Understanding Compression Sensitivity in Kazakh Named Entity Recognition: The Roles of Entity Frequency and Subword Fragmentation

Dilnaz Nurumbayeva
Supervisor: Akmaral Kuatbayeva, Assistant Professor, Astana IT University

August 13, 2026

Abstract

This paper explores the use of techniques such as knowledge distillation, pruning, and architectural compression as an approach towards significantly reducing computational expenses associated with NLP models while maintaining their overall accuracy. However, average statistics do not necessarily reflect cases when compressed models fail. Previous studies conducted in the field of computer vision show that compression is likely to have a disparate effect on the examples that belong to the long tail of data distribution (Hooker et al., 2019). However, it is unclear whether this trend occurs in the case of named entity recognition (NER) tasks in low-resource morphologically rich languages.

For this purpose, the Kazakh Named Entity Recognition Dataset (KazNERD; Yeshpanov et al., 2022) is used to compare the performance of compressed models for the Kazakh NER task, starting from a baseline transformer model and finishing with a light-weight BiLSTM-CRF architecture, in terms of F1-score, number of parameters, inference time and memory usage. Based on the obtained benchmark results, two hypotheses are tested: (H1) compression has a disparate effect on rare entity classes, and (H2) entities that are fragmented into multiple subword tokens are more sensitive to compression. Hypothesis H2 incorporates previous findings related to compression bias (Hooker et al., 2019) and the role of subword fragmentation as a predictor of poor performance in underrepresented languages (Rust et al., 2021).

For 24 classes with sufficient representation, Hypothesis 1 (H1) is supported: class frequency is significantly negatively associated with compression sensitivity, both for the distilled model ($r = -0.43$, $p = .034$) and for the BiLSTM-CRF model ($r = -0.58$, $p = .003$). Hypothesis 2 (H2) is not supported both for the distilled model ($r = 0.05$, $p = .82$) and for the BiLSTM-CRF model ($r = 0.18$, $p = .40$). Frequency remains a significant predictor of compression sensitivity even when both predictors are considered simultaneously in a multiple regression, while fertility does not. The key contribution of this study consists in uncovering the Kazakh NER classes susceptible to model compression and demonstrating that frequency rather than subword fragmentation is the main factor behind their susceptibility.

Keywords: named entity recognition, model compression, low-resource NLP, Kazakh, subword tokenization, long-tail degradation

1 Definitions

- **Named Entity Recognition (NER):** A computational task of identifying and classifying stretches of text into predefined semantic types, such as person, organization, event, etc. (Li et al., 2020).
- **Model Compression:** Methods, including knowledge distillation, pruning, quantization, and architecture simplification, that reduce the number of parameters or computational cost of a model while trying to retain its task performance (Sanh et al., 2019).
- **Long-Tail Degradation:** The issue noted by Hooker et al. (2019) where compressed models perform comparably to their non-compressed counterparts in terms of aggregate accuracy, yet fail disproportionately on an under-represented set of cases, known as Pruning Identified Exemplars (PIEs).

- **Compression Sensitivity:** In this paper, the proportional drop in F1-score per class for a given named entity type as a function of model compression; used as the dependent variable in Hypotheses 1 & 2.
- **Tokenizer fertility/fragmentation:** The mean number of subwords (e.g., WordPieces or BPEs) generated per word or span of entities by a tokenizer; higher fertility means more fragmentation (Rust et al., 2021).
- **Agglutinative morphology:** Morphological type, exemplified by the Kazakh language, in which grammatical meanings are expressed via adding morphemes sequentially to a word stem, resulting in complex and highly informative words.

2 Introduction

2.1 Background: Efficient NLP and Model Compression

The current state-of-the-art natural language processing (NLP) paradigm is transfer learning from pre-trained transformers, which have been successfully applied to most benchmark problems and produced significant accuracy improvements (Devlin et al., 2019). While these improvements are substantial, they come at the cost of increased computational overhead, which manifests itself through the requirement for expensive hardware and energy consumption during training and serving of the models, unevenly spread among different research organizations and institutions (Strubell et al., 2019). As NLP models move from benchmarks to production environments, this overhead turns from a purely theoretical concern into a practical limitation.

