

# A SMART Mnemonic Sounds like “Glue Tonic”: Mixing LLMs with Student Feedback to Make Mnemonic Learning Stick

Nishant Balepur<sup>1</sup> Matthew Shu<sup>2</sup> Alexander Hoyle<sup>1</sup> Alison Robey<sup>3</sup>  
Shi Feng<sup>4</sup> Seraphina Goldfarb-Tarrant<sup>5</sup> Jordan Boyd-Graber<sup>1</sup>

<sup>1</sup>University of Maryland <sup>2</sup>Yale University <sup>3</sup>SUNY Empire State University

<sup>4</sup>George Washington University <sup>5</sup>Cohere

[nbalepur@umd.edu](mailto:nbalepur@umd.edu) [jbog@umiacs.umd.edu](mailto:jbog@umiacs.umd.edu)

## Abstract

Keyword mnemonics are memorable explanations that link new terms to simpler keywords. Prior work generates mnemonics for students, but they do not train models using mnemonics students prefer and aid learning. We build SMART, a mnemonic generator trained on feedback from real students learning new terms. To train SMART, we first fine-tune LLaMA-2 on a curated set of user-written mnemonics. We then use LLM alignment to enhance SMART: we deploy mnemonics generated by SMART in a flashcard app to find preferences on mnemonics students favor. We gather 2684 preferences from 45 students across two types: **expressed** (inferred from ratings) and **observed** (inferred from student learning), yielding three key findings. First, expressed and observed preferences disagree; what students *think* is helpful does not always capture what is *truly* helpful. Second, Bayesian models can synthesize complementary data from multiple preference types into a single effectiveness signal. SMART is tuned via Direct Preference Optimization on this signal, which resolves ties and missing labels in the typical method of pairwise comparisons, augmenting data for LLM output quality gains. Third, mnemonic experts assess SMART as matching GPT-4 at much lower deployment costs, showing the utility of capturing diverse student feedback to align LLMs in education.<sup>1</sup>

## 1 Mnemonics Aid Vocabulary Learning

Keyword mnemonics promote efficient and engaging vocabulary (vocab) learning (Benge and Robbins, 2009). These tools help students learn a new term’s meaning (e.g. *Benevolent*) by relating it to a simpler keyword (e.g. *Benevolent sounds like benefit*), and explaining how the keyword and term are linked (e.g. *A boss giving employee benefits is kind, which is the meaning of benevolent*) (Pressley et al., 1982). Students use mnemonics to prepare for exams like the GRE (Fairbanks, 1977) which involve

mastering hundreds of terms (Khan, 2009). Despite their utility, writing mnemonics is tedious, requiring vocabulary expertise and creativity to make memorable keyword links (Siriganjanavong, 2013).

To ease these burdens, prior work automatically generates keyword mnemonics (Savva et al., 2014). However, most works design keyword extractors (Anonthanasap et al., 2015), omitting the explanations linking keywords to terms that enable effective mnemonic use (Raugh and Atkinson, 1975). Large Language Models (LLMs) are apt for writing explanations, a difficult task that tests if LLMs can combine vocabulary (Huang et al., 2022a), phonology (Suvarna et al., 2024), commonsense (Davis, 2023), and creativity (Tian et al., 2024) to help students learn (§6.3). While promising, existing works only *prompt* LLMs (Lee and Lan, 2023) and lack training on *student feedback* to guide LLMs toward mnemonics students prefer and benefit learning.

In pursuit of student-guided mnemonics, we propose SMART, which employs Student Mnemonic Alignment to generate keyword mnemonics that aid the **Recall of Terms**. (Figure 1). To train SMART, we first get data from MnemonicDictionary (Memiliapp, 2007), a site where users submit mnemonics that they find helpful. We collect a high-quality subset of submitted mnemonics to fine-tune LLaMA-2 70B (Figure 1, left) as our initial model (§2). To enhance SMART, we draw from LLM alignment, which improves LLMs via tuning to preference labels that capture which of two LLM outputs users favor (Casper et al., 2023). We gather preferences by sampling mnemonics from our initial model and deploying them to students in a flashcard app (§3).

There are many ways to collect preferences, and Bansal et al. (2024) show that pairwise rankings and Likert ratings yield conflicting labels on which LLM output is favored. To study preference agreement in education (Figure 1, mid), we also gather pairwise and Likert annotations, which we define as **expressed preferences**: those inferred from user

<sup>1</sup><https://github.com/nbalepur/Mnemonic>

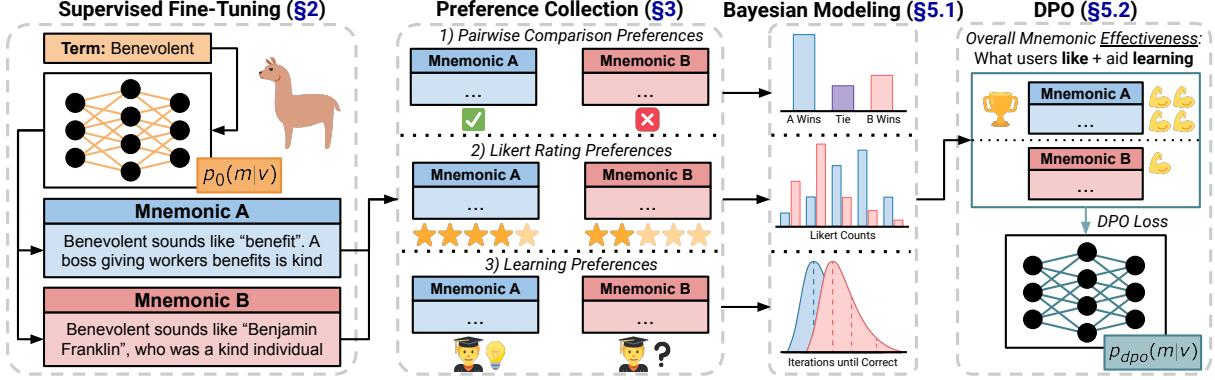


Figure 1: SMART overview. We fine-tune LLaMA-2 70B for the initial SMART model (§2). We then collect three preference types: pairwise, rating, and learning (§3). Finally, a Bayesian model synthesizes mnemonic effectiveness from all three preferences (§5.1) and we use this signal to align SMART via Direct Preference Optimization (§5.2).

ratings. Expressed preferences measure what users *think* are more helpful, but to see if this agrees with what *truly* helps users, we introduce **observed preferences**: those inferred from observable goals (e.g. learning) as users interact with outputs. We collect observed preferences via the mean time users need to learn a term while studying with its mnemonic, a proxy for mnemonic short-term learning efficacy.

Over three months, 45 students gave 2684 preferences on mnemonic pairs. To decide how to align SMART, we study the relation of preference types. Expressed and observed preferences disagree (§4), so what students *think* helps them learn differs from what *truly* helps them learn. These preference types represent equally valuable goals (§5): an effective and helpful mnemonic should be non-harmful and liked by users (expressed), but also aid learning (observed). Thus, we design a Bayesian model (Gelman and Hill, 2006) that learns mnemonic effectiveness via feedback from all preference types (§5.1). We compare mnemonics by effectiveness to elect a winning mnemonic in the pair (Figure 1, right) and tune SMART with Direct Preference Optimization (Rafailov et al., 2024, DPO) on this signal (§5.2).

We assess SMART through several experiments. Fine-tuning and DPO enhance SMART, so aligning LLMs to student preferences improves educational text (§6.1). Further, combining all preferences via Bayesian modeling can resolve ties or missing labels in the typical method of pairwise comparisons, augmenting DPO data for mnemonic quality gains (§6.2). Multiple preferences can be gathered in one app, sometimes with no extra annotations (§3.2.2). So if resources allow, we advise collecting multiple preferences to study the relation of complementary alignment objectives, which can then be combined

for data augmentation to improve LLM outputs.

Lastly, two mnemonic experts assess mnemonics (§6.3) from SMART, GPT-4, and a freelance creative writer from Upwork, finding: 1) SMART matches the SOTA LLM GPT-4, showing the utility of student feedback; and 2) Our writer’s keywords are much simpler and their explanations are more imageable compared to GPT-4 and SMART, motivating mnemonic generation as a difficult task and giving insights into feedback types (simplicity, imageability) that can also be collected to better align LLMs for downstream tasks. Our contributions are:

- 1) We design SMART, an LLM mnemonic generator aligned by feedback from real-world students.
- 2) We analyze expressed and observed preferences, finding that the LLM outputs students *think* help them learn is not what *actually* helps them learn.
- 3) We align SMART with multiple preferences via Bayesian modeling, which can break preference ties for DPO output quality gains and results in a more efficient keyword mnemonic generator that matches the state-of-the-art LLM GPT-4.
- 4) We release the first fine-tuning and preference datasets to aid research in mnemonic generation.

## 2 An Initial SMART Mnemonic Model

Given a vocabulary (vocab) term  $v$  (e.g. **Benevolent**), we desire a keyword mnemonic  $m$  students can use to remember the meaning of  $v$ . For optimal benefits (McDaniel and Pressley, 1984),  $m$  should link to a similarly sounding and simpler keyword  $k$  (e.g. *Benevolent sounds like benefit*), and then explain how  $k$  and  $v$  are linked (e.g. *A boss who gives their employees benefits is kind—or benevolent*).

We now train an initial SMART model to generate keyword mnemonics. We collect a high-quality

dataset of user-written mnemonics (§2.1, §2.2) and fine-tune SMART on this mnemonic dataset (§2.3).

## 2.1 Data Collection

A dataset with vocab terms and mnemonics does not exist, so we curate new datasets to facilitate mnemonic research. We use vocab words from the Graduate Records Examination (GRE), a graduate admissions exam that students prepare for by learning hundreds of vocab terms (Nayak et al., 2017). Mnemonics have been used to help students learn GRE vocabulary (Fairbanks, 1977; Pi et al., 2021).

We base our dataset on 2380 public English GRE terms  $\mathcal{V}$  from seven tutoring sites (Kotchian, 2019). We find a mnemonic for each term from MnemonicDictionary (Memliapp, 2007), a site where users submit keyword mnemonics for vocab terms. Users can also vote on mnemonics, which we later use to find high-quality mnemonics (§2.2). With permission from the owners, we collect 13955 candidate MnemonicDictionary mnemonics for our dataset.

## 2.2 Identifying High-Quality Mnemonics

The user-submitted mnemonics collected from §2.1 are noisy, but a subset of high-quality data would better train SMART for mnemonic generation (Xia et al., 2024). MnemonicDictionary users upvote or downvote mnemonics, so upvote ratio could find high-quality data, but this metric does not consider all upvotes given (Powell et al., 2017). Thus, following Hoffart et al. (2019), we build a Bayesian model to learn the probability  $q_i$  of mnemonic  $m_i$  being high-quality, based on the upvote  $\nu_{u,i}$  and downvote  $\nu_{d,i}$  counts on  $m_i$ . We assume mnemonics with higher  $q_i$  have more upvotes, so we model  $\nu_{u,i}$  as a Binomial distribution with probability  $q_i$ :

$$q_i \sim \text{Beta}(\alpha = 2, \beta = 8), \quad (1)$$

$$\nu_{u,i} \sim \text{Binomial}(\nu_{u,i} + \nu_{d,i}, q_i), \quad (2)$$

which has prior  $\alpha = 2, \beta = 8$ , as our brief manual assessment found that  $\sim 20\%$  of the mnemonics are high-quality. We estimate  $q_i$  via No U-Turn Sampling (Hoffman et al., 2014, NUTS). Pairs  $(v_i, m_i)$  with the 1000-highest  $q_i$  values form the fine-tuning dataset  $\mathcal{D}_{ft}$  for SMART (details in Appendix A).

