

Against All Odds: Overcoming Typology, Script, and Language Confusion in Multilingual Embedding Inversion Attacks

Yiyi Chen, Russa Biswas, Heather Lent, Johannes Bjerva

Department of Computer Science, Aalborg University, Copenhagen, Denmark
yiyic, rubi, hcle, jbjerva@cs.aau.dk

Abstract

Large Language Models (LLMs) are susceptible to malicious influence by cyber attackers through intrusions such as adversarial, backdoor, and embedding inversion attacks. In response, the burgeoning field of LLM Security aims to study and defend against such threats. Thus far, the majority of works in this area have focused on monolingual English models; however, emerging research suggests that multilingual LLMs may be more vulnerable to various attacks than their monolingual counterparts. While previous work has investigated embedding inversion over a small subset of European languages, it is challenging to extrapolate these findings to languages from different linguistic families and with differing scripts. To this end, we explore the security of multilingual LLMs in the context of embedding inversion attacks and investigate cross-lingual and cross-script inversion across 20 languages, spanning over 8 language families and 12 scripts. Our findings indicate that languages written in Arabic and Cyrillic scripts are particularly vulnerable to embedding inversion, as are languages within the Indo-Aryan language family. We further observe that inversion models tend to suffer from language confusion, sometimes significantly reducing the efficacy of an attack. Accordingly, we systematically explore this bottleneck for inversion models, uncovering predictable patterns attackers could leverage. Ultimately, this study aims to further the field’s understanding of the outstanding security vulnerabilities facing multilingual LLMs and raise awareness for the languages most at risk of negative impact from these attacks.

Code — https://github.com/siebeniris/vec2text_exp

Models — <https://huggingface.co/yiyic>

Extended version — <https://arxiv.org/abs/2408.11749>

Introduction

Vector databases store texts encoded into *embeddings*, and represent a 1.5 billion USD market globally as of 2023, which is expected to grow to an impressive 4 billion within the next 5 years (Markets and Markets 2023). In more concrete contexts, services like Pinecone and Qdrant,¹ store embeddings for retrieval in applications like retrieval-augmented generation (RAG) (Lewis et al. 2020). In RAG,

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

¹<https://www.pinecone.io/> and <https://qdrant.tech/>

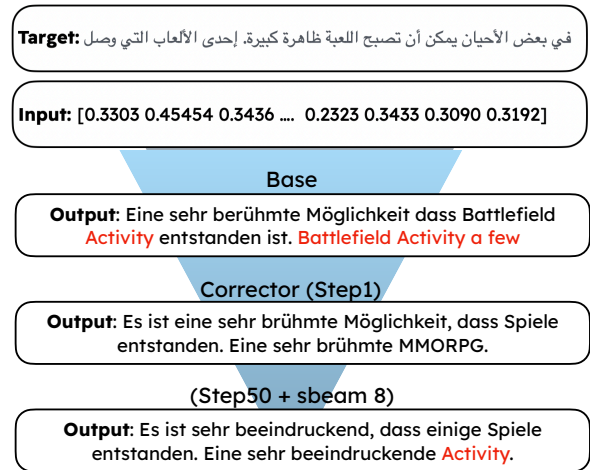


Figure 1: Inverting Arabic texts (target) using inversion models trained on multilingual encoder with German (source), showing **language confusion** across steps (red).

user queries are embedded via LLMs and sent to a vector database for semantic search, while large-scale embeddings of text data are also stored for semantic search. The security of this technology is increasingly critical for providers and customers depending on vector databases for their businesses. Presently, vector databases are known to have one outstanding security vulnerability: text embeddings can be reverted into original text, even without prior knowledge of the embedding model, thus constituting a major threat to privacy (Song and Raghunathan 2020; Höhmann, Rettinger, and Kugler 2021; Hayet, Yao, and Luo 2022; Morris et al. 2023; Li, Xu, and Song 2023; Chen, Lent, and Bjerva 2024). In other words, text embeddings are no more secure than plain text (Morris et al. 2023). A malicious actor needs only to intercept the vectors to carry out an embedding inversion attack, whereby potentially sensitive information is leaked.

As with most NLP problems, large English datasets have served as the primary focus of research on embedding inversion attacks. Emerging work, however, highlights how the field’s over-reliance on English comes at a grave cost to the general security of Large Language Models (LLMs). For ex-

ample, Yong, Menghini, and Bach (2024) demonstrate how ChatGPT’s built-in safety features can be easily by-passed by prompting in lower-resourced languages, while OpenAI’s defences function as expected for English and other highly-resourced languages. Meanwhile, Wang et al. (2024a) show how low-resource languages can be weaponized to poison models in backdoor attacks against machine translation systems. In the context of embedding inversion attacks, security may also be compromised by failing to consider a diversity of languages, especially as vector database providers may leverage multilingual LLMs to serve clients around the globe. Chen, Lent, and Bjerva (2024) find that multilingual LLMs are more vulnerable to embedding inversion attacks than their monolingual counterparts. However, their study includes only four highly-resourced European languages, restricted to Latin script. Conclusions drawn over a small set of relatively homogeneous languages cannot be trivially extended to massively multilingual LLMs, thus stressing the need for investigations of embedding inversion attacks with languages of diverse linguistic families and varying scripts.

In this work, we explore the nuances of embedding inversion attacks against multilingual models across 20 languages from 8 language families written in 12 scripts. We find that the success of inversion attacks is largely influenced by combinations of language family and script, ultimately showing that some languages are more vulnerable to attacks than others. Beyond elucidating threats facing multilingual encoders for vector databases, we analyze a common hindrance to the performance of embedding inversion: *language confusion*, a term recently coined by Marchisio et al. (2024), defined as a limitation of LLMs where the models are often unable to consistently generate text in the user’s desired language, or the appropriate language given the context. Fig. 1 illustrates different vulnerability levels in inversion attacks and language confusion. Specifically, Arabic textual embeddings (input) are inverted by a model trained on German embeddings, with the final output producing a mostly coherent German sentence containing an English term. Language confusion is understood as an “erroneous behavior” across the whole of NLP, reported in LLM instruction tuning and is detrimental to generation performance in terms of intelligibility in general (Marchisio et al. 2024). We show that language confusion can be predicted and controlled for, unfortunately to the benefit of would-be attackers.

Our contributions:

1. We explore the vulnerability of monolingual versus multilingual LLMs, including a wide variety of languages, and LLMs of differing architectures and training objectives, such that results can be better generalizable for modern massively multilingual LLMs.
2. We investigate embedding inversion attacks across 8 language families, comparing within and across-family and find that the Indo-Aryan family was the most vulnerable. Notably, inverting Punjabi texts produces meaningful sentences only in the in-family training setting compared to training it independently.
3. We examine attacks across 12 scripts, finding that scripts differ in vulnerability to inversion attacks. Our findings

indicate that Arabic and Cyrillic scripts were the most susceptible to attack. In-script training boosts Urdu inversion attack performance, doubling the BLEU score.

4. We identify language confusion as a bottleneck to cross-lingual inversion attacks. A systematic investigation uncovers that language confusion can be predicted using various language characteristics, such as script and script-directionality. Moreover, the prediction of language confusion can directly aid attack performance.

Related Work

Textual Embedding Inversion Inversion attacks on text embeddings have increased rapidly, with attack accuracy approaching 100% (Höhmman, Rettinger, and Kugler 2021; Hayet, Yao, and Luo 2022; Li, Xu, and Song 2023; Morris et al. 2023) whereas previously only 50–70% of tokens were achievable (Song and Raghunathan 2020). However, progress has mainly focused on English, leaving the security of multilingual embedding models largely unexplored. The first study to explore embedding inversion attacks in a multilingual context is limited to European languages and Latin script, highlighting a significant gap in LLM security (Chen, Lent, and Bjerva 2024). Tokenization methods in multilingual LLMs disproportionately allocate subwords to Latin script (Wu and Dredze 2020), suggesting that embedding inversion attack success may differ across languages depending on script and language. Finally, another notable limitation of the study by Chen, Lent, and Bjerva (2024) is the reliance on parallel data. As LLMs are known to memorize information (Shokri, Stronati, and Shmatikov 2016; Carlini et al. 2018; Nasr, Shokri, and Houmansadr 2019), studies on exclusively parallel text preclude the opportunity to study information leakage across languages (Ki, Park, and Kim 2024). As a consequence, the field’s threat assessment of multilingual LLM’s may not capture the full severity.

