“Newspaper Eat” Means “Not Tasty”: A Taxonomy and Benchmark for Coded Language in Real-World Chinese Online Reviews

Ruyuan Wan[1] **Changye Li**[2] **Ting-Hao ‘Kenneth’ Huang**[1]
[1]The Pennsylvania State University [2]University of Washington
[1]{rjw6289, txh710}@psu.edu [2]changyel@uw.edu

Abstract

Coded language is an important part of human communication. It refers to cases where users intentionally encode meaning so that the surface text differs from the intended meaning and must be decoded to be understood. Current language models handle coded language poorly. Progress has been limited by the lack of real-world datasets and clear taxonomies. This paper introduces **CodedLang**, a dataset of 7,744 Chinese Google Maps reviews, including **900 reviews with span-level annotations of coded language**. We developed a seven-class taxonomy that captures common encoding strategies, including phonetic, orthographic, and cross-lingual substitutions. We benchmarked language models on coded language detection, classification, and review rating prediction. Results show that even strong models can fail to identify or understand coded language. Because many coded expressions rely on pronunciation-based strategies, we further conducted a phonetic analysis of coded and decoded forms. Our code and dataset are publicly available [1]. Together, our results highlight coded language as an important and underexplored challenge for real-world NLP systems.

1 Introduction

Coded language is an important way people communicate with each other (Ji and Knight, 2018). Instead of expressing meaning directly, writers intentionally encode their intent so that the surface text differs from the intended meaning and must be decoded by an informed reader. Such practices are common in everyday communication, yet they remain understudied in natural language processing (NLP), where models typically assume that meaning is conveyed transparently in literal text.

Online review platforms provide a clear real-world context in which coded language emerges. While reviews influence consumer decisions and public reputation, writing negative reviews can expose users to social pressure, retaliation, or reduced visibility. Among Chinese-speaking users on Google Maps, we observed a recurring strategy: embedding negative evaluations inside seemingly positive or neutral reviews using coded language. These reviews rely on homophones, phonetic substitutions, transliteration, emoji, and cross-lingual cues that obscure meaning from casual readers, machine translation systems, and moderation tools, while remaining interpretable to in-group audiences. For example, a review stating “**Newspaper eat**” appears nonsensical in English but encodes a negative message through a phonetic substitution of the Chinese phrase “不好吃” (bu4 hao3 chi1, meaning **not tasty**). When spoken quickly, “bu4 hao3 chi1” can sound like “bao4 chi1” (报吃), which literally translates to “Newspaper Eat” in English, where “bao4” means newspaper and “chi1” means eat. Figure 1 shows another representative

Figure 1: Example of creative coded language in a Google Maps restaurant review. Top: the original user review in Chinese, as posted on Google Maps. Left: the machine-generated English translation produced by Google Translate. Right: the inferred underlying meaning obtained by decoding phonetic substitutions.

[1]https://github.com/Crowd-AI-Lab/CodedLang

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9433–9446
July 2-7, 2026 ©2026 Association for Computational Linguistics

example, where machine translation systems fail to recover the intended meaning. These examples illustrate how coded language creates a systematic gap between surface form and communicative intent that current NLP systems struggle to bridge.

To study this phenomenon, we curated **CODEDLANG**, a dataset of 7,744 real-world Chinese Google Maps reviews, including **900 reviews containing coded language with span-level annotations** (Section 3). We also developed a seven-class taxonomy that captures common encoding strategies, including homophonic, phonetic, and cross-lingual substitutions (Section 3.3). Using CODEDLANG, we evaluated modern language models on three NLP tasks (Section 4) and further analyzed the phonetic properties of coded expressions (Section 5). This work treats coded language as a systematic and understudied form of real-world communication and provides a grounded dataset and taxonomy for examining how NLP systems handle intentionally encoded meaning in practice.

2 Related Work

Natural Language Processing Methods for Coded Languages. Prior NLP work showed that coded language is hard to analyze due to its dependence on context, cultural knowledge, polysemy, and limited labeled data (Ji and Knight, 2018). Decoding could be further complicated in multilingual settings, where translation systems and multilingual language models misinterpret coded expressions due to misleading cross-language word similarity (Kallini et al., 2025). Open challenges included balancing opacity and interpretability, modeling personalized and multilingual coding strategies, and handling the rapid evolution of code phrases. While these challenges were known, prior work lacked task formulations and datasets grounded in real-world review platforms. Our work addressed this gap by formulating coded reviews as an NLP task and introducing CODEDLANG, a curated dataset of real-world coded reviews.

Existing studies focused on encoding and decoding mechanisms. Ji and Knight (2018) showed how users encoded meaning to bypass censorship while remaining intelligible to in-group audiences, supporting functions such as discussing sensitive topics and expressing negative opinions indirectly. Kim et al. (2021) found that Korean speakers on Google Maps used morphological, phonological, optical, and semantic strategies to make reviews intelligible only to in-group readers. Cho and Kim (2021) also categorized intentionally noisy text by interpretation boundaries. However, these studies did not capture how users systematically hide negative sentiment inside positive or neutral reviews, nor did they provide span-level annotations for this behavior. Coded language has also been studied in political discourse as dog whistles, which are expressions that appear innocuous to general audiences but convey specific meanings to targeted in-groups (Mendelsohn et al., 2023; Xu et al., 2021). This line of work focused on ideological signaling rather than everyday consumer communication and did not consider how coded language interacted with ratings or machine translation.

Phonetic and Orthographic Substitution. Early NLP work treated phonetic and orthographic variation as noise to be normalized (Saito et al., 2014). More recent studies have shown that such variations were deliberately used as coding strategies. In Chinese internet language, homophonic substitutions and character-level perturbations encode meaning implicitly and require phonological reasoning and shared background knowledge for interpretation (Xu et al., 2021; Ma et al., 2025b). Benchmarks showed that large language models (LLMs) struggle with these substitutions and often rely on memorized patterns rather than systematic reasoning (Ma et al., 2025b,a). Related work in content moderation also showed that homophones, emoji substitutions, and visually similar characters degraded toxic language detection performance (Xiao et al., 2024). Recent work also explored countermeasures, showing that simple LLM-based approaches could partially counter algospeak (Fillies and Paschke, 2024). However, this line of work treats surface manipulation as an attack rather than a communication strategy. Our work instead studies how real users use phonetic and orthographic substitutions to express negative evaluations while avoiding detection.