## 2.3 Model Fine-Tuning

The dataset  $\mathcal{D}_{ft}$  has term/mnemonic pairs  $(v, m)$ , so we can use  $\mathcal{D}_{ft}$  to train an initial seq2seq SMART model  $p_0(m | v)$  to create  $m$  from  $v$ . Upon inspection, we find some quality issues in the mnemonics,

so we use GPT-4 to clean grammar errors in  $m$  via a 0-shot prompt, and discard any  $m$  with offensive or overly culturally-specific text (see Appendix A.1). We end up with 889 pairs for fine-tuning SMART.

Each  $(v, m) \in \mathcal{D}_{ft}$  forms prompt  $\mathcal{P} = \text{“Term: } v \backslash \text{nMnemonic:”}$  and output text  $m$ . Our initial model  $p_0(m | v)$  fine-tunes LLaMA-2 70B (Touvron et al., 2023) to minimize the cross-entropy loss  $\mathcal{L}_{CE}$  of predicting tokens  $m_j \in m$  given  $\mathcal{P}$ :

$$\mathcal{L}_{CE} = \sum_{j=1}^{|m|} \log p(m_j | m_1, \dots, m_{j-1}, \mathcal{P}). \quad (3)$$

We use QLoRA (Dettmers et al., 2023) to minimize  $\mathcal{L}_{CE}$ . All parameters are listed in Appendix B.1.

## 3 Collecting Mnemonic Preferences

Only fine-tuning does not explicitly guide SMART toward mnemonics that users prefer and help them learn—our overall goal. Thus, we use alignment (Ziegler et al., 2019): tuning LLMs to preference labels capturing which outputs users favor. To align SMART, we need a preference dataset  $\mathcal{D}_{pref}$  with entries of a term  $v$ , mnemonic pair  $(m_A, m_B)$  for  $v$  created by the initial model  $p_0(m | v)$ , and preference label  $y \in \{A, B, \text{tie}\}$  noting the mnemonic in the pair users favor. To build  $\mathcal{D}_{pref}$ , we create mnemonic pairs (§3.1), define our preference labels (§3.2), and describe our user study details (§3.3).

### 3.1 Generating Mnemonic Pairs

For  $\mathcal{D}_{pref}$ , we first need mnemonic pairs  $(m_A, m_B)$  created by model  $p_0(m | v)$  for many terms  $v$ . In preference datasets, researchers sample candidate LLM responses to match the abilities of the target LLM (Bai et al., 2022), so the mnemonics should have *high probability* in our model  $p_0(m | v)$ . Further, to ensure the mnemonics in the pair are not too similar, which would often result in preference ties, we seek a *diverse* pair of mnemonics; this gives the user distinct choices, yielding clearer preferences.

We combine these two objectives through best-of- $n$  sampling (Nakano et al., 2021), which samples  $n$  LLM outputs and picks the one with the best score from a reward model. We define a reward  $\pi_{pair}(m_A, m_B, v)$  that returns the sum of  $m_A$  and  $m_B$  sequence probabilities from  $p_0(m | v)$ , minus the ROUGE-1 (Lin, 2004) of  $m_A$  and  $m_B$ , assessing mnemonic diversity (Shaib et al., 2024). The reward favors mnemonics with high sequence probability and low word overlap. To create mnemonic

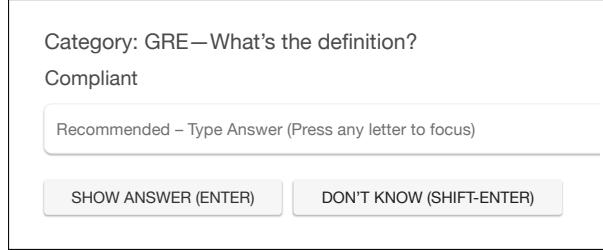


Figure 2: Screenshot from our web-based flashcard app after a user is presented a GRE vocabulary flashcard.

pairs, we sample 1000 terms  $\mathcal{V}_{pref} \subset \mathcal{V}$  not used in  $\mathcal{D}_{train}$ . For each term  $v \in \mathcal{V}_{pref}$ , we sample five mnemonics  $\mathcal{M} = \{m\}^5 \sim p_0(m | v)$  with 0.3 temperature. We take  $(m_A, m_B) \in \mathcal{M} \times \mathcal{M}$  with the best  $\pi_{pair}(m_A, m_B, v)$  score as the pair for  $v$ .

### 3.2 Preference Label Collection

While preferences are often elicited through crowd-worker sites, we decide to collect preferences from students who can learn vocab using our mnemonics. Flashcard software can aid mnemonic use (Tuite et al., 2012), so we host our user annotation schema within a flashcard app to gather student preferences  $y \in \{A, B, \text{tie}\}$  for the mnemonic pairs in §3.1.

Flashcards have two sides and while studying, users read the front of the card and answer what is on the back. In our app, users study flashcards  $f_v$  with a term  $v$  as the front of the card and type its definition on the back (Figure 2). In one session, users study a set of flashcards with 5 to 50 terms the user has not yet studied, and continue studying until they correctly type each term’s definition. We use TF-IDF (Sparck Jones, 1972) with a cutoff of 0.15 to check if the user’s typed definition matches the ground-truth definition, which the user can override if they disagree with the metric’s prediction. If a user answers  $f_v$  correctly, it is removed from the pool of cards left to study. If answered incorrectly, they see a mnemonic from the pair (§3.1) for  $v$  to aid learning (Figure 3), and the card remains in the set of cards to study. Thus, for each card  $f_v$ , we can count how many turns  $n \in \mathbb{Z}^+$  the user needed to answer  $f_v$  correctly. We use the KAR<sup>3</sup>L (Shu et al., 2024) model and flashcard learning interface to select the set of new flashcards to show users.

Researchers often use one method to collect preferences. But diverse methods, like pairwise comparisons and Likert ratings, can yield conflicts on which outputs are favored (Ethayarajh and Jurafsky, 2022; Bansal et al., 2024), and may also give complementary signals for LLM output quality (§5). To

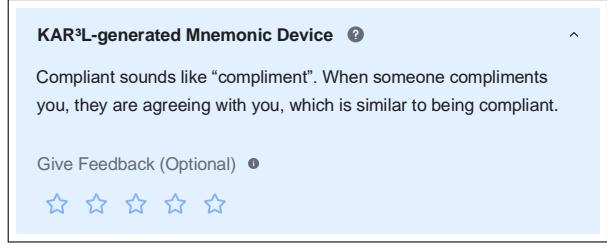


Figure 3: Screenshot of UI to collect Likert ratings.

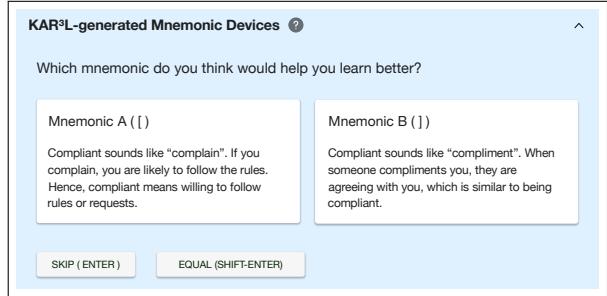


Figure 4: Screenshot of UI for pairwise comparisons.

see how diverse schema impact preferences in education, we collect three different preference labels grouped into two types: **expressed** and **observed**.

#### 3.2.1 Expressed Preferences

We define **expressed preferences** as those inferred from explicit user ratings—the most common preference type (Casper et al., 2023). We collect two expressed preferences: Likert ratings (Harpe, 2015) and pairwise comparisons (Bozóki et al., 2013).

For Likert ratings, if a user sees a mnemonic  $m_X$  after answering card  $f_v$  *incorrectly*, they rate  $m_X$  on a 5-Likert scale (Figure 3). We call this preference label  $y_{rate}$  and set  $y_{rate} = A$  if the average Likert rating of  $m_A$  is higher than  $m_B$  across users (same for B and tie). For each  $f_v$ , users see only one of  $m_A$  or  $m_B$ , so their rating cannot be biased by having already seen the other in the pair.

For pairwise preferences, if  $f_v$  is answered *correctly*, the user picks the mnemonic in  $(m_A, m_B)$  they think would help them learn best (Figure 4). Users can pick one mnemonic or mark them equal. We call this preference  $y_{pair}$  and set it to the most voted option. Order is shuffled for position biases.

#### 3.2.2 Observed Preferences

Expressed preferences measure the outputs users *think* are more helpful, but they do not capture what is *truly* more helpful for user goals (e.g. learning). Such preferences are undefined, so we propose **observed preferences**—those inferred from observed outcomes of user interactions with model outputs.

Preference Pairs	Raw Agreement	Sample Size
$(y_{pair}, y_{rate})$	0.675	80
$(y_{rate}, y_{learn})$	0.507	73
$(y_{pair}, y_{learn})$	0.407	59

Table 1: Raw agreement of preference types. Expressed preferences  $y_{pair}$  and  $y_{rate}$  have some disagreement, but agreement between expressed and observed preferences ( $y_{rate}$  vs  $y_{learn}$  and  $y_{pair}$  vs  $y_{learn}$ ) is even lower.

As keyword mnemonics can improve short-term recall (Wang et al., 1992), we base observed preferences on the mean turns  $t_X$  users studying with  $m_X$  need to correctly answer its flashcard  $f_v$ , as calculated in §3.2. This measure forms a proxy for short-term learning. For a given pair  $(m_A, m_B)$ , if  $t_A < t_B$ , we call this preference label  $y_{learn}$  and set  $y_{learn} = A$ , as  $m_A$  helps users learn the definition of  $v$  quicker than  $m_B$  (same for B and tie).  $y_{learn}$  is collected automatically as users study.

### 3.3 User Study Details

We deploy all mnemonics from  $\mathcal{D}_{pref}$  and have 47 English-speaking students from exam preparation forums, Google ads, and university courses study in our app. To filter noisy annotators, we add random quality checks in pairwise comparisons, where one mnemonic is clearly low-quality. If any user picks the low-quality mnemonic in the pairwise comparison, their preferences are omitted for analysis and model training. All users, including the two users who failed our quality checks, are awarded \$50. In three months, 45 students gave us 2684 preferences for 752 mnemonic pairs and 472 unique pairs (Table 5, details in Appendix A.2). We omit pairs with two or fewer labels so the mean labels per pair is 3.57, following the method of using three or more preferences to curb noise (Bai et al., 2022; Ji et al., 2024). Users are referred to just by numerical ID.

## 4 Preference Analysis

We study the relation of our preference types (§3.2) and uncover that students cannot fully predict what aids learning (§4.1, §4.2). Thus, we conclude that a mnemonic’s overall helpfulness cannot be captured by just one preference type, inspiring the design of our final model that combines all preferences (§5).

### 4.1 Are Preference Types Equivalent?

To see if our preference labels capture equivalent information, we compute the agreement of preferences (e.g.  $y_{pair}$  vs  $y_{rate}$ ) for the same mnemonics.

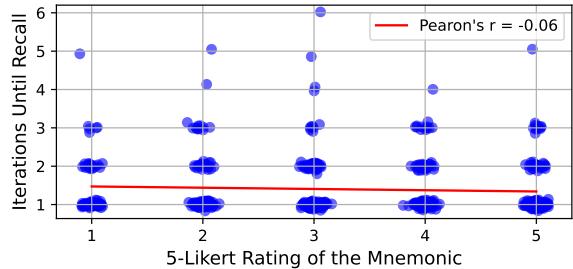


Figure 5: Correlation between user mnemonic ratings and turns needed for the same user to recall the term when studying with said mnemonic (jittered). Users cannot predict which mnemonics will best help them learn.