In order to resolve this problem, many efforts have been made in the field of efficient natural language processing and model compression with the aim of achieving high performance at minimal model sizes and computation times. Knowledge distillation, a technique in which a small-sized "student" model is taught to mimic a large "teacher" model, proved especially effective: according to Sanh et al. (2019), DistilBERT obtains 97% of BERT's performance with 40% fewer parameters and 60% less latency. This idea has since been extended to NER specifically. Zhou et al. (2024) distill large language models into a lightweight, open-domain NER model (UniversalNER), while Zaratiana et al. (2023) proposed GLiNER, a lightweight generalist NER model, seen as a lightweight counterpart of LLM prompting,confirming that lightweight NER is an active research direction, though neither work identifies which entity types a compressed model trades away. Moreover, a BiLSTM-CRF architecture (Lample et al., 2016), significantly simpler than a transformer, serves as an important baseline, as it achieves similar NER performance without any pretrained backbone. Thus, the space of model compression design is not limited to one method and the areas within this space where the model succeeds and fails for some task are empirical to a large extent.

2.2 Kazakh Named Entity Recognition

This problem is especially acute for languages with poor resources and rich morphology. Kazakh, an agglutinative Turkic language with over 13 million speakers, has long been overlooked in NLP research, as expected based on the general disparity in languages' technical development highlighted by Joshi et al. (2020). Recent studies confirm this disparity: according to Maxutov et al. (2024), modern language models perform significantly worse when working with Kazakh compared to other, more resourceful languages. In the case of model compression for high-resource languages, the focus is on efficiency, while Kazakh NLP technologies are needed precisely where GPU is unavailable (for instance, in local educational establishments and administrative organizations). In other words, the compression of models is mandatory for Kazakh.

The problem of lacking data for Kazakh NER was mitigated by the introduction of the KazNERD dataset created by Yeshpanov et al. (2022): 112,702 sentences with 136,333 entities belonging to 25 classes; transformer models achieve 97.22% exact match F1 there as a baseline. However, these researchers continue to build Kazakh NLP infrastructure with new datasets (open-domain question answering (Yeshpanov et al., 2024) and parallel machine translation corpora (Yeshpanov et al., 2024b)), though without addressing the compression problem. Thus, the questions mentioned above are not

addressed. Despite the fact that the creation of the KazNERD dataset is a great advance per se, it does not take into account the way performance changes during compression, as well as the effect on entity classes. This paper takes those results as a starting point: a compression benchmark for KazNERD is necessary but not sufficient, and serves here as infrastructure for the hypotheses that follow.

2.3 Research Gap

Outside natural language processing, the main theoretical examination of the hypothesis that compression is not a neutral technique with regard to accuracy comes from Hooker et al. (2019). The authors have demonstrated that by comparing models with significantly different numbers of parameters but similar overall accuracy measures, one can see that compression doesn't harm accuracy in a uniform way. Instead, it causes harm to a subset of underrepresented examples. In consequence, the task of analysis has been shifted from measuring the magnitude of accuracy loss to finding out the particular examples impacted and the reasons behind their susceptibility to compression. Later, this theory was empirically corroborated outside the realm of image classification. Holste et al. (2023) noticed that class frequency reliably predicted both the likelihood and the magnitude of degradation resulting from pruning, the mechanism considered in this study in relation to named entity recognition (NER). There is another, independent, line of inquiry addressing a separate but related vulnerability characteristic of multilingual and morphologically complex languages. Rust et al. (2021) found that the number of subwords generated from a token through tokenizer fertility is a reliable predictor of the performance of the model in underrepresented languages, which Petrov et al. (2023) have framed as the systemic effect of imposing the "tokenization tax" on non-Latin-script and morphologically rich languages, including Kazakh. Neither of these lines of research, taken separately, establishes whether the combination of these vulnerabilities would make subword-fragmented entities especially susceptible to compression.

