Hybrid Symbolic-Neural Domain Adaptation via SymbSteer. Markov-Guided Prompting and Decoding for Resource-Efficient Language Model Steering

Zbigniew H. Gontar
SGH Warsaw School of Economics
Warsaw, Poland *zgonta@sgh.waw.pl*

Beata Gontar
University of Lodz
Lodz, Poland *beata.gontar@uni.lodz.pl*

Abstract

Adapting large language models (LLMs) to formal, low-resource domains-such as public procurement or regulatory writing-remains a significant challenge, particularly in non-English contexts. We present a lightweight hybrid framework that combines symbolic 3-gram Markov models with neural generation using DistilGPT2. The approach introduces symbolic guidance in two stages: domain-specific few-shot prompting and decoding-time probability adjustment. This enables domain-consistent generation without model retraining. Evaluated on Polish public procurement documents and deployed on CPU-only infrastructure, the method improves domain fidelity, structure, and semantics, as measured by BLEU, ROUGE-L, and BERTScore. The proposed framework offers a scalable, inference-only alternative to fine-tuning for generating formal texts under strict resource constraints.

Keywords: domain-adaptation, Markov models, few-shot prompting, guided decoding, resource-efficient NLP.

1. Introduction

Adapting large language models (LLMs) to formal, low-resource domains—such as public procurement or regulatory writing—poses unique challenges. These domains feature rigid syntax, repetitive structures, and specialized terminology, often in under-resourced languages like Polish or Hungarian. Unlike general-purpose corpora, they demand stylistic and structural fidelity that standard prompting fails to capture [14].

While fine-tuning or continued pretraining offers one solution, these approaches are often infeasible due to computational and storage costs, especially in constrained settings like public sector infrastructure [12]. Recent methods like LoRA [8] offer parameter-efficient alternatives, but still require access to model internals and GPU resources. Inference-only methods that maintain domain alignment without retraining are therefore highly desirable.

We introduce SymbSteer, a hybrid symbolic-neural framework for low-resource domain adaptation. It uses a simple trigram Markov model to (1) generate domain-representative few-shot prompts and (2) apply decoding-time token biasing. Combined with DistilGPT2, this setup enables domain-consistent generation on CPU-only infrastructure without updating model weights.

We evaluate our method on a corpus of Polish public procurement texts, demonstrating improved structural and semantic alignment through BLEU, ROUGE-L, and BERTScore.

2. Background and Related Work

Natural Language Generation (NLG) is a fundamental challenge in natural language processing (NLP), especially when applied to formal, domain-specific texts such as legal, procurement, or regulatory documents. These texts are characterized by rigid syntax, repetitive structures, and specialized terminology, complicating syntactic coherence and semantic fidelity in machine-generated outputs [3].

Traditional approaches to text generation include statistical language models [16] (e.g., n-grams, Markov chains), deep neural architectures (e.g., LSTMs, RNNs), and more recently, large-scale Transformer-based pre-trained language models (PLMs) like GPT, BERT and Falcon [6], [9], [21]. While PLMs have significantly advanced the fluency and coherence of generated text, they often fail to internalize strict domain constraints without extensive supervised fine-tuning, which is infeasible in many real-world low-resource contexts [12].

As a viable alternative, few-shot prompting techniques have emerged, in which domain-specific examples are embedded in the prompt to simulate learning without parameter updates. However, such methods can fail when token order and formal consistency are crucial.

Symbolic and statistical methods such as Markov chains have a long tradition in modeling local linguistic patterns. A 3-gram Markov model, for instance, estimates transition probabilities between word sequences based on short context windows, effectively capturing domain-consistent phraseology [16]. These models are interpretable, computationally inexpensive, and well-suited to domains with formulaic language, such as public tenders or legal drafts [19].

Recent studies suggest that combining symbolic priors with neural decoders—via either logit steering or synthetic prompting—can guide large language models more effectively in low-data settings [15]. Controlled decoding strategies (e.g., plug-and-play models) have demonstrated that modifying output distributions at generation time can significantly enhance alignment with target constraints, even without fine-tuning [2].

3. Method - SymbSteer Framework

3.1. Overview

SymbSteer is a hybrid symbolic-neural framework designed for domain adaptation without model retraining. It uses a trigram Markov model trained on a small, domain-specific corpus to provide symbolic guidance at two stages of the generation pipeline: (1) input prompting and (2) decoding. These components operate independently of the LLM's architecture and enable low-resource, inference-only adaptation.

