Split-Wise Evaluation for Turkish Light Verb Construction Detection

Sercan Karakaş
University of Chicago
skarakas@uchicago.edu

Yusuf Şimşek
Fırat University
ysimsek@firat.edu.tr

Relevant UniDive working groups: WG1, WG2, WG3, WG4

1 Introduction

Multiword expressions (MWEs) pose a persistent challenge for NLP because they often resist strictly compositional analysis while exhibiting substantial lexical, semantic, and morphosyntactic variability (Sag et al., 2002; Odijk, 2013; Ramisch, 2015; Savary et al., 2017; Barman et al., 2024; Mititelu et al., 2025; Karakaş and Şimşek, 2026). This challenge is particularly acute in Turkish, where rich inflectional morphology, flexible argument realization, and highly productive verb–nominal predicate formation generate considerable surface variation (Oflazer, 1994; Oflazer et al., 2004; Butt, 2010; Uçar, 2010). Following the PARSEME guidelines for verbal multiword expressions, we treat light verb constructions (LVCs) as predicate-like noun–verb expressions in which the nominal element contributes the core predicational content, while the verbal element is semantically light or partially bleached (Savary et al., 2017). In Turkish, however, the same verb may also occur in fully literal, compositional uses, yielding minimal contrasts between idiomatic and literal readings that are difficult to resolve from lexical identity alone.

In this paper, we use Turkish LVCs as a controlled diagnostic for testing whether models recover predicate-level meaning or instead rely primarily on shallow lexical defaults. Specifically, we compare a supervised Turkish encoder baseline with three instruction-tuned LLMs under zero-shot, one-shot, and few-shot prompting regimes. The results reveal a strong effect of prompting regime: zero-shot LLMs behave conservatively and miss many positive LVC cases, one-shot prompting can lead to overcorrection, and few-shot prompting improves calibration while remaining highly model-dependent. More broadly, the paper argues that Turkish LVCs provide a valuable benchmark for evaluating meaning-sensitive MWE detection, since aggregate accuracy alone can obscure whether a model has genuinely learned the literal–idiomatic contrast.

2 Data and Experimental Design

We train the encoder models on nine Turkish UD treebanks (ud-, a,b,c,d,e,f,h,g,i). Candidate LVCs are extracted from `compound:lvc` where available and otherwise from noun–verb `compound` relations with manual filtering (ud-, k,j), yielding 82,319 sentences and 9,491 LVC instances. This operationalization is intended to be compatible with the PARSEME view of LVCs as verbal MWEs whose overall predicate meaning cannot be reduced to an ordinary literal use of the verb alone (Savary et al., 2017). The diagnostic NLVC condition was designed to contrast these cases with sentences in which the same verbs retain their literal lexical meaning.

Evaluation uses a separate 147-item diagnostic set balanced across LVC, NLVC, and RANDOM conditions (n=49 each). The generated diagnostic evaluation dataset is presented in Appendix A. The NLVC items reuse the same target verbs as the positives under literal readings, blocking simple verb-based heuristics and forcing models to distinguish lexicalized from literal uses. All items were validated by three annotators. Because the set is diagnostic rather than i.i.d., we interpret the results as evidence about controlled decision-boundary behavior rather than broad in-the-wild performance (Ribeiro et al., 2020; Gardner et al., 2020; Kiela et al., 2021; Yang et al., 2022; Zhao et al., 2024; He et al., 2025; Mayne et al., 2025).

The task is sentence-level binary classification: `[1]` for LVC and `[0]` otherwise. We report split-wise and overall accuracy; the fuller analysis also includes false positives, false negatives, precision, and recall, since aggregate accuracy alone can hide systematic failure on the positive class.

3 Models

We fine-tune BERTurk 32K and 128K cased models with a binary classifier over the final-layer `[CLS]` representation, using the original setup: an 80/20 stratified split, dropout 0.2, learning rate 2×10^{-5}, batch size 32, weight decay 0.01, and early stopping. These serve as supervised Turkish baselines and provide a useful comparison point for prompted models.