Cross-Lingual and Cross-Script Transfer The majority of the world’s languages lack sufficient data for NLP applications (Joshi et al. 2020). Cross-lingual transfer offers a solution to this issue, as models trained on resource-rich languages can be applied to low-resource languages (Conneau et al. 2020; Hu et al. 2020). While some studies suggest that cross-lingual transfer is more effective within the same language families, other work points to sub-word evenness (SuE) as a more critical factor in successful transfer (Pelloni, Shaitarova, and Samardzic 2022). Furthermore, transferring knowledge to languages with unknown scripts has proven exceedingly difficult (Anastasopoulos and Neubig 2019), prompting exploration of transliteration to mitigate script-related challenges (Hermjakob, May, and Knight 2018; Murikinati, Anastasopoulos, and Neubig 2020). The implications of cross-lingual and cross-script transfer for LLM security warrant further investigation.

Language Confusion Language confusion is a significant limitation of LLMs, where a model cannot consistently generate text in the desired language or appropriate language given the context (Marchisio et al. 2024). This can result in full-response confusion, line-level confusion, and word-level confusion (Vu et al. 2022; Marchisio et al. 2024). In

embedding inversion attacks, Chen, Lent, and Bjerva (2024) attributes language confusion to using a multilingual black-box encoder. However, in this work, we find that language confusion can occur when with *both* a multilingual and monolingual black-box encoder. Language confusion as a vulnerability is also supported by recent work, whereby it has already been proven an effective weapon for jailbreaking state-of-the-art LLMs (Song et al. 2024).

Methodology

In this work, we consider a scenario where a malicious attacker has illegitimate access to stolen embeddings and API access to the black-box encoder. To gauge the vulnerability of multilingual models against black-box inversion embedding attacks, we implement inversion attacks on a diverse setting of languages across language families and scripts. The attack scenario is formally defined as follows: given a text sequence x and a black-box encoder ϕ , an external inversion model ψ aims to recover the text x from its embeddings $\phi(x)$. The objective can be formalized as such that:

$$\psi(\phi(x)) \approx \phi^{-1}(\phi(x)) = x. \quad (1)$$

Since sentence embeddings result from pooling token representations, the mapping $\phi(x)$ is non-injective, making it impossible for the attacker model ψ to precisely invert ϕ (Li, Xu, and Song 2023). To learn the approximation of ψ , we learn a distribution $p(x|e; \theta)$ from a given dataset to invert $e = \phi(x)$ via maximum likelihood (Morris et al. 2023):

$$\theta = \arg \max_{\hat{\theta}} E_{x \sim D} [p(x | \phi(x); \hat{\theta})]. \quad (2)$$

This forms a text-to-text generation task, resulting in the trained **base model**. However, this approach alone is insufficient. To improve the inversion performance, a corrector model is implemented, where ϕ is queried using the output of the base model, and new *hypothesis embeddings* $\hat{e}^{(t)} = \phi(x^{(t)})$ is computed in iteration t with which a new correction text $x^{(t+1)}$ is generated. The corrector aims to find text \hat{x} that maximizes cosine similarity with the target embedding $\phi(x) : \hat{x} = \arg \max_x \cos(\phi(x), e)$. At step $(t+1)$, the model concatenates the previous output $\hat{x}^{(t)}$, hypothesis embeddings $\hat{e}^{(t)}$, and target embeddings e^t as in the equation (3) in the methodology section. The model is defined by marginalizing over intermediate hypotheses (Morris et al. 2023):

$$p(x^{(t+1)}|e) = \sum_{x^{(t)}} p(x^{(t)}|e)p(x^{(t+1)}|e, x^{(t)}, \hat{e}^{(t)}) \quad (3)$$

We train a **corrector** model (step1) where $x^{(1)}$ is the correction of $x^{(0)}$, and apply further steps with beam search. Results are reported for the base model, corrector (step1), and after 50 steps with beam search (width 8).

Language Models Previous studies have explored language models ϕ , both monolingual such as GTR and multilingual such as multilingual-e5 (ME5) but the backbone of ψ is the encoder-decoder T5 (Raffel et al. 2020), pre-trained on

four Germanic and Romance languages in Latin script, i.e., English, German, Spanish and French. To explore languages across diverse settings of scripts, we employ multilingual T5 (MT5) as the external encoder-decoder to train the inversion model, pre-trained in 102 languages, including the above languages except Meadow Mari. ME5 is also used as our multilingual language black-box encoder, to experiment across settings. The vulnerability between monolingual and multilingual language models is also compared by adopting GTR as ϕ to train the inversion model in German, ALEPH-BERT (Seker et al. 2022), Hebrew, and TEXT2VEC Chinese. This work uses LLMs sourced from Hugging Face (HF).²

Sentence Representations Prior research shows that BERT’s intermediate layers capture a hierarchy of linguistic information, with surface features in lower layers, syntactic features in the middle, and semantic features in the upper layers (Jawahar, Sagot, and Seddah 2019). Huang et al. (2021) find that averaging the first and last layers ($s^{[1,12]} = \frac{1}{2}(s^1 + s^{12})$) outperforms the [CLS] token and other layer combinations in downstream tasks like sentence semantics. We investigate the existing methods that use last hidden state, mean of all layers, and [CLS] in inversion tasks (Morris et al. 2023; Chen, Lent, and Bjerva 2024; Zhuang et al. 2024) and extend it with $s^{[1,12]}$, revealing that $s^{[1,12]}$ delivers superior attack performance, which is used for model training and evaluation in this paper.

Languages To explore the scenarios of multi-script and multi-family inversion attacks, we curate languages written in diverse scripts in several language families. Eventually, we simulate the inversion attacks in 20 languages across 8 families and 12 scripts. We train inversion models and evaluate languages and the combinations of which, including 12 languages **Arabic** (Semitic; Latn), **Urdu** (Indo-Aryan; Arab), **Kazakh** (Turkic; Cyril),³ **Mongolian** (Mongolic; Cyril), **Hindi** (Indo-Aryan; Deva), **Gujarati** (Indo-Aryan; Gujr), **Punjabi** (Indo-Aryan; Guru), **Chinese** (Sino-Tibetan; Hani), **Hebrew** (Semitic; Hebr), **Japanese** (Japonnic; Jpan), **German** (Germanic; Latn), and **Turkish** (Turkic; Latn). In addition, we evaluate on another languages **Amharic** (Semitic; Ethi), **Sinhala** (Indo-Aryan; Sinh), **Korean** (Koreanic; Hang), **Finnish** (Uralic; Latn), **Hungarian** (Uralic; Latn), **Yiddish** (Germanic; Hebr), **Maltese** (Semitic; Latn), and **Meadow Mari** (Uralic; Cyril).⁴

Dataset We randomly sample clean and deduplicated data from CulturaX (Nguyen et al. 2024), a trillion token dataset in 167 languages. Due to data size limitations for certain

²HF: sentence-transformers/gtr-t5-base, intfloat/multilingual-e5-base, google/mt5-base, imvladikon/sentence-transformers-alephbert, shibing624/text2vec-base-chinese-paraphrase

³Kazakh has historically been written in the Latin, Arabic, and Cyrillic scripts. Although the Kazakh government has announced plans to switch from Cyrillic to Latin script by 2031, we work with Kazakh written in Cyrillic, as this has been the predominant writing form over the last century.

⁴Language name (Language family, ISO 15924 script code). Full descriptions of languages are in SUPPL. doc.

languages, such as Indo-Aryan languages, i.e., Hindi, Gujarati, Punjabi and Urdu, we curate 600K samples for each language for training, while for other languages there are 1M samples for each train set, i.e., Arabic, Kazakh, Mongolian, Chinese, Hebrew, Japanese, German and Turkish. We consider 500 samples for each language in all the above-mentioned 20 languages for evaluation.