3.2. Symbolic Prompt Generator

We first train a simple trigram Markov model on a curated corpus of formal procurement documents. This model captures domain-specific token transition patterns, which reflect legal phrasing and fixed syntactic structures. We use it to generate synthetic sequences that resemble in-domain language and insert them as few-shot examples at the start of the LLM's context.

Unlike human-curated prompts, these synthetic examples are automatically sampled and lightweight, enabling adaptation even with minimal labeled data. They help prime the LLM to follow the syntactic and stylistic conventions of the domain, enhancing formal consistency during inference.

3.3. Symbolic Decoding Controller

Beyond prompting, we introduce decoding-time control by incorporating symbolic likelihoods into token selection. At each step of generation, the LLM's output probabilities are blended with probabilities from the Markov model. This reweighting biases the decoder toward transitions consistent with domain-specific patterns.

Rather than altering the LLM's weights, we apply a simple adjustment to the predicted token scores before sampling. The degree of symbolic influence is tunable via a blending parameter, allowing fine-grained trade-offs between fluency and structural fidelity. This adjustment integrates seamlessly with existing decoding methods (e.g., top-k, nucleus sampling) and requires no architectural changes.

At each decoding step, we blend the token probabilities from the LLM and the Markov model using:

$$P'_{final}(w_t) \propto P_{LM}(w_t|w_{<t})^{\lambda} \cdot P_{Markov}(w_t|w_{t-1}, w_{t-2})^{1-\lambda} \tag{1}$$

where $\lambda \in [0,1]$ controls the balance between fluency and symbolic conformity.

3.4. Combined System

The two mechanisms—symbolic prompting and Markov-guided decoding—operate in tandem. Prompting sets the tone and structure for initial generation, while decoding correction maintains alignment throughout the output. Together, they reinforce phrase consistency, prevent style drift, and ensure outputs remain domain-appropriate across long sequences.

Because both steps occur during inference, the system remains lightweight and easily deployable in environments lacking GPU acceleration or training capabilities. This makes SymbSteer especially suited for constrained applications such as government NLP or low-budget institutional research.

4. Symbolic Prior Extraction from Domain Corpus

We construct a trigram Markov chain model trained on a curated corpus of Polish public procurement documents to obtain a lightweight symbolic representation of domain-specific linguistic structure. These texts include formal tenders and legal notices related to university construction projects-a subdomain that exemplifies highly regular syntax, formal institutional phrasing, and constrained semantic intent. Such characteristics make this dataset an ideal testbed for exploring symbolic modeling of structural patterns in formal domains [5].

The final corpus contains 52,867 tokens, from which we extract 23,188 unique trigram sequences. Each trigram consists of a pair of consecutive words (w_i,w_{i+1}), called the prefix, followed by a successor token w_{i+2}. The model estimates the probability of encountering a particular successor word given the prefix, thereby encoding a representation of the local syntactic flow prevalent in the domain. The resulting transition graph forms a first-order approximation of domain-specific language sufficient to capture repetitive constructions, formal legal clauses, and typical clause openers [16].

Formally, for each trigram (w_i,w_{i+1},w_{i+2}), we estimate the transition probability:

$$P(w_{i+2}|w_i, w_{i+1}) = \frac{count(w_i,w_{i+1},w_{i+2})}{\sum_{w'} count(w_i,w_{i+1},w')} \tag{2}$$

This probability estimates the likelihood of token w_{i+2} appearing after the pair (w_i,w_{i+1}), computed by normalizing the observed count of the full trigram over all possible successors following the same prefix. While this structure lacks long-range syntactic or discourse awareness, it is sufficient to model surface-level regularities and local cohesion-a frequent characteristic of domain-specific documentation [19].

We leverage the resulting symbolic model in two complementary roles:

- We sample sequences from the trained Markov chain to create synthetic, domain-representative utterances. These synthetic examples reflect common collocations and phrase templates seen in public tenders and are embedded into few-shot prompts. The goal is to prime the language model with structurally valid and stylistically accurate inputs, enhancing domain alignment before generation begins.
- During generation with DistilGPT2, we dynamically adjust the model's output logits based on the Markov model's transition probabilities. At each time step, tokens likely to occur (given the recent context) according to the Markov model receive a positive bias, while unlikely tokens are penalized. This ensures that

the model's output distribution follows the correct continuation, the model weights are not altered, and the gradient is updated. This method supports top-k or nucleus sampling [7].

