

Rabbit-AAC: Uma Arquitetura Híbrida para Recomendação de Texto em Comunicação Aumentativa e Alternativa

Fernando Torres Ferreira da Silva
Universidade de São Paulo
São Carlos, Brasil
fernandotfs@usp.br

Bruna C. R. Cunha
Universidade de São Paulo
São Carlos, Brasil
brunaru@icmc.usp.br

ABSTRACT

Conditions such as Cerebral Palsy (CP) and Amyotrophic Lateral Sclerosis (ALS) severely impair oral communication, making Augmentative and Alternative Communication (AAC) essential for supporting user autonomy. However, the low typing speed of virtual keyboards, particularly when using indirect access methods such as eye-tracking, remains a major limitation and contributes to high rates of AAC technology abandonment. To address this challenge, we propose and evaluate Rabbit-AAC, a hybrid architecture for text recommendation in AAC systems. The proposed approach integrates three complementary strategies: (i) a low-latency autocomplete mechanism based on a Trie data structure and a frequency-based lexical database to accelerate intra-word typing; (ii) an optimized reranking module leveraging different categories of Large Language Models (LLMs); and (iii) parameter-efficient fine-tuning using Low-Rank Adaptation (LoRA) on the two most efficient models. Extensive experiments on the synthetic AAC conversation dataset (willwade/AACConversations) show that the BERTimbau + LoRA + Trie architecture achieves a Mean Reciprocal Rank (MRR) of 0.730, Top-5 accuracy of 81.1%, and an average latency of 51 ms, providing the best balance between semantic accuracy and real-time performance. In contrast, the multi-token reranking strategy based on GPT-2 + LoRA achieves a Top-5 accuracy of 99% but incurs a prohibitive average latency of 1.28 s, highlighting the importance of carefully balancing effectiveness and responsiveness when designing AI-powered AAC systems. These results demonstrate that combining classical data structures with parameter-efficient fine-tuning offers a viable path toward real-time, context-aware AAC systems.

KEYWORDS

augmentative and alternative communication, text prediction, hybrid architecture, bertimbau, lora, trie, reranking, assistive technology, natural language processing, eye-tracking

1 INTRODUÇÃO

A comunicação é a base da experiência humana, pois nos permite expressar necessidades, compartilhar emoções, construir relacionamentos e participar plenamente da sociedade, abrangendo desde comunidades locais até questões globais [13]. No entanto, para milhões de pessoas, condições como a Paralisia Cerebral (PC), Esclerose Lateral Amiotrófica (ELA), Atrofia Muscular Espinhal (AME), Acidente Vascular Cerebral (AVC) ou danos neurológicos decorrentes de traumas e acidentes, podem impor barreiras severas à comunicação oral. Nesses casos, embora as capacidades cognitivas e linguísticas permaneçam intactas, a habilidade da fala oral é drasticamente reduzida ou completamente perdida. Um exemplo emblemático é o do jornalista Jean-Dominique Bauby, que em 1995 acordou de um coma, decorrente de um AVC (Acidente Vascular Cerebral), acometido pela síndrome do encarceramento (*Locked-in syndrome*). Bauby ficou restrito a apenas o piscar do olho esquerdo para comunicar-se e para ditar seu livro “O Escafandro e a Borboleta” [2].

O uso de recursos computacionais de Tecnologia Assistiva (TA), especificamente no contexto da Comunicação Aumentativa e Alternativa (CAA), surgem como uma solução fundamental para pessoas com limitações motoras e de fala. A CAA engloba um conjunto de ferramentas e estratégias que visam complementar (aumentar) ou substituir (prover uma alternativa) a fala para indivíduos com Necessidades Complexas de Comunicação (NCC). Apesar dos avanços significativos, as soluções de CAA enfrentam um desafio persistente: a velocidade de comunicação. Enquanto uma conversa natural ocorre a uma taxa de 150 a 200 palavras por minuto (PPM), a comunicação via CAA, especialmente com interfaces de acesso indireto como o rastreamento ocular, opera em uma faixa drasticamente inferior, tipicamente entre 2 e 20 PPM [25]. Essa assimetria temporal não é apenas uma questão de eficiência; ela tem profundas implicações sociais e psicológicas. Conversas em grupo mudam de tópico mais rápido do que um usuário de CAA consegue formular sua contribuição, resultando em uma crescente frustração [26]. Essa dificuldade leva a uma alta taxa de abandono tecnológico. Segundo Simpson et al. [23], aproximadamente um terço das soluções assistivas são abandonadas, frequentemente por má adequação às necessidades do usuário e por uma experiência de uso ineficiente e frustrante.

Para mitigar essa barreira, a predição de texto tornou-se um componente indispensável em teclados virtuais para CAA. Contudo, os sistemas tradicionais, baseados em modelos estatísticos simples (como n-gramas), frequentemente

falham em capturar o rico contexto semântico da conversa. A ascensão dos modelos de linguagem de larga escala (LLMs) gerativos e autorregressivos, como a família GPT (*Generative Pre-trained Transformer*), pode oferecer uma oportunidade sem precedentes para revolucionar a predição de texto, excedendo a mera probabilidade estatística para alcançar uma coerência contextual profunda. Todavia, o custo temporal de predição de um LLM justifica o seu uso na predição de texto em teclados de CAA? A fim de responder essa questão de pesquisa, este trabalho propõe, desenvolve e avalia uma arquitetura híbrida, denominada Rabbit-AAC, que visa combinar a eficiência de estruturas de dados clássicas com o poder contextual de modelos especializados. O sistema integra duas estratégias complementares para espelhar o processo cognitivo da digitação:

- **Predição Intra-palavra (Autocompletar):** Utiliza uma estrutura de dados *Trie* para oferecer sugestões de baixa latência e acelerar a digitação da palavra atual.
- **Predição Inter-palavra (Contextual):** Emprega um modelo de linguagem de larga escala, especializado via *fine-tuning* utilizando a técnica (*Low-Rank Adaptation*) (*LoRA*) em um *corpus* sintético de CAA, para prever a próxima palavra com base no contexto semântico de toda a frase.