We exclude labels denoting a tie, focusing instead on labels that show a clear preference towards one mnemonic. Table 1 shows that the expressed preferences  $y_{pair}$  and  $y_{rate}$  have moderate agreement (0.675), aligning with Bansal et al. (2024), who also uncover disagreement in pairwise and rating preferences. But notably, the agreement between expressed and observed preferences is much lower (0.507 and 0.407), so asking students which outputs they think are more helpful does not always capture what is truly more helpful for learning.

### 4.2 Can Users Predict their Learning?

To study preference type disagreement at the student level, we see if a user’s rating of a mnemonic predicts the total turns needed for the user to learn the vocab term linked to that mnemonic. If users can predict this well, we would see a strong negative correlation, with lower ratings indicating more turns needed, but we find little ( $r = -0.06$ ) correlation (Figure 5). Prior work shows that students struggle to identify the *study strategies* that best aid learning (McCabe, 2011; Yan et al., 2016), and we uncover that students struggle to do the same for *study items*, further showcasing that expressed preferences cannot fully capture learning outcomes.

## 5 Training a SMART-er Mnemonic Model

Our goal of collecting student preference labels for mnemonic pairs is to identify more *helpful* or *effective* mnemonics, using this signal to guide SMART’s outputs. But as preference types disagree (§4), *how should we identify more effective mnemonics*?

Pairwise comparisons are typically used for this purpose, but they do not always match our goal of aiding learning (§4.1). Further, when pairwise preferences are missing or have a tie, we could draw from other preferences to break these ties and iden-

tify the better mnemonic, instead of discarding the pair for training (§6.2). Conversely, using observed preferences to select more effective mnemonics is promising as it matches our learning goal, but also using expressed preferences could help us avoid bizarre or offensive (Kroll and Tu, 1988) mnemonics that may still aid learning (see Appendix D.1).

Given the multi-faceted nature of mnemonic effectiveness, we develop a Bayesian model to learn effectiveness from all preferences and tune SMART via Direct Preference Optimization (Rafailov et al., 2024) on this learned effectiveness signal (§5.2).

### 5.1 Learning Mnemonic Effectiveness

For mnemonic pairs  $(m_A, m_B) \in \mathcal{D}_{pref}$ , we seek to find the more effective mnemonic. We intuit that mnemonic effectiveness is a latent value that can be modeled via feedback from all three of our preferences. Bayesian models are well-suited for this task as they capture annotator noise in feedback more effectively than aggregation (Wang et al., 2023).

Thus, we design a Hierarchical Bayesian model (Gelman and Hill, 2006) to estimate mnemonic effectiveness. We seek to learn  $\mathbb{P}(\theta_A > \theta_B)$ , the probability mnemonic  $m_A$  is more effective than mnemonic  $m_B$ . To do so, we model mnemonics  $m_{A,i}$  and  $m_{B,i}$  by latent effectiveness parameters  $\theta_{A,i}$  and  $\theta_{B,i}$ , which are assigned uniform priors:

$$\theta_{A,i}, \theta_{B,i} \sim \text{Beta}(1, 1). \quad (4)$$

We assume  $\theta_{A,i}$  and  $\theta_{B,i}$  influence observed feedback in our three preferences: pairwise  $y_{pair}$ ; rating  $y_{rate}$ ; and learning  $y_{learn}$ , which we outline below.

For **pairwise preferences**, let  $\mathcal{C}_i = \{c_1, \dots, c_n\}$  be the pairwise ratings of  $(m_{A,i}, m_{B,i})$ , where  $c_i \in \{\text{A, B, tie}\}$ . If  $\theta_{A,i} > \theta_{B,i}$ , we assume  $\mathcal{C}_i$  has more A preferences. To model  $\mathcal{C}_i$  from  $\theta_{A,i}$  and  $\theta_{B,i}$ , we compute the sigmoid ( $\sigma$ ) of a linear transform of  $\theta_{A,i}$  as the probability  $p_{A,i}^{pair} = \mathbb{P}(\text{A} \in \mathcal{C}_i)$ :

$$\alpha_{pair}, \beta_{pair} \sim \text{Normal}(0, 1), \quad (5)$$

$$p_{A,i}^{pair} = \sigma(\alpha_{pair} \cdot \theta_{A,i} + \beta_{pair}), \quad (6)$$

and same for  $p_{B,i}^{pair}$ . We then model  $\mathcal{C}_i$  as a Bradley-Terry model with ties (Davidson, 1970), where  $\mathbb{P}(\text{tie} \in \mathcal{C}_i)$  depends on a uniform latent value  $\tau$ :

$$\tau \sim \text{Beta}(1, 1), \quad (7)$$

$$p_i^{pair} = \frac{[p_{A,i}^{pair}; p_{B,i}^{pair}; \tau]}{p_{A,i}^{pair} + p_{B,i}^{pair} + \tau}, \quad (8)$$

$$\mathcal{C}_i \sim \text{Multinomial}(n, p_i^{pair}). \quad (9)$$

For **rating preferences**, let  $\mathcal{R}_{A,i} = \{r_1, \dots, r_5\}$  be cumulative counts of Likert ratings for  $m_{A,i}$ , where  $m_{A,i}$  has  $r_j$  votes less than or equal to rating  $j$ . We assume  $m_{A,i}$  with higher effectiveness  $\theta_{A,i}$  has higher ratings. We model  $\mathcal{R}_{A,i}$  as a multinomial distribution, parameterized by a linear transformation of  $\theta_{A,i}$  to a 5-length probability distribution:

$$\alpha_{rate}, \beta_{rate} \sim \text{Normal}(0, 1)^5, \quad (10)$$

$$p_{A,i}^{rate} = \sigma(\alpha_{rate} \cdot \theta_{A,i} + \beta_{rate}), \quad (11)$$

$$\mathcal{R}_{A,i} \sim \text{Multinomial}(\Sigma \mathcal{R}_{A,i}, p_{A,i}^{rate}), \quad (12)$$

and do the same for  $\mathcal{R}_{B,i}$ .

For **learning preferences**,  $\mathcal{T}_{A,i} = \{t_1, \dots, t_m\}$  is the distribution of turns users need to recall the term with  $m_{A,i}$ , where  $t_j \in \mathbb{Z}^+$ . We assume  $m_{A,i}$  with higher effectiveness  $\theta_{A,i}$  yields fewer  $t_j$  needed. Every turn count  $t_j$  is the tries until a success, so we model each  $t_j$  as a Geometric distribution parameterized by a linear transformation of  $\theta_{A,i}$ :

$$\alpha_{learn}, \beta_{learn} \sim \text{Normal}(0, 1), \quad (13)$$

$$p_{A,i}^{learn} = \sigma(\alpha_{learn} \cdot \theta_{A,i} + \beta_{learn}), \quad (14)$$

$$t_j \sim \text{Geometric}(p_{A,i}^{learn}), \quad (15)$$

and same for  $\mathcal{T}_{B,i}$ . We learn all variables via NUTS (Hoffman et al., 2014) for 1000 epochs. Parameters converge across five chains (Appendix B.4), meaning our model consistently estimates effectiveness.

### 5.2 Aligning SMART with Student Preferences

We now use the learned effectiveness of mnemonics  $(m_A, m_B)$  in the preference dataset  $\mathcal{D}_{pref}$  to align SMART. Among many alignment methods, we adopt Direct Preference Optimization (Rafailov et al., 2024, DPO), which tunes LLMs to preferences without requiring explicit reward modeling or reinforcement learning steps. Alternatives like Proximal Policy Optimization (Schulman et al., 2017) need extensive parameter tuning and are thus harder to reproduce (Huang et al., 2022b).

DPO requires dataset entries with a prompt  $x$  and winning/losing outputs  $y_w/y_l$ , where  $y_w/y_l$  are “good”/“bad” outputs for  $x$ . We set  $x$  to the term  $v$  in  $\mathcal{D}_{pref}$  with its mnemonics  $(m_A, m_B)$  as outputs. The mnemonic with higher effectiveness ( $\theta_A$  vs  $\theta_B$  from §5.1) is  $y_w$ , and the other is  $y_l$ . With this data, we update our initial model  $p_0(m | v)$  ( $\pi_0$  below) to align a better SMART model  $p_{dpo}(m | v)$  ( $\pi$  below) with DPO, which minimizes the loss  $\mathcal{L}_{dpo}$ :

$$\mathcal{L}_{dpo} = \mathbb{E}_{\substack{x, y_w, y_l \\ \sim \mathcal{D}_{pref}}} \left[ \ln \sigma \left( \beta \ln \frac{\pi(y_w | x)}{\pi_0(y_w | x)} - \beta \ln \frac{\pi(y_l | x)}{\pi_0(y_l | x)} \right) \right]. \quad (16)$$

SMART minimizes  $\mathcal{L}_{dpo}$  using QLoRA (Dettmers et al., 2023). Appendix B.1 lists all parameters.

## 6 How Smart Are SMART’s Mnemonics?

We now assess SMART’s mnemonics for 500 terms  $\mathcal{V}_{test} \subset \mathcal{V}$  not used in  $\mathcal{D}_{train}$  or  $\mathcal{D}_{pref}$ . SMART is aligned using a combination of three preference metrics: pairwise comparisons, Likert ratings, and learning. Due to space and data limits, we mainly evaluate via the most popular of the three metrics: pairwise comparisons (Casper et al., 2023). Thus, our evaluation reveals how using multiple preference labels (MPLs) affects *pairwise* output quality. We acknowledge that an evaluation across all preference metrics would be insightful (§9) and hope future works extend this direction with our datasets.

Given the costs of human pairwise evaluations, we adopt a common practice having GPT-4 judge which of two model-created mnemonics is higher quality (Chiang et al., 2023; Liu et al., 2023). GPT-4 has 80% agreement with users (Appendix B.3) on 200 held-out mnemonic pairs, near the 81% human agreement in MT-bench (Zheng et al., 2024), so GPT-4 agrees with user pairwise mnemonic ratings. To curb position bias (Wang et al., 2024), we compare mnemonics in both orders, only marking that one model wins if GPT-4 picks the model’s mnemonic in both orders, otherwise marking a tie.

We first use GPT-4 to compare mnemonic quality of SMART ablations (§6.1, §6.2). We then have mnemonic experts evaluate SMART’s mnemonics to inform future work (§6.3). We also present examples of SMART’s mnemonics in Appendix D.4.

### 6.1 Ablation Study

We ablate SMART (Figure 1) to verify that both fine-tuning and DPO improve mnemonic quality. Our fine-tuned model  $p_0(m|v)$  generates higher-quality mnemonics versus the few-shot LLaMA model  $p_{fs}(m|v)$  prompted using the ten  $\mathcal{D}_{ft}$  examples with the highest latent quality (§2.2), and same for  $p_{dpo}(m|v)$  versus  $p_0(m|v)$  (Table 2). Both of the steps improve pairwise mnemonic quality, confirming DPO can align LLMs with student preferences to enhance LLM outputs in education.