There has been little research on the vulnerability of particular named entities in morphologically complex low-resource languages to model compression, and whether that risk is predictable from independently studied factors such as class frequency and subword fragmentation. This is the gap in the literature that this paper aims to fill.

2.4 The Present Study

The study is carried out in two clearly distinguishable stages. During the first stage, the engineering benchmark of KazNERD is set up, where a full-scale transformer-based model is compared against its increasingly compressed variants, a distilled transformer and a BiLSTM-CRF baseline without a pretrained backbone, regarding the following metrics: F1-score, number of parameters, inference time, and memory footprint. The benchmark is treated as crucial infrastructure and not scientific contribution here. In the second core stage, the benchmark is used to test two falsifiable hypotheses on the way the compression affects model performance:

- **H1:** The compression more adversely affects the recognition of rare entity classes than frequent ones, expanding the long-tail degradation problem (Hooker et al., 2019) from image classification to structured prediction in a low-resource language.
- **H2:** The compression more adversely affects the recognition of fragmented entities (those represented by multiple subword tokens) than unfragmented ones. In other words, this hypothesis links together two concepts, namely, compression bias (Hooker et al., 2019) and the effect of subword fragmentation (Rust et al., 2021).

Thus, through explicit separation of benchmarking experiments from hypothesis testing, theoretically-grounded diagnosis of which Kazakh named entities suffer most from compression is provided.

2.5 Novelty

In this study, the baseline, the distillation method, and the BiLSTM-CRF module follow conventional methods (Conneau et al., 2020; Sanh et al., 2019; Lample et al., 2016). The paper makes three major contributions. First, this work is the first to explore together and compare the accounts of long-tail compression-bias (Hooker et al., 2019; Holste et al., 2023) and tokenizer-fertility (Rust et al., 2021;

Petrov et al., 2023) in the domain of structured sequence labeling.Second, it evaluates two architecturally independent forms of capacity reduction (a same-family distilled transformer and a from-scratch, backbone-free BiLSTM-CRF) against each other, so a frequency effect found in only one model cannot be mistaken for a general compression effect. Recent NER-compression work such as UniversalNER (Zhou et al., 2024) and GLiNER (Zaratiana et al., 2023) reports a single compressed architecture without this cross-architecture, per-class comparison. Third, this analysis is done on Kazakh, an agglutinative low-resource language, for which recent research efforts (Yeshpanov et al., 2024; Maxutov et al., 2024) have increased the number of Kazakh NLP models, but not their safe compressibility.

3 Methodology

3.1 Dataset

The current research uses the KazNERD dataset (Yeshpanov et al., 2022): 112,702 sentences, 136,333 entities, and 25 classes. The original training/validation/test ratio (90,228 / 11,167 / 11,307 sentences) is kept to maintain compatibility with the 97.22% baseline. Frequencies of classes in the training set vary between 23,551 (CARDINAL) and 6 (NON_HUMAN). All classes with less than 30 instances in the training set are disregarded from hypothesis testing, since per-class F1 score becomes statistically insignificant in such settings.

3.2 Models

Three points along the compression axis are considered. (1) **Baseline**: `xlm-roberta-base` fine-tuned end-to-end (12 layers, approximately 278M parameters). (2) **Distilled**: a 6-layer student model based on each second teacher layer (1, 3, 5, 7, 9, 11), following Sanh et al. (2019), trained using the combined loss $\lambda \cdot \mathrm{CE} + (1-\lambda) \cdot T^2 \cdot \mathrm{KL}(\text{teacher} \| \text{student})$, where $\lambda = 0.5$ and $T = 2.0$. The student adopts the teacher's tokenizer, so subword tokenization is consistent across the baseline and the distilled model. Architecture is the only manipulated factor here. (3) **BiLSTM-CRF**: word embeddings are trained from scratch, concatenated with character-level BiLSTM, followed by the linear-chain CRF. Each model is trained with 3 random seeds.