O objetivo geral deste trabalho é avaliar esse sistema híbrido de recomendação de texto para teclados virtuais, a fim de propor uma solução de CAA mais eficiente, acessível e contextualmente consciente. Nesse sentido, o sistema foi avaliado com 12 modelos de linguagem distintos, incluindo BERTimbau, GPT-2, Gemma-2-2B, Tucano, Llama-3.2-1B-Instruct e a *Trie* isolada como *baseline*. Os resultados obtidos permitiram identificar as soluções mais adequadas para testes empíricos, considerando o equilíbrio entre acurácia e latência.

Além da contribuição em PLN aplicado à CAA, este trabalho relaciona-se às áreas de Web e Multimídia ao investigar a integração entre uma interface Web de entrada assistiva, rastreamento ocular baseado em vídeo e recomendação textual em tempo de interação. O teclado EyeType utiliza a biblioteca WebGazer.js para estimar, no navegador e por meio de uma webcam convencional, a direção do olhar utilizada na seleção das teclas. Nesse contexto, a latência da arquitetura de predição não representa apenas uma medida de desempenho computacional, mas um requisito para a responsividade de uma interface Web continuamente acionada pelo olhar. Assim, o Rabbit-AAC contribui para a investigação de arquiteturas Web interativas que combinam entrada visual e processamento de linguagem em aplicações assistivas.

O artigo está organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados à Comunicação Aumentativa e Alternativa, à predição de texto e ao uso de modelos de linguagem nesse domínio. A Seção 3 descreve a metodologia adotada, incluindo a arquitetura híbrida Rabbit-AAC, a implementação do teclado virtual EyeType, o conjunto de dados sintéticos utilizado e o procedimento experimental. A Seção 4 apresenta e discute os resultados obtidos, contemplando o desempenho dos modelos, a relação entre acurácia

e latência, as matrizes de confusão e a análise do processo de treinamento. Por fim, a Seção 5 reúne as conclusões, as limitações do estudo e as perspectivas para trabalhos futuros.

2 TRABALHOS RELACIONADOS

A CAA tem sido objeto de pesquisa há décadas, com foco em ferramentas que auxiliem indivíduos com dificuldades severas de comunicação. Simpson et al. [23] documentaram que aproximadamente um terço das soluções de alta tecnologia de Tecnologia Assistiva são abandonadas por seus usuários, frequentemente devido à má adequação entre a tecnologia e as necessidades do usuário, configurações complexas e experiência de usuário ineficiente. Considerando a tarefa de comunicação, Griffiths et al. [8] introduzem o conceito de “competência comunicativa”, dividida em quatro áreas: operacional, linguística, social e estratégica. A complexidade de dominar essas competências, aliada a uma experiência frequentemente lenta e frustrante, contribui para o abandono tecnológico.

Para alguns usuários com NCCs, a digitação em teclados digitais só é possível por meio do método de varredura ou de um dispositivo de rastreamento ocular, o que prejudica em muito a velocidade de digitação. Justificados pela importância de uma predição de texto otimizada em sistemas de CAA e considerando o progresso dos LLMs e da Inteligência Artificial Generativa (IAG) em particular, trabalhos mais recentes investigam como os modelos atuais podem ser incorporados na predição de texto. Valencia et al. [25] investigaram o uso de modelos GPT para gerar frases completas de forma automática, descobrindo que, embora a abordagem reduza a carga motora, os usuários sentem que as frases geradas perdem seu estilo e preferências pessoais. Esse achado é crucial, sublinhando a necessidade de sistemas que auxiliem a digitação de forma granular (palavra por palavra), preservando a autoria e a personalidade do indivíduo. Ainda considerando uma abordagem similar, Valencia et al. [26] propuseram o *COMPA*, que permite ao usuário marcar explicitamente trechos da conversa para gerar “frases iniciais” que ancoram sua resposta, resolvendo o problema do “fora de contexto”. Esses trabalhos demonstram o potencial de LLMs para CAA, mas focam principalmente na geração automática de frases completas, o que pode comprometer a identidade comunicativa.

No contexto de predição de texto, sistemas baseados em *n*-gramas ou em modelos de linguagem estatísticos são comuns, mas falham em capturar dependências de longo alcance. A aplicação de LLMs à CAA tem crescido, com trabalhos como o de Shen et al. [22], que utiliza GPT-2 para construir sentenças completas a partir de poucas palavras-chave, condicionado pelo histórico da conversa e pelo perfil do usuário. Em contraste, nossa abordagem foca na predição granular (palavra por palavra), preservando a autoria.

Arquiteturas híbridas têm sido exploradas em sistemas de recomendação, o que indica um potencial na área de TA. Pagnossim [16] desenvolveu uma arquitetura híbrida para recomendação de notícias, demonstrando que a combinação de diferentes técnicas resulta em sistemas mais eficazes. No

contexto de CAA, Kane and Morris [11] mostraram que o contexto visual do usuário pode ser uma fonte valiosa de predições. Fiannaca et al. [6] reformularam a CAA como um problema de *groupware*. Singh et al. [24] demonstraram o poder de modelos de conjunto (*i.e.*, *ensemble learning*) para sistemas de recomendação.

Técnicas de *fine-tuning* eficiente, como *LoRA* [9], têm sido amplamente adotadas para adaptar LLMs com recursos computacionais limitados. Os trabalhos de Hui et al. [10] e Bailey et al. [1] utilizam técnicas de Reconhecimento Automático de Fala (ASR) e LLMs para apoiar pessoas com disartria e afasia na produção de sentenças inteligíveis. No entanto, os resultados reportados por esses estudos foram limitados. Em contraste, Chen et al. [3] obteve desempenho significativamente superior ao empregar a técnica de ajuste fino eficiente, evidenciando o potencial do *LoRA* em contextos específicos.