Experimental Setup Each inversion model consists of a base and a corrector, each trained for 100 epochs with a learning rate of $2e - 5$, epsilon of $1e - 6$, and 1000 warm-up steps on a constant schedule. Models use a batch size of 256 on data with 32 tokens and are trained on 4 AMD MI250 GPUs and 56 CPU cores using distributed training. The slowest model takes 4 days to complete in these settings. **[Baselines]** For each of the above-mentioned 12 languages, we train and evaluate inversion models using the corresponding data (ref. Table 1).

[In-Script] To explore inversion attacks within scripts, we train and evaluate on language pairs written in Arabic, Cyrillic, Latin and Hani-Jpan scripts⁵ (ref. Table 3).

[In-Family] Moreover, to better understand the transfer learning within language families in inversion attacks, we train languages in pairs from three language families, i.e., Semitic, Turkic and Indo-Aryan (ref. Table 4).

[Control] As a control group, we train the combinations of language pairs from Turkic and Indo-Aryan language families (ref. Table 5).

The same data samples used in the baseline models are applied to the in-script and in-family inversion model training. For the control group, we matched the number of samples to the smallest dataset.

Word Tokenization and Evaluation To measure the matching words between the target and inverted texts, we use word tokenizers according to the languages, Specifically, jieba⁶ (Chinese), Hebrew Tokenizer⁷ (Hebrew), Kipwipiepy (Lee 2024) (Korean), fugashi (McCann 2020) (Japanese), and NLTK Word Tokenizer (Bird and Loper 2004) for other languages. For evaluation, we use word-matching metrics such as BLEU (Post 2018), ROUGE (Lin 2004), and Token F1 (TF1), which calculates the multi-class F1 score between predicted and actual tokens. And, cosine similarity (COS) is employed to compare the true and hypothesis embeddings.

Language Confusion While previous work in language confusion is directed at prompting in diverse languages using instruct models (Marchisio et al. 2024), our experiment is in the setup, where an inversion model inverts the language embeddings in diverse languages extracted from a black-box encoder. Suppose an inversion model is trained on a set of $n \in \mathbb{N}$ languages $L_s = \{l_1, \dots, l_n\}$, and the text embeddings to be inverted are in a set of $m \in \mathbb{N}$ languages $L_t = \{l_1, \dots, l_m\}$. In the context of textual embedding inversion, we investigate language confusion in two settings:

⁵Due to the overlapping of Kanji characters between Chinese and Japanese, we experiment on this combination.

⁶<https://github.com/fxsjy/jieba>

⁷<https://github.com/YontiLevin/Hebrew-Tokenizer>

a) Monolingual generation, where the inversion model inverts the language embeddings in the languages included in training, i.e., $L_t \subseteq L_s$; **b) Cross-lingual generation**, where the languages of the target language embeddings are not included in the training data of the inversion model, i.e., $L_t \not\subseteq L_s$ and $L_t \cap L_s = \emptyset$.

To model and predict patterns in language confusion, we leverage the following aspects in the context of the inversion attacks: i) Stages of evaluating inversion attacks, i.e., base, corrector (step1) and (Step 50 + sbeam 8); ii) Language Characteristics, such as **Language Script (S)**, **Language Family (F)**, **Directionality of the Script (LR)**⁸, and **Word Order of Subject, Object and Verb(WO)** (Dryer and Haspelmath 2013); iii) COS; iv) Comparison of Languages in **Train Data (T)** and Evaluation Data with (i), (ii) and (iii).

As shown in Fig. 1, language confusion can happen at different levels, e.g., word level and line level. Because of loan words and code-switching as such, a mixture of languages can be contained in different levels of texts. To detect different levels of language confusion, we use off-the-shelf language identification (LID) tools.⁹ We define **Line-level Language Confusion** under the context of inversion security as the *probabilities* of the detected languages in which the inverted texts are decoded at line level, while **Word-level Language Confusion** is defined as such at word level. Using the language detector, we define the language space as the set including all the above-mentioned 20 languages and others (etc.). As illustrated in Fig. 2, using the inversion model trained on monolingual and multilingual embeddings with German to invert Chinese texts, during different stages, the embeddings in Chinese are inverted into texts in different proportions of English, German and other languages.

After removing missing values, we have 170 samples for monolingual LLMs as ϕ and 1789 samples for multilingual LLMs, across monolingual and cross-lingual generation settings. We split the data into 80% for training and 20% for testing and use Random Forest¹⁰ to train a regression model. This model predicts the probabilities of the languages in which the inverted embeddings are decoded based on the features described earlier (see details in SUPPL.). Our goal is to identify which features are more conducive to predicting language identities and to determine if there is a pattern between monolingual and multilingual LLMs.

Results on Inversion Attacks

Baselines We train and evaluate inversion models on ME5 embeddings across 12 languages. As detailed in Table 1, the performance of attacking textual embeddings improves progressively from the base model to the corrector model (Step 1), and further with 50 steps and a beam search width of 8. An exception is observed with inverting Punjabi text em-

⁸Directionality of scripts can include Left-to-Right (LTR), Right-to-Left (RTL), Top-to-Bottom and Bottom-to-Top. In this work, we focus on the scripts written in directions LTR and RTL.

⁹<https://github.com/pemistahl/lingua-py> and <https://github.com/zafercavdar/fasttext-langdetect>

¹⁰<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html>

Lang. (Script)	#Tok.	#Pred. Tok.	Tok. F1	BLEU	ROUGE	COS
Arabic (Arab)						
Base	13.48	13.27	40.06	17.31	92.93	0.9310
Corrector (Step1)	13.48	13.02	43.66	19.12	92.96	0.9292
(Step 50+sbeam8)	13.48	13.17	<u>48.18</u>	22.93	94.15	0.9232
Urdu (Arab)						
Base	15.03	14.77	49.71	20.86	78.57	0.9612
Corrector (Step1)	15.03	14.72	51.64	22.76	79.27	0.9736
(Step 50+sbeam8)	15.03	14.85	<u>54.66</u>	25.04	80.19	0.9712
Kazakh (Cyril)						
Base	13.35	13.16	53.12	30.69	86.39	0.9712
Corrector (Step1)	13.35	13.12	57.46	34.62	87.29	0.9794
(Step 50+sbeam8)	13.35	13.21	<u>65.74</u>	44.50	89.44	0.9860
Mongolian (Cyril)						
Base	12.00	11.74	50.01	28.21	82.36	0.9818
Corrector (Step1)	12.00	11.70	55.36	32.96	85.00	0.9523
(Step 50+sbeam8)	12.00	11.85	<u>64.67</u>	43.63	88.58	0.9742
Hindi (Deva)						
Base	15.21	14.98	50.98	19.10	73.40	0.9576
Corrector (Step1)	15.21	14.77	52.78	20.06	74.25	0.9443
(Step 50+sbeam8)	15.21	14.77	<u>55.79</u>	22.29	76.00	0.9197
Gujarati (Gujr)						
Base	11.07	10.85	43.19	18.74	87.57	0.9787
Corrector (Step1)	11.07	10.71	45.51	20.15	87.65	0.9856
(Step 50+sbeam8)	11.07	10.81	<u>48.17</u>	22.35	88.43	0.9893
Punjabi (Guru)						
Base	11.07	10.97	<u>59.65</u>	31.10	91.39	0.9882
Corrector (Step1)	11.07	21	0.03	0.43	0	0.7355
(Step 50+sbeam8)	11.07	21.00	0.03	0	0	0.7190
Chinese (Hani)						
Base	5.81	6.06	27.84	16.73	81.82	0.9483
Corrector (Step1)	5.81	5.88	28.33	17.12	82.29	0.9701
(Step 50+sbeam8)	5.81	5.74	<u>29.2</u>	17.50	83.35	0.9543
Hebrew (Hebr)						
Base	13.94	13.67	38.89	14.84	91.63	0.9646
Corrector (Step1)	13.94	13.50	41.81	16.32	92.30	0.9588
(Step 50+sbeam8)	13.94	13.68	<u>45.1</u>	18.77	93.04	0.9477
Japanese (Jpan)						
Base	2.93	3.03	9.47	9.31	80.08	0.9496
Corrector (Step1)	2.93	2.94	9.46	8.93	80.41	0.9576
(Step 50+sbeam8)	2.93	2.87	<u>9.61</u>	8.60	81.66	0.9630
German (Latn)						
Base	17.26	16.76	55.46	25.88	57.22	0.9460
Corrector (Step1)	17.26	16.85	56.69	27.35	58.78	0.9563
(Step 50+sbeam8)	17.26	16.80	<u>61.85</u>	31.35	64.62	0.9982
Turkish (Latn)						
Base	14.66	14.36	44.05	18.15	49.40	0.9498
Corrector (Step1)	14.66	14.26	46.33	20.62	52.51	0.9650
(Step 50+sbeam8)	14.66	14.30	<u>50.38</u>	24.51	57.51	0.9309