5. Experimental Setup

5.1. Dataset

We use a curated corpus of Polish public procurement documents, characterized by repetitive phrasing, legal structure, and constrained vocabulary. The domain features long average sentence lengths (~25 words), a low type-token ratio (0.19), and over 40% trigram repetition — indicating high structural regularity typical of formal administrative texts [5], [9].

These properties make the dataset ideal for evaluating symbolic domain adaptation. The Markov model is trained on 52,867 tokens, capturing local word transitions that reflect the domain's rigid linguistic patterns.

5.2. Models and Baselines

We use DistilGPT2-small (82M parameters)), a distilled variant in the GPT family inspired by the principles of DistilBERT [18], for its efficient inference on CPU-only hardware, without retraining or gradient updates. Our approach operates entirely at inference time, allowing symbolic prompting and decoding to adapt generation behavior without modifying the model.

All experiments were conducted using Databricks Community Edition (CPU-only). This setup demonstrates feasibility for real-world deployments in constrained environments [4].

We compare four configurations to isolate each component's contribution described in section 6.1 (Table 1.):

Table 1. Overview of model variants used in evaluation. Each configuration varies in the use of symbolic prompting and/or decoding to steer domain-specific generation.

Model Variant	Description
Vanilla Prompting	Few-shot prompt manually selected, no symbolic augmentation.
Symbolic Prompting	Few-shot prompt generated via Markov sampling.
Markov Decoding	Symbolic bias during token selection, no prompting.
Full Hybrid	Combined symbolic prompting + decoding.

5.3. Evaluation Metrics

To evaluate the effectiveness of our hybrid framework, we adopted the metrics - BLEU, ROUGE-L, and BERT Score - each of which captures different aspects of generation quality and provides a comprehensive view of the model's performance in the target domain [11], [17].

We assessed model outputs using the following:

- BLEU (Bilingual Evaluation Understudy) measures the precision of n-gram overlaps between the generated output and a reference text. It evaluates how many n-gram sequences in the generated text appear in the reference, capturing lexical accuracy and fluency at multiple levels. BLEU incorporates a brevity penalty (BP) that discourages under-generation to prevent the model from generating excessively short outputs. Formally, BLEU is defined as:

$$BLEU = BP \cdot exp(\sum_{n=1}^{N} w_n log(p_n)) \tag{3}$$

where p_n is modified precision for n-grams of size n, w_n is weight assigned to each n-gram size (typically uniform), and BP is brevity penalty, calculated as:

$$BP = \begin{cases} 1 & if c > r \\ e^{1-r/c} & if\ c \leq r \end{cases} \tag{4}$$

Here, c is the length of the candidate output, and r is the length of the reference.

BLEU is most effective at measuring surface-level fluency and phrase alignment, making it a good fit for formulaic text domains like public procurement.

- ROUGE -L (Recall-Oriented Understudy for Gisting Evaluation) focuses on recall by computing the longest common subsequence (LCS) between the reference and the generated text. Unlike BLEU, which emphasizes exact n-gram matches, ROUGE -L captures non-contiguous but ordered overlaps, making it more suitable for evaluating overall structural alignment. Formally, it is defined as:

$$ROUGE - L = \frac{LCS(X,Y)}{length\, of\, reference} \tag{5}$$

where LCS(X,Y) is the longest subsequence in both sequences X and Y [10].

- BERT Score evaluates semantic similarity between generated and reference texts using contextualized embeddings from a pre-trained BERT model. It measures how closely each token in the generated output aligns-embedding-wise-with its most similar token in the reference. Unlike BLEU and ROUGE, which rely on surface overlap, BERT Score enables soft alignment, rewarding outputs that preserve meaning even if exact wordings differ.

$$BERTScore = \frac{1}{|X|} \sum_{x \epsilon X} \max_{y \epsilon Y} \cos\big(\phi(x), \phi(y)\big) \tag{6}$$

where ϕ(·) is BERT-based contextual embedding function, X,Y are token sets from candidate and reference texts, and cos(·,·) is cosine similarity [22]. This metric is particularly valuable in legal or regulatory text generation, where exact phrasal matching is less important than preserving meaning and logical form.

6. Results

6.1. Overview of Results

To assess the contribution of each component in our hybrid symbolic-neural framework, we systematically compared four generation configurations. These setups were chosen to isolate the individual effects of symbolic prompting and decoding and evaluate their combined synergy. Each method was evaluated using the same model architecture (DistilGPT2) and tested on the domain-specific dataset of Polish public procurement texts.