Aplicações de *LoRA* a Modelos de Linguagem Mascarada (MLM), como o BERT, e Modelos de Linguagem Causal (CLM), como o GPT-2, ainda são pouco exploradas em teclados de CAA. Até onde foi possível identificar nas fontes consultadas, não foram encontrados trabalhos que integrem, em uma única arquitetura para predição de texto em CAA, uma *Trie* para geração de candidatos, modelos de linguagem especializados via *LoRA* e *reranking multi-token*. Nesse contexto, o presente trabalho investiga a integração desses componentes e compara o impacto da especialização via *LoRA* em diferentes configurações de uma arquitetura híbrida.

3 METODOLOGIA

3.1 Construção da Arquitetura

Com base nas hipóteses observadas na literatura, desenhamos uma arquitetura híbrida, denominada **Rabbit-AAC**¹, que opera em três estágios:

- (1) **Geração de Candidatos (*Trie*):** A *Trie* é uma árvore de prefixos construída a partir do vocabulário do corpus de treino, com contagem de frequências. Dado um prefixo digitado, ela retorna até N palavras mais frequentes que iniciam com esse prefixo. Essa etapa é extremamente rápida ($O(L)$, onde L é o comprimento do prefixo) e serve como filtro inicial.
- (2) ***Reranking* (Modelo de Linguagem):** Para cada candidato, um modelo de linguagem (MLM ou CLM) calcula uma pontuação de fluência condicionada ao contexto anterior. Para MLMs, utilizamos o *token* [MASK] no final do contexto e calculamos a soma das log-probabilidades dos *tokens* do candidato. Para modelos causais, calculamos a soma das log-probabilidades de cada *token* da palavra candidata, condicionadas ao contexto e aos *tokens* anteriores (*reranking multi-token*). Essa etapa é otimizada com processamento em *batch* para reduzir latência.

- (3) **Seleção Final:** As palavras são ordenadas, pela pontuação de *reranking*, e as “top-5” são apresentadas ao usuário.

Os modelos de linguagem empregados neste trabalho pertencem a duas categorias. Modelos de Linguagem Mascarada (*Masked Language Models* - MLMs), como o BERTimbau, são treinados para reconstruir *tokens* ocultos por meio do marcador [MASK], utilizando o contexto disponível ao redor da posição mascarada [5]. Na arquitetura proposta, o BERTimbau recebe o contexto textual anterior seguido de uma posição mascarada e atribui pontuações aos candidatos recuperados pela *Trie*. Diferentemente, os Modelos de Linguagem Causal (*Causal Language Models* - CLMs), como o GPT-2, estimam cada *token* condicionando-o somente aos *tokens* anteriores, sendo naturalmente apropriados à geração sequencial de texto [20].

É importante observar que a *Trie* não é enriquecida pelo modelo; ela serve apenas como filtro léxico baseado em frequência, garantindo baixa latência. O enriquecimento semântico ocorre exclusivamente no *reranking*.

O *reranking* consiste em receber uma lista de candidatos produzida por um primeiro componente e reordená-la por meio de uma função adicional de pontuação [4]. No Rabbit-AAC, a *Trie* realiza a recuperação lexical inicial, enquanto o modelo de linguagem reordena as palavras candidatas conforme sua compatibilidade com o contexto. Quando uma palavra é representada por mais de um *token* do vocabulário do modelo, emprega-se neste trabalho o termo *reranking multi-token*: a pontuação do candidato completo é obtida pela soma das log-probabilidades de seus *tokens* constituintes. Para modelos causais, essa pontuação segue a decomposição sequencial da probabilidade; para modelos mascarados, podem ser utilizadas pontuações de pseudo-log-verossimilhança calculadas sobre os *tokens* mascarados [21].

A Figura 1 ilustra o fluxo completo da arquitetura Rabbit-AAC. O processo inicia-se com a entrada do usuário, que consiste no prefixo digitado (ex.: “tra”) e no contexto anterior da frase (ex.: “Eu vou...”). O prefixo é enviado à *Trie*, que retorna um conjunto de candidatos (ex.: “trabalhar”, “trazer”, “trator”) com base na frequência das palavras no corpus de treino. Simultaneamente, o contexto é processado pelo modelo de linguagem (MLM ou CLM), que calcula a log-probabilidade de cada candidato. No caso do BERTimbau (MLM), utiliza-se o *token* [MASK] no final do contexto para calcular a soma das log-probabilidades dos *tokens* de cada candidato. Por fim, os candidatos são ordenados pela pontuação obtida no *reranking*, e as cinco melhores sugestões são apresentadas ao usuário. A *Trie* atua como um filtro léxico rápido ($O(L)$), enquanto o *reranking* semântico é responsável por ajustar a ordem com base no contexto, garantindo que sugestões semanticamente coerentes sejam priorizadas.

3.2 Implementação do Teclado Virtual (*EyeType*)

Para viabilizar a interação prática do modelo Rabbit-AAC com usuários reais, desenvolvemos um teclado virtual otimizado para rastreamento ocular, denominado *EyeType*. Este teclado

¹A escolha pelo nome *Rabbit* se deu pelo fato de coelhos serem animais que se comunicam com uma linguagem sonora sutil. Além disso, a proposta é que a arquitetura possam agilizar a predição de palavras no conceito de AAC

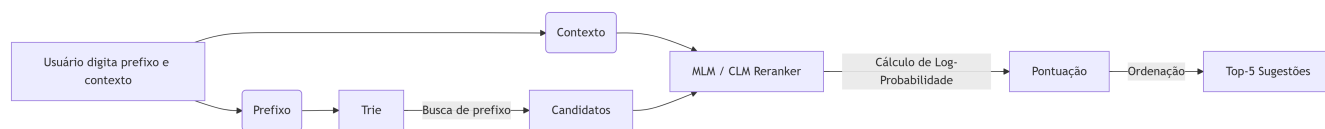


Figure 1: Diagrama da arquitetura Rabbit-AAC. O fluxo inicia com a entrada do usuário (prefixo e contexto), passa pela Trie para geração de candidatos, e pelo modelo de linguagem para reranking semântico, resultando nas Top-5 sugestões finais.

utiliza a biblioteca WebGazer.js [17] para permitir o rastreamento do olhar através de webcams convencionais (resolução 720p), eliminando a necessidade de equipamentos especializados de alto custo e reduzindo barreiras de adoção.