### 6.2 DPO with Multiple Preference Labels

We investigate the effectiveness of training with DPO using MPLs for pairwise mnemonic quality through two research questions, outlined below:

Model A/B Pair	A Wins	Tie	B Wins
$p_0(m v), p_{fs}(m v)$	<b>0.76*</b>	0.13	0.11
$p_{dpo}(m v), p_0(m v)$	<b>0.29*</b>	0.53	0.18

Table 2: GPT judgement of our ablations. Significantly better models (Binomial,  $p < 0.005$ ) are **bold** with \*. Our fine-tuning and DPO steps both improve SMART.

Model A/B Pair	A Wins	Tie	B Wins
$p_{bayes}(m v), p_{pair}(m v)$	0.19	0.60	0.21
$p_{dpo}(m v), p_{pair}(m v)$	<b>0.28*</b>	0.54	0.18

Table 3: GPT judgement of DPO models. Significantly better models (Binomial,  $p < 0.005$ ) are **bold** with \*. Multiple preferences can break ties in singular preferences for mnemonic quality gains (bottom row).

**Q1—Do MPLs harm pairwise metrics?** One concern of optimizing on MPLs with DPO is that the model will produce lower-quality mnemonics compared to a model using pairwise labels, as the latter optimizes just on the evaluation metric. To test this, we first select a subset of preference data  $\mathcal{D}_{pair} \subset \mathcal{D}_{pref}$ <sup>2</sup> with the pairwise preference  $y_{pair}$ . We then train two DPO models on  $\mathcal{D}_{pair}$  when  $y_{pair} \neq \text{tie}$ :  $p_{bayes}(m|v)$ , training on the Bayesian label  $y_{bayes}$  from §5.1, and  $p_{pair}(m|v)$ , training on  $y_{pair}$ . Despite having 20% disagreement in  $y_{pair}$  and  $y_{bayes}$  on the winning mnemonic, the two models are judged to generate mnemonics with equal quality on  $\mathcal{V}_{test}$  (Table 3, top). Thus, DPO training with MPLs does not always degrade LLM output quality on singular preference metrics.

**Q2—Can MPLs augment data?** If we train DPO with just  $y_{pair}$ , we must discard preference data when  $y_{pair} = \text{tie}$  or no  $y_{pair}$  exists. While we could collect more  $y_{pair}$  labels with another user study, we investigate whether  $y_{bayes}$  can directly resolve the missing or tied  $y_{pair}$  labels using the other preferences to elect winning mnemonics, augmenting our training data without collecting any more pairwise comparison data. We compare  $p_{pair}(m|v)$ , which trains on 348  $y_{pair}$  labels, to our full model  $p_{dpo}(m|v)$ , which trains on 117 extra pairs when  $y_{bayes}$  breaks a tie in  $y_{pair}$  and twelve extra pairs without  $y_{pair}$  labels.  $p_{dpo}(m|v)$  has significantly better ( $p < 0.005$ ) mnemonics than  $p_{pair}(m|v)$  (Table 3, bottom), meaning that MPLs can effectively augment DPO training data over pairwise preferences for output quality gains.

<sup>2</sup>To have enough label disagreement,  $\mathcal{D}_{pref}$  in this analysis also adds mnemonics with two labels (see Appendix A.3).

**Takeaway:** Since optimizing on  $y_{bayes}$  matches  $y_{pair}$  in non-ties and improves output quality by resolving ties and missing  $y_{pair}$  labels, we advise collecting MPLs if resources allow. This is feasible as MPLs can be collected in a single app, which is often cheaper than another user study to break ties, especially as some labels (e.g. *ylearn*) can be gathered without explicit annotations. Such efforts can help researchers study complementary alignment objectives (§4) and even boost LLM output quality.

### 6.3 Qualitative Evaluation

For a detailed evaluation, we have two mnemonic researchers assess our mnemonics, split into keyword and explanation quality. For *keyword quality*, we ask two yes/no questions: 1) Does the keyword sound like the term? (**Phonetic Similarity**); 2) Is the keyword simpler<sup>3</sup> than the term? (**Simplicity**). For *explanation quality*, we rate mnemonic explanations out of five on: 1) **Clarity**: Ease of understanding; 2) **Strength**: The obviousness of the explanation’s association of the keyword and the term (Hall et al., 1981); 3) **Imageability**: The ability to evoke mental imagery (Campos et al., 2011). We argue clarity and strength are most important for explanation quality, ensuring students can understand and create strong memory links from terms to keywords. We use imageability as it *can* affect memory (Groninger, 1971), but an imageable explanation is still unmemorable if it is unclear or low strength.<sup>4</sup>

We assess mnemonic keyword and explanation quality for 50  $\mathcal{V}_{test}$  terms created by a professional writer from Upwork, 10-shot GPT-4, and SMART,  $p_{dpo}(m | v)$ . We also compare SMART’s keyword quality to Transphoner (Savva et al., 2014), a SOTA mnemonic keyword extractor. Transphoner is hard to reproduce, so we use 50 terms and Transphoner outputs released by the authors for this comparison.

**Keyword Quality:** SMART has slightly better keywords than Transphoner (higher simplicity, equal PS), meaning LLMs are strong alternatives for keyword extractors (Figure 6, left). SMART also produces much simpler keywords than GPT-4, but with lower PS (Figure 6, middle). Suvarna et al. (2024) reveals a large gap (37% accuracy) when prompting LLaMA-2 13B and GPT-4 for rhyme generation which also assesses PS; our gap is just 8%, so fine-tuning and human feedback can help LLMs address

<sup>3</sup>A keyword is simpler if a user not knowing the term could likely know the keyword (*torpor* is not simpler than *torpid*).

<sup>4</sup>For example, *Ben Franklin* for *benevolent* in Figure 1.

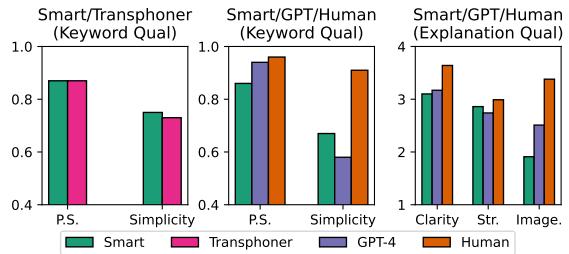


Figure 6: Expert qualitative evaluation of mnemonics. SMART matches GPT-4 at much lower deployment costs, but our human writer largely surpasses all models, especially in keyword simplicity and imageability.

phonetic weaknesses. Lastly, our human writer surpasses both models in keyword quality, with large simplicity gaps. Thus, systems that simplify text with LLMs (summarizers, topic models) may benefit from explicit feedback on word simplicity.

**Explanation Quality:** SMART matches GPT-4’s explanations (lower clarity, but higher strength), meaning student feedback lets smaller LLMs like SMART compete with large SOTA LLMs (Figure 6, right). Our writer again has the best explanations, especially in imageability over SMART, as SMART is not explicitly guided toward imageable outputs. Imageability does not entail memorability, but it may be useful for memorability tasks (story-telling, advertising) to use imageability feedback, as even GPT-4 shows a clear weakness in imageability.

**Takeaway:** Training LLMs with student feedback results in mnemonics with keyword quality matching SOTA extractive methods while also generating explanations. It also allows SMART, a smaller LLM, to compete with GPT-4. This is a significant feat, as SMART can provide students with mnemonics as effective as GPT-4 at much lower deployment costs—an attractive feature for educators looking to implement LLM educational tools. Further, since GPT-4 is stronger than LLaMA, using student feedback with GPT-4 could yield even better mnemonics. Lastly, our expert writes better mnemonics than LLMs, specifically in keyword simplicity and imageability. This motivates mnemonic generation as a challenging task and gives insights into feedback that could be used to guide LLMs in similar tasks.

## 7 Related Work

Below, we review relevant literature on mnemonic generation (§7.1) and human preferences (§7.2).

## 7.1 Mnemonic Generation

Mnemonics help users learn information, such as passwords (Yang et al., 2016; Song et al., 2019), vocabulary (Dundes, 1961; Levin et al., 1992), and medical facts (Ajayi et al., 2019; Leeds et al., 2020). There are many mnemonic types, including song (Hayes, 2009; Werner, 2018), acronym (McCabe et al., 2013; Li et al., 2021), and keywords (Campos et al., 2004, 2011). We study keyword mnemonics, which link complex terms to simpler keywords.

The effort of manually writing mnemonics has led to mnemonic generation research. Early works use phonetic similarity and multi-score ranking to find keywords (Savva et al., 2014; Anonthanasap and Leelanupab, 2015; Anonthanasap et al., 2015), but these methods do not explain how the keyword is linked to the fact. Recent works prompt LLMs to generate mnemonic explanations (Lee and Lan, 2023; Wong and Wolf, 2024), but we are the first to collect fine-tuning and preference data to generate mnemonics guided by real-world student feedback.

Prior education work has found that students cannot predict which study strategies, such as blocked versus interleaved practicing, will best help them learn (McCabe, 2011; Yan et al., 2016). We discover that students struggle to do the same for individual study items (§4.2), such as mnemonics.

## 7.2 Human Preferences

Recent work aligns LLMs with preference data capturing what humans prefer (Stiennon et al., 2020). Alignment methods include reinforcement learning with reward models (Christiano et al., 2017; Ziegler et al., 2019), selecting high-quality data (Sanh et al., 2022; Zhou et al., 2024), and augmenting LM loss with rewards (Yuan et al., 2023; Rafailov et al., 2024). Preferences have been used for sentiment generation (Maas et al., 2011), summaries (Völske et al., 2017), and dialogue safety (Bai et al., 2022), but we are the first to study them in mnemonics.

Our work also follows recent efforts to measure issues in preferences, such as preference agreement (Ethayarajh and Jurafsky, 2022; Bansal et al., 2024) and annotator biases (Peng et al., 2022; Wan et al., 2023). In contrast, we distinguish between expressed and observed preferences and show expressed preferences do not capture what truly helps users. We are also similar to works that collect preferences across varied demographics (Kirk et al., 2024) and use reinforcement learning regularization to align models to diverse preferences (Xue

et al., 2023), but we are the first to collect diverse preferences in an education setting and combine them via Bayesian modeling (Yang et al., 2024).

In social science, several works find that a human’s stated preferences, elicited from survey responses, do not always agree with revealed preferences, the human’s actual behavior (Urama and Hodge, 2006; Hoderlein and Stoye, 2014; De Corte et al., 2021). We show that the same applies in education, as pairwise comparisons and Likert ratings for perceived learning efficacy (expressed preferences) have low agreement (§4.1) with what truly helps users learn (observed preferences). Mozannar et al. (2024) find a similar trend in coding, where programmer preferences do not correlate with their actual productivity. As a result, we hope our work will lead researchers to reevaluate how we should measure helpfulness in preference data collection to design models that truly help users downstream.

## 8 Conclusion

We design SMART, the first keyword mnemonic generator guided by student feedback. SMART is trained on new fine-tuning and preference datasets, both of which are released. While curating data, we reveal low agreement in **expressed** preferences and our introduced **observed** preferences, showing that students cannot predict their learning. Combining expressed and observed preferences via DPO and Bayesian modeling yields a smaller, more efficient mnemonic model matching GPT-4. However, our human writer surpasses both models, especially in keyword simplicity and explanation quality, motivating mnemonic generation as a challenging task. To further challenge LLMs, researchers could explore personalizing mnemonics for students, adapting SMART to different domains, languages, and modalities, or optimizing on other learning signals (e.g. *long-term* learning) via student feedback.

While expressed preferences are currently used for alignment, they do not measure what truly helps users. Thus, there is a need for more work in capturing observed preferences and user goals. Ensuring LLMs pursue such goals safely requires alignment methods to steer LLMs toward *both* expressed and observed preferences, and we design a method to combine them; our method also resolves ties and missing labels to augment datasets. We hope our study of the disagreements and benefits of diverse mnemonic preferences will motivate future work in safely aligning LLM outputs to true user needs.