3.3 Metrics

We provide metrics for entity-level F1 (exact span and entity type match, IOB2 scheme) both overall and per class for the held-out test set. Also, the number of parameters, latency during inference, and peak GPU memory during a single-sentence forward pass are measured for the engineering benchmark.

3.4 Predictors

Class frequency is defined as the logarithm of the number of samples in the training set for each class. Subword fertility is defined as the average number of subwords per entity span (using the `xlm-roberta-base` tokenizer) divided by the span size in words. Correlation between the two predictors is weak and not significant ($r = -0.15$, $p = .48$, $n = 24$ classes). Thus, H1 and H2 can be tested separately.

3.5 Hypothesis Testing

Per-class compression sensitivity is defined as the relative F1 drop with respect to baseline, $(F1_{\text{base}} - F1_{\text{compressed}})/F1_{\text{base}}$, for each individual compressed model. H1 is tested via the correlation between frequency and sensitivity; H2 is tested via the correlation between fertility and sensitivity. Due to the small number of preserved classes ($n \approx 24$), one corpus-wide test per hypothesis is used instead of separate per-class tests, which precludes multiple-comparison correction at the cost of power. As a robustness check, a multiple regression of sensitivity on both predictors jointly is also reported.

3.6 Training Details

In order to reproduce the experiments, each experiment used the same fixed hyperparameters for all models and all three random seeds. Baseline: optimizer AdamW, learning rate 2×10^{-5}, batch size 32, 6 epochs of training, 500 warmup steps, weight decay 0.001. Distillation: the same optimizer parameters

as in baseline, but learning rate 5×10^{-5} (initialization is less deep in this case, therefore, a little bigger learning rate is helpful), batch size 32, 6 epochs, 500 warmup steps, weight decay 0.001, $\lambda = 0.5$ and $T = 2.0$ as stated in Section 3.2. BiLSTM-CRF: optimizer Adam, learning rate 1×10^{-3}, batch size 32, 20 epochs (the model is trained from scratch and needs more epochs to converge compared to fine-tuning), gradient clipping norm 5.0, word embeddings dimension 100, character embeddings dimension 30, character-BiLSTM hidden dimension 25, word-BiLSTM hidden dimension 200 per direction. No hyperparameter tuning was performed. All values were fixed in advance rather than tuned on the validation set, so reported results reflect a single configuration per model.

4 Results

4.1 Engineering Benchmark

Table 1 shows the three models in terms of their overall test F1 and efficiency. The baseline model has an F1 score of 96.6%, which corresponds to the reported range for xlm-roberta-based models on KazNERD (Yeshpanov et al., 2022). The distilled model has an approximate F1 score of 95.6% (the support-weighted mean across the 24 classes retained for hypothesis testing, excluding NON_HUMAN) using only 6 out of the 12 transformer layers of the baseline model. Because this figure is computed on a 24-class subset rather than the full 25-class test set used for the baseline and BiLSTM-CRF, the 95.6% value is not directly comparable to the other two overall F1 scores and should be read as an approximation restricted to the classes used for hypothesis testing.

The number of parameters of the distilled model (235 million parameters) is quite modest in comparison with the number of parameters of the baseline (278 million parameters). The parameter ratio is 1.18×. The embedding table with 250,000 tokens of xlm-roberta-base has 192 million parameters out of 278 million parameters and does not reduce when the encoder layers are reduced. Thus, reducing the number of layers by half provides only slight reduction in the number of parameters even though half of the computation in the transformers is cut off. Hence, latency of inference becomes the more relevant efficiency metric: despite the modest 1.18× parameter reduction, the distilled model is roughly 1.87× faster per forward pass (8.9 ms vs. 4.8 ms, averaged over three random seeds and 100 runs per seed), confirming that halving the transformer layers delivers a real latency benefit that parameter count alone would understate.