3 CODEDLANG Dataset and Taxonomy

We curated CODEDLANG, a dataset of 7,744 online reviews, of which 900 contain coded language. Each coded-language review includes span-level annotation(s) indicating the coding strategy, as defined by our taxonomy. This section overviews the construction and analysis of CODEDLANG.

We defined **coded language** as *the intentional encoding or obfuscation of meaning, in which the*

Coded Language Class	Definition	Example
Ambiguous Homophone	Existing lexical phrases whose surface form is a valid expression, but whose intended meaning differs from the literal meaning via homophones	“炒鸡”(chao3 ji1, fried chicken) = 超级 (chao1 ji2, super)
Non-Lexical Homophone	Homophonic characters that do not form a valid lexical expression on the surface, but can be decoded phonetically into a conventional phrase.	“灰常”(hui1 chang2, grey often) = 非常 (fei1 chang2, very)
Phonetic Substitution	Use of phonetic symbols, including Pinyin, Zhuyin, or Roman letters, to substitute for standard Chinese characters based on pronunciation.	“tm” = 他妈 (ta1 ma1, damn); “好ㄘ”(hao3 c, good) = 好吃 (hao3 chi1, tasty)
Emoji Substitution	Emoji replace textual content and convey an encoded meaning that differs from the literal emoji semantics.	“🌶️🐔” = 辣鸡 (la4 ji1, spicy chicken) = 垃圾 (la1 ji1, trash)
Orthographic Substitution	Encoding through visual or orthographic similarity between characters and symbols.	“口乞”(kou3 qi3, mouth begging) = 吃 (chi1, eat)
Cross-Lingual Phonetic Encoding	Encoding meaning through cross-lingual phonetic approximation, often involving transliteration or translation of homophones.	“古德(gu3 de2, old virtue)” = Good; “Newspaper eat” = 报吃 (bao4 chi1, newspaper eat) = 不好吃(bu4 hao3 chi1, not tasty)
Cipher	The whole sentence is a nonsensical expression without a clear literal semantic meaning.	“qqQQaQ！你燃yfyi y f r”

Table 1: Taxonomy of coded language classes with definitions and representative examples. Examples are presented as they appear in the original reviews, with the decoded form, Pinyin, and English meanings provided in parentheses. Cipher instances are not decoded, as they are not directly interpretable.

surface form differs from the intended meaning and decoding is required for interpretation. Importantly, not all socially opaque language qualifies as coded language. Dialect, jargon, and multilingual text can be socially opaque because they rely on shared community knowledge, but they are not considered coded language unless the meaning is intentionally disguised through an explicit encoding mechanism, such as phonetic, orthographic, or symbolic substitutions. In contrast, coded language is socially opaque because it is encoded: understanding it requires actively decoding the surface form to recover the intended meaning.

3.1 Data Sources and Pre-Processing

Initial Coded-Language Examples. We first observed news articles reporting Chinese-speaking users employing coded language to express dissatisfaction in Google Reviews for overseas businesses (Qing, 2024; Fan, 2024; Chen, 2025). These cases were originally shared on social media platforms such as Xiaohongshu (Rednote) and Threads, where users discussed the opacity and in-group interpretability of such reviews and shared additional examples they had encountered.[2] Therefore, we collected 23 reviews containing coded language that were shared on social media or reported in news articles. These reviews serve as an initial seed set, providing concrete examples of real-world coded expressions and guiding our large-scale mining of low-frequency coded reviews.

Real-World Google Maps Reviews. Building on this seed set, we further screened for coded-language instances in massive Google Maps reviews. We used two Google Maps review datasets: *(i)* 1.77 million restaurant reviews (Li et al., 2022) and *(ii)* 666 million local reviews (Yan et al., 2023). These reviews were multilingual and covered U.S. businesses, including restaurants and hotels. Each review included a user ID, a business ID, the original review text, ratings, and an English translation of the review when the original was not in English. We then selected reviews containing at least one Chinese character, which yielded 5,391 Chinese-language reviews from the restaurant review dataset and 112,521 from the local review dataset. This set included reviews written in Simplified and Traditional Chinese, with some mixed with English and other languages.[3] Section 3.2 described our

[2]For example, a Xiaohongshu post titled “This is the first time I’ve seen real Chinese encryption hahaha”, and a Threads post said “Saw a review that only Taiwanese people can understand, thank you, kind soul!”, both highlighted community awareness of such coded expressions.

[3]Here, *Traditional Chinese* refers to written Chinese in traditional characters, including varieties such as Taiwan Man-

process of finding reviews that contain coded languages from this set.

3.2 Constructing CODEDLANG Dataset

We used an iterative, human-in-the-loop mining process to identify coded languages in large-scale Google Maps reviews. Two of the authors served as annotators in this process.

Pilot Study. We collected 23 gold examples of coded language from social media and news media discussions (sources described in Section 3.1). To estimate real-world prevalence, we manually annotated 5,391 U.S. restaurant reviews (created in Section 3.1). We identified only 19 reviews (0.35%) that used coded language, confirming that it is a rare phenomenon in this domain. We also found that most coded expressions occur at the *phrase level* rather than the sentence level, meaning they can often be decoded using limited local context. A dictionary-based table lookup covering common phrases, combined with substantial human validation, should capture a large portion of real-world cases.

Step 1: Seed Dictionary Construction. We first annotated the specific text spans that contained coded language in all examples (*e.g.*,"虾" in "真的是虾到爆") from Pilot Study, where 虾 (xia1, shrimp) is a homophonic substitution for 瞎 (xia1, absurd), and then aggregated these annotated spans to form a dictionary. The initial dictionary contained 100 coded spans.

Step 2: Candidate Retrieval and Validation. Using this dictionary, we applied string matching across the entire review corpus to retrieve candidate reviews containing dictionary entries. To further expand coverage, we also included reviews whose English machine translations contained Pinyin, a phonetic system for Chinese that uses the Latin alphabet to represent pronunciation, since this indicates potential machine translation challenges. All candidate reviews were then manually validated.

Step 3: Dictionary Expansion and Iterative Bootstraping. The validated reviews from Step 2 went through the same span-annotation process described in Step 1, and the newly annotated spans were added to the dictionary. The updated dictionary was then used to retrieve additional coded-language reviews from the corpus, and this process was repeated iteratively. We continued the bootstrapping process until convergence, defined as an iteration in which no new validated coded-language spans were identified and no additional coded-language reviews were retrieved. To further improve coverage, we also incorporated all word entries from an external homophone dictionary derived from Weibo and Tieba data (Ma et al., 2025b) during this iterative process.

