenko. 2025. Adaptive retrieval without self-knowledge? bringing uncertainty back home. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6355–6384, Vienna, Austria. Association for Computational Linguistics.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2024. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 10862–10878. Association for Computational Linguistics.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. `http://openai.com/blog/chatgpt`.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Jason Phang, Phil Yeres, Jesse Swanson, Haokun Liu, Ian F. Tenney, Phu Mon Htut, Clara Vania, Alex Wang, and Samuel R. Bowman. 2020. `jiant` 2.0: A software toolkit for research on general-purpose text understanding models. `http://jiant.info/`.

Sergey Pletenev, Maria Marina, Nikolay Ivanov, Daria Galimzianova, Nikita Krayko, Mikhail Salnikov, Vasily Konovalov, Alexander Panchenko, and Viktor Moskvoretskii. 2025. Will it still be true tomorrow? multilingual evergreen question classification to improve trustworthy QA. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 8603–8620, Suzhou, China. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 101–108. Association for Computational Linguistics.

Elisei Rykov, Valerii Olisov, Maksim Savkin, Artem Vazhentsev, Kseniia Titova, Alexander Panchenko, Vasily Konovalov, and Julia Belikova. 2025a. SmurfCat at SemEval-2025 task 3: Bridging external knowledge and model uncertainty for enhanced hallucination detection. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 1034–1045, Vienna, Austria. Association for Computational Linguistics.

Elisei Rykov, Kseniia Petrushina, Maksim Savkin, Valerii Olisov, Artem Vazhentsev, Kseniia Titova, Alexander Panchenko, Vasily Konovalov, and Julia Belikova. 2025b. When models lie, we learn: Multilingual span-level hallucination detection with PsiloQA. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 11663–11682, Suzhou, China. Association for Computational Linguistics.

Maksim Savkin and Vasily Konovalov. 2024. Tuning-free discriminative nearest neighbor few-shot intent detection via consecutive knowledge transfer. In *Recent Trends in Analysis of Images, Social Networks and Texts*, pages 96–110, Cham. Springer Nature Switzerland.

Maksim Savkin, Anastasia Voznyuk, Fedor Ignatov, Anna Korzanova, Dmitry Karpov, Alexander Popov, and Vasily Konovalov. 2024. DeepPavlov 1.0: Your gateway to advanced NLP models backed by transformers and transfer learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 465–474, Miami, Florida, USA. Association for Computational Linguistics.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv: 2503.19786*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and Thang Luong. 2024. FreshLLMs: Refreshing large language models with search engine augmentation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13697–13720, Bangkok, Thailand. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *CoRR*, abs/2412.13663.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

A Latency

Model	**Params**	**PsiloQA**	**EvergreenQA**	**TextDetox**
Open-source language-models				
Qwen2.5-7B-Instruct	7B	990	36	74
Qwen2.5-32B-Instruct	32B	2756	130	196
Qwen2.5-72B-Instruct	72B	3000	190	225
gpt-oss-120b	120B	2164	539	675
Transformer encoder models				
EG-E5-large	560M	–	6.3	–
ModernBERT-base (our)	395M	**8.7**	<u>5.9</u>	–
mmBERT-base (our)	395M	<u>8.8</u>	–	–
RoBERTa-large	355M	–	–	<u>6.3</u>
BERT-base (our)	110M	–	**0.4**	**1.4**

Table 6: Model performance in milliseconds per task sample across different tasks.

B Training Hyperparameters

Hyperparameter	**Value**
`Batch size`	8
`Optimizer`	AdamW
`Learning rate`	1×10^{-5}
`Weight decay`	1×10^{-3}
`Adam betas`	(0.9, 0.999)
`Adam epsilon`	1×10^{-6}
`Clip norm`	1.0
`Max epochs`	6
`Model selection metric`	F1-weighted

Table 7: Training hyperparameters for Hallucination Detector.

Hyperparameter	**Value**
`Max sequence length`	512
`Batch size`	16
`Optimizer`	AdamW
`Learning rate`	1×10^{-5}
`Weight decay`	1×10^{-6}
`Adam betas`	(0.9, 0.999)
`Adam epsilon`	1×10^{-6}
`Clip norm`	1.0
`Min. learning rate`	2×10^{-6}
`Learning rate drop patience`	15
`Learning rate drop factor`	1.5
`Max epochs`	20
`Model selection metric`	F1-weighted

Table 8: Training and model hyperparameters for EvergreenQA classifier.