The correlation between the parameter count and the latency is non-monotonic in the case of all three models. BiLSTM-CRF, with its 39× lower parameter count than the baseline, does not show a consistently lower latency in per-sentence processing (10.7-20.3 ms vs. 7.9-10.8 ms for the baseline). A possible reason behind this is the difference in the architecture itself, rather than just the parameter count: while the BiLSTM performs the computations recurrently, token-by-token, the self-attention mechanism of the transformer processes the whole sequence at once and CRF Viterbi decoding adds up some more sequential steps. The recurrence penalty makes the difference for GPU-based single-sentence inference in favor of fewer parameters irrelevant, even if those parameters are much smaller. Moreover, the BiLSTM-CRF model shows a higher variance in the latency value depending on the seed (one seed achieves almost twice the latency as the other two), most likely caused by temporary GPU contention rather than any intrinsic feature of the architecture. The values presented above are exact numbers, without any smoothing. In this way, this is another relevant point to make in reference to the discussion of how "fewer parameters serve as a proxy for efficiency" in Section 4.1: while lower parameters help in reducing latency in case of the distilled model (due to parallel computation of the transformer), it does not apply to BiLSTM-CRF (with its non-parallel computation), so efficiency measure depends on the context of deployment. GPU memory consumption, in contrast, correlates with parameter count as expected: BiLSTM-CRF takes 47-56 MB of it, which is about 18× less than the distilled model and over 20× less than the baseline.

4.2 H1: Class Frequency and Compression Sensitivity

Figure 1 shows compression sensitivity as a function of log class frequency for both compressed models, across the 24 classes retained for analysis. The relationship is negative and statistically significant for

Table 1: Engineering benchmark across the three models. Overall F1 for the baseline and BiLSTM-CRF is computed by seqeval strict-mode evaluation on the full test set. The distilled model's F1 is a support-weighted average over the 24 classes retained for hypothesis testing. Latency and peak memory are single-sentence GPU forward-pass measurements (min–max across 3 seeds, 100 runs each after 10 warmup runs for latency; peak allocated memory per seed for memory).

Model	Overall F1	Params	Latency (ms)	Peak Mem. (MB)	Backbone
Baseline (xlm-roberta-base)	96.6%	≈278M	7.9–10.8	1068.0–1068.7	pre-trained
Distilled (6-layer student)	≈95.6%	≈235M	4.3–5.8	906.4	pre-trained (student)
BiLSTM-CRF	90.6%	7.16M	10.7–20.3	46.6–55.7	none (from scratch)

both: for the distilled model, Pearson $r = -0.43$, $p = .034$; for BiLSTM-CRF, Pearson $r = -0.58$, $p = .003$ (Spearman $\rho = -0.70$, $p < .001$). Rarer classes lose proportionally more F1 under compression than frequent ones, for both compression methods. The effect is noticeably stronger for BiLSTM-CRF, the more aggressively compressed model, which is consistent with the long-tail degradation account: greater capacity reduction produces a larger frequency-linked gap.