Let M be the Markov model and G be the base LLM (DistilGPT2). The configurations are defined as follows.

We compared:

- Vanilla Prompting. A standard few-shot prompting baseline. DistilGPT2 is conditioned with a manually selected domain-relevant prompt without any symbolic augmentation. This serves as a lower-bound baseline representing general-purpose usage of the model in the absence of domain steering.

$$Vanila\ Prompting = Prompt(G) \tag{7}$$

- Markov few-shot prompting only. In this setup, the model is primed with few-shot examples automatically generated by sampling from the Markov model. These synthetic sequences mirror the domain's structural patterns, allowing the model to internalize them at inference time. However, no symbolic control is applied during decoding, allowing us to assess the effect of prompting alone [19].

$$Few - Shot\ Prompting = Prompt\big(G, Sample(M)\big) \tag{8}$$

- Markov decoding only. Here, no synthetic prompt examples are provided. Instead, the output logits of the model are dynamically adjusted during generation using transition probabilities from the Markov model. This method applies symbolic constraints in real-time, steering output token selection according to domain-specific structural expectations [16].

$$Markov\ Decoding = Decode\big(G, Bias(M)\big) \tag{9}$$

- Full hybrid model (proposed). Our complete symbolic-neural integration strategy. The model is primed with synthetic Markov-generated examples, and its token-level generation is simultaneously influenced by Markov-conditioned decoding. This setup combines initial structure-aligned conditioning with continuous, symbolic correction throughout the generation pipeline.

$$Full\ Hybrid = Prompt\big(G, Sample(M)\big) + Decode\big(G, Bias(M)\big) \tag{10}$$

Each configuration explores a different intersection of symbolic control and neural generation. By keeping the base model G fixed and modulating the symbolic interface via M, we isolate the impact of prompt design, decoding intervention, and their joint effect on domain alignment [1], [4].

Our hybrid framework outperformed all baseline configurations across BLEU, ROUGE-L, and BERTScore. Symbolic guidance via prompting and decoding significantly improved both surface-level fluency and semantic consistency relative to domain-specific references.

Table 2. Automatic evaluation scores for each model variant. The hybrid approach outperforms all baselines across BLEU, ROUGE-L, and BERTScore metrics, confirming the effectiveness of dual symbolic guidance.

Model Variant	BLEU	ROUGE-L	BERTScore
Vanilla Prompting	0.42	0.47	0.81
Markov Prompting Only	0.48	0.53	0.84
Markov Decoding Only	0.50	0.55	0.86
Full Hybrid (Ours)	0.58	0.63	0.89

The hybrid model achieved up to +15% improvement in BLEU/ROUGE over the baseline and an +8% increase in BERTScore, indicating better alignment with domain phrasing and meaning (Table 2.).

6.2. Qualitative Observations

Generated outputs from the hybrid model exhibited:

- Canonical phrasing (e.g., “zgodnie z dokumentacją przetargową”),
- Fewer grammatical anomalies,
- Better clause structuring in legal-style sentences.

These outputs adhered more consistently to formal norms, a key requirement in the regulatory and procurement domain.

6.3. Ablation Insights

Ablation experiments confirm the complementary effect of the two symbolic components:

- Prompting only improved structural initialization and framing.
- Decoding only reduced mid-sentence drift and reinforced token-level

regularity.

- Combined, they yielded the strongest results, confirming the value of symbolic guidance across both input and output stages.

Even in CPU-only settings, the symbolic components enabled strong domain control without any model fine-tuning. This underscores the viability of inference-only adaptation for formal, resource-constrained applications.

7. Discussion & Limitations

Our results show that even simple symbolic models—like 3-gram Markov chains—can substantially enhance the domain alignment of language model outputs. The proposed hybrid approach demonstrates that symbolic prompting and decoding can compensate for the lack of fine-tuning, especially in resource-constrained settings such as public-sector NLP or non-English domains.

By capturing local syntactic regularities, the symbolic model helps the neural generator remain stylistically consistent with formal domain expectations. Crucially, the entire pipeline runs on CPU-only infrastructure with no model retraining, making it deployable in low-compute environments.

However, our method has limitations. The 3-gram model captures only short-range dependencies and lacks global context or document-level structure. This restricts its ability to model complex legal logic, nested clauses, or discourse-level coherence. In some cases, it may also overfit common patterns, reducing lexical diversity.

Future directions include integrating more expressive symbolic components, such as probabilistic context-free grammars (PCFGs) [20], constraint-based models, or discourse-aware structures such as Rhetorical Structure Theory (RST) [13]. Adaptive symbolic weighting—where the influence of the prior shifts dynamically during generation—may further improve stylistic control without sacrificing fluency.

8. Conclusions

We introduced a lightweight hybrid framework for steering language model outputs using symbolic priors. By combining Markov-based prompting and decoding-time reweighting, our approach improves domain fidelity without retraining or additional model parameters.

The system is deployable on CPU-only hardware and requires no annotated training data, making it suitable for low-resource domains like Polish public procurement. This work contributes to scalable, interpretable, and accessible NLP for formal and structured text generation, and offers a promising direction for symbolic-neural collaboration in domain adaptation.

References

1. Brown, T. et al., Language models are few-shot learners. In: Advances in Neural Information Processing Systems (NeurIPS), 33, pp.1877–1901 (2020)
2. Dathathri, S. et al., Plug and play language models: A simple approach to controlled text generation. In: International Conference on Learning Representations (ICLR) (2020)
3. Deckers, R. and Lago, P., Systematic literature review of domain-oriented specification techniques. Journal of Systems and Software, 192, pp.111415 (2022)
4. Dettmers, T. et al., QLoRA: Efficient finetuning of quantized LLMs. arXiv preprint arXiv:2305.14314 (2023)
5. Felizzola, H., Gomez, C., Arrieta, N., Jerez, V., Erazo, Y., Camacho, G., Enhancing transparency in public procurement: A data-driven analytics approach, Information Systems, Volume 125, 102430,https://doi.org/10.1016/j.is.2024.102430, (2024)
6. Fields, J., Chovanec, K. and Madiraju, P., A survey of text classification with

transformers. IEEE Access, 12, pp.6518–6531 (2024)

7. Holtzman, A., Buys, J., Du, L., Forbes, M., Choi, Y., The curious case of neural text degeneration. In: International Conference on Learning Representations (ICLR) (2020)
8. Hu, E. et al., LoRA: Low-Rank Adaptation of Large Language Models. arXiv preprint arXiv:2106.09685 (2021)
9. Jurafsky, D., Martin, J.H., Speech and Language Processing (3rd ed.). Prentice Hall (2025)
10. Lin, C.-Y., ROUGE: A package for automatic evaluation of summaries. In: Proceedings of the ACL-04 Workshop on Text Summarization Branches Out, pp. 74–81, (2004)
11. Lu, J., Shao, Y., Xu, Y. and Neubig, G., NeuroLogic A*esque Decoding: Constrained Text Generation with Lookahead Heuristics. In: Proceedings of ACL-IJCNLP, pp. 6670–6683 (2021)
12. Ma, W., Ho, S.Y., Sentiment-devoid lexicons: A novel method for domain-specific textual analysis in business and governance documents. Information Management, 62(1) (2025)
13. Mann, W.C., Thompson, S.A., Rhetorical Structure Theory: Toward a functional theory of text organization. Text-Interdisciplinary Journal for the Study of Discourse, 8(3), pp.243–281 (1988)
14. Naveed H., Arora C., Khalajzadeh H., Grundy J., Haggag O., Model driven engineering for machine learning components: A systematic literature review, Information and Software Technology, Vol. 169, 107423 (2024)
15. Panchendrarajan, R., Zubiaga, A., Synergizing machine learning & symbolic methods: A survey on hybrid approaches to NLP. Expert Systems with Applications. Preprint (submitted) (2024)
16. Pandey, A.K., Roy, S.S., Natural language generation using sequential models: A survey. Neural Processing Letters, 55(6) (2023)
17. Papineni, K., Roukos, S., Ward, T., Zhu, W.-J., BLEU: A method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318 (2002)
18. Sanh, V., Debut, L., Chaumond, J., Wolf, T., DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 , (2019)
19. Shakhovska, K., Dumyn, I., Kryvinska, N., Kagita, M.K., An approach for a next-word prediction for Ukrainian language. Wireless Communications and Mobile Computing, 2021, Article ID 5886119 (2021)
20. Stolcke, A., An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. Computational Linguistics, 21(2), pp.165–201 (1995)
21. Wolf, T. et al., Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45 (2020)
22. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q. and Artzi, Y., BERTScore: Evaluating text generation with BERT. In: International Conference on Learning Representations (ICLR) (2020)