Table 1: Text Reconstruction in Multiple Languages, sorted by the alphabetic order of language scripts. For each language, the best performing TF1 and COS are underlined, and the best BLEU is **bolded**.

beddings, which achieve the highest performance in BLEU and TF1 at the base model but deteriorate significantly when using the corrector. Overall, multilingual embeddings are more susceptible to attacks in languages written in Cyrillic compared to those in Arabic, Latin, or Hebrew scripts, particularly in Haqniqdoq and Japanese scripts. Despite being trained on a smaller dataset, Indo-Aryan languages are more vulnerable than Chinese, Japanese, and Hebrew. This finding aligns with previous work by Wang et al. (2024b), which found that Hindi and Bengali suffered from more safety issues than other languages, in the context of adversarial attacks.

Monolingual vs. Multilingual LLMs To compare the vulnerability of monolingual embeddings and multilingual embeddings, we train and evaluate German, Hebrew, and

Lang. (Script)	Tok. F1		BLEU		COS	
	MONO	MULTI	MONO	MULTI	MONO	MULTI
Chinese (Hani)						
Base	21.87	27.84	6.53	16.73	0.8723	0.9484
Corrector (Step 1)	22.05	28.33	6.19	17.12	0.8548	0.9701
(Step 50+sbeam 8)	21.75	29.2	6.01	17.50	0.8576	0.9543
Hebrew (Hebr)						
Base	34.17	38.89	8.99	14.84	0.8601	0.9646
Corrector (Step 1)	36.27	41.81	9.95	16.32	0.8773	0.9588
(Step 50+sbeam 8)	37.98	45.1	10.56	18.77	0.9018	0.9477
German (Latn)						
Base	39.86	55.46	15.07	25.88	0.7985	0.9460
Corrector (Step 1)	39.83	56.69	14.52	27.35	0.9083	0.9563
(Step 50+sbeam 8)	42.17	61.85	15.54	31.35	0.9718	0.9982

Table 2: MONO and MULTI evaluate Text Reconstruction with monolingual embeddings and multilingual embeddings respectively. The best results for metrics are in **bold**.

Chinese on both monolingual (MONO) and multilingual (MULTI) LLMs. While in Chen, Lent, and Bjerva (2024), with the same settings of MONO and MULTI, trained on English samples, the best reconstruction performance on MONO excels MULTI by 14.33% in BLEU and by 5.81% in TF1. While trained on German samples, inverting MONO embeddings under-performs across the metrics compared to MULTI, as shown in Table 2. Overall, while the performance improves steadily across languages across the phases of evaluations in both word matching metrics and cosine similarities, ME5 is more vulnerable compared to MONO LLMs with different architectures (see details of the LLMs in SUPPL.). Also consistent with the findings from Chen, Lent, and Bjerva (2024), the cosine similarities are uniformly higher for MULTI than MONO. Overall, German is more vulnerable in both monolingual and multilingual settings, compared to Chinese and Hebrew.

Attacking In-script Textual Embeddings Since cross-lingual transfer learning is pervasive in multilingual NLP, and previous work has only touched upon Latin script in text embedding inversion, we investigate whether incorporating multiple languages in the same script would boost the inversion attack performance. Although the Japanese writing script has a more diverse set of forms (i.e., kanji and kana) compared to Chinese, they share a fair amount of vocabulary in the same script, in kanji and hanzi. To minimize the influence of transfer learning due to shared language families, the languages in each script are selected from different language families. As shown in Table 3, we train the inversion model in Arabic script with Arabic and Urdu, Cyrillic with Kazakh and Mongolian and Latin with German and Turkic, in addition to the Hani-Jpan combination with Chinese and Japanese. Overall, the inversion attack performance improves across languages when they are trained together with the same-script languages across stages, except in rare cases such as inverting German and Turkish embeddings with the base model. Language embeddings in Cyrillic are the most vulnerable compared to other scripts while Japanese and Chinese appear to be less vulnerable than other evaluated languages. Notably, the performance of attacking Urdu text embeddings is boosted notably by 100% in BLEU

	Lang. (Script)	# Tok.	# Pred. Tok.	Tok. F1	BLEU	ROUGE	COS
Arabic Script	Arabic						
	Base	13.48	13.37	41.91↑	17.97↑	93.69	0.9416
	Corrector (Step1)	13.48	13.24	47.13↑	21.48↑	94.06	0.9332
	(Step 50+sbeam8)	13.48	13.30	<u>55.58↑</u>	28.73↑	94.68	0.9243
	Urdu						
	Base	15.03	14.93	61.39↑	31.05↑	88.61	0.9735
Corrector (Step1)	15.03	14.83	66.7↑	36.90↑	90.41	0.9618	
(Step 50+sbeam8)	15.03	14.90	<u>76.18↑</u>	50.13↑	92.80	0.9832	
Hani-Japan Script	Chinese						
	Base	5.81	6.08	28.79↑	17.10↑	82.52	0.9475
	Corrector (Step1)	5.81	5.90	30↑	18.01↑	84.82	0.9549
	(Step 50+sbeam8)	5.81	5.71	<u>31.66↑</u>	19.86↑	87.80	0.9476
	Japanese						
	Base	2.93	2.94	9.85↑	9.41↑	80.64	0.9535
Corrector (Step1)	2.93	2.84	10.63↑	8.99↑	83.23	0.9698	
(Step 50+sbeam8)	2.93	2.94	<u>11.58↑</u>	10.25↑	85.61	0.9666	
Cyrillic Script	Kazakh						
	Base	13.35	13.17	54.29↑	31.39↑	87.08	0.9682
	Corrector (Step1)	13.35	13.08	61↑	37.62↑	89.83	0.9836
	(Step 50+sbeam8)	13.35	13.21	<u>72.37↑</u>	52.44↑	92.58	0.9863
	Mongolian						
	Base	12.00	11.75	53.56↑	31.63↑	85.14	0.9716
Corrector (Step1)	12.00	11.67	59.74↑	37.47↑	87.15	0.9777	
(Step 50+sbeam8)	12.00	11.82	<u>71.96↑</u>	51.62↑	91.22	0.9985	
Latin Script	German						
	Base	17.26	16.90	53.56↓	24.20↓	55.35	0.9545
	Corrector (Step1)	17.26	17.00	58.2↑	27.44↑	60.23	0.9594
	(Step 50+sbeam8)	17.26	16.96	<u>68.42↑</u>	37.66↑	71.11	0.9958
	Turkish						
	Base	14.66	14.47	42.96↓	17.29↓	48.97	0.9595
Corrector (Step1)	14.66	14.30	50.08↑	23.36↑	56.26	0.9637	
(Step 50+sbeam8)	14.66	14.39	<u>59.68↑</u>	33.33↑	66.35	0.9680	

Table 3: Multilingual Text Reconstruction within Script. For each language, the best TF1 score is underlined, and the best BLEU score is **bolded**. The color box indicates the highest inversion boost compared to the baseline.

score and 39.37% in TF1 compared to the baseline. Arabic, Urdu, Kazakh and Turkish are further investigated in the scenarios of in-family inversion attacks. Overall, the in-script training improves performance more than in-family training, except for Turkish (ref. Table 3 and 4).