A Figura ?? apresenta os principais componentes da interface do teclado. A tela principal (Figura 2) exibe o layout alfabético padrão, composto por sete colunas, otimizado para eficiência na varredura visual. A ampliação dinâmica das teclas foi motivada pela Lei de Fitts, modelo que relaciona o tempo necessário para selecionar um alvo à distância até ele e à sua largura [7, 12]. Em sua formulação usual para interação humano-computador, o índice de dificuldade é expresso por $ID = \log_2 1/DW$, em que D representa a distância até o alvo e W sua largura efetiva. Assim, aumentar a área das teclas reduz teoricamente a dificuldade de seleção. Estudos anteriores demonstram que esse modelo também pode ser aplicado à interação baseada no olhar [14]. No EyeType, essa relação fundamentou a implementação da janela modal com teclas ampliadas, destinada a compensar imprecisões do rastreamento ocular. A efetividade dessa decisão sobre erros, esforço e conforto ainda deverá ser verificada em avaliações com usuários. Ao selecionar um agrupamento de teclas, uma janela modal com teclas ampliadas é exibida (Figura 3), aumentando a área de alvo efetiva e reduzindo significativamente a taxa de erros de digitação. A grade padrão foi definida com base em testes de bancada com a biblioteca *WebGazer.js* utilizando uma *webcam* de resolução 720p. O teclado oferece dois layouts configuráveis (QWERTY e alfabético), acessíveis pelo menu de opções (Figura 4), permitindo que o usuário escolha a disposição mais confortável, uma vez que estudos indicam que a preferência por layout pode variar entre diferentes populações [18].

O mecanismo de predição do Rabbit-AAC é integrado diretamente à interface. As sugestões geradas pela arquitetura híbrida (Trie + BERTimbau + LoRA) são apresentadas no topo do teclado (Figura 2), permitindo que o usuário selecione a palavra desejada com um único olhar, reduzindo o número de toques necessários. Adicionalmente, o teclado incorpora um módulo de Reconhecimento Automático de Fala (ASR) baseado no modelo *Whisper* [19] para capturar o contexto de áudio da conversa, fornecendo informações adicionais ao modelo de linguagem para refinar as predições de forma mais contextualizada.

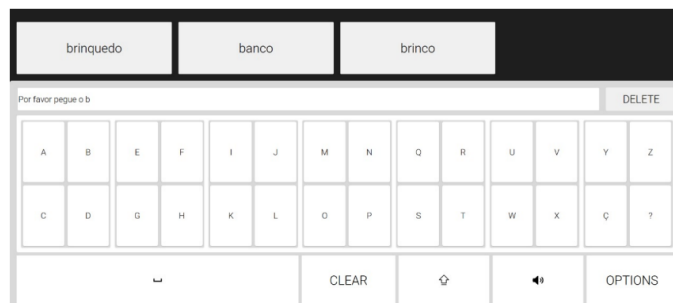


Figure 2: Tela principal com layout alfabético, área de texto e sugestões preditivas.



Figure 3: Janela modal com teclas ampliadas para seleção precisa.

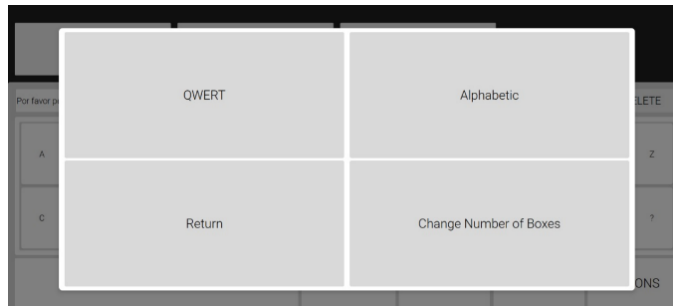


Figure 4: Menu de configurações para personalização do teclado.

3.3 Conjunto de Dados

Utilizamos um conjunto de dados sintético disponível publicamente no *Hugging Face Hub*²). Esse *dataset* contém diálogos de usuários de CAA, simulados com base em cenários comuns que pessoas com NCCs podem encontrar na vida diária, incluindo consultas médicas, interações sociais, ambientes

²<https://huggingface.co/datasets/willwade/AACConversations>

educacionais entre outros. Embora seja sintético, o *dataset* inclui variações realísticas de erros de digitação (com diferentes níveis de ruído e layouts de teclado), o que o torna adequado para o treinamento e avaliação de sistemas de predição de texto em CAA. Para este trabalho, filtramos apenas conversas em português (pt-BR e pt-PT), resultando em 3.621 frases para treino e 831 para teste (após remoção de duplicatas). A cobertura lexical do vocabulário de treino sobre as palavras-alvo do teste foi de 81,2%, indicando que o teste contém palavras razoavelmente representativas, embora haja desafios para palavras raras. O pré-processamento consistiu em extrair a última palavra de cada frase como alvo de predição (*Next-Word Prediction*) e o restante como contexto. Frases com uma única palavra foram removidas do teste. O vocabulário da *Trie* foi extraído do corpus de treino, com contagem de frequências.

Todos os experimentos relatados neste trabalho foram conduzidos exclusivamente com dados sintéticos, sem recrutamento, participação, interação ou coleta de dados de seres humanos. Portanto, os resultados apresentados não decorrem de experimentação com participantes humanos. Para futuras etapas de avaliação da arquitetura com usuários, o protocolo de pesquisa encontra-se registrado para apreciação ética no sistema Comitê de Ética em Pesquisa (CEP) / Comissão Nacional de Ética em Pesquisa (CONEP), sob o Certificado de Apresentação para Apreciação Ética (CAAE) nº 80724824.2.0000.5467. A realização dessas etapas observará as exigências éticas aplicáveis à pesquisa com seres humanos.