As a result, CODEDLANG contained **900 reviews containing coded language** out of 7,744 manually annotated reviews. These 7,744 items include the 5,391 annotated restaurant reviews from the Pilot Study, as well as cases surfaced throughout the iterations.

Annotating Coded Language Classes. Based on the accumulated coded-language instances, we developed a taxonomy of seven coded-language categories (Section 3.3) and annotated each review using multi-label class assignments. During annotation, annotators first identified whether a review contained coded language, then marked the exact text spans that were coded. Each coded span, consisting of one or more characters, was assigned a single category corresponding to its primary encoding strategy. It is possible for a single span to involve multiple encoding mechanisms. But for annotation consistency and downstream evaluation clarity, we adopted a one-to-one span–category mapping, assigning each span to its dominant encoding strategy. Our annotation decision was based on both surface form and intended meaning. While some expressions may involve layered transformations, we categorize them according to the most salient observable mechanism. For example, the expression " 铁🍔🍔" derives from the phrase "铁憨憨" (tie3 han1 han1, meaning "silly"). The transformation involves an intermediate phonetic substitution 憨 (han1, silly) to 汉 (han4), to 汉堡 (han4 bao3, hamburger) which is 🍔, but the final surface form primarily manifests as emoji substitution. Therefore, it is labeled under emoji substitution in our taxonomy. A review could contain multiple spans using different strategies and thus receive multiple coded-language labels. This annotation process resulted in 184 unique coded spans across the dataset. While each span was assigned a single encoding category, its decoded meaning could vary depending on contextual usage.

Inner-Annotator Agreements. During the iterative bootstrapping phase, the first author manu-

darin and Cantonese.

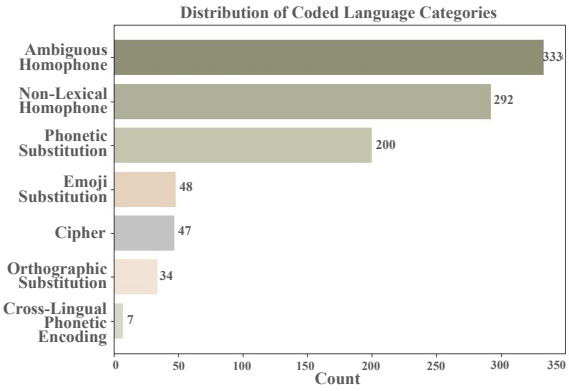

Figure 2: Distribution of coded language categories in the annotated dataset.

ally labeled all items returned by the process and included all labeled items, both coded and non-coded, in CODEDLANG. The second author then independently annotated the entire CODEDLANG. They achieved high agreement on whether a review contained coded language (Cohen's kappa = 0.99), which was calculated on the annotated data before resolving disagreements.. At the span level, annotators also showed strong consistency, with 98% of identified spans overlapping between annotators. Recurring disagreements primarily arose in distinguishing dialectal variation from intentionally coded language. In some cases, regional phonetic spellings resembled coded substitutions, requiring annotators to determine whether a form reflected deliberate encoding or ordinary dialectal expression. For example, 里格楞 was a candidate of coded expression, but was later identified as a Beijing dialect term meaning 'pull strings'. We labeled it as non-coded language, as its surface form directly reflects its intended meaning, despite posing an interpretation barrier for those unfamiliar with the dialect. All disagreements were resolved through discussion to reach consensus, which determined the final released labels.

3.3 Taxonomy of Coded Language

We inductively refined the decision boundaries of each category through repeated discussions between the two annotators through multiple rounds of review. Table 1 summarizes the final, converged taxonomy, including concise definitions and representative examples. Figure 2 shows the distribution of coded-language categories. Homophone-based strategies are the most prevalent, followed by phonetic and orthographic substitutions, while cipher-like expressions and cross-lingual phonetic encoding are rarer.

Model	Precision ↑	Recall ↑	F_1 ↑
GPT-5-mini	0.69	0.80	0.74
Gemini-2.5-Flash	0.67	**0.92**	0.77
DeepSeek-V3.2	0.77	0.79	**0.78**
Qwen2.5-7B-Instruct	**0.80**	0.43	0.56
Llama-3.1-8B-Instruct	0.35	0.68	0.46

Table 2: Coded language detection performance across models. Best scores in each column are in **bold**.

4 Downstream Benchmark Tasks

Using CODEDLANG, we evaluated the capability of LLMs to handle real-world coded language across three NLP tasks: *(i)* coded language detection, *(ii)* coded language classification, and *(iii)* review rating prediction.

4.1 Coded Language Detection

Experimental Setups. We assessed LLMs' ability to detect coded language in reviews using a binary classification task. We benchmarked a diverse set of modern English and Chinese LLMs, including GPT-5-mini (OpenAI, 2025), Gemini-2.5-Flash (Comanici et al., 2025), Qwen2.5-7B-Instruct (Qwen et al., 2025), DeepSeek-V3.2 (DeepSeek-AI et al., 2025) and Llama-3.1-8B-Instruct (Meta, 2024). Models were instructed to identify whether a review contains coded language and output a binary prediction. Our prompt included the definition of coded language along with few-shot examples drawn from our taxonomy. We evaluated classification performance using precision, recall, and F_1 score.

Results. From the Table 2, we found DeepSeek-V3.2 has a balanced tradeoff between precision and recall, and has the highest F_1 score in the coded language detection. Gemini-2.5-Flash has the highest recall, which captures the majority of coded expressions at the cost of reduced precision. On the other side, Qwen2.5-7B-Instruction has the highest precision. Comparing across four models, there is no single model that dominates across all metrics.

4.2 Coded Language Classification

Experimental Setups. We formulated coded language classification as a multi-label classification task, where a review may apply multiple coded language strategies to encode its message. Our prompts provide concise definitions of each coded-language category along with a few-shot representative examples. Given a review, models are instructed to produce a structured output with three

Coded Language Class	P ↑	R ↑	F_1 ↑	N
Ambiguous Homophone	0.48	0.46	0.47	333
Non-Lexical Homophone	0.62	0.57	0.59	292
Phonetic Substitution	0.47	0.56	0.51	200
Emoji Substitution	0.36	0.50	0.42	48
Cipher	**0.84**	0.66	**0.74**	47
Orthographic Substitution	0.32	0.94	0.48	34
Cross-Lingual Phonetic Encoding	0.25	**1.00**	0.40	7

Table 3: Multi-label coded language classification performance across categories by `DeepSeek-V3.2`. The highest (lowest) scores in each column are highlighted in **bold** (underlined).

fields: (1) classes, a list of coded-language categories applied in the review; (2) evidence, a list of exact text spans copied from the review that support the classification; and (3) a brief explanation justifying the decision. The output classes and the evidence of the coded span would be empty if no coded language is detected. Models are evaluated against annotations derived from our taxonomy and measured using precision, recall, and F_1 score.