Attacking In-Family Text Embeddings In addition to script being a significant factor in enhancing inversion attack performance, we also investigate the impact of a shared language family in this context. We train and evaluate various language pairs selected from the Semitic, Turkic, and Indo-Aryan language families, as shown in Table 4. Overall, inversion attack performance in word-matching metrics improves for all languages across evaluation steps, with the exception of Gujarati at the corrector step1 and Hindi with the base model. Notably, the corrector model trained solely on Punjabi fails to produce meaningful output when inverting Punjabi text embeddings in the baseline scenario (ref. Table 1). However, when trained alongside other Indo-Aryan languages, the performance improves dramatically. For instance, when Punjabi is trained with Gujarati, the BLEU score for the corrector model at step 50 with a beam search width of 8 jumps from 0 to 52.89. Additionally, Turkish text embedding inversion performs better in in-family training compared to in-script training (ref. Table 3). Overall, training the embedding inversion model with languages from the same language family significantly enhances performance. This indicates the vulnerability of low-resource languages which are related to high-resource ones.

	Lang. (Script)	#Tok.	#Pred. Tok.	Tok. F1	BLEU	ROUGE	COS
Semitic	Arabic (Arab)						
	Base	13.48	13.34	42.75↑	18.56↑	93.64	0.9434
	Corrector (Step1)	13.48	13.14	46.46↑	20.97↑	93.97	0.9572
	(Step 50+sbeam8)	13.48	13.25	<u>52.27↑</u>	25.61↑	94.35	0.9140
	Hebrew (Hebr)						
	Base	13.94	13.65	39.88↑	15.02↑	93.36	0.9583
Corrector (Step1)	13.94	13.49	43.6↑	17.26↑	93.87	0.9707	
(Step 50+sbeam8)	13.94	13.57	<u>48.4↑</u>	20.30↑	94.06	0.9730	
Turkic	Kazakh (Cyril)						
	Base	13.35	13.14	55.07↑	31.62↑	87.83	0.9755
	Corrector (Step1)	13.35	13.20	61.07↑	37.85↑	89.82	0.9685
	(Step 50+sbeam8)	13.35	13.24	<u>69.71↑</u>	49.94↑	92.56	0.9815
	Turkish (Latn)						
	Base	14.66	14.41	45.32↑	19.31↑	51.52	0.9399
Corrector (Step1)	14.66	14.34	51.07↑	23.96↑	57.22	0.9597	
(Step 50+sbeam8)	14.66	14.41	<u>59.58↑</u>	33.62↑	66.10	0.9683	
Indo-Aryan	Hindi (Deva)						
	Base	15.21	15.02	51.07↑	19.47↑	73.95	0.9596
	Corrector (Step1)	15.21	14.88	54.97↑	22.72↑	75.65	0.9733
	(Step 50+sbeam8)	15.21	15.04	<u>62.12↑</u>	28.72↑	79.20	0.9728
	Gujarati (Gujr)						
	Base	11.07	10.77	43.91↑	18.99↑	88.82	0.9824
Corrector (Step1)	11.07	10.74	48.97↑	22.12↑	89.52	0.9870	
(Step 50+sbeam8)	11.07	10.89	<u>56.08↑</u>	29.54↑	91.42	0.9925	
Indo-Aryan	Hindi (Deva)						
	Base	15.21	15.03	53.21↑	20.62↑	74.83	0.9476
	Corrector (Step1)	15.21	14.91	57.07↑	23.73↑	77.00	0.9389
	(Step 50+sbeam8)	15.21	14.98	<u>62.31↑</u>	28.89↑	79.03	0.9115
	Punjabi (Guru)						
	Base	11.07	10.90	63.6↑	35.78↑	93.05	0.9872
Corrector (Step1)	11.07	10.87	68.1↑	40.61↑	93.80	0.9903	
(Step 50+sbeam8)	11.07	10.97	<u>75.37↑</u>	50.80↑	95.08	0.9967	
Indo-Aryan	Gujarati (Gujr)						
	Base	11.07	10.81	44.1↑	19.47↑	87.90	0.9824
	Corrector (Step1)	11.07	10.67	36.63↓	13.49↓	85.59	0.9838
	(Step 50+sbeam8)	11.07	10.84	<u>50.66↑</u>	24.22↑	89.74	0.9880
	Punjabi (Guru)						
	Base	11.07	11.00	63.43↑	34.64↑	93.21	0.9883
Corrector (Step1)	11.07	10.90	68.56↑	41.07↑	94.39	0.9887	
(Step 50+sbeam8)	11.07	10.99	<u>76.51↑</u>	52.89↑	95.43	0.9957	
Indo-Aryan	Gujarati (Gujr)						
	Base	11.07	10.79	45.42↑	19.88↑	87.62	0.9763
	Corrector (Step1)	11.07	10.72	49.72↑	23.27↑	88.52	0.9850
	(Step 50+sbeam8)	11.07	10.83	<u>56.89↑</u>	30.34↑	90.55	0.9949
	Urdu (Arab)						
	Base	15.03	14.84	54.63↑	24.23↑	83.28	0.9621
Corrector (Step1)	15.03	14.78	57.69↑	27.35↑	85.43	0.9709	
(Step 50+sbeam8)	15.03	14.84	<u>62.8↑</u>	32.99↑	87.40	0.9824	
Indo-Aryan	Hindi (Deva)						
	Base	15.21	15.01	51.43↑	18.84↓	73.20	0.9427
	Corrector (Step1)	15.21	14.88	55.22↑	21.57↑	75.84	0.9772
	(Step 50+sbeam8)	15.21	14.98	<u>61.21↑</u>	27.93↑	78.82	0.9771
	Urdu (Arab)						
	Base	15.03	14.75	53.32↑	23.87↑	83.86	0.9728
Corrector (Step1)	15.03	14.87	57.19↑	26.84↑	85.60	0.9683	
(Step 50+sbeam8)	15.03	15.01	<u>63.15↑</u>	32.81↑	87.79	0.9831	
Indo-Aryan	Punjabi (Guru)						
	Base	11.07	10.89	60.99↑	33.19↑	91.76	0.9878
	Corrector (Step1)	11.07	10.62	38.94↑	13.98↑	88.27	0.9739
	(Step 50+sbeam8)	11.07	10.89	<u>59.99↑</u>	30.86↑	92.18	0.9878
	Urdu (Arab)						
	Base	15.03	14.77	52.14↑	22.88↑	81.98	0.9689
Corrector (Step1)	15.03	14.72	55.6↑	25.54↑	84.85	0.9447	
(Step 50+sbeam8)	15.03	14.79	<u>61.6↑</u>	31.39↑	86.67	0.9810	

Table 4: Text Reconstruction within Language Family. For each language, the best TF1 is underlined, and the best BLEU is **bolded**. The color box indicates the highest inversion boost compared to the baseline.

Control Group in Text Embedding Inversion To rule out the possibility that performance gains are merely due to an increased amount of training data compared to baseline models, we conduct experiments with a control group where languages from the Indo-Aryan and Turkic families are mixed. As shown in Table 5, when paired with Kazakh, the inversion performance of Indo-Aryan languages improves compared to the baseline, with the exception of Urdu when using corrector models. However, the improvement is not as