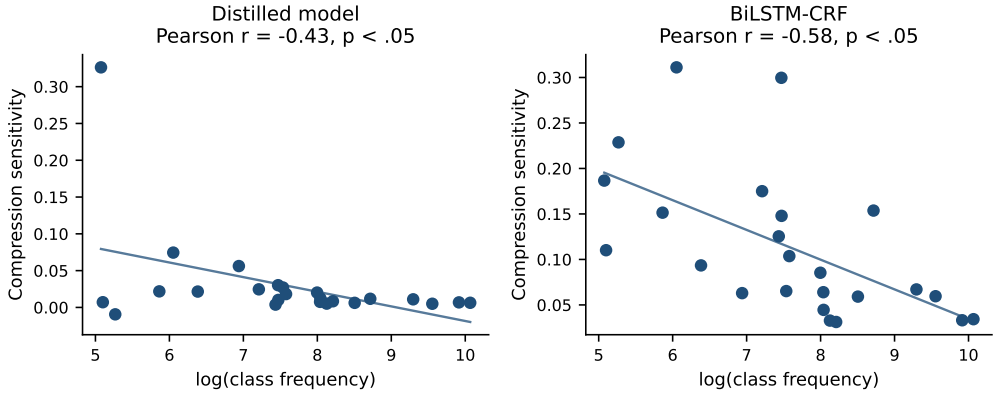


Figure 1: Compression sensitivity (relative F1 drop from baseline) as a function of log class frequency, for the distilled model (left) and BiLSTM-CRF (right). Each point is one of 24 entity classes with ≥30 training instances. Lines show ordinary least-squares fits.

4.3 H2: Subword Fertility and Compression Sensitivity

Figure 2 below shows the same relationship but with subword fertility replacing frequency as the predictor. For the distilled model, there is no correlation at all (Pearson $r = 0.05$, $p = .82$). In the BiLSTM-CRF model, again, there is no correlation between sensitivity and subword fertility by either metric (Pearson $r = 0.18$, $p = .40$; Spearman $\rho = 0.38$, $p = .070$). Hence, hypothesis H2 is not supported by either compressed model, adding more weight to the null hypothesis than a bivariate test on one architecture would have.

4.4 Multiple Regression

Table 2 shows the results of multiple regression analyses with frequency and fertility predicting sensitivity in both models. The regression coefficient for frequency remains negative and significant in the presence of fertility in both models, while the fertility coefficient is again small and insignificant. This implies that, within the dataset in question, the effects seen in bivariate hypothesis H1 can be attributed to frequency and are not just proxies for some other factor (note that the predictors are only weakly correlated with each other, as explained in Section 3.4).

4.5 A Qualitative Case: ADAGE

ADAGE is a perfect example for this pattern, being the most compression-sensitive class while undergoing distillation, losing 32.6% of its base F1 (53.9% to 36.3%), even though its average fertility (1.94 subwords per span) is one of the lowest in comparison with the other 24 classes (for instance, DISEASE's

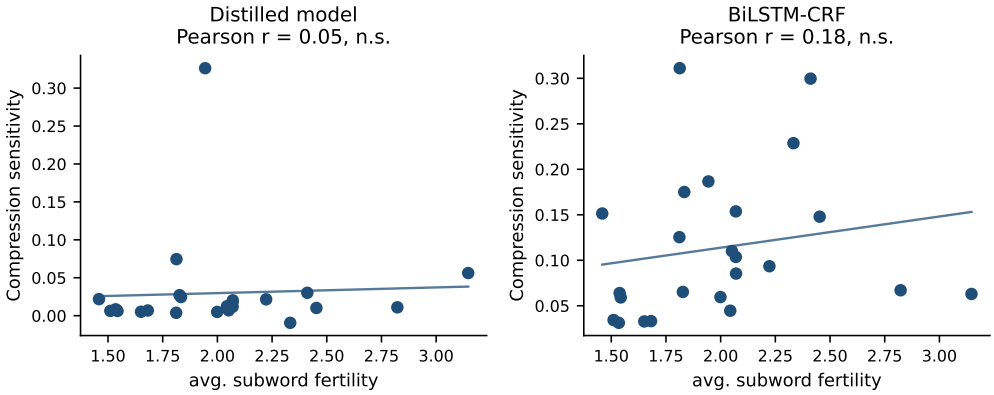


Figure 2: Compression sensitivity as a function of average subword fertility, for the distilled model (left) and BiLSTM-CRF (right).

Table 2: Multiple regression of compression sensitivity on log-frequency and fertility, jointly, for each compressed model ($n = 24$).