We also evaluate three instruction-tuned LLMs

via Ollama—`llama3.1:8b`, `gptoss:20B`, and `Qwen2.5:14B`—with sentence-level binary prompts and low-temperature decoding. We compare zero-shot, one-shot, and few-shot prompting, asking whether demonstrations improve LVC detection or mainly shift the threshold for predicting the positive label (Brown et al., 2020; Liu et al., 2023). This setup lets us compare supervised adaptation against in-context adaptation on the same controlled benchmark.

4 Experiments and Results

4.1 Experiment 1: Zero-shot prompting

The first experiment compares zero-shot LLMs against the supervised BERTurk baselines. The main pattern is a sharp asymmetry between negative and positive conditions. In zero-shot, all three LLMs perform well on RANDOM and NLVC items but largely fail on LVC positives. Llama 3.1 8B shows the strongest version of this pattern, with near-ceiling performance on negatives but complete collapse on LVCs. GPT-OSS-20B and Qwen 2.5 14B behave similarly, though with slightly better positive recall. In practical terms, the models default to the safer negative label unless given explicit evidence that positive cases exist and should be recognized. The results obtained from BERTurk and zero-shot prompting are provided in Appendix B.

This makes pooled accuracy misleading. Because two of the three evaluation splits are negative, a conservative model can still look moderately successful overall while missing the phenomenon the benchmark is designed to test. By contrast, the supervised BERTurk baselines remain much more balanced across conditions, especially BERTurk-128k, which performs strongly on LVC positives while preserving high negative accuracy. The gap suggests that task-specific Turkish supervision makes the literal–idiomatic distinction more directly accessible than prompting alone in the zero-shot setting.

4.2 Experiment 2: One-shot prompting

The second experiment tests whether a minimal in-context demonstration corrects the zero-shot false-negative bias. It does, but not in a uniform way. Instead, one-shot prompting produces strongly model-specific calibration shifts. Llama 3.1 8B moves from near-total rejection of the positive class to a much more liberal strategy: its LVC accuracy rises sharply, but its performance on both negative splits drops substantially. This indicates over-prediction of the LVC label. Qwen 2.5 14B shows the opposite tendency. It remains extremely strong on negatives and improves less dramatically on positives, suggesting a more conservative threshold. GPT-OSS-20B is the most balanced model under one-shot prompting, improving substantially on LVCs without collapsing on the negative splits. The tabulated results of the three models evaluated under one-shot prompting are provided in Appendix C.

The interpretation is that one-shot demonstrations do not simply "teach" the task. They can also reset the model's decision bias. A single positive example may be enough to convince one model that LVCs should be predicted much more often, while another model still treats them as exceptional and continues to prefer the negative class. In the original analysis, chi-square tests with Holm correction show reliable model differences across the splits in this regime, which reinforces the point that the prompt is interacting with model family rather than producing a stable universal improvement.

4.3 Experiment 3: Few-shot prompting

The third experiment asks whether a richer prompt can reduce the extremes observed in one-shot evaluation. Overall, the answer is yes. Few-shot prompting largely removes the zero-shot always-negative failure mode and softens the strongest one-shot distortions. GPT-OSS-20B and Qwen 2.5 14B both achieve strong overall performance under few-shot prompting, and both become substantially more competitive with the supervised baselines. Llama 3.1 8B preserves high LVC recall, but remains noticeably less calibrated on the negative splits, which shows that additional demonstrations help but do not fully eliminate family-specific response biases. The results are presented in Appendix D.

The broader pattern across all three experiments is therefore not monotonic "more prompting is better," but rather a shift from conservatism to more balanced behavior, with different models stabilizing at different points. Zero-shot prompting under-predicts positives. One-shot prompting can overcorrect. Few-shot prompting usually improves calibration, but the resulting balance between false positives and false negatives remains model-dependent. This is why split-wise reporting is essential: the same overall score can arise from very different underlying behaviors.