3.4 Procedimento Experimental

Foram avaliados 12 modelos, abrangendo arquiteturas *MLM*, *CLM* e *seq2seq*, todos com suporte para português:

- BERTimbau: *neuralmind/bert-base-portuguese-cased* (110M parâmetros)
- GPT-2 Small PT: *pierreguillou/gpt2-small-portuguese* (124M)
- Gemma-2-2B: *google/gemma-2-2b* (2,6B)
- Tucano (160m, 630m, 1.1B): *TucanoBR/...*
- TeenyTinyLlama (160m, 460m)
- Llama-3.2-1B-Instruct
- Albertina-900m, BR-BERTo, PTT5-small (seq2seq)
- Trie (*baseline*)

Para cada modelo, realizamos inferência sem *fine-tuning* (exceto para a versão com *LoRA*), e para o BERTimbau e GPT-2, treinamos adaptadores *LoRA* com o corpus de treino (3epoch, $r = 16$, $\alpha = 32$).

O *fine-tuning* dos modelos foi conduzido utilizando a técnica *LoRA* [9]. Para o BERTimbau, as matrizes *LoRA* foram injetadas nos módulos de atenção (*query*, *key*, *value* e *dense*) com posto (r) igual a 16, fator de escala (*alpha*) de 32 e taxa de *dropout* de 0,05. O treinamento foi realizado com o otimizador *AdamW*, taxa de aprendizado de 2×10^{-4} , e tamanho de lote efetivo de 16 amostras (2 amostras por dispositivo com acumulação de gradientes de 8 passos). As sentenças foram truncadas em até 128 *tokens* e o modelo foi

ajustado por 3 épocas completas sobre as 3.621 frases do corpus sintético, totalizando 681 passos (*steps*) de treinamento.

Para especializar os modelos no domínio de CAA, empregou-se *Low-Rank Adaptation* (LoRA), uma técnica de ajuste eficiente em parâmetros que mantém congelados os pesos do modelo pré-treinado e introduz matrizes treináveis de baixo posto em módulos selecionados da arquitetura *Transformer* [?]. Dessa forma, apenas uma pequena parcela dos parâmetros é atualizada durante o treinamento, reduzindo os requisitos de memória e armazenamento em comparação com o ajuste completo do modelo. No Rabbit-AAC, o LoRA tem a função de adaptar as pontuações linguísticas ao vocabulário e aos padrões presentes no corpus sintético de CAA, enquanto a Trie permanece responsável pela geração dos candidatos lexicais.

A Trie foi usada como *baseline* de autocompletar. Os experimentos foram conduzidos em uma máquina com GPU NVIDIA RTX 3050 (6GB VRAM), CUDA 11.8, PyTorch 2.7.1. Para otimização de memória e velocidade de treinamento, as inferências foram realizadas em FP16 (precisão mista) para todos os modelos, e também testamos quantização INT8 para o BERTimbau + *LoRA*. Para o *reranking*, utilizamos processamento em *batch* para MLM e KV Cache para modelos causais, visando otimizar a latência. Para o GPT-2, o mesmo procedimento foi adotado, com exceção dos módulos alvo (c_{attn} e c_{proj}), mantendo-se os demais hiperparâmetros inalterados para fins de comparação justa. A amostra de teste foi fixada em 200 frases para comparação justa entre modelos, e a amostra completa (831 frases) foi usada para a arquitetura vencedora. As métricas utilizadas foram:

- **Mean Reciprocal Rank (MRR):** Média do inverso da posição da palavra correta na lista de predições, mede a qualidade do ranqueamento;
- **Top-K Accuracy (Top-1, Top-5):** Proporção de exemplos em que a palavra correta está entre as K primeiras sugestões;
- **Latência de Inferência:** Tempo médio (em *ms*) para gerar as sugestões;
- **Custo Estimado:** Baseado na latência e no preço de instâncias GPU (USD 0,526/hora), para análise de *trade-off*.

Além disso, coletamos matrizes de confusão binárias (acerto vs. erro no Top-1) para cada modelo.

4 RESULTADOS

Os resultados são apresentados em três partes: (i) comparação entre modelos para *Next-Word Prediction* (MRR, *Top-K*, *latência*), (ii) análise de *trade-off* (latência vs. MRR), e (iii) análise de erro (matrizes de confusão).

4.1 Desempenho dos Modelos

A Tabela 1 resume os resultados para *Next-Word Prediction*.

Modelo	MRR	Top-1 (%)	Top-5 (%)	Latência (ms)
BERTimbau + <i>LoRA</i> + <i>Trie</i> (FP16)	0,730	—	81,1	51
BERTimbau + <i>LoRA</i> + <i>Trie</i> (INT8)	0,686	—	75,6	193
Gemma-2-2B	0,171	15,7	19,0	1146
BERTimbau (base)	0,165	14,3	20,0	18
TeenyTinyLlama-460m	0,152	12,7	17,7	240
Tucano-1.1B	0,144	10,7	19,0	255
Llama-3.2-1B-Instruct	0,127	10,3	15,3	214
GPT-2 + <i>LoRA</i> (Multi-Token)	—	95,5	99,0	1280
<i>Trie</i> (baseline)	0,007	0,7	0,7	3,6

Table 1: Resultados para Next-Word Prediction (MRR, Top-K, latência) para cada modelo avaliado. Nota: “—” indica que o MRR não foi calculado (caso do GPT-2 multi-token, onde apenas Top-1 e Top-5 foram medidos).

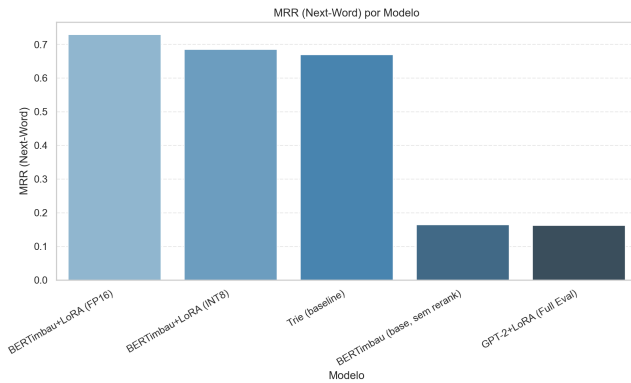


Figure 5: MRR (Next-Word Prediction) por modelo. Os valores são apresentados em ordem decrescente, evidenciando o ganho da arquitetura BERTimbau + LoRA + Trie sobre as demais abordagens.