Model	Freq. coef.	p	Fertility coef.	p
Distilled	−0.0200	**.040**	−0.0027	.931
BiLSTM-CRF	−0.0319	**.004**	0.0180	.602

fertility is 3.15, yet it has lost far less F1 under compression). ADAGE is actually the second-rarest class kept in the analysis (n = 160 training instances). Thus, the result contradicts H2 and becomes a good example of H1 in action.

5 Discussion

These results confirm Hypothesis 1, whereas Hypothesis 2 is not supported in its initial formulation. Class frequency is a strong predictor of compression sensitivity, regardless of whether it concerns moderate model compression using distillation or aggressive compression through a different architecture (BiLSTM-CRF). This allows us to extend the general long-tail degradation framework developed by Hooker et al. (2019) and recently validated for medical imaging by Holste et al. (2023) from classification to structured and low-resource sequence labeling. We validate this generalization through two different compression methods, making our claim stronger in that it is about compression per se and not any specific method.

The strengthened H1 effect under BiLSTM-CRF compared to distillation is consistent with the idea that frequency-related degradation scales with the extent of model compression, though with only two compression points and no formal test of the difference between the two correlation coefficients, this should be read as suggestive rather than a confirmed dose-response relationship (see Limitations). Even so, there is a trade-off not visible from the average F1 score that concerns the balance between how much the model is compressed and how skewed the errors are distributed.

The null hypothesis for Hypothesis 2 is a meaningful outcome rather than lack of evidence. Previously, Rust et al. (2021) and Petrov et al. (2023) have shown fertility to be a predictor of performance across languages differing in fertility patterns. Our work, on the contrary, is concerned with a single language and asks whether fertility predicts entity performance within this language. A possible explanation is that the agglutinative nature of Kazakh makes all entities' spans break into multiple parts roughly equally across classes, thus decreasing fertility variation and leaving frequency, spanning over three orders of magnitude, as the main factor of variation. The ADAGE example in Section 4.5 shows how it works: despite being a class with a low fertility level, it suffers high compression sensitivity only because of its low frequency. Under BiLSTM-CRF compression, the classes most sensitive to compression (LAW and FACILITY) are intermediate on fertility.

6 Limitations

The current study identifies several limitations that have bearing on the results. This study relies on around 24 class-level data points, limiting statistical power both in general and for the near-significance effects. Two compressed models under consideration do not lie along a single "compression axis," as distillation is architecture-wise similar to the baseline, while BiLSTM-CRF is a completely different, backbone-less architecture. As such, the H1 result must be treated as two separate pieces of evidence rather than a dose-response curve. Latency estimates for BiLSTM-CRF were noticeably more sensitive to changes in random seeds compared to any transformer model (Section 4.1). This difference is likely driven by measurement error (e.g., temporary GPU load) rather than architecture features. Thus, the range of latencies should be interpreted accordingly. Fertility was estimated using one tokenizer (xlm-roberta-base) and will vary when other tokenizers are used. Lastly, the current study only tests a single dataset and a single language, making generalizability to other agglutinative or low-resource languages an open question.

7 Conclusion

In this paper, we introduce a three-point compression benchmark for Kazakh NER and use it to test two hypotheses about the reasons behind the uneven failures of compressed models across classes of entities. We find that class frequency significantly predicts sensitivity of compression for both distilled transformer and from-scratch BiLSTM-CRF, lending support to the notion that compression of models trained in low-resource, morphologically rich languages is associated with long tail degradation. At the same time, subword fragmentation does not predict sensitivity after controlling for frequency, refuting H2. From the practical standpoint, organizations that apply compressed Kazakh NER models should not rely solely on aggregate F1 measure: rare classes like legal terms, facilities, and events will suffer disproportionally from compression even in case of relatively good performance on aggregate F1. Future research should generalize this analysis to other low-resource morphologically rich languages and test whether the frequency-linked degradation effect can be mitigated with certain techniques.