5 Discussion

The experiments support two main conclusions. First, Turkish LVC detection is strongly prompt-sensitive: the same LLM can shift from missing most positives to over-predicting them depending on the demonstration regime. Second, the supervised encoder is more stable overall. Although this is not a fully symmetric comparison, since BERTurk uses task-specific supervision and the LLMs rely only on prompting, the contrast is

still informative. Turkish LVCs therefore provide a compact diagnostic benchmark for lexicalized predicate meaning and show why pooled accuracy alone can be misleading for linguistically structured tasks.

6 Conclusion

We presented a controlled evaluation of Turkish LVC detection comparing supervised Turkish encoders with instruction-tuned LLMs under zero-shot, one-shot, and few-shot prompting. Zero-shot LLMs perform well on negatives but largely fail on positives, one-shot prompting often introduces strong calibration shifts, and few-shot prompting improves performance while preserving clear differences across model families. Supervised BERTurk remains the most stable baseline, while carefully designed prompts allow some LLMs to approach it on controlled judgments.

References

a. UD Turkish Atis. Universal Dependencies treebank hub page. Accessed: 2025-12-16.

b. UD Turkish BOUN. Universal Dependencies treebank hub page. Accessed: 2025-12-16.

c. UD Turkish FrameNet. Universal Dependencies treebank hub page. Accessed: 2025-12-16.

d. UD Turkish GB. Universal Dependencies treebank hub page. Accessed: 2025-12-16.

e. UD Turkish IMST. Universal Dependencies treebank hub page. Accessed: 2025-12-16.

f. UD Turkish Kenet. Universal Dependencies treebank hub page. Accessed: 2025-12-16.

g. UD Turkish Penn. Universal Dependencies treebank hub page. Accessed: 2025-12-16.

h. UD Turkish PUD. Universal Dependencies treebank hub page. Accessed: 2025-12-16.

i. UD Turkish Tourism. Universal Dependencies treebank hub page. Accessed: 2025-12-16.

j. Universal dependencies: `compound`. Universal Dependencies relations documentation. Accessed: 2025-12-16.

k. Universal dependencies: `compound:lvc`. Universal Dependencies relations documentation. Accessed: 2025-12-16.

A. Barman, D. Saha, and A. R. Pal. 2024. An approach for maintaining structural uniformity of multiword expressions. In *2024 Fourth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*. IEEE.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Miriam Butt. 2010. The light verb jungle: still hacking away. In Mengistu Amberber, Brett Baker, and Mark Harvey, editors, *Complex Predicates: Cross-linguistic Perspectives on Event Structure*. Cambridge University Press.

Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hanna Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2758–2768, Online. Association for Computational Linguistics.

Linyang He, Qiaolin Wang, Xilin Jiang, and Nima Mesgarani. 2025. Layer-wise minimal pair probing reveals contextual grammatical-conceptual hierarchy in speech representations. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Suzhou, China. Association for Computational Linguistics.

Sercan Karakaş and Yusuf Şimşek. 2026. From lemmas to dependencies: What signals drive light verbs classification? In *Proceedings of the Second Workshop Natural Language Processing for Turkic Languages (SIGTURK 2026)*, pages 220–227, Rabat, Morocco. Association for Computational Linguistics.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*.

Harry Mayne, Ryan Othniel Kearns, Yushi Yang, Andrew M. Bean, Eoin D. Delaney, Chris Russell, and Adam Mahdi. 2025. LLMs don't know their own decision boundaries: The unreliability of self-generated counterfactual explanations. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Suzhou, China. Association for Computational Linguistics.

Verginica Barbu Mititelu, Voula Giouli, Georgiana Korvel, Chaya Liebeskind, Nino Lobzhanidze, Rusudan Makhachashvili, Stella Markantonatou, Maja Markovic, and Ivelina Stoyanova. 2025. Survey on lexical resources focused on multiword expressions for the purposes of NLP.

In *Proceedings of the 21st Workshop on Multiword Expressions (MWE 2025)*, pages 41–57, Albuquerque, New Mexico, USA. Association for Computational Linguistics.