Results. We reported the results of the coded-language classification using DeepSeek-V3.2. Results for other models are provided in the Appendix A.2. Table 3 reports multi-label classification performance across coded language categories. To account for class imbalance, we reported per-class performance metrics, allowing a granular assessment across categories. Cipher expressions achieved the highest F_1 and precision, indicating that they are easier to identify than other categories. Cross-Lingual Phonetic Encoding had the highest recall but lowest precision and F_1, which tells that the model can not identify this category clearly. This might be influenced by the limited sample size of the Cross-Lingual Phonetic Encoding Category. The Homophones, Phonetic, Orthographic, and Emoji Substitutions achieved F_1 scores ranging from 0.42 to 0.59, indicating that these phenomena remained challenging for current models.

4.3 Review Rate Prediction

Experimental Setups. We examine review rate prediction to understand how coded language affects models' ability to infer users' intended star ratings. While ratings are commonly treated as proxies for sentiment, reviews that use coded language may express evaluative intent indirectly, introducing additional ambiguity. Considering the misalignment between ratings and review content, we manually validated the pairs of ratings and reviews. We found that ratings are generally consistent with review sentiment, with only seven reviews rated four or five that express negative content.

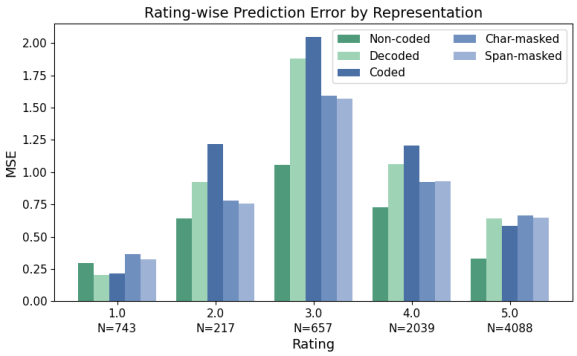

Figure 3: Rating-wise mean squared error (MSE) for review rating prediction under different textual representations. Results were predicted by `GPT-5-mini`. Sample sizes for each rating level were shown below the x-axis, reflecting imbalance across ratings.

The review rating prediction was evaluated on Mean Squared Error (MSE). To investigate the effect of coded language, we construct multiple formats of the same review text: (1) Non-coded reviews as original; (2) Coded reviews as original; (3) Character-based masked reviews, where each coded characters is replaced with a [CODE] token; (4) Span-based masked reviews, where consecutive coded tokens are replaced with a single [CODE-SPAN]; and (5) Decoded reviews, where coded expressions are replaced with their intended meaning using a dictionary derived from annotation.

Results. Across all models in Table 4, the MSE on coded reviews is larger than on non-coded reviews, indicating higher prediction error under coded language. It shows that coded language disrupted sentiment signals relevant for rating prediction. GPT-5-mini showed the most robust performance across review formats, with consistently low MSE, so we further examined its predictions stratified by rating groups.

Figure 3 presents rating-wise prediction error across different textual representations of reviews. Prediction errors were low for extreme ratings (1 and 5), indicating that strongly expressed sentiment was relatively easy to infer even in the presence of coded or masked language. In contrast, 3-star reviews exhibited the highest error, reflecting the inherent ambiguity of mixed or neutral evaluative intent. Notably, for intermediate ratings (2-4), masked reviews have lower error than decoded and coded reviews, suggesting that coded expressions can introduce misleading surface cues and that literal decoding does not always clarify intent.

Model	Non-coded ↓	Decoded ↓	Span-Masked ↓	Coded ↓	Char-Masked ↓
GPT-5-mini	**0.52**	0.55	0.65	0.68	0.71
Gemini-2.5-Flash	**0.63**	0.69	0.73	0.70	0.74
DeepSeek-V3.2	**0.47**	0.82	1.16	0.80	1.08
Qwen2.5-7B-Instruct	**0.90**	1.27	1.60	1.90	1.67
Llama-3.1-8B-Instruct	**0.67**	0.89	0.93	1.30	1.23

Table 4: Review rating prediction performance (mean squared error (MSE), lower is better) across different review types. The best performance in each row is highlighted in **bold**, and the worst is underlined. Overall, LLMs perform worse when processing reviews that contain coded language.

Coded Language Class	CER-Pinyin	CER-IPA
Ambiguous Homophone	0.49	0.57
Non-Lexical Homophone	**0.30**	**0.37**
Phonetic Substitution	0.61	0.68
Emoji Substitution	0.58	0.63
Cross-Lingual Phonetic Encoding	1.23	2.46

Table 5: Character error rates (CER) between coded expressions and their decoded forms. Lower CER indicates greater pronunciation similarity. The lowest (highest) CER are highlighted in **bold** (underlined).

Coded Language Class	MSE
Ambiguous Homophone	0.31
Non-Lexical Homophone	0.43
Phonetic Substitution	0.51
Emoji Substitution	0.56
Cross-Lingual Phonetic Encoding	0.86

Table 6: Mean squared error (MSE) between GPT5-mini predicted and user-assigned ratings across coded language classes.

As an additional analysis, we found that coded language was significantly associated with lower ratings in Google Reviews ($p < 0.001$). Coded language tended to appear in relatively concise reviews under extreme sentiment conditions, even though negative reviews were longer overall ($p < 0.001$). Appendix A.1 shows the detailed analysis.

5 Phonetics Analysis

LLMs often performed poorly on coded language, but it remained unclear which factors drive this performance drop. In CODEDLANG, many coded reviews relied on pronunciation-based strategies, which raised an interesting question: **Does phonetic similarity between coded expressions and their intended meanings make coded language easier for LLMs to process?** We conducted a set of analyses to answer this question.