	Lang.(Script)	#Tok.	#Pred. Tok.	Tok. F1	BLEU	ROUGE	COS
Gujarati-Kazakh	Gujarati (Gujr)						
	Base	11.07	10.82	45.34 \uparrow	19.75 \uparrow	88.21	0.9854
	Corrector(Step1)	11.07	10.68	49.03 \uparrow	22.94 \uparrow	90.25	0.9804
	(Step 50+sbeam8)	11.07	10.77	<u>54.98\uparrow</u>	29.79\uparrow	91.38	0.9809
	Kazakh (Cyril)						
	Base	13.35	13.21	49.34 \downarrow	26.08 \downarrow	85.22	0.9767
Corrector(Step1)	13.35	13.09	53.47 \downarrow	30.31 \downarrow	87.05	0.9800	
(Step 50+sbeam8)	13.35	13.14	<u>60.4\downarrow</u>	38.20\downarrow	89.89	0.9841	
Hindi-Kazakh	Hindi (Deva)						
	Base	15.21	14.97	52.16 \downarrow	19.58 \downarrow	74.22	0.9441
	Corrector(Step1)	15.21	14.74	44.82 \downarrow	15.00 \downarrow	68.00	0.9584
	(Step 50+sbeam8)	15.21	15.06	<u>56.96\downarrow</u>	23.99\uparrow	76.15	0.9502
	Kazakh (Cyril)						
	Base	13.35	13.10	48.01 \downarrow	25.31 \downarrow	84.23	0.9563
Corrector(Step1)	13.35	13.02	52.68 \downarrow	30.18 \downarrow	87.54	0.9729	
(Step 50+sbeam8)	13.35	13.10	<u>59.38\downarrow</u>	36.57\downarrow	89.07	0.9635	
Kazakh-Punjabi	Kazakh (Cyril)						
	Base	13.35	13.06	48.34 \downarrow	25.60 \downarrow	84.63	0.9609
	Corrector(Step1)	13.35	13.09	51.49 \downarrow	28.79 \downarrow	84.97	0.9662
	(Step 50+sbeam8)	13.35	13.04	<u>56.8\downarrow</u>	34.49\downarrow	86.12	0.9824
	Punjabi (Guru)						
	Base	11.07	11.04	63.44 \uparrow	34.30 \uparrow	92.94	0.9880
Corrector(Step1)	11.07	10.93	65.92 \uparrow	37.28 \uparrow	93.89	0.9901	
(Step 50+sbeam8)	11.07	10.95	<u>71.67\uparrow</u>	46.65\uparrow	95.19	0.9942	
Kazakh-Urdu	Kazakh (Cyril)						
	Base	13.35	13.11	46.87 \downarrow	24.84 \downarrow	84.18	0.9619
	Corrector(Step1)	13.35	21.00	0.2 \downarrow	0.03 \downarrow	0.00	0.7604
	(Step 50+sbeam8)	13.35	21.00	<u>0.2\downarrow</u>	0.03\downarrow	0	0.7528
	Urdu (Arab)						
	Base	15.03	14.81	<u>52.35\uparrow</u>	22.58\uparrow	83.44	0.9657
Corrector(Step1)	15.03	21.00	0.08 \downarrow	0.01 \downarrow	0	0.7687	
(Step 50+sbeam8)	15.03	21.00	<u>0.08\downarrow</u>	0.01 \downarrow	0	0.7821	
Gujarati-Turkish	Gujarati (Gujr)						
	Base	11.07	10.75	42.64 \downarrow	18.84 \uparrow	87.55	0.9838
	Corrector(Step1)	11.07	10.74	47.51 \uparrow	21.70 \uparrow	89.16	0.9849
	(Step 50+sbeam8)	11.07	10.86	<u>54.49\uparrow</u>	28.19\uparrow	91.61	0.9892
	Turkish (Latn)						
	Base	14.66	14.42	41.08 \downarrow	16.10 \downarrow	46.29	0.9437
Corrector(Step1)	14.66	14.37	45.07 \downarrow	19.37 \downarrow	51.09	0.9657	
(Step 50+sbeam8)	14.66	14.44	<u>50.17\downarrow</u>	24.29\downarrow	57.46	0.9593	
Hindi-Turkish	Hindi (Deva)						
	Base	15.21	15.00	53.57 \uparrow	20.65 \uparrow	76.03	0.9518
	Corrector(Step1)	15.21	14.88	55.97 \uparrow	23.05 \uparrow	77.09	0.9743
	(Step 50+sbeam8)	15.21	15.00	<u>61.34\uparrow</u>	28.59\uparrow	80.74	0.9917
	Turkish (Latn)						
	Base	14.66	14.49	42.85 \downarrow	17.11 \downarrow	48.13	0.9413
Corrector(Step1)	14.66	14.26	44.69 \downarrow	19.50 \downarrow	51.37	0.9738	
(Step 50+sbeam8)	14.66	14.35	<u>51.08\downarrow</u>	25.58\downarrow	58.27	0.9498	
Punjabi-Turkish	Punjabi (Guru)						
	Base	11.07	10.91	<u>62.93\uparrow</u>	34.19\uparrow	92.68	0.9878
	Corrector(Step1)	11.07	10.56	38.58 \uparrow	13.69 \uparrow	88.89	0.9711
	(Step 50+sbeam8)	11.07	10.87	54.33 \uparrow	25.37 \uparrow	92.28	0.9904
	Turkish (Latn)						
	Base	14.66	14.28	41.97 \downarrow	16.75 \downarrow	47.29	0.9298
Corrector(Step1)	14.66	14.31	44.25 \downarrow	18.89 \downarrow	50.71	0.9445	
(Step 50+sbeam8)	14.66	14.45	<u>49.2\downarrow</u>	23.52\downarrow	56.18	0.9463	
Turkish-Urdu	Turkish (Latn)						
	Base	14.66	14.49	41.65 \downarrow	16.70 \downarrow	47.32	0.9373
	Corrector(Step1)	14.66	14.34	44.93 \downarrow	19.32 \downarrow	51.26	0.9631
	(Step 50+sbeam8)	14.66	14.39	<u>50.05\downarrow</u>	23.95\downarrow	56.61	0.9083
	Urdu (Arab)						
	Base	15.03	14.74	53.79 \uparrow	23.51 \uparrow	83.57	0.9574
Corrector(Step1)	15.03	14.70	56.03 \uparrow	26.27 \uparrow	85.51	0.9632	
(Step 50+sbeam8)	15.03	14.86	<u>61.6\uparrow</u>	31.77\uparrow	87.82	0.9841	

Table 5: Text Reconstruction in Control Group. For each language, the best TF1 is underlined, the best BLEU is **bolded**.

significant as that observed with in-family training (ref. Table 4). Conversely, the performance of inverting Kazakh text embeddings diminishes with these combinations. A similar trend is observed in inversion models trained on combinations of Turkish and Indo-Aryan languages, except when using the combination of Turkish and Hindi text embeddings. In this case, the reconstruction of Turkish texts also improves performance in word-matching metrics, possibly due to overlapping vocabulary between Hindi and Turkish (Singh 2024). Overall, this rules out the size of the training data as the decisive factor in enhancing inversion attack performance; instead, shared script and language family are

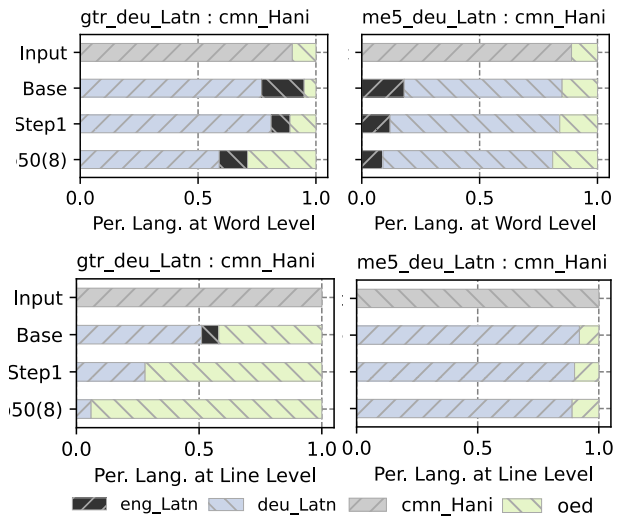


Figure 2: Language Confusion of reconstructing Chinese (Hani) texts using inversion models trained on monolingual (left) and multilingual (right) encoders with German (Latn) at Word Level (top) and Line Level (bottom).

necessary contributors.

Language Confusion Analysis

When GTR is used for the black-box encoder with German training data as depicted in Fig. 2 (left), the output languages at both word and line levels are a mix of German, English, and other languages, with proportions varying across different steps. However, with ME5 as the black-box encoder, the output is more likely to remain in the source language (i.e., German) at both levels. We also observe the target and hypothesis embeddings of inverted Chinese texts across different stages using inversion models trained on monolingual and multilingual embeddings in German. For multilingual embeddings, the target and hypothesis embeddings form distinct clusters, consistent with the findings that multilingual embeddings in different languages tend to occupy their respective subspaces. In contrast, for monolingual embeddings, the hypothesis embeddings are clustered together during inference steps, while at the base model stage, they are more sparsely distributed (see details in SUPPL.).