## 9 Limitations

One limitation is that our fine-tuning and preference datasets are relatively small. Despite this, our datasets both improve the quality of mnemonics from SMART (§6.1), following the recent paradigm of LIMA (Zhou et al., 2024) which suggests that small, high-quality datasets can be used to align and improve LLMs. Further, regardless of size, our preference dataset results in an insightful analysis of the relation between expressed and observed preferences (§4). If more students study vocab in our app, we will update and release both of our mnemonic datasets accordingly, resulting in larger datasets to facilitate mnemonic research.

GPT-4 as a judge can result in biases. We used GPT-4 due to the high cost of human annotations, which has also become a standard practice (Chiang et al., 2023; Liu et al., 2023; Touvron et al., 2023; Chiang and Lee, 2023; Dettmers et al., 2023). While using GPT-4, we adopt best practices for robust evaluation: 1) only evaluating on SMART to avoid self-recognition (Panickssery et al., 2024); 2) using DSPy to limit the sensitivity of prompt perturbations (Khattab et al., 2024); 3) ensuring GPT-4 has high agreement with humans on held-out mnemonic comparisons (Zheng et al., 2024); and 4) running inference on both orders of mnemonic pairs to curb position bias (Wang et al., 2024).

Lastly, our final model for SMART optimizes on a combination of multiple preference labels: pairwise comparisons, Likert ratings, and learning. While training and evaluating DPO models on all three objectives independently would be insightful, we are limited by our data. Thus, we focus on the most popular of the three—pairwise preferences—and our evaluation reveals how optimizing with multiple preference labels can impact metrics based on just one preference label. We encourage future work to use our preference data to explore how optimizing on multiple preferences impacts metrics based on each type of preference. To further motivate works in this direction, we have mnemonic experts evaluate SMART’s mnemonics to inform which types of feedback (e.g. keyword simplicity, explanation imageability) could be collected and trained with to improve mnemonics (§6.3).

## 10 Ethical Considerations

While optimizing directly on observed preferences for downstream applications is promising, we advise researchers to take caution for deploying such

models in a way that impacts users. Even our relatively harmless objective of optimizing on learning can result in bizarre mnemonics that perpetuate harms (Appendix D.1), and we imagine that in other domains, these consequences could be more severe. For example, training an LLM in the news domain optimized on user clicks can result in misinformation (Milano et al., 2020), while optimizing directly on time spent looking at an advertisement can yield harmful, addictive content.

We took precautions to avoid these harms, such as filtering out offensive mnemonics before fine-tuning (§2.3), and our protocols were approved by an Institutional Review Board to mitigate any risks during our user study. Thus, while we advocate for research in observed preferences for LLM alignment, we also urge researchers to consider the consequences of optimizing on these objectives before deployment in user-facing applications.

## Acknowledgements

We would like to thank members of the CLIP lab at the University of Maryland and external collaborators for their discussions of this work, including David Martinez, Naina Balepur, Sandra Sandoval, Yoo Yeon Sung, and Rachel Rudinger. We also thank GregMat, GMAT Club, and Memli App for their feedback on our user study design and for allowing us to use their testing materials in our research. This material is based upon work supported by the National Science Foundation under Grant No. IIS-2403436 (Boyd-Graber) and DGE-2236417 (Balepur). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Annotations and cloud computing resources were made possible by a gift from Adobe Research.

## References

- Toluwalase A Ajayi, David Shaw, and Kyle P Edmonds. 2019. Feasibility and effectiveness of a mnemonic approach to teach residents how to assess goals of care. *Journal of Palliative Medicine*, 22(6):696–701.
- Orapin Anonthanasap, Monticha Ketna, and Teerapong Leelanupab. 2015. *Automated english mnemonic keyword suggestion for learning japanese vocabulary*. In *2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pages 638–643.
- Orapin Anonthanasap and Teerapong Leelanupab. 2015.

Imnem: Interactive mnemonic word suggestion using phonetic algorithms. In *Proceedings of the 20th International Society on Artificial Life and Robotics, ser. AROB'15*, pages 316–321.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Hritik Bansal, John Dang, and Aditya Grover. 2024. **Peering through preferences: Unraveling feedback acquisition for aligning large language models**. In *The Twelfth International Conference on Learning Representations*.

Cindy Benge and ME Robbins. 2009. Using keyword mnemonics to develop secondary students' vocabularies: A teacher's action research. *Journal of Language and Literacy Education*, 6(1):93–104.

Sándor Bozóki, Linda Dezső, Attila Poesz, and József Temesi. 2013. Analysis of pairwise comparison matrices: an empirical research. *Annals of Operations Research*, 211:511–528.

Alfredo Campos, Angeles Amor, and María Angeles González. 2004. The importance of the keyword-generation method in keyword mnemonics. *Experimental psychology*, 51(2):125–131.

Alfredo Campos, Estefanía Camino, and María José Pérez-Faballo. 2011. Using the keyword mnemonics method among adult learners. *Educational Gerontology*, 37(4):327–335.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémie Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, Tony Tong Wang, Samuel Marks, Charbel-Raphael Segerie, Micah Carroll, Andi Peng, Phillip Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, Anand Siththanjan, Max Nadeau, Eric J Michaud, Jacob Pfau, Dmitrii Krasheninnikov, Xin Chen, Lauro Langosco, Peter Hase, Erdem Biyik, Anca Dragan, David Krueger, Dorsa Sadigh, and Dylan Hadfield-Menell. 2023. **Open problems and fundamental limitations of reinforcement learning from human feedback**. *Transactions on Machine Learning Research*. Survey Certification.

Cheng-Han Chiang and Hung-yi Lee. 2023. **Can large language models be an alternative to human evaluations?** In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%\* chatgpt quality. See <https://vicuna.lmsys.org> (accessed 14 April 2023), 2(3):6.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Roger R Davidson. 1970. On extending the bradley-terry model to accommodate ties in paired comparison experiments. *Journal of the American Statistical Association*, 65(329):317–328.

Ernest Davis. 2023. Benchmarks for automated commonsense reasoning: A survey. *ACM Computing Surveys*, 56(4):1–41.

Kaat De Corte, John Cairns, and Richard Grieve. 2021. Stated versus revealed preferences: An approach to reduce bias. *Health economics*, 30(5):1095–1123.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Alan Dundes. 1961. mnemonic devices. *Midwest Folklore*, 11(3):139–147.

Kawin Ethayarajh and Dan Jurafsky. 2022. **The authenticity gap in human evaluation**. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6056–6070, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Marilyn M Fairbanks. 1977. Vocabulary instruction at the college/adult levels: A research review. In *Proceedings of the Annual Conference of the Western College Reading Association*, volume 10, pages 19–29. Taylor & Francis.

Andrew Gelman and Jennifer Hill. 2006. *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press.

Lowell D Groninger. 1971. Mnemonic imagery and forgetting. *Psychonomic Science*, 23(2):161–163.

James W Hall, Kim P Wilson, and Richard J Patterson. 1981. Mnemotechnics: Some limitations of the mnemonic keyword method for the study of foreign language vocabulary. *Journal of Educational Psychology*, 73(3):345.

Spencer E Harpe. 2015. How to analyze likert and other rating scale data. *Currents in pharmacy teaching and learning*, 7(6):836–850.

Orla C Hayes. 2009. The use of melodic and rhythmic mnemonics to improve memory and recall in elementary students in the content areas. *Online Submission*.

Stefan Hoderlein and Jörg Stoye. 2014. Revealed preferences in a heterogeneous population. *Review of Economics and Statistics*, 96(2):197–213.

Janine Christin Hoffart, Sebastian Olschewski, and Jörg Rieskamp. 2019. Reaching for the star ratings: A bayesian-inspired account of how people use consumer ratings. *Journal of Economic Psychology*, 72:99–116.

Matthew D Hoffman, Andrew Gelman, et al. 2014. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623.

Jie Huang, Hanyin Shao, Kevin Chen-Chuan Chang, Jinjun Xiong, and Wen-mei Hwu. 2022a. [Understanding jargon: Combining extraction and generation for definition modeling](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3994–4004, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shengyi Huang, Rousslan Fernand Julien Dossa, Antonin Raffin, Anssi Kanervisto, and Weixun Wang. 2022b. [The 37 implementation details of proximal policy optimization](#). In *ICLR Blog Track*. <Https://iclr-blog-track.github.io/2022/03/25/ppo-implementation-details/>.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2024. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36.

Novall Khan. 2009. *Conquering Vocab: The Most Effective Way to Learn GRE Vocabulary in the Shortest Amount of Time*. Novall Khan.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan A, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2024. [DSPy: Compiling declarative language model calls into state-of-the-art pipelines](#). In *The Twelfth International Conference on Learning Representations*.

Hannah Rose Kirk, Alexander Whitefield, Paul Röttger, Andrew Bean, Katerina Margatina, Juan Ciro, Rafael Mosquera, Max Bartolo, Adina Williams, He He, et al. 2024. The prism alignment project: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language models. *arXiv preprint arXiv:2404.16019*.

Vince Kotchian. 2019. [Vince’s GRE Vocab Compilation and Curation](#). Online.

Neal EA Kroll and Shi-Fen Tu. 1988. The bizarre mnemonic. *Psychological Research*, 50(1):28–37.

Jaewook Lee and Andrew Lan. 2023. Smartphone: Exploring keyword mnemonic with auto-generated verbal and visual cues. In *International Conference on Artificial Intelligence in Education*, pages 16–27. Springer.

F Stuart Leeds, Kareem M Atwa, Alexander M Cook, Katharine A Conway, and Timothy N Crawford. 2020. Teaching heuristics and mnemonics to improve generation of differential diagnoses. *Medical education online*, 25(1):1742967.

Joel R Levin, Mary E Levin, Lynette D Glasman, and Margaret B Nordwall. 1992. Mnemonic vocabulary instruction: Additional effectiveness evidence. *Contemporary Educational Psychology*, 17(2):156–174.

Bin Li, Fei Xia, Yixuan Weng, Xiusheng Huang, Bin Sun, and Shutao Li. 2021. Psg: prompt-based sequence generation for acronym extraction. In *The AAAI-22 Workshop on Scientific Document Understanding*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. [G-eval: NLG evaluation using gpt-4 with better human alignment](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. [Learning word vectors for sentiment analysis](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Jennifer McCabe. 2011. Metacognitive awareness of learning strategies in undergraduates. *Memory & cognition*, 39:462–476.

Jennifer A McCabe, Kelsey L Osha, Jennifer A Roche, and Jonathan A Susser. 2013. Psychology students’ knowledge and use of mnemonics. *Teaching of psychology*, 40(3):183–192.

Mark A McDaniel and Michael Pressley. 1984. Putting the keyword method in context. *Journal of Educational Psychology*, 76(4):598.

Memliapp. 2007. [Mnemonic dictionary](#). Online.

Silvia Milano, Mariarosaria Taddeo, and Luciano Floridi. 2020. Recommender systems and their ethical challenges. *Ai & Society*, 35:957–967.