Measuring Pronunciation Similarity. We measured phonetic similarity between coded expressions and their decoded meanings. We used

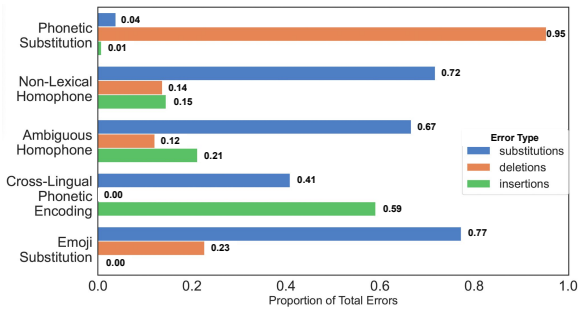

Figure 4: The average error rate for substitutions, deletions, and insertions per class using Pinyin representations, ranked by the total number of errors per class.

tools to annotate Pinyin (Huang et al., 2025) and International Phonetic Alphabet (IPA) representations (Taubert, 2025) for each coded-decoded span pair in CODEDLANG, excluding the Cipher and Orthographic Substitution classes. Tones in Pinyin were annotated from 1 to 4 and mapped to IPA tone contours accordingly (1 to ˥, 2 to ˧˥, 3 to ˨˩˦, and 4 to ˥˩). We used character error rate (CER), a common evaluation metric in automatic speech recognition (ASR), to quantify the phonetic similarity between coded and decoded spans. In ASR, CER is computed as the normalized sum of substitutions (S), deletions (D), and insertions (I) over the total number of characters (N) between the ASR-generated utterance (hypothesis) and verbatim utterance (reference) (Morris et al., 2004) (Equation 1). In our analysis, a lower CER indicates greater pronunciation similarity between the coded expression and its decoded form. We treated each phoneme in IPA and syllable in Pinyin as a character, and computed pairwise CER for each pair of decoded span (reference) and coded span (hypothesis). Some pinyin abbreviations consist only of consonant initials, which cannot be directly mapped to IPA without a vowel. In such cases, we follow the common convention used by Chinese speakers when reading pinyin initials aloud, using pronunciation for each alphabet letter without tone to ensure pronounceability before converting to IPA. This avoids gener-

ating transduction error [UNK] tokens during IPA conversion, while we acknowledge that this transformation may introduce higher IPA error rate.

$$\text{CER} = \frac{S + D + I}{N} \tag{1}$$

Figure 5 illustrates how CER is computed using Pinyin and IPA representations for the same coded expression and its decoded form, including substitution, deletion, and insertion errors.

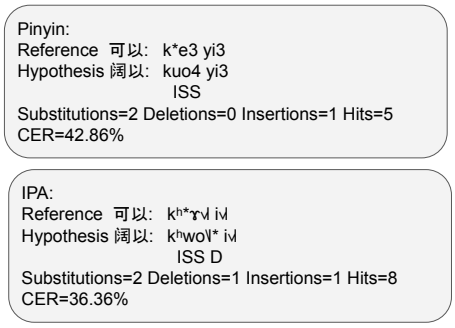

Figure 5: Example of CER based on Pinyin and IPA representations.

5.1 Findings

Phonetic Error Patterns Across Categories. As shown in Table 5, cross-lingual phonetic encoding yielded the highest CER, highlighting the difficulty of cross-language pronunciation mapping. Non-lexical homophones showed the lowest CER, reflecting relatively high phonetic similarity despite semantic ambiguity. Figure 4 shows the average substitution, deletion, and insertion error rates across coded language categories computed using Pinyin, following Equation 2. Corresponding results using IPA are reported in Appendix A.3.

$$\text{Error Rate} = \frac{\text{total \# of S/I/D}}{\text{total \# of S+I+D}} \tag{2}$$

We observed that the Phonetic Substitution class reached the highest number of deletions for both Pinyin and IPA. This observation was primarily driven by the use of Pinyin initials, which reduced multi-syllabic words to only Pinyin consonants and subsequently led to failures in the IPA conversion. For example, "rmb" was abbreviated from **r**en2 **m**in2 **b**i4 (Chinese Yuan). We also observed a cross-lingual case where "ww" was coded for 哈哈 (haha, laughter), similar to lol (lots of laughs). The origin of w was abbreviated from わら (warai, laughter) in Japanese. In addition, substitutions reached the second most frequent error type across all coded language categories. For example, **j**in1 ho**u4** **xia2** (金厚呷, non-lexical homophones for 'really delicious' in Taiwanese) was substituted for **zhe**n1 ha**o3** **chi1** (真好吃, really delicious). This suggested a high degree of phonetic interference, in which the target sounds were replaced by phonetically similar but intentionally incorrect variants.

Phonetic Similarity and Rating Prediction Error. We found that greater phonetic distortion in coded language was associated with higher rating prediction error, but this relationship varied across coded language categories and remained modest overall. The MSE between the GPT-5-mini predicted rating and the user-assigned rating are shown in the table 6. The Spearman's ρ between MSE and average CER reached 0.80 for both Pinyin and IPA, suggesting that more character errors were associated with more variance between predicted ratings and user-assigned ratings. Yet, the correlations did not reach conventional levels of statistical significance (p-value = 0.104).

The Cross-Lingual Phonetic Encoding class achieved the highest MSE and CER for both Pinyin and IPA, which may be due to the lack of cross-lingual context and the limited data size. Compared to Ambiguous Homophones, Non-Lexical Homophones achieved lower CER under both Pinyin and IPA, but obtained higher MSE in rating prediction. This suggested that, despite higher pronunciation similarity, the loss of lexicalized meaning in Non-Lexical Homophones impairs downstream pragmatic interpretation more.

6 Discussion

Expressiveness vs. Interpretability in Coded Language. Our data shows a highly skewed distribution in coded language usage: among 184 unique coded spans, 111 appear only once, while a small number of forms are used repeatedly. The most frequent span, "滴"(di1, drip), appears 102 times and serves as a phonetic substitution for "的"(de, of), often used to convey a playful, cute tone. This pattern suggests that while the majority of coded expressions are context-specific, a small subset becomes widely shared and stabilized through repeated use. This observation aligns with prior work showing that popular code words rely on unconventional character combinations and simple, memorable, affective, or sarcastic expressions (Zhang et al., 2014). At the other extreme, some unique spans in our dataset were difficult even for human annotators to interpret. For exam-

ple "🐘人🈸一🐑🈴姜" used emojis to replace homophone words. The review reads as (xiang4 ren2 shen1 yi**1** yang**2** de**2** jiang1, elephant human apply one sheep get ginger), which actually means (xiang4 ren2 shen1 yi**2** yang**4** de jiang1, the ginger was like ginseng).

There is a trade-off between expressiveness and interpretability in coded language. Codes that are easy to recognize, humorous, or representative are more likely to be widely adopted and circulate as shared memes, but their obfuscatory or boundary-setting function weakens as they spread. In contrast, niche or opaque codes preserve stronger hiding effects but require greater interpretive effort and shared background knowledge. Prior work shows that human-created code words are perceived as funnier than machine-generated ones and are therefore more likely to be adapted over time (Zhang et al., 2015). Similarly, Wan et al. (2025) shows that boundary-making strategies based on linguistic tricks lose effectiveness as audiences diversify, forcing users to continually invent new, more creative, and less transparent codes.