Author Contributions

This is an individual-authored study. The research questions, hypotheses, study design, which includes predictors, compression-sensitivity measure, and pre-registered minimum support threshold, were developed by D.N. The author modified published code for the baseline fine-tuning model (Yeshpanov et al., 2022), programmed BiLSTM-CRF and knowledge distillation pipelines ab initio, ran all training, benchmarking, and statistical analyses, produced all figures and tables, and wrote the manuscript. This study was conducted under the guidance of Akmaral Kuatbayeva (Assistant Professor, Astana IT University).

References

1. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 8440–8451). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.747
2. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (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, Volume 1* (pp. 4171–4186). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423
3. Holste, G., Jiang, Z., Jaiswal, A., Hanna, M., Minkowitz, S., Legasto, A. C., Escalon, J. G., Steinberger, S., Bittman, M., Shen, T. C., et al. (2023). How does pruning impact long-tailed multi-label medical image classifiers? In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 663–673). Springer. https://arxiv.org/abs/2308.09180
4. Hooker, S., Courville, A., Clark, G., Dauphin, Y., & Frome, A. (2019). What do compressed deep neural networks forget? *arXiv preprint*. https://arxiv.org/abs/1911.05248
5. Joshi, P., Santy, S., Budhiraja, A., Bali, K., & Choudhury, M. (2020). The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*

(pp. 6282–6293). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.560
6. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 260–270). Association for Computational Linguistics. https://doi.org/10.18653/v1/N16-1030
7. Li, J., Sun, A., Han, J., & Li, C. (2020). A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering, 34*(1), 50–70. https://doi.org/10.1109/TKDE.2020.2981314
8. Maxutov, A., Myrzakhmet, A., & Braslavski, P. (2024). Do LLMs speak Kazakh? A pilot evaluation of seven models. In *Proceedings of the First Workshop on Natural Language Processing for Turkic Languages (SIGTURK 2024)* (pp. 81–91). Association for Computational Linguistics. https://aclanthology.org/2024.sigturk-1.8/
9. Petrov, A., La Malfa, E., Torr, P., & Bibi, A. (2023). Language model tokenizers introduce unfairness between languages. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*. https://arxiv.org/abs/2305.15425
10. Rust, P., Pfeiffer, J., Vulić, I., Ruder, S., & Gurevych, I. (2021). How good is your tokenizer? On the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Volume 1* (pp. 3118–3135). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.243
11. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint*. https://arxiv.org/abs/1910.01108
12. Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 3645–3650). Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1355
13. Yeshpanov, R., Khassanov, Y., & Varol, H. A. (2022). KazNERD: Kazakh named entity recognition dataset. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference* (pp. 417–426). European Language Resources Association. https://aclanthology.org/2022.lrec-1.44
14. Yeshpanov, R., Efimov, P., Boytsov, L., Shalkarbayuli, A., & Braslavski, P. (2024). KazQAD: Kazakh open-domain question answering dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)* (pp. 9645–9656). ELRA and ICCL. https://lrec.elra.info/lrec2024-main-0843
15. Yeshpanov, R., Polonskaya, A., & Varol, H. A. (2024b). KazParC: Kazakh parallel corpus for machine translation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)* (pp. 9633–9644). ELRA and ICCL. https://aclanthology.org/2024.lrec-main.842/
16. Zaratiana, U., Tomeh, N., Holat, P., & Charnois, T. (2023). GLiNER: Generalist model for named entity recognition using bidirectional transformer. *arXiv preprint*. https://arxiv.org/abs/2311.08526
17. Zhou, W., Zhang, S., Gu, Y., Chen, M., & Poon, H. (2024). UniversalNER: Targeted distillation from large language models for open named entity recognition. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR 2024)*. https://dblp.org/rec/conf/iclr/Zhou00CP24.html