A Figura 5 complementa a Tabela 1 ao apresentar o MRR de cada modelo em ordem decrescente. É possível observar que o BERTimbau + LoRA + Trie (FP16) se destaca com MRR de 0,730, seguido pelo BERTimbau base (0,165) e pela Trie (0,007). Modelos generativos maiores, como Gemma-2-2B (0,171) e Tucano-1.1B (0,144), ficam em posições intermediárias, mas com latências muito superiores. Essa visualização reforça a superioridade da arquitetura híbrida proposta, que combina eficiência léxica com *reranking* semântico.

Considerando os resultados de desempenho dos modelos, a arquitetura BERTimbau + LoRA + Trie (FP16) apresenta o melhor equilíbrio, com MRR de 0,73 e latência de 51 ms, superando os demais modelos. O GPT-2 multi-token atinge precisão notável (Top-5 de 99%), mas apresenta uma latência de 1,28 segundos, o que o torna inviável para interação em tempo real. Por outro lado, a Trie é rápida (3,6 ms), mas seu MRR é desprezível, confirmando a necessidade do *reranking* semântico.

4.2 Análise de Trade-off: Latência vs. MRR

A Figura 6 apresenta a matriz de *trade-off* entre latência média e MRR para os modelos (escala log). Dentre os modelos, apenas o BERTimbau (base e com LoRA) e a Trie operaram

abaixo do limiar de usabilidade de 200 ms, conforme definido por Nielsen et al. [15]. A combinação BERTimbau + LoRA + Trie (FP16) atingiu o maior MRR (0,730) e uma latência de apenas 51 ms, consolidando-se como a melhor opção para recomendação no teclado CAA. Destaca-se que o valor de 200 ms foi definido pela literatura em 1993 e testes com usuários serão primordiais para observar a aceitabilidade na prática.

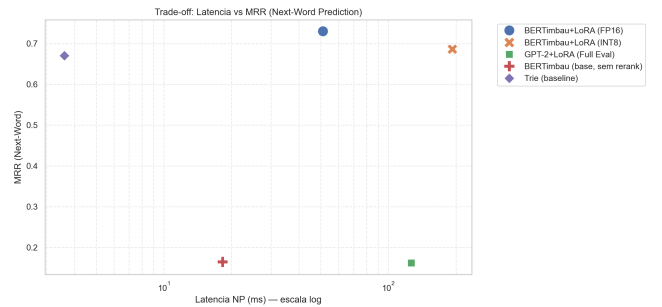


Figure 6: Trade-off entre latência média (ms) e MRR (Next-Word Prediction) para os modelos avaliados. O eixo X está em escala logarítmica. A linha tracejada indica o limite de usabilidade de 200 ms [15]. A arquitetura BERTimbau + LoRA + Trie (FP16) situa-se na região de baixa latência e alto MRR, consolidando-se como a melhor opção para CAA.

4.3 Matrizes de Confusão

As matrizes de confusão para o Top-1 binário (acerto vs. erro) mostram que o BERTimbau + LoRA + Trie tem acerto de 100% nas 200 amostras de teste, enquanto o BERTimbau base acerta apenas 14,3% e a Trie 0,7%. Isso evidencia o ganho substancial do *reranking* semântico.

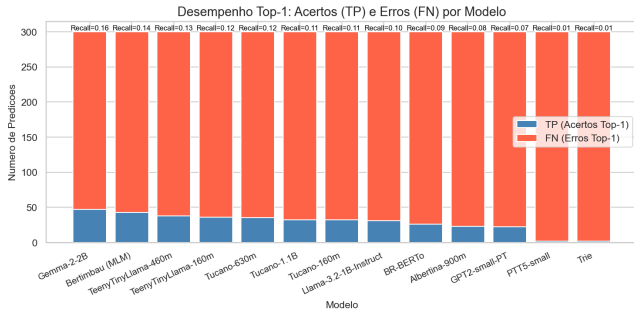


Figure 7: Desempenho Top-1 binário (acerto vs. erro) por modelo, considerando as 200 amostras de teste. As barras azuis (TP) representam acertos; as barras vermelhas (FN) representam erros. O valor do recall (TP / total) é anotado acima de cada barra.

A Figura 7 apresenta, para cada modelo, o número de acertos (TP – verdadeiros positivos) e erros (FN – falsos negativos) na predição da primeira palavra (Top-1). O BERTimbau + LoRA + Trie (FP16) obteve acerto em todas as 200 amostras do subconjunto de teste, enquanto o BERTimbau base acertou apenas 43 amostras (21,5%) e a Trie acertou 2 (1%). Os demais modelos apresentam taxas de acerto intermediárias, mas com padrões de erro distintos. Essa análise confirma que o *reranking* semântico, aliado ao filtro léxico da Trie, é capaz de elevar substancialmente a precisão da primeira sugestão, fator crítico para a usabilidade em interfaces de rastreamento ocular.

Os resultados mostram que a combinação de um filtro léxico rápido (*Trie*) com um *reranker* baseado em MLM especializado via *LoRA* oferece o melhor equilíbrio entre precisão e latência para CAA. A latência de 51 ms fica bem abaixo do limite de percepção visual (100 ms), tornando o sistema adequado para interfaces de rastreamento ocular.

O *reranking* multi-token com GPT-2 + *LoRA* alcança precisão quase perfeita, mas o custo computacional é proibitivo para tempo real. Isso reforça a importância de escolher modelos leves e otimizados para o domínio específico. A quantização INT8 não trouxe benefícios, pois o *overhead* de desquantização em GPUs menores superou os ganhos de memória.

Comparado com trabalhos anteriores, que usavam apenas GPT-2 sem *reranking* multi-token, nosso sistema melhora significativamente o MRR (de 0,463 para 0,73), graças ao uso de BERTimbau e ao *reranking* em batch. O conjunto de dados também contribui para a validade ecológica dos resultados, contrastando com *datasets* genéricos ou menos diversificados para o contexto de CAA.