Multilingualism of Coded Language. Our analysis further shows that coded language often operates across languages and writing systems, creating barriers to interpretation for both native speakers and language models. Some coded expressions exploited cross-script visual similarity rather than phonetic or semantic equivalence, such as using Kanji or Chinese characters whose shapes resemble Latin letters (*e.g.*, "爪丫" encoding "MY"). More broadly, Chinese comprises a family of languages and dialects with varying alignments between written and spoken forms. For example, written Cantonese closely reflects spoken Cantonese and differs substantially from Mandarin in syntax and lexicon, introducing additional variation in coded expressions. Chinese online communication also draws on sustained contact with neighboring languages such as Japanese and Korean, leading to the use of transliteration-based codes. For example, "卡哇伊" (ka3 wa1 yi1, card wow they) means かわいい (Kawaii, cute) from Japanese. These multilingual and cross-script practices highlight that understanding coded language often requires shared community knowledge spanning multiple linguistic systems, further complicating both human annotation and automated interpretation.

7 Conclusion

Coded language is a common communicative strategy in which intended meaning diverges from surface form and requires decoding to be understood. We presented CODEDLANG, a dataset of 7,744 Chinese Google Maps reviews with span-level annotations and a seven-class taxonomy capturing phonetic, orthographic, and cross-lingual encoding strategies. We evaluated language models on coded language detection, classification, and review rating prediction. Our results showed that even strong models frequently fail to detect coded language, and phonetic distortions contribute to downstream prediction errors. These findings highlight coded language as a persistent challenge for real-world NLP systems and motivate further work.

8 Limitations

Although our work represents an important first step toward understanding and modeling coded language, it has several limitations.

- First and most significantly, our data curation relied on an iterative bootstrapping process that was inherently constrained by what LLMs and existing lexicons could surface. If a coded expression never appeared in the seed dictionaries we used and was not recognized by the LLMs during bootstrapping, it was missed. This bias was difficult to avoid when constructing datasets for low-frequency linguistic phenomena and fundamentally limited coverage of rare, creative, or highly novel forms of coded language.

- Second, the dataset was temporally and geographically bounded. Our collection endpoint was 2022, and the data focused on online reviews in the United States. Because coded language evolves over time and varies across regions, platforms, and communities, newer forms or region-specific patterns emerging after this cutoff were not captured. In the long run, we plan to update CODEDLANG to cover additional languages, genres, and time frames.

- Third, annotation and interpretation were inevitably shaped by the annotators' linguistic backgrounds and community knowledge. As a result, some coded expressions may have been missed or misinterpreted, particularly

those that were highly personalized, culturally specific, or dependent on insider context.

- Fourth, the operational scope of coded language in this paper was informed by collective authorial decisions made during taxonomy development and data curation. We were confident that such cases were rare, given that the authors have professional training in linguistics and are native speakers of Chinese. However, it remained possible that a different group of researchers, drawing on alternative theoretical perspectives or community norms, might have made different judgments about which cases to include or exclude.

- Finally, our analysis prioritized identifying broad patterns and evaluating downstream impacts of coded language rather than constructing an exhaustive inventory. Accordingly, we did not claim to capture all possible coded expressions in online communication.

9 Ethics Statement

Our work analyzed naturally occurring coded language in online reviews to better understand how meaning is expressed beyond literal text. Some coded expressions are intentionally used to obscure meaning or to restrict interpretability to specific audiences. We recognized that, in some cases, decoding such expressions could run counter to the original communicative intent of individual writers. We are aware of and familiar with prior scholarship on ethical considerations in research using public online data, including discussions of participant perspectives and ethical tensions when analyzing social media content in public research contexts, such as studies of Twitter user perceptions of research ethics (Fiesler and Proferes, 2018) and systematic reviews of ethical questions in research using Reddit data (Fiesler and Proferes, 2018).

To mitigate ethical concerns, we focused on a relatively low-risk domain, namely publicly available online reviews. All data were drawn from public sources and analyzed in aggregate, without attempts to infer, identify, or attribute sensitive information to specific individuals.

Our goal was not to expose or target individual users, but to study coded language as a linguistic and social phenomenon that plays an important role in human communication. By examining coded language, we aimed to support more faithful and equitable interpretation of authentic user communication in NLP systems, rather than to undermine or suppress user expression.

Acknowledgments

We acknowledge support from the Linguistic Diversity Across the Lifespan Graduate Research Traineeship Program (NSF Grant No. 2125865) awarded to Ruyuan Wan, as well as the pilot-study funding from the Center for Language Science for this project. We also thank the anonymous reviewers, Dr. Kenton Murray, Dr. Mina Lee and her lab, the Cornell NLP Lab, Dr. Qunfang Wu, Lan Gao, and Lu Xian for their constructive feedback.

References

Jialin Chen. 2025. Coded language emerges in taiwanese online reviews. A Taiwanese user's highly encrypted review of an overseas restaurant attracts 1.61 million viewers.

Won Ik Cho and Soomin Kim. 2021. Google-trickers, yaminjeongeum, and leetspeak: An empirical taxonomy for intentionally noisy user-generated text. In *Proceedings of the seventh workshop on noisy user-generated text (W-NUT 2021)*, pages 56–61.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, and 245 others. 2025. Deepseek-v3.2: Pushing the frontier of open large language models. *Preprint*, arXiv:2512.02556.

Mia Fan. 2024. Disappointed chinese diners are writing google maps business reviews in secret codes. Accessed May 15, 2025.

Casey Fiesler and Nicholas Proferes. 2018. "participant" perceptions of twitter research ethics. *Social Media + Society*, 4(1):2056305118763366.

Jan Fillies and Adrian Paschke. 2024. Simple llm based approach to counter algospeak. In *Proceedings of the 8th Workshop on Online Abuse and Harms (WOAH 2024)*, pages 136–145.

Huang Huang, Xiaoquan Kong, and 1 others. 2025. pypinyin: A python library for converting chinese characters to pinyin. https://github.com/mozillazg/python-pinyin. Version 0.55.0.

Heng Ji and Kevin Knight. 2018. Creative language encoding under censorship. In *Proceedings of the first workshop on natural language processing for internet freedom*, pages 23–33.