Hyperparameter	**Value**
`Max sequence length`	64
`Batch size`	64
`Optimizer`	AdamW
`Adam betas`	(0.9, 0.999)
`Adam epsilon`	1×10^{-6}
`Learning rate`	1×10^{-5}
`Learning rate drop patience`	5
`Learning rate drop factor`	2.0
`Max epochs`	100
`Model selection metric`	F1-weighted

Table 9: Training and model hyperparameters for Toxicity Classification.

C LLM Judge Variance

Model	Params	Temp	IoU (%)	AP (%)
Qwen2.5-7B-Instruct	7B	0.3	36.97 ± 0.62	61.90 ± 1.03
Qwen2.5-32B-Instruct	32B	0.3	46.77 ± 0.33	64.49 ± 0.73
gpt-oss-120B	120B	1.0	42.29 ± 0.55	58.14 ± 0.50

Table 10: Variance of LLM-as-a-judge hallucination detection across 5 independent runs. Each model is evaluated with 3-shot prompting on the PsiloQA (Rykov et al., 2025b) test set (all 14 languages). We report character-level macro-averaged IoU and micro-averaged AP (mean $\pm$ std over 5 runs). Qwen models use temperature 0.3; gpt-oss-120B uses temperature 1.0.

Model	Params	Temp	F1-weight
Qwen2.5-7B-Instruct	7B	1.0	77.66 ± 0.27
Qwen2.5-32B-Instruct	32B	0.3	79.62 ± 0.12

Table 11: Variance of LLM-as-a-judge hallucination detection across 5 independent runs. Each model is evaluated on the `TextDetox` (Dementieva et al., 2024) test set. We report F1-weighted (mean $\pm$ std over 5 runs).

D Prompt Templates for LLM Baselines

Toxicity Classification Prompt

You are a safety classifier that determines whether a single user message is toxic or non-toxic.

Toxic content includes insults, hate speech, threats, explicit harassment, or demeaning language toward an individual or group. Non-toxic content is neutral, polite, or otherwise acceptable.

Read the message carefully and decide if it is **Toxic** or **Non-toxic**. At the end, respond with a single line in the form:
`Classification: Toxic`
or
`Classification: Non-toxic`.

Examples:
`{few_shot}`

Message: `{input_text}`

Classification:

Figure 3: Prompt for binary toxicity classification.

Hallucination Detection Prompt

You are an expert hallucination detector for question answering with retrieved context.

Given:
– a context passage from Wikipedia,
– a user question, and
– an LLM answer,

you must identify all hallucinated spans in the answer.

A hallucinated span is any part of the answer that:
– contradicts the context, or
– introduces specific factual details that are not supported by the context or by the gold answer.

Your output must be the model answer text where you wrap every hallucinated span in `[HAL]` and `[/HAL]` tags.

CRITICAL INSTRUCTIONS:
– Do not change, rephrase, re order, or truncate the answer.
– Do not add new information.
– Only insert `[HAL]` before and `[/HAL]` after hallucinated spans.
– If there is no hallucination, return the answer unchanged (with no `[HAL]` tags).

Examples:
`{few_shot}`

Return only the model answer text, where hallucinated spans are wrapped in `[HAL]` and `[/HAL]` tags. Do not add any explanation or commentary.

Knowledge source: `{passage}`
Question: `{question}`
Answer: `{answer}`
Answer with highlighted spans:

Figure 4: Prompt for span level hallucination detection.

Evergreen Detection Prompt

You are a helpful assistant that classifies questions based on their temporality.

There are two classes:
Immutable: the answer almost never changes over time (for example, historical facts, birth years, names of past events).
Mutable: the answer typically changes over the course of several years or less (for example, current leaders, upcoming events, latest statistics).

Think carefully about each question and decide whether it is Immutable or Mutable. At the end, answer with exactly one line of the form:
`Classification: Immutable`
or
`Classification: Mutable`.

Examples:

`{few_shot}`

Question: `{input_question}`

Classification:

Figure 5: Prompt for Evergreen classification.