Hussein Mozannar, Valerie Chen, Mohammed Alsobay, Subhro Das, Sebastian Zhao, Dennis Wei, Manish Nagireddy, Prasanna Sattigeri, Ameet Talwalkar, and David Sontag. 2024. The realhumaneval: Evaluating large language models’ abilities to support programmers. *arXiv preprint arXiv:2404.02806*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Nihal V. Nayak, Tanmay Chinchore, Aishwarya Hanumanth Rao, Shane Michael Martin, Sagar Nagaraj Simha, G. M. Lingaraju, and H. S. Jamadagni. 2017. **V for vocab: An intelligent Flashcard application.** In *Proceedings of ACL 2017, Student Research Workshop*, pages 24–29, Vancouver, Canada. Association for Computational Linguistics.

Arjun Panickssery, Samuel R Bowman, and Shi Feng. 2024. Llm evaluators recognize and favor their own generations. *arXiv preprint arXiv:2404.13076*.

Andi Peng, Besmira Nushi, Emre Kiciman, Kori Inkpen, and Ece Kamar. 2022. Investigations of performance and bias in human-ai teamwork in hiring. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 12089–12097.

Zhongling Pi, Fangfang Zhu, Yi Zhang, and Jiumin Yang. 2021. An instructor’s beat gestures facilitate second language vocabulary learning from instructional videos: Behavioral and neural evidence. *Language Teaching Research*, page 13621688211039023.

Derek Powell, Jingqi Yu, Melissa DeWolf, and Keith J. Holyoak. 2017. **The love of large numbers: A popularity bias in consumer choice.** *Psychological Science*, 28(10):1432–1442. PMID: 28825874.

Michael Pressley, Joel R Levin, and Harold D Delaney. 1982. The mnemonic keyword method. *Review of Educational Research*, 52(1):61–91.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Michael R Raugh and Richard C Atkinson. 1975. A mnemonic method for learning a second-language vocabulary. *Journal of Educational Psychology*, 67(1):1.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multi-task prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Manolis Savva, Angel X Chang, Christopher D Manning, and Pat Hanrahan. 2014. Transphoner: Automated mnemonic keyword generation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3725–3734.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. **Proximal policy optimization algorithms.** *ArXiv*, abs/1707.06347.

Chantal Shaib, Joe Barrow, Jiuding Sun, Alexa F Siu, Byron C Wallace, and Ani Nenkova. 2024. Standardizing the measurement of text diversity: A tool and a comparative analysis of scores. *arXiv preprint arXiv:2403.00553*.

Matthew Shu, Nishant Balepur, Shi Feng, and Jordan Boyd-Graber. 2024. Karl: Knowledge-aware retrieval and representations aid retention and learning in students. *arXiv preprint arXiv:2402.12291*.

Vanlee Siriganjanavong. 2013. The mnemonic keyword method: Effects on the vocabulary acquisition and retention. *English Language Teaching*, 6(10):1–10.

Jianhua Song, Degang Wang, Zhongyue Yun, and Xiao Han. 2019. Alphapwd: A password generation strategy based on mnemonic shape. *IEEE Access*, 7:119052–119059.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. **Learning to summarize with human feedback.** In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc.

Ashima Suvarna, Harshita Khandelwal, and Nanyun Peng. 2024. **PhonologyBench: Evaluating phonological skills of large language models.** In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 1–14, Bangkok, Thailand. Association for Computational Linguistics.

Yufei Tian, Abhilasha Ravichander, Lianhui Qin, Ronan Le Bras, Raja Marjieh, Nanyun Peng, Yejin Choi, Thomas Griffiths, and Faeze Brahman. 2024. **MacGyver: Are large language models creative problem solvers?** In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5303–5324, Mexico City, Mexico. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwala Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Kathleen Tuite, Timothy Pavlik, Sandra B. Fan, Tyler Robison, Alexander Jaffe, Yun-En Liu, Erik Andersen, and Steven Tanimoto. 2012. **Picard: a creative and social online flashcard learning game**. In *Proceedings of the International Conference on the Foundations of Digital Games*, FDG '12, page 231–234, New York, NY, USA. Association for Computing Machinery.

Kevin C Urama and Ian D Hodge. 2006. Are stated preferences convergent with revealed preferences? empirical evidence from nigeria. *Ecological Economics*, 59(1):24–37.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. **TL;DR: Mining Reddit to learn automatic summarization**. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, Copenhagen, Denmark. Association for Computational Linguistics.

Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. 2023. Poisoning language models during instruction tuning. In *International Conference on Machine Learning*, pages 35413–35425. PMLR.

Alvin Y Wang, Margaret H Thomas, and Judith A Ouellette. 1992. Keyword mnemonic and retention of second-language vocabulary words. *Journal of Educational Psychology*, 84(4):520.

Jiashuo Wang, Haozhao Wang, Shichao Sun, and Wanjie Li. 2023. **Aligning language models with human preferences via a bayesian approach**. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. **Large language models are not fair evaluators**. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.

Riah Werner. 2018. Music, movement and memory: Pedagogical songs as mnemonic aids. *TESOL Journal*, 9(4):1–11.

Keith A Wollen and Matthew G Margres. 1987. Bizarreness and the imagery multiprocess model. In *Imagery and related mnemonic processes: Theories, individual differences, and applications*, pages 103–127. Springer.

Heidi Wong and Elissa Wolf. 2024. Large language model (llm) generated personalized mnemonics.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. **LESS: Selecting influential data for targeted instruction tuning**. In *ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models*.

Wanqi Xue, Bo An, Shuicheng Yan, and Zhongwen Xu. 2023. Reinforcement learning from diverse human preferences. *arXiv preprint arXiv:2301.11774*.

Veronica X Yan, Elizabeth Ligon Bjork, and Robert A Bjork. 2016. On the difficulty of mending metacognitive illusions: A priori theories, fluency effects, and misattributions of the interleaving benefit. *Journal of Experimental Psychology: General*, 145(7):918.

Adam X Yang, Maxime Robeyns, Thomas Coste, Jun Wang, Haitham Bou-Ammar, and Laurence Aitchison. 2024. Bayesian reward models for llm alignment. *arXiv preprint arXiv:2402.13210*.

Weining Yang, Ninghui Li, Omar Chowdhury, Aiping Xiong, and Robert W Proctor. 2016. An empirical study of mnemonic sentence-based password generation strategies. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 1216–1229.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. **RRHF: Rank responses to align language models with human feedback**. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhiuhuan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

## A Dataset Details

### A.1 Cleaning Noisy Mnemonics

#### Warning: This subsection contains an example of an offensive mnemonic.

We use Bayesian modeling to obtain a high-quality subset of student-submitted mnemonics on MnemonicDictionary for fine-tuning (§2.2), but we still find grammar mistakes in the mnemonics, and some of the mnemonics can be considered offensive or too culturally specific. To fix the grammar issues, we ask GPT-4 (web interface) to “Fix the spelling and grammar mistakes in these mnemonics [paste 50 mnemonic devices]”. This converts mnemonics like “when two gender’s male & female end up together .. they produce or give rise to a CHILD.” to the more structured form “When two genders, male and female, come together, they produce or give rise to a child.”

We also manually remove the 111 mnemonics we thought could be harmful to users or less understood due to being too specific to being related to a certain culture. For example, we consider the mnemonic: *Spurn sounds like 's + porn.'* The mnemonic implies disdainful rejection of inappropriate content. offensive, while the mnemonic: *Glower can be associated with Gulshan Grover, imagining him glaring angrily at the hero* is too culturally specific. In future works on mnemonic generation, it could be interesting to personalize the generated mnemonics through aspects like culture.

### A.2 $\mathcal{D}_{ft}$ and $\mathcal{D}_{pref}$ Dataset Details

In Tables 4 and 6, we provide descriptions of the columns in  $\mathcal{D}_{ft}$  and  $\mathcal{D}_{pref}$ , respectively. For  $\mathcal{D}_{pref}$ , we filter out terms and mnemonics with less than two annotations, resulting in the summary statistics described in §3.3, and summarized in Table 5. Along with the filtered subset, we will release the entire dataset of human preferences. We will also continue to release data if users study vocabulary with mnemonics in our app. The list of 500 vocab terms  $\mathcal{V}_{test}$  used for testing will also be released.

Finally, in Table 7, we quantify the noise of annotator ratings in  $\mathcal{D}_{pref}$ . We use average entropy and variance as each instance in our dataset can be annotated by a different number of annotators, and there is no guarantee that the same annotator will be rating each of the mnemonics. We find that the average variance and entropy of our annotations are significantly lower than random chance.

Our datasets are based on publicly available

GRE vocab words and mnemonics from MnemonicDictionary, and both were used as intended by the original authors. None of our datasets contain personal information, as users are referred to just by ID. All collected mnemonics are in English.

### A.3 Experiment Dataset Splits

When conducting the analysis of human preferences and training most DPO models, we filter all mnemonic pairs with less than two human preference annotations to form  $\mathcal{D}_{pref}$ . For the pairwise preferences, this means that we filtered mnemonics where the sum of votes for A and B was less than or equal to two. However, for the comparison between  $p_{bayes}(m|v)$  and  $p_{pair}(m|v)$  in §6.2, we also use mnemonic pairs with exactly two votes. We did this because otherwise, the Bayesian label  $y_{bayes}$  and pairwise label  $y_{pair}$  had very high agreement in which mnemonic in the pair was winning, so we would not be able to meaningfully study the differences between optimizing on just pairwise or multiple preferences. Essentially, our filtering strategy removed the annotator noise in the pairwise preferences, so the clear pairwise preference towards one mnemonic caused  $y_{bayes}$  to nearly always be in agreement with  $y_{pair}$ . However, by adding back noisy annotations, we were able to better study the differences between  $y_{bayes}$  and  $y_{pair}$ , as these two labels ended up with 20% disagreement.

## B Model and Experiment Details

### B.1 Training Setup

All of our models are trained for a maximum of 24 hours using eight NVIDIA RTX: A5000 GPUs. In practice, we find that both fine-tuning and Direct Preference Optimization (DPO) converge in around 6 hours. Parameters were manually selected. Fine-tuning and DPO were both implemented using the `trl` library<sup>5</sup> using a 90/10 train/evaluation split on their respective datasets. Both fine-tuning and DPO use QLoRA with LLaMA-2 (70B) with 8-bit quantization,  $\alpha = 32$ , a dropout of 0.05, a bias of 0, and update the default attention query and value projection layers. We use 5 training epochs for fine-tuning. We perform DPO with a training batch size of 1, a beta value of 0.1, a maximum prompt length of 16 tokens, a maximum output length of 64 tokens, and use the accuracy of the reward model as the metric for the best model. We use 5 training epochs for each DPO model (Table 3)

<sup>5</sup><https://huggingface.co/docs/trl>

All unspecified hyperparameters are the default values. All evaluations are from a single run.

## B.2 Decoding Strategy

We implement greedy decoding (no sampling) when generating mnemonics in §6. For an input vocabulary word  $v$ , we generate its mnemonic with the prompt: `### Term: $v$ \n### Mnemonic:  $v$  sounds like` to ensure the mnemonic follows the two-step process of: 1) linking  $v$  to a simpler keyword  $k$ ; and 2) generating an explanation that rationalizes how  $v$  and  $k$  are connected.