Julie Kallini, Dan Jurafsky, Christopher Potts, and Martijn Bartelds. 2025. False friends are not foes: Investigating vocabulary overlap in multilingual language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 21138–21154.

Soomin Kim, Changhoon Oh, Won Ik Cho, Donghoon Shin, Bongwon Suh, and Joonhwan Lee. 2021. Trkic g00gle: Why and how users game translation algorithms. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW2):1–24.

Jiacheng Li, Jingbo Shang, and Julian McAuley. 2022. Uctopic: Unsupervised contrastive learning for phrase representations and topic mining. *arXiv preprint arXiv:2202.13469*.

Jianfei Ma, Zhaoxin Feng, Emmanuele Chersoni, Huacheng Song, and Ziqi Zhang. 2025a. Phonothink: Improving large language models' reasoning on chinese phonological ambiguities. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 19018–19033.

Jianfei Ma, Zhaoxin Feng, Huacheng Song, Emmanuele Chersoni, and Zheng Chen. 2025b. Reasoning or memorization? investigating llms' capability in restoring chinese internet homophones. In *Proceedings of the 3rd Workshop on Towards Knowledgeable Foundation Models (KnowFM)*, pages 120–139.

Julia Mendelsohn, Ronan Le Bras, Yejin Choi, and Maarten Sap. 2023. From dogwhistles to bullhorns: Unveiling coded rhetoric with language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15162–15180.

Meta. 2024. Llama-3.1-8b-instruct. https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct. Accessed: 2026-04-20.

Andrew Morris, Viktoria Maier, and Phil Green. 2004. From wer and ril to mer and wil: improved evaluation measures for connected speech recognition.

OpenAI. 2025. Gpt-5 mini model documentation. https://platform.openai.com/docs/models/gpt-5-mini. Accessed: 2026-01.

Zhu Qing. 2024. 'Newspaper eat:' How Chinese students hide warnings on Google reviews. Accessed May 15, 2025.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Itsumi Saito, Kugatsu Sadamitsu, Hisako Asano, and Yoshihiro Matsuo. 2014. Morphological analysis for japanese noisy text based on character-level and word-level normalization. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1773–1782.

Stefan Taubert. 2025. pinyin-to-ipa: A tool for converting pinyin to ipa. https://github.com/stefantaubert/pinyin-to-ipa. Version 1.0.0.

Ruyuan Wan, Lingbo Tong, Tiffany Knearem, Toby Jia-Jun Li, Ting-Hao'Kenneth' Huang, and Qunfang Wu. 2025. Hashtag re-appropriation for audience control on recommendation-driven social media xiaohongshu (rednote). In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–25.

Yunze Xiao, Yujia Hu, Kenny Choo, and Roy Lee. 2024. Toxicloakcn: Evaluating robustness of offensive language detection in chinese with cloaking perturbations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6012–6025.

Canwen Xu, Wangchunshu Zhou, Tao Ge, Ke Xu, Julian McAuley, and Furu Wei. 2021. Blow the dog whistle: A chinese dataset for cant understanding with common sense and world knowledge. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2139–2145.

An Yan, Zhankui He, Jiacheng Li, Tianyang Zhang, and Julian McAuley. 2023. Personalized showcases: Generating multi-modal explanations for recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2251–2255.

Boliang Zhang, Hongzhao Huang, Xiaoman Pan, Heng Ji, Kevin Knight, Zhen Wen, Yizhou Sun, Jiawei Han, and Bulent Yener. 2014. Be appropriate and funny: Automatic entity morph encoding. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 706–711.

Boliang Zhang, Hongzhao Huang, Xiaoman Pan, Sujian Li, Chin-Yew Lin, Heng Ji, Kevin Knight, Zhen Wen, Yizhou Sun, Jiawei Han, and 1 others. 2015. Context-aware entity morph decoding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 586–595.

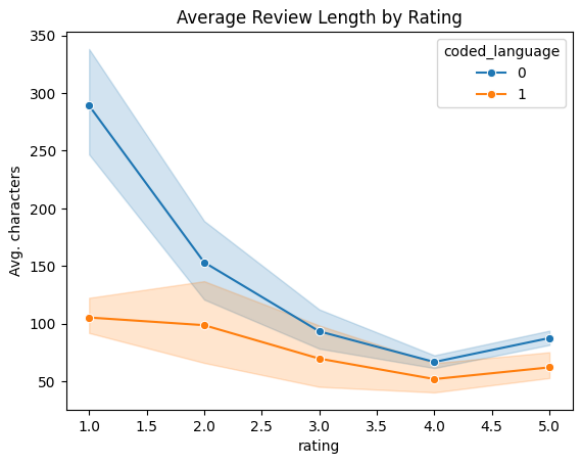

Figure 6: The average review length by rating, comparing coded and non-coded reviews.

A Appendix

A.1 Statistics Analysis

We examine whether coded language is associated with user ratings in Google Reviews. Using a chi-square test of independence, we find a significant association between coded language and rating distribution ($p < 0.001$), with a moderate effect size (Cramér's $V = 0.29$). A Mann–Whitney U test further shows that reviews containing coded language have significantly lower rating ranks than non-coded reviews ($p < 0.001$). Consistent with these findings, an ordinal logistic regression indicates a negative association between coded language presence and rating ($\beta = -0.89$, $p < 0.001$), suggesting that coded language is more likely to appear in lower-rated reviews.

We further analyze the relationships among review length, coded language use, and rating scores. Because the review length distribution is highly skewed and heavy-tailed, we employ nonparametric statistical tests throughout. We first examine the relationship between review length and rating across the full corpus. A Kruskal–Wallis test shows that the distributions of review lengths differ significantly across the five rating levels ($p < 0.001$). Consistent with this result, Spearman's rank correlation reveals a weak but significant negative monotonic relationship between rating and review length ($\rho = -0.12, p < 0.001$), indicating that lower-rated (more negative) reviews tend to be longer overall. We next test whether coded language use is associated with review length when aggregating across all ratings. A Mann–Whitney U test comparing coded and non-coded reviews shows no significant difference in length distributions ($p = 0.18$),

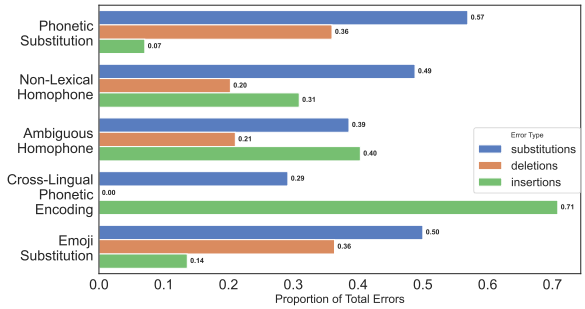

Figure 7: The average error rate for substitutions, deletions, and insertions per class using IPA representations, ranked by the total number of errors per class.

suggesting that coded language is not globally associated with review length. However, when stratifying by rating, length differences emerge at the extremes: coded reviews are significantly shorter than non-coded reviews at both 1 and 5 ratings ($p < 0.001$), while differences at intermediate ratings (2–4) are not statistically significant. This pattern indicates that the length effect of coded language is conditional on sentiment intensity rather than a uniform stylistic property. Therefore, negative reviews tend to be longer in general, while coded language is associated with more concise expression under extreme sentiment conditions, as also shown in Figure 6.