Patterns of Language Confusion This phenomenon of language confusion is pervasive and seems to be random at first. However, inversion attack results show there are strong correlations between the improved performance with training models using languages with shared scripts and language families. The language of training data, evaluation data, and different evaluation stages may serve as baseline variables for predicting language patterns, acting as control variables for the baseline model. In addition, shared language family and scripts and also the cosine similarity of the target and hypothesis embeddings play a role in language confusion. We use the combinations of COS, F, S, LR, LR(T), WO plus the variables from baseline, to predict the language of the inversion output using Random Forest Regression, with Mean

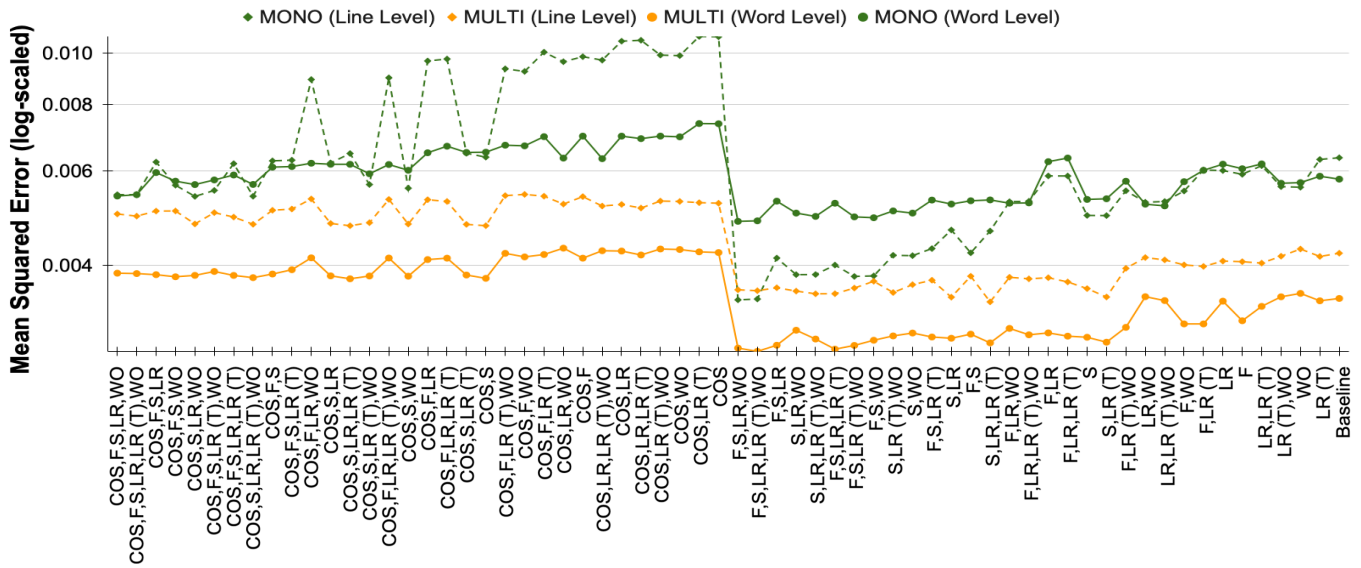


Figure 3: Language Confusion in inverting **Monolingual (MONO)** and **Multilingual (MULTI)** textual embeddings with Control Variables (X-axis) from Linguistic Characteristics.

Squared Error (MSE) as evaluation metrics. As shown in Fig. 3, all combinations of control variables with COS predict languages with larger error for both monolingual and multilingual LLMs and also for both word and line levels, while consistently, the features such as F, S, LR, LR(T) and WO renders better predictions across LLMs and levels.

Conclusion

Our work highlights the fact that vulnerabilities in multilingual embedding attacks disproportionately affect low-resource languages. This risk is exacerbated by the fact that language confusion, which can be exploited by attackers, is shown to be both predictable and solvable. We prove this by showing directly that attackers drastically increase the efficacy of their embedding inversion attacks, when specifically accounting for language confusion (i.e., by training on in-family and in-script languages), making it an appealing exploit for malicious actors to potentially leverage. In this context, we argue that a deeper understanding of language confusion, including its connection to linguistic characteristics, could offer critical insights into the mechanics of multilingual attacks. Ultimately, the security of LLMs and guarantees for safety for all end-users relies on the inclusion of diverse languages in LLM security research. Multilinguality poses unique challenges, and solutions offered for high-resource languages are not necessarily useful for those with fewer resources. Without prioritizing diversity in this emerging field, LLMs will remain vulnerable, with marginalized communities worldwide facing disproportionate risk as LLMs become more widespread.

Ethics Statement

This work investigates attacks on multilingual embedding models and the phenomenon of language confusion, with

the goal of highlighting vulnerabilities across languages representing diverse scripts and typologies. We encourage the community to incorporate a broader range of languages in NLP security research. Some languages we examine, such as Yiddish and Meadow Mari, are particularly vulnerable. To minimize potential harm, *we do not train any attacks* on these languages, instead focusing solely on evaluation to determine their current threat level. The language models used in this study are open-source, ensuring that this research does not pose an immediate threat to EaaS providers, who are likely using private models. Additionally, we do not experiment with sensitive data, ensuring that no real-world harm results from this work.

Computational Constraints Despite reducing the training data size for each inversion model by 60% to 80%, allowing us to investigate significantly more languages compared to previous studies (Morris et al. 2023; Chen, Lent, and Bjerva 2024), the computational overhead remains substantial. A key limitation of this work is the resource-intensive nature of the experiments, which required approximately 40,000 GPU hours. While expanding this research to include additional languages would further escalate computational costs, we strongly advocate prioritizing NLP security across diverse languages to ensure equitable advancements beyond English.

LLM’s Cross-lingual Vulnerabilities Defending multilingual LLMs has proven particularly challenging against attacks such as jailbreak (He et al. 2024), backdoor attacks (Wang et al. 2024a), and fine-tuning (Poppi et al. 2024). We echo these findings by examining multilingual LLMs in embedding inversion attacks across a linguistically diverse set of languages. A major finding of this work is that language confusion can be directly weaponized as an exploit against multilingual LLMs, and as such, we provide further

analysis to better understand the underlying patterns of language confusion. However, the trade-off between LLM vulnerabilities and LLM performance still needs to be investigated thoroughly in multilingual settings to devise effective defense mechanisms, which we leave for future work.

Acknowledgements

YC, HCL, and JB are funded by the Carlsberg Foundation, under the Semper Ardens: Accelerate programme (project nr. CF21-0454). We further acknowledge the support of the AAU AI Cloud and express our gratitude to DeiC for providing computing resources on the LUMI cluster (project no. DeiC-AAU-N5-2024085-H2-2024-28). We extend our thanks to Sighvatur Sveinn Davíðsson and Frederik Petri Svenningsen for their assistance with AI Cloud, and to Esther Ploeger and Mike Zhang for their discussions related to this work.