## B.3 GPT-4 Classifier Implementation

Our classifier to judge the quality of mnemonics is implemented with DSPy<sup>6</sup> (Khattab et al., 2024) and based on `gpt-4-turbo-2024-04-09`. To train and evaluate this classifier, we select 250 pairs of mnemonics for the same vocabulary term from MnemonicDictionary with the highest difference in our latent quality scores (§2.2), indicating a set of mnemonics with clear human preferences. We use 25 random training examples and 25 random validation examples to optimize this prompt with DSPy and after optimization, we run inference with the classifier on a held-out set of the 200 remaining examples. The prompt is optimized using bootstrap few-shot with random search with a maximum of 3 bootstrapped demos, a maximum of 3 labeled demos, and 10 candidate programs. As features, the classifier uses the two mnemonics to choose from, the vocab term, and a sample sentence containing the vocabulary word from WordsAPI.<sup>7</sup> This choice of inputs was selected by assessing validation set accuracy while adding different vocabulary features, including the definition of the word, synonyms, antonyms, and part-of-speech information. The instruction given in the DSPy signature is: *Given a vocabulary term, a sentence using the term, and two candidate mnemonics (Mnemonic A and Mnemonic B), classify whether Mnemonic A or Mnemonic B is a better mnemonic device. Output just the letter of the better mnemonic ("A" or "B").* Our classifier prompt will be released.

## B.4 Bayesian Model Evaluation

In this section, we evaluate the quality of our Hierarchical Bayesian model. Since our goal was to estimate the latent effectiveness of mnemonics, we

first assess the convergence of our learned parameters across chains. All learned parameters have an  $\hat{r}$  under 1.01 and an effective sample size over 1000, indicating strong convergence. Further, our final Bayesian preference label (which mnemonic has a higher effectiveness score) across chains has a Krippendorff's  $\alpha$  over 0.75, indicating strong agreement and convergence. Finally, in Figure 9, we display the log-likelihood values for our observed data across iterations, finding that they converge.

While convergence is more important to assess the quality of a Bayesian model that learns latent values, we also assess the generalizability of our model. We first train our model on 80% of our data and run inference on the remaining 20% for evaluation. In Figure 10, we compare the log-likelihood of predicting the observed data on the training and evaluation splits. For 3/5 of our observed data types, we find a non-significant difference between the log-likelihoods (2-sample  $t$ -test). The only significant difference is in the data associated with the learning preferences  $y_{learn}$ , further suggesting that modeling observed preferences is a challenging and interesting direction for future research.

## B.5 Obtaining Bayesian Preference Labels

To get the final latent overall mnemonic effectiveness values for training the DPO model  $p_{dpo}(m|v)$ , we average the latent variables over all five chains post-burn-in. For the ablation study where we compare optimizing on  $y_{pair}$  versus  $y_{bayes}$ , we similarly obtain the final latent mnemonic effectiveness values by averaging the latent variables over all five chains post-burn-in, but this time we just use a random sample of the epochs (i.e. thinning). By taking a sample of each chain instead of using the entire chains, we introduce more variability and disagreement in the labels, allowing us to more meaningfully study the difference between optimizing on pairwise preferences versus all preference labels.

## C User Study Details

### C.1 Annotator Instructions

During our user studies, we ensure to provide ample instructions to annotators. On the home page of our flashcard learning app, users can view our Institutional Review Board documents, which detail the purpose of the user study and how user data will be collected and used. Further, we provide users instructions to help them rate the quality of mnemonic devices (Figure 7), which can be viewed

<sup>6</sup><https://dspy-docs.vercel.app/>

<sup>7</sup><https://www.wordsapi.com/>

at any time throughout the user study. All annotators are English speakers. Users were aware that they were participants in a research study and as participation was voluntary, compensation is fair.

## C.2 Qualitative Evaluation Details

In our qualitative evaluation, we compare our full model trained with DPO on Bayesian labels versus 10-shot GPT-4 (gpt-4-turbo-2024-04-09), where examples were chosen according to the highest latent quality scores in  $\mathcal{D}_{train}$  (§2.2). We also compare against expert human-written mnemonics to serve as an upper bound on mnemonic quality. These mnemonics were written by a professional copyeditor and creative writer with a Bachelor of Science degree that we hired on Upwork. As part of the interview, we asked the writer to produce two sample mnemonics to ensure the mnemonics would be high quality, and the annotator was paid a high rate of \$3 per mnemonic (around \$60/hr), which is fair for the participant demographic.

Our annotators who rated the keyword and explanation quality of these mnemonics are both researchers in memory and mnemonic research (one post-doc and one assistant professor). The instructions given to these annotators are shown in Figure 8. Annotators were paid at a rate of \$50/hr, which is fair for the participant demographic. Our annotators showcase moderate agreement, highlighting the subjective nature of mnemonic generation (Table 8). Numerical tabular versions of the results from Figure 6 are in Tables 9 and 10.

## C.3 Learning Time Distribution

We provide the distribution of learning time, measured in turns needed to recall the definition of the vocabulary term, for the winning and losing mnemonics in the pair. As expected, the winning mnemonics have a significantly lower ( $p = 0.05$ ) average number of turns needed until the definition is recalled, showing a clear gap in short-term learning efficacy between mnemonics in our pairs.

## D Detailed Analysis and Results

### D.1 Offensive Mnemonics can Aid Learning

**Warning: This subsection contains an example of an offensive mnemonic.**

In §5, we describe our rationale for using both expressed and observed preferences; while observed preferences often reflect our downstream goal, expressed preferences ensure that this goal is

achieved in a safe manner. For example, if we were to optimize mnemonics just on learning, which is our downstream goal, we may produce bizarre or offensive mnemonics, since these mnemonics have been shown to help students learn (Wollen and Margres, 1987). However, expressed preferences are a more reliable method to detect these offensive mnemonics, and is thus likely why the majority of LLM alignment methods for dialogue safety rely on expressed preferences as training data.

To illustrate this point, we present the following mnemonic generated by the initial SMART model during the user study, which was flagged as one of two offensive mnemonics: *Obtuse sounds like "abuse". If you abuse someone, they may not understand the situation, just like an obtuse person who is slow to understand.* As noted by our annotator, this mnemonic: *"may be insensitive to people who have felt abuse in their lives and feel as if the mnemonic is calling them slow"*. Through expressed preferences, this mnemonic received a Likert rating of 1 and received 0 votes in the pairwise comparison. However, with observed preferences, the student studying with this mnemonic learned the term in just one iteration. Thus, while observed preferences would suggest that this mnemonic is highly effective for learning, the expressed preferences show that this mnemonic may be offensive or harmful to users, motivating our use of all preference labels for enhancing mnemonic generation.

### D.2 Are Bayesian Tie-Breaks Good?

We previously found that using multiple preference labels to break ties within singular preference labels, improving LLM output quality with DPO (§6.2). To ensure that these ties are better than random tie breaks, we use our GPT-4 classifier to compare the winning and losing mnemonics from our Bayesian labels (i.e. mnemonics with higher and lower effectiveness) where the majority vote in the pairwise setting is Tie. On the set of ties, GPT-4 states that our “winning” mnemonic is better 40% of the time, tied with the losing mnemonic 31% of the time, and is worse 29% of the time. Thus, even though humans marked these LLM outputs as tied, we were able to draw from other preference labels to identify a winning mnemonic, and GPT-4 also tends to agree that this mnemonic is higher quality.

For context, when students mark a mnemonic as “winning” from pairwise comparisons (i.e. non-tie), GPT-4 states this mnemonic is higher-quality

than the alternative in 51% of cases, tied in 22% of cases, and is lower-quality in 26% of cases.

### D.3 DPO Models versus Fine-Tuning

In Table 12, we use GPT-4 to judge the mnemonics produced by each of the DPO configurations used in §6.2 versus the SMART model just using fine-tuning  $p_0(m|v)$ . We find that DPO improves the mnemonic quality of each of these models.

### D.4 Mnemonic Examples

In Table 13, we provide examples of high-quality keyword mnemonic devices generated by our final SMART model. We also show some examples of low-quality mnemonics from our model and highlight areas for improvement in Table 14.

Column	Description	Num Unique
term	Vocabulary term	822
mnemonic	Mnemonic for the vocabulary term	889

Table 4: Descriptions of columns in  $\mathcal{D}_{train}$ .

<u>Examples of Good and Bad Mnemonics</u>	
<b>Good Mnemonic</b>	<ul style="list-style-type: none"> <li>• Good keyword link</li> <li>• Clear explanation</li> <li>• Easy to understand</li> <li>• Memorable</li> <li>• Correct definition</li> </ul>
<b>Incorrect Definition</b>	Pithy sounds like "witty." Thus, pithy means clever or smart
<b>Unclear</b>	Pithy sounds like "pity." A person who is pitied is often pitied and pitied people are given short messages
<b>Circular Keyword</b>	Pithy sounds like "pith"...
<b>Low Memorability (Subjective)</b>	Pithy sounds like "pit." Pits can be short. Pithy means brief.
<b>Keyword Doesn't "Sound Like" Term</b>	Pithy sounds like "short"...
<b>Poor Keyword Explanation</b>	Pithy sounds like "pity." People are often pitied when they mess up. Pithy means brief and meaningful.

Figure 7: Instructions given to students to help them rate the quality of mnemonic devices for expressed preferences.

Task Overview
In this evaluation task, you will be presented with a term and an associated mnemonic (or just a keyword). Your role is to assess the quality of the keyword used in the mnemonic and the quality of the mnemonic's explanation.
<b>Part 1: Keyword Quality Assessment</b>
For each term and mnemonic provided, answer the following yes/no questions to evaluate the keyword quality:
<ul style="list-style-type: none"> <li>- <u>Sounds Like Original</u>: Does the keyword sound like the original term?</li> <li>- <u>Too Similar to Original</u>: Is the keyword too similar to the original term?</li> <li>- <u>Simpler than Original</u>: Is the keyword simpler than the original term?</li> </ul>
<b>Part 2: Explanation Quality Assessment</b>
Rate the following aspects of the mnemonic's explanation on a Likert scale from 1 to 5 (where 1 is the lowest and 5 is the highest):
<ul style="list-style-type: none"> <li>- <u>Clarity / Understandability</u>: How clear and understandable is the explanation? Are there significant grammar or coherence issues?</li> <li>- <u>Associative Strength</u>: How strong is the association between the keyword and term in the explanation? How directly and effectively does the explanation link the term and keyword?</li> <li>- <u>Imageability</u>: How well can you visualize the explanation? Does it create a mental image?</li> </ul>

Figure 8: Instructions to expert educators when rating the keyword and explanation quality of mnemonic devices.