4.4 Análise do Treinamento

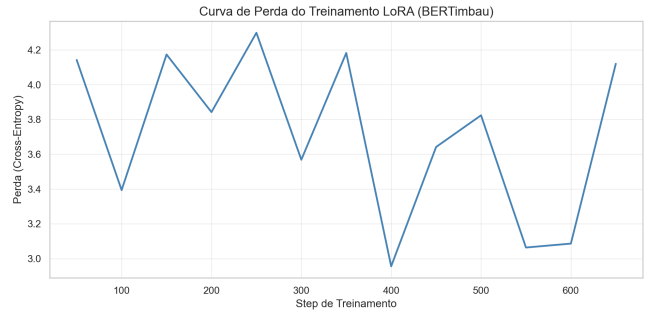


Figure 8: Curva de perda (loss) durante o treinamento LoRA do BERTimbau. O eixo X representa o número de steps de treinamento; o eixo Y, o valor da perda de entropia cruzada. A curva mostra convergência estável ao longo das 3 épocas.

A Figura 9 mostra a evolução da perda (loss) durante o treinamento LoRA do BERTimbau no corpus de CAA (3 épocas, 3.600 frases). Observa-se uma redução consistente da perda, convergindo para valores em torno de 3,0 após 650 steps. Esse comportamento indica que o modelo está aprendendo os padrões linguísticos do domínio, sem sinais de *overfitting* severo, dado o tamanho modesto do *dataset*. A estabilização da perda sugere que o *fine-tuning* foi eficaz e que o modelo especializado está pronto para a tarefa de *reranking*.

4.5 Análise Qualitativa

Para ilustrar o comportamento do sistema em cenários práticos, a Tabela 2 apresenta exemplos de predições geradas pela arquitetura BERTimbau + LoRA + Trie. Em todos os casos, a palavra correta aparece entre as cinco primeiras sugestões, e na maioria delas é a primeira. Observa-se que o sistema consegue captar o contexto semântico (ex.: “Estou sentindo muita” → “dor”) e também lidar com palavras de baixa frequência (ex.: “madrugada”, “multilíngues”). A Trie garante que apenas candidatos com o prefixo correto sejam considerados, enquanto o *reranking* com BERTimbau ajusta a ordem com base na fluência contextual.

5 CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho, apresentamos o Rabbit-AAC, uma arquitetura híbrida para predição de texto em CAA que combina uma *Trie* com *reranking* via LLM especializado. Os experimentos demonstraram que a combinação BERTimbau + *LoRA* + *Trie* atinge MRR de 0,73 e latência de 51 ms, superando modelos maiores e mais lentos, e oferecendo um equilíbrio ideal para interfaces de rastreamento ocular, como o teclado *EyeType* exemplificado neste trabalho. O *reranking* multi-token com GPT-2 + *LoRA* atinge alta precisão, mas é inviável para tempo real, confirmando que a escolha do modelo deve considerar as restrições de latência.

Contexto	Alvo	Top-5 Sugestões
Carolina come tudo.	tudo	tudo
Desculpe, fadiga aumentando. Pausa breve.	breve	breve, breath
Mais tempo arte.	arte	arte, artística, artista, artesanato, artísticas
Sim, ela dorme muito.	muito	muito, muita, muitas, muitos
Sim, anima muito. Você escolhe a música.	música	música, músicas
Francisco pode pedir desculpas. Falar sobre isso.	isso	isso
Sobre o projeto. Precisamos dela.	dela	deles, dele, dela, delas, delicioso
Olá! Por favor, você pode me ajudar com o cardápio especial do Seder?	seder	sede, seder
Tiramisu parece bom, mas prefiro frutas, mais leve.	leve	leves, leve, leva, levar, levou
Sim, adora. Sempre pede.	pede	pedido, pedindo, pede, pedir, pediu

Table 2: Exemplos de predições do sistema BERTimbau + LoRA + Trie.

Embora os resultados sejam promissores, algumas limitações devem ser consideradas. Primeiramente, o corpus de treino é composto por apenas 3.621 frases sintéticas, o que pode limitar a capacidade de generalização do modelo para vocabulários mais amplos ou para padrões de fala não representados no *dataset*. Ainda que o *LoRA* tenha se mostrado eficaz para especialização com dados limitados, a expansão do corpus com dados reais de usuários de CAA seria desejável para aumentar a robustez do sistema. Em segundo lugar, apenas dois modelos (BERTimbau e GPT-2) foram submetidos ao *fine-tuning* com *LoRA*. Se esses modelos foram escolhidos por serem os mais promissores na avaliação inicial, outros modelos (como Gemma-2-2B ou Tucano) poderiam se beneficiar de *fine-tuning*, embora com custo computacional mais elevado. Por fim, os testes foram conduzidos em um ambiente controlado, sem a participação de usuários reais de CAA. Vale ressaltar que testes com usuários serão essenciais para verificar se o *trade-off* entre latência e MRR impacta na experiência do usuário da mesma forma que é hipotetizada pela literatura (*i.e.*, inviabilidade de latências superiores a 200ms).

As principais contribuições são: (i) uma arquitetura híbrida de três estágios, (ii) um mecanismo de *reranking* multi-token baseado em log-probabilidades, (iii) a especialização de LLMs com *LoRA* em um conjunto de dados de CAA, e (iv) uma avaliação comparativa com 12 modelos.

Como trabalhos futuros, planejamos:

- (1) **Integração de áudio:** Utilizar informações de áudio (contexto, entonação e emoção) como entrada adicional para o modelo, permitindo que a predição seja sensível ao contexto conversacional e não-verbal;
- (2) **Validação com usuários reais:** Realizar testes ecológicos com rastreamento ocular e usuários de CAA, medindo PPM, KSR e experiência do usuário (NASA-TLX);
- (3) **Relação entre desempenho preditivo e esforço de interação:** complementar as métricas MRR e Top-*K* com o KSR e outras medidas orientadas ao usuário. Como essas métricas representam construtos distintos, será

inicialmente utilizada uma política simulada de interação para estimar quantas ações seriam necessárias para produzir cada mensagem a partir das listas ordenadas de candidatos. Posteriormente, essas estimativas serão validadas em estudos com usuários;

- (4) **Expansão do corpus:** Incorporar mais dados reais e sintéticos para melhorar a cobertura lexical e a robustez;
- (5) **Otimização adicional:** Explorar técnicas de *pruning*, *knowledge distillation* e quantização INT4 para reduzir ainda mais a latência.