References

- Anastasopoulos, A.; and Neubig, G. 2019. Pushing the Limits of Low-Resource Morphological Inflection. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.
- Bird, S.; and Loper, E. 2004. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, 214–217. Barcelona, Spain: Association for Computational Linguistics.
- Carlini, N.; Liu, C.; Kos, J.; Erlingsson, Ú.; and Song, D. 2018. The Secret Sharer: Measuring Unintended Neural Network Memorization & Extracting Secrets. *CoRR*, abs/1802.08232.
- Chen, Y.; Lent, H.; and Bjerva, J. 2024. Text Embedding Inversion Security for Multilingual Language Models. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 7808–7827. Bangkok, Thailand: Association for Computational Linguistics.
- Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; and Stoyanov, V. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8440–8451. Online: Association for Computational Linguistics.
- Dryer, M. S.; and Haspelmath, M., eds. 2013. *WALS Online (v2020.3)*. Zenodo.
- Hayet, I.; Yao, Z.; and Luo, B. 2022. Invernet: An Inversion Attack Framework to Infer Fine-Tuning Datasets through Word Embeddings. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Findings of the Association for Computational Linguistics: EMNLP 2022*, 5009–5018. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.
- He, X.; Wang, J.; Xu, Q.; Minervini, P.; Stenetorp, P.; Rubinstein, B. I.; and Cohn, T. 2024. TuBA: Cross-Lingual Transferability of Backdoor Attacks in LLMs with Instruction Tuning. *arXiv preprint arXiv:2404.19597*.
- Hermjakob, U.; May, J.; and Knight, K. 2018. Out-of-the-box Universal Romanization Tool uroman. In Liu, F.; and Solorio, T., eds., *Proceedings of ACL 2018, System Demonstrations*, 13–18. Melbourne, Australia: Association for Computational Linguistics.
- Höbmann, J.; Rettinger, A.; and Kugler, K. 2021. In-vBERT: Text Reconstruction from Contextualized Embeddings used for Derived Text Formats of Literary Works. *CoRR*, abs/2109.10104.
- Hu, J.; Ruder, S.; Siddhant, A.; Neubig, G.; Firat, O.; and Johnson, M. 2020. XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalisation. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 4411–4421. PMLR.
- Huang, J.; Tang, D.; Zhong, W.; Lu, S.; Shou, L.; Gong, M.; Jiang, D.; and Duan, N. 2021. WhiteningBERT: An Easy Unsupervised Sentence Embedding Approach. In Moens, M.-F.; Huang, X.; Specia, L.; and Yih, S. W.-t., eds., *Findings of the Association for Computational Linguistics: EMNLP 2021*, 238–244. Punta Cana, Dominican Republic: Association for Computational Linguistics.
- Jawahar, G.; Sagot, B.; and Seddah, D. 2019. What Does BERT Learn about the Structure of Language? In Korhonen, A.; Traum, D.; and Màrquez, L., eds., *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3651–3657. Florence, Italy: Association for Computational Linguistics.
- Joshi, P.; Santy, S.; Budhiraja, A.; Bali, K.; and Choudhury, M. 2020. The State and Fate of Linguistic Diversity and Inclusion in the NLP World. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6282–6293. Online: Association for Computational Linguistics.
- Ki, D.; Park, C.; and Kim, H. 2024. Mitigating Semantic Leakage in Cross-lingual Embeddings via Orthogonality Constraint. In Zhao, C.; Mosbach, M.; Atanasova, P.; Goldfarb-Tarrent, S.; Hase, P.; Hosseini, A.; Elbayad, M.; Pezzelle, S.; and Mozes, M., eds., *Proceedings of the 9th Workshop on Representation Learning for NLP (ReplANLP-2024)*, 256–273. Bangkok, Thailand: Association for Computational Linguistics.
- Lee, M. 2024. Kiwi: Developing a Korean Morphological Analyzer Based on Statistical Language Models and Skip-Bigram. *Korean Journal of Digital Humanities*, 1(1).
- Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33: 9459–9474.

- Li, H.; Xu, M.; and Song, Y. 2023. Sentence Embedding Leaks More Information than You Expect: Generative Embedding Inversion Attack to Recover the Whole Sentence. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Findings of the Association for Computational Linguistics: ACL 2023*, 14022–14040. Toronto, Canada: Association for Computational Linguistics.
- Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 74–81. Barcelona, Spain: Association for Computational Linguistics.
- Marchisio, K.; Ko, W.-Y.; Bérard, A.; Dehaze, T.; and Ruder, S. 2024. Understanding and Mitigating Language Confusion in LLMs. *arXiv preprint arXiv:2406.20052*.
- Markets and Markets. 2023. Vector Database Market by Offering (Solutions and Services), Technology (NLP, Computer Vision, and Recommendation Systems), Vertical (Media & Entertainment, IT & ITes, Healthcare & Life Sciences) and Region - Global Forecast to 2028. *Top Market Reports*.
- McCann, P. 2020. fugashi, a Tool for Tokenizing Japanese in Python. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, 44–51. Online: Association for Computational Linguistics.
- Morris, J.; Kuleshov, V.; Shmatikov, V.; and Rush, A. 2023. Text Embeddings Reveal (Almost) As Much As Text. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.
- Murikinati, N.; Anastasopoulos, A.; and Neubig, G. 2020. Transliteration for Cross-Lingual Morphological Inflection. In Nicolai, G.; Gorman, K.; and Cotterell, R., eds., *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*.
- Nasr, M.; Shokri, R.; and Houmansadr, A. 2019. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE.
- Nguyen, T.; Nguyen, C. V.; Lai, V. D.; Man, H.; Ngo, N. T.; Derroncourt, F.; Rossi, R. A.; and Nguyen, T. H. 2024. CulturaX: A Cleaned, Enormous, and Multilingual Dataset for Large Language Models in 167 Languages. In Calzolari, N.; Kan, M.-Y.; Hoste, V.; Lenci, A.; Sakti, S.; and Xue, N., eds., *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 4226–4237. Torino, Italia: ELRA and ICCL.
- Pelloni, O.; Shaitarova, A.; and Samardzic, T. 2022. Subword Evenness (SuE) as a Predictor of Cross-lingual Transfer to Low-resource Languages. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 7428–7445. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.
- Poppi, S.; Yong, Z.-X.; He, Y.; Chern, B.; Zhao, H.; Yang, A.; and Chi, J. 2024. Towards Understanding the Fragility of Multilingual LLMs against Fine-Tuning Attacks. *arXiv preprint arXiv:2410.18210*.
- Post, M. 2018. A Call for Clarity in Reporting BLEU Scores. In Bojar, O.; Chatterjee, R.; Federmann, C.; Fishel, M.; Graham, Y.; Haddow, B.; Huck, M.; Yepes, A. J.; Koehn, P.; Monz, C.; Negri, M.; Nèveol, A.; Neves, M.; Post, M.; Specia, L.; Turchi, M.; and Verspoor, K., eds., *Proceedings of the Third Conference on Machine Translation: Research Papers*, 186–191. Brussels, Belgium: Association for Computational Linguistics.
- Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140): 1–67.
- Seker, A.; Bandel, E.; Bareket, D.; Brusilovsky, I.; Greenfield, R.; and Tsarfaty, R. 2022. AlephBERT: Language Model Pre-training and Evaluation from Sub-Word to Sentence Level. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- Shokri, R.; Stronati, M.; and Shmatikov, V. 2016. Membership Inference Attacks against Machine Learning Models. *CoRR*, abs/1610.05820.
- Singh, K. 2024. Contribution of Turkic Languages in the Evolution in and Development of Hindustani Languages. Accessed: 2024-08-16.
- Song, C.; and Raghunathan, A. 2020. Information Leakage in Embedding Models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, CCS '20*, 377–390. New York, NY, USA: Association for Computing Machinery. ISBN 9781450370899.
- Song, J.; Huang, Y.; Zhou, Z.; and Ma, L. 2024. Multilingual blending: Llm safety alignment evaluation with language mixture. *arXiv preprint arXiv:2407.07342*.
- Vu, T.; Barua, A.; Lester, B.; Cer, D.; Iyyer, M.; and Constant, N. 2022. Overcoming Catastrophic Forgetting in Zero-Shot Cross-Lingual Generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 9279–9300.
- Wang, J.; Xu, Q.; He, X.; Rubinstein, B. I. P.; and Cohn, T. 2024a. Backdoor Attack on Multilingual Machine Translation. *arXiv:2404.02393*.
- Wang, W.; Tu, Z.; Chen, C.; Yuan, Y.; Huang, J.-t.; Jiao, W.; and Lyu, M. 2024b. All Languages Matter: On the Multilingual Safety of LLMs. In Ku, L.-W.; Martins, A.; and Srikanth, V., eds., *Findings of the Association for Computational Linguistics ACL 2024*.
- Wu, S.; and Dredze, M. 2020. Are All Languages Created Equal in Multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, 120–130.
- Yong, Z.-X.; Menghini, C.; and Bach, S. H. 2024. Low-Resource Languages Jailbreak GPT-4. *arXiv:2310.02446*.
- Zhuang, S.; Koopman, B.; Chu, X.; and Zuccon, G. 2024. Understanding and Mitigating the Threat of Vec2Text to Dense Retrieval Systems. *arXiv preprint arXiv:2402.12784*.