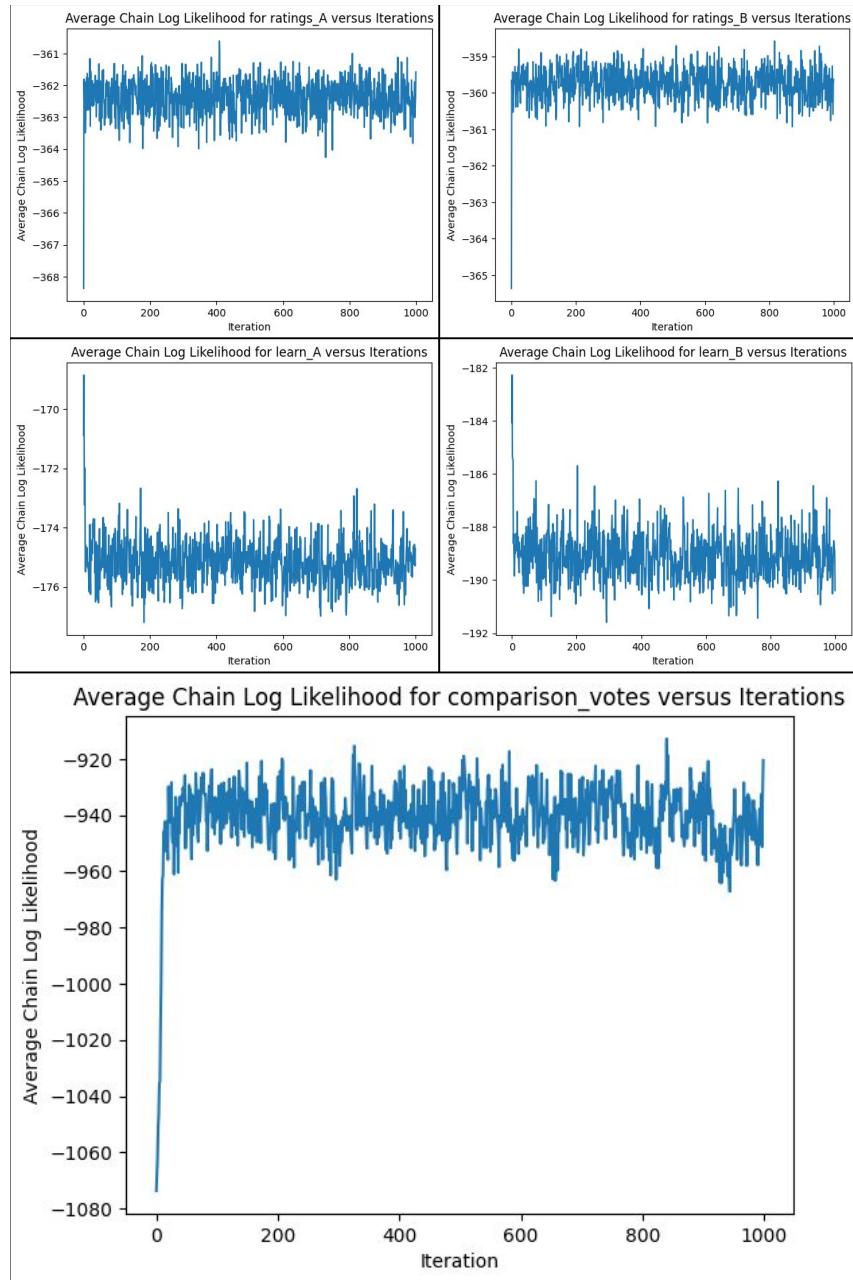


Figure 9: Log likelihood convergence of observed data in our Bayesian model.

Preference Type	# Annotations	# Pairs	Average Annotations / Pair
Pairwise ( $y_{pair}$ )	1693	460	3.68
Rating ( $y_{rate}$ )	389	121	3.21
Learning ( $y_{learn}$ )	602	170	3.54

Table 5: Summary statistics of preference annotations.

Column	Description	Num Unique
term	Vocabulary term	472
mnemonic_A	Mnemonic A for the vocabulary term	472
mnemonic_B	Mnemonic B for the vocabulary term	472
pairwise_A_votes	Number of users who picked Mnemonic A in the pairwise comparison	11
pairwise_B_votes	Number of users who picked Mnemonic B in the pairwise comparison	9
pairwise_tie_votes	Number of users who picked “tie” in the pairwise comparison	5
A likert_ratings	List of Likert ratings from 1-5 denoting the quality of Mnemonic A	5
B likert_ratings	List of Likert ratings from 1-5 denoting the quality of Mnemonic B	5
A_learn_iterations	List of turns from 1 to $\infty$ the student needed to learn the term with Mnemonic A	7
B_learn_iterations	List of turns from 1 to $\infty$ the student needed to learn the term with Mnemonic B	8

Table 6: Descriptions of columns in  $\mathcal{D}_{pref}$ .

Feedback	Metric Used	Preference Agreement	Random Agreement
Comparison	Average Entropy	0.802	1.222
Rating	Average Variance	0.778	1.331
Learning	Average Variance	0.323	2.179

Table 7: Quantifying annotation noise in  $\mathcal{D}_{pref}$ .

Model Qualitative Comparison		Cohen's $\kappa$
SMART vs Transphoner (PS)		0.381
SMART vs Transphoner (Simplicity)		0.526
SMART vs GPT-4 vs Human (PS)		0.497
SMART vs GPT-4 vs Human (Simplicity)		0.538
SMART vs GPT-4 vs Human (Clarity)		0.428
SMART vs GPT-4 vs Human (Strength)		0.356
SMART vs GPT-4 vs Human (Imageability)		0.681

Table 8: Agreement of our annotators across all qualitative evaluations (model comparisons and aspects). We measure agreement through Cohen  $\kappa$ , using quadratic weighting for the Likert scale ratings.

Keyword Quality		
Model	Phonetic Similarity	Simplicity
SMART	0.87	0.75
Transphoner	0.87	0.73

Table 9: Numerical tabular version of qualitative evaluation of SMART versus Transphoner from Table 6.

Model	Keyword Quality		Explanation Quality		
	Phonetic Similarity	Simplicity	Clarity	Strength	Imageability
SMART	0.86	0.67	3.10	2.86	1.91
GPT-4	0.94	0.58	3.17	2.74	2.52
Human	0.96	0.91	3.64	2.99	3.38

Table 10: Numerical tabular version of qualitative evaluation of SMART versus GPT-4 versus Humans from Table 6.

Percentile	Winning Mnemonic Turns	Losing Mnemonic Turns
0	1.00	1.25
25	1.00	1.60
50	1.00	2.00
75	1.50	2.00
100	3.00	7.00
<b>Average</b>	<b>1.24</b>	<b>2.13</b>

Table 11: Distribution of learning time (Turns) for winning and losing mnemonics.

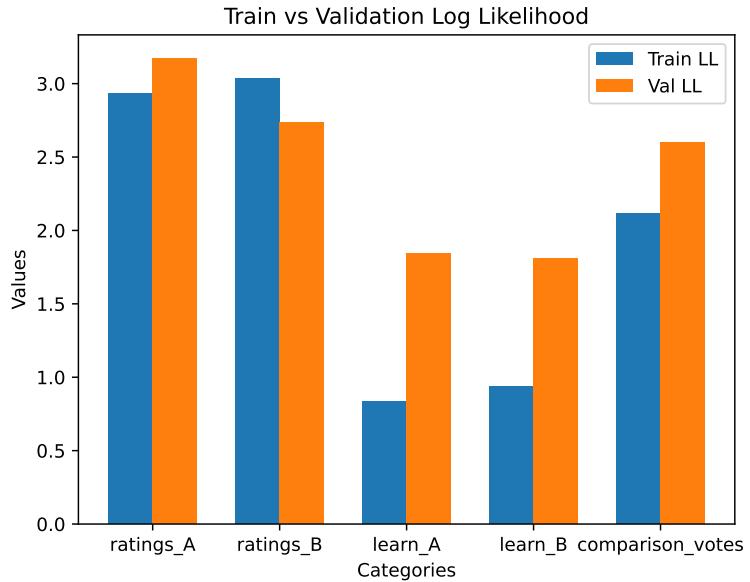


Figure 10: Comparison of Bayesian model log likelihood of observed data on training versus validation set.

Model A/B Pair	A Wins	Tie	B Wins
$p_{pair}(m v)$ (Table 3, top), $p_0(m v)$	<b>0.26</b>	0.54	0.20
$p_{bayes}(m v)$ (Table 3, top), $p_0(m v)$	<b>0.22</b>	0.58	0.20
$p_{pair}(m v)$ (Table 3, bottom), $p_0(m v)$	<b>0.24</b>	0.58	0.18
$p_{dpo}(m v)$ (Table 3, bottom), $p_0(m v)$	<b>0.29</b>	0.53	0.18

Table 12: GPT judgement of DPO models versus fine-tuning. The better model is in **bold**.

Term	Generated Keyword Mnemonic
Lionized	Lionized sounds like "lion-eyes," envisioning a lion being admired for its eyes. Lionized means to be admired or treated like a celebrity.
Empiricism	Empiricism sounds like "emperor"; an emperor may rely on experience rather than theory to rule. Empiricism is based on experience rather than theory.
Dwindling	Dwindling sounds like "wind" + "lingering"; when the wind lingers, it gradually decreases in strength. This relates to dwindling, which means decreasing in size, amount, or strength.
Glowered	Glowered sounds like "glow" + "red". Imagine a red glowing face, which could represent anger or a glowered expression.
Static	Static sounds like "stick", which is stationary or unmoving. Static electricity is also stationary, as it does not move. Both relate to the concept of being stationary or unmoving.
Sanctimony	Sanctimony sounds like "saint" + "money"; saints are supposed to be holy, but some may be hypocritical and greedy for money. This hypocrisy and false piety is sanctimony.
Goosebumps	Goosebumps sounds like "ghost bumps," which can happen when you see a ghost. Goosebumps are bumps on the skin caused by fear or cold.
Resurgence	Resurgence sounds like "re-surgery"; after a surgery, one may experience a resurgence of pain. Resurgence refers to a renewed increase or rise..
Moot	Moot sounds like "mute"; when something is mute, it's debatable whether it's on or off. Similarly, a moot point is debatable.
Subordinate	Subordinate sounds like "sub-order", implying a lower rank or position. Subordinates are lower in rank or position, and are subject to the authority of others.
Upbraided	Upbraided sounds like "up" + "braid". Imagine a teacher scolding a student for not braiding her hair properly. This scolding or reprimanding is upbraiding.
Despoiled	Despoiled sounds like "despite spoiled." Imagine someone spoiling your plans despite your efforts, leaving you feeling despoiled.
Caterwauling	Caterwauling sounds like "cat + war + yelling," imagining a cat yelling during war. Caterwauling refers to loud, unpleasant noise.
Escalate	Escalate sounds like "escalator," which goes up, representing an increase or rise. Escalate means to increase or rise in intensity or severity.
Compromise	Compromise sounds like "com + promise". When you make a promise, you may have to compromise on something to keep it. For example, you may have to compromise on your sleep to keep a promise to meet someone early in the morning.
Deftness	Deftness sounds like "leftness," which can be associated with left-handed people who are often known for their dexterity or deftness. This can help remember the meaning of deftness as skillful or adroit.
Haptic	Haptic sounds like "happy tic", which can be associated with a happy tickle, relating to touch. Haptic refers to the sense of touch.
Parenthetical	Parenthetical sounds like "parenthesis," which is used to add extra information in a sentence. Parenthetical means providing additional information in a sentence.
Fillip	Fillip sounds like "fill up"; when you fill up a glass, it gives a sudden push to the liquid, similar to a fillip. A fillip is a sudden push or stimulus.
Kowtow	Kowtow sounds like "cow tow"—imagine a cow towing a car, symbolizing submission or obeisance. Kowtow means to show respect or submission by bowing or kneeling.

Table 13: Examples of 20 high-quality keyword mnemonics generated by our final model  $p_{dpo}(m|v)$ .

Term	Generated Keyword Mnemonic
Affiliated	Affiliated sounds like " <b>affiliation</b> ," which is a connection or association with a group or organization. Being affiliated means being connected or associated with something.
Evinced	Evinced sounds like "evidence"; evidence is proof, which is what evinced means. <b>Evidence is also a synonym of evinced</b>
Complicit	Complicit sounds like " <b>compliant</b> " + " <b>licit</b> " (licit means legal). If someone is compliant with legal actions, they are involved in them. Complicit means involved in a wrongdoing.
Quintessential	Quintessential sounds like "quint" (five) + "essential". <b>Five essentials are required to make a perfect dish</b> , representing the quintessential.
Peons	Peons sounds like " <b>pions</b> " ( <b>pions are subatomic particles</b> ). Pions are subatomic particles, so peons are subordinate workers.

Table 14: Examples of low-quality keyword mnemonics generated by our final model  $p_{dpo}(m|v)$ . Prominent issues and areas for improvement are highlighted in **red**.