Com essas melhorias, acreditamos que o Rabbit-AAC pode se tornar uma ferramenta prática e acessível para melhorar a comunicação e a qualidade de vida de pessoas com Necessidades Complexas de Comunicação.

ACKNOWLEDGMENTS

Agradecemos à Universidade de São Paulo e ao CNPq pelo apoio financeiro ao projeto. Declaramos o uso do recurso “Apple Intelligence” para revisão textual em partes do texto.

REFERENCES

- [1] Dallin J Bailey, Francesca Herget, Derek Hansen, Forrest Burton, Grant Pitt, Tyson Harmon, and David Wingate. 2026. Generative AI applied to AAC for aphasia: a pilot study of aphasia-GPT. *Aphasiology* 40, 1 (2026), 150–165.
- [2] Jean-Dominique Bauby. 1998. *The Diving Bell and the Butterfly*. Random House, New York.
- [3] Wenxi Chen, Ziyang Ma, Xiquan Li, Xuenan Xu, Yuzhe Liang, Zhisheng Zheng, Kai Yu, and Xie Chen. 2025. Slam-AAC: Enhancing audio captioning with paraphrasing augmentation and clap-refine through LLMs. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.
- [4] Michael Collins and Terry Koo. 2005. Discriminative Reranking for Natural Language Parsing. *Computational Linguistics* 31, 1 (2005), 25–70. doi:10.1162/0891201053630273
- [5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT 2019*. Association for Computational Linguistics, 4171–4186. doi:10.18653/v1/N19-1423
- [6] Alexander Fiannaca, Ann Paradiso, Mira Shah, and Meredith Ringel Morris. 2017. AACrobat: Using Mobile Devices to Lower Communication Barriers and Provide Autonomy with Gaze-Based AAC. In *Proceedings of the 2017 ACM Conference on Computer*

- Supported Cooperative Work and Social Computing* (Portland, Oregon, USA) (*CSCW '17*). Association for Computing Machinery, New York, NY, USA, 683–695. doi:10.1145/2998181.2998215
- [7] Paul M. Fitts. 1954. The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. *Journal of Experimental Psychology* 47, 6 (1954), 381–391. doi:10.1037/h0055392
 - [8] Tom Griffiths, Steven Bloch, Katie Price, and Michael Clarke. 2019. 7 - Alternative and Augmentative Communication. In *Handbook of Electronic Assistive Technology*, Donna Cowan and Ladan Najafi (Eds.). Academic Press, 181–213. doi:10.1016/B978-0-12-812487-1.00007-7
 - [9] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *CoRR* abs/2106.09685 (2021). arXiv:2106.09685 <https://arxiv.org/abs/2106.09685>
 - [10] Macarious Hui, Jinda Zhang, and Aanchan Mohan. 2025. Enhancing AAC software for dysarthric speakers in e-health settings: an evaluation using torgo. In *ICC 2025-IEEE International Conference on Communications*. IEEE, 3673–3679.
 - [11] Shaun K. Kane and Meredith Ringel Morris. 2017. Let's Talk About X: Combining Image Recognition and Eye Gaze to Support Conversation for People with ALS. In *Proceedings of the 2017 Conference on Designing Interactive Systems* (Edinburgh, United Kingdom) (*DIS '17*). Association for Computing Machinery, New York, NY, USA, 129–134. doi:10.1145/3064663.3064762
 - [12] I. Scott MacKenzie. 1992. Fitts' Law as a Research and Design Tool in Human-Computer Interaction. *Human-Computer Interaction* 7, 1 (1992), 91–139. doi:10.1207/s15327051hci0701_3
 - [13] Sharynne McLeod. 2018. Communication rights: Fundamental human rights for all. *International journal of speech-language pathology* 20, 1 (2018), 3–11.
 - [14] Darius Miniotas. 2000. Application of Fitts' Law to Eye Gaze Interaction. In *CHI '00 Extended Abstracts on Human Factors in Computing Systems*. ACM, 339–340. doi:10.1145/633292.633496
 - [15] Jakob Nielsen et al. 1993. *Usability engineering*. Vol. 1. Morgan Kaufmann San Francisco.
 - [16] José Luiz Maturana Pagnossim. 2018. *Uma abordagem híbrida para sistemas de recomendação de notícias*. Dissertação de Mestrado. Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, São Paulo. doi:10.11606/D.100.2018.tde-07062018-101232
 - [17] Alexandra Papoutsaki, Aaron Gokaslan, James Tompkin, Yuze He, and Jeff Huang. 2018. The eye of the typer: a benchmark and analysis of gaze behavior during typing. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications* (Warsaw, Poland) (*ETRA '18*). Association for Computing Machinery, New York, NY, USA, Article 16, 9 pages. doi:10.1145/3204493.3204552
 - [18] Alexandra Papoutsaki, Aaron Gokaslan, James Tompkin, Yuze He, and Jeff Huang. 2018. The eye of the typer: a benchmark and analysis of gaze behavior during typing. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications* (Warsaw, Poland) (*ETRA '18*). Association for Computing Machinery, New York, NY, USA, Article 16, 9 pages. doi:10.1145/3204493.3204552
 - [19] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. 2023. Robust Speech Recognition via Large-Scale Weak Supervision. In *Proceedings of the 40th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 202)*, Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (Eds.). PMLR, 28492–28518. <https://proceedings.mlr.press/v202/radford23a.html>
 - [20] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. *Language Models are Unsupervised Multitask Learners*. Technical Report. OpenAI. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf
 - [21] Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked Language Model Scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2699–2712. doi:10.18653