A.2 Additional Coded Language Classification Results

Table 7 reports per-class precision, recall, and F1 across GPT-5-mini, Gemini-2.5-Flash, Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct.

A.3 IPA-based Phonetic Analysis

Figure 7 presents character error rate analyses computed using IPA representations, corresponding to the IPA-based results reported in the main paper.

A.4 Prompt Templates

A.4.1 Coded Language Detection and Classification

```
You are annotating whether a review
contains coded language.
Definition:
"Coded language" refers to intentional
encoding or obfuscation of meaning.
The surface form and the intended hidden
meaning are different, and decoding
or inference is required to recover
the meaning.
IMPORTANT:
```

Coded Language Class	GPT-5-mini			Gemini-2.5-Flash			Qwen2.5-7B-Instruct			Llama-3.1-8B-Instruct		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Ambiguous Homophone	0.66	0.61	0.63	0.57	0.61	0.59	0.45	0.23	0.30	0.25	0.27	0.26
Non-Lexical Homophone	0.65	0.54	0.59	0.68	0.73	0.70	0.38	0.30	0.34	NaN	0.00	NaN
Phonetic Substitution	0.50	0.95	0.65	0.51	0.97	0.67	0.48	0.16	0.24	0.38	0.45	0.41
Emoji Substitution	0.75	0.25	0.38	0.77	0.83	0.80	0.10	0.08	0.09	0.09	0.58	0.16
Cipher	0.86	0.81	0.84	0.78	0.85	0.81	0.45	0.12	0.19	0.04	0.13	0.06
Orthographic Substitution	0.30	0.91	0.45	0.46	0.97	0.63	0.17	0.65	0.27	0.02	0.85	0.05
Cross-Lingual Phonetic Encoding	0.05	1.00	0.10	0.04	1.00	0.08	0.09	0.57	0.16	0.06	0.71	0.11

Table 7: Per-class coded language classification performance across models.

```
Not all socially opaque language is
coded language. Dialect, jargon, or
multilingual text alone does NOT count
as coded language unless encoding is
involved.
Coded language classes (multi-label
allowed):

1. ambiguous_homophone
  An existing word or phrase whose
  literal meaning differs from its
  hidden meaning.
  Examples:
  - 炒鸡 → 超级 (very)
  - 辣鸡 → 垃圾 (trash)
  - 稀饭 → 喜欢 (like)

2. non_lexical_homophone
  Homophonic characters or tokens
  that do NOT form a valid lexical
  expression on the surface.
  Examples:
  - 灰常 (非常)
  - 菇凉 (姑娘)
  - 开森 (开心)

3. Phonetic Substitution
  Use of pinyin, phonetic symbols,
  or roman letters to represent
  words or meanings instead of
  full characters.
  Examples:
  - nb （牛逼）
  - tm （他妈）
  - 好ㄘ（好吃）
  - e度（呢度）

4. Emoji
  Emoji replaces textual content and
  conveys a hidden meaning DIFFERENT
  from its literal emoji meaning.
  Coded examples:
  - [鸡emoji]贼 → 鸡贼
  - ①般 → 一般
  - 铁[汉堡emoji][汉堡emoji] → 铁憨憨
  Non-coded (DO NOT label as coded):
  - 值[得emoji] → (值得)
  - [满emoji]意 → 满意
  - 酱油[鸡emoji] → (酱油鸡)

5. orthographic
  Encoding through visual or orthographic
  similarity.
  Examples:
  - 爪丫 → MY
  - 艹 → 草
  - N0 G00D → NO GOOD

6. transliteration
  Phonetic transliteration across
  languages.
  Examples:
  - 古德 → good
  - 歪瑞古德 / 歪立古德 → very good
  - 纱头 → shuttle
  - ball ball → 求求

7. cipher
  The sentence has no clear literal
  semantic meaning and appears
  intentionally nonsensical or
  cipher-like.
  Example:
  - qqQQaQ！你燃yfyi y f r

Exclusions (NOT coded language):
- Pure dialectal expressions without
encoding
- Pure multilingualism without
phonetic or orthographic substitution
- Emoji used literally without hidden
meaning
```

```
Your task:
- Decide whether coded language is
present.
- If present, identify ALL applicable
coded language classes.
- Extract exact spans from the review
that constitute coded language.
Return ONLY valid JSON with the
following keys:
- label: 1 if coded language is
present, else 0
- classes: list of coded language
classes (empty if label=0)
- evidence: list of exact spans
copied from the review
(empty if label=0)
- short_reason: <= 20 words
explaining the decision

Review:
<<<{text}>>>
```

A.4.2 Rating Prediction

Original Reviews

```
You will be given a review.
Estimate the user's intended star rating
on a 1-5 scale.

Rules:
- Output JSON ONLY.
- The JSON must have exactly one field:
"rating".
- The value of "rating" must be a number
between 1 and 5.
- Do not include any explanation or
extra fields.

Review:
{text}
```

Masked Reviews

```
You will be given a review.
Estimate the user's intended star rating
on a 1-5 scale.

Note:
- The token [CODE] indicates masked
coded language in the original review.
- Treat [CODE] as an unknown placeholder

Rules:
- Output JSON ONLY.
- The JSON must have exactly one field:
"rating".
- The value of "rating" must be a number
between 1 and 5.
- Do not include any explanation or extra
fields.

Review:
{text}
```

Decoded Reviews

```
You will be given a review.
Estimate the user's intended star rating
on a 1-5 scale.

Note:
- Some reviews may contain unusual
or nonsensical expressions.

Rules:
- Output JSON ONLY.
- The JSON must have exactly one field:
"rating".
- The value of "rating" must be a
number between 1 and 5.
- Do not include any explanation or
extra fields.

Review:
{text}
```