Jan Odijk. 2013. Identification and lexical representation of multiword expressions. In Peter Spyns and Jan Odijk, editors, *Essential Speech and Language Technology for Dutch: Results by the STEVIN-programme*, pages 233–245. Springer, Berlin, Heidelberg.

Kemal Oflazer. 1994. Two-level description of Turkish morphology. *Literary and Linguistic Computing*, 9(2):137–148.

Kemal Oflazer, Özlem Çetinoğlu, and Bilge Say. 2004. Integrating morphology with multi-word expression processing in Turkish. In *Proceedings of the Workshop on Multiword Expressions: Integrating Processing*, pages 64–71, Barcelona, Spain. Association for Computational Linguistics.

Carlos Ramisch. 2015. *Multiword Expressions Acquisition: A Generic and Open Framework*. Springer.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In *Computational Linguistics and Intelligent Text Processing (CICLing)*.

Agata Savary, Carlos Ramisch, Silvio Cordeiro, Federico Sangati, Veronika Vincze, Behrang QasemiZadeh, Marie Candito, Fabienne Cap, Voula Giouli, Ivelina Stoyanova, and Antoine Doucet. 2017. The PARSEME shared task on automatic identification of verbal multiword expressions. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 31–47, Valencia, Spain. Association for Computational Linguistics.

Aygül Uçar. 2010. Light verb constructions in Turkish dictionaries: Are they submeanings of polysemous verbs? *Dil ve Edebiyat Dergisi / Journal of Linguistics and Literature*, 7(1):1–17.

Guanqun Yang, Mirazul Haque, Qiaochu Song, Wei Yang, and Xueqing Liu. 2022. TestAug: A framework for augmenting capability-based NLP tests. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3480–3495, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Raoyuan Zhao, Abdullatif Köksal, Yihong Liu, Leonie Weissweiler, Anna Korhonen, and Hinrich Schuetze. 2024. SynthEval: Hybrid behavioral testing of NLP models with synthetic CheckLists. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7017–7034, Miami, Florida, USA. Association for Computational Linguistics.

A Example Appendix

Sentence Turkish	Sentence English	Sentence Type
Kitapları rafa düzenli bir şekilde koydu.	He put the books on the shelf in an orderly manner.	NLVC Lexical Controls
Bu davranışıyla herkese karşı tavır koydu.	With this behavior, he put up an attitude against everyone.	LVC Positives
Kimse defterlerinde stok istemiyor.	No one wants stock in their books.	Random

Table 1: Structure of the test dataset and examples

B Zero-Shot Results

Model	Random	NLVC	LVC	Overall
Llama3.1-8B	0.980	0.959	0.000	0.646
GPT-OSS-20B	0.939	1.000	0.061	0.667
Qwen2.5-14B	0.918	0.857	0.122	0.633
BERTurk-32k (clf)	0.980	0.816	0.673	0.823
BERTurk-128k (clf)	0.980	0.816	0.796	0.864

Table 2: Experiment 1 success rates (0–1). LLM rows are zero-shot prompting; BERTurk rows are supervised classifier baselines (clf). Each condition has $n = 49$ items; Overall pools 147 items.

C One-Shot Results

Model	Random	NLVC	LVC	Overall
GPT-OSS-20B	0.898	0.735	0.837	0.823
Llama 3.1 8B	0.469	0.286	0.878	0.544
Qwen 2.5 14B	0.959	1.000	0.490	0.816

Table 3: Experiment 2 success rates (0–1). Each split has $n = 49$ items; Overall pools $N = 147$ items.

D Few-Shot Results

Model	Random	NLVC	LVC	Overall
GPT-OSS-20B	0.918	0.755	0.857	0.844
Llama 3.1 8B	0.510	0.612	0.878	0.667
Qwen 2.5 14B	0.878	0.980	0.714	0.857

Table 4: Experiment 3 success rates (0–1). Each condition has $n = 49$ items; Overall pools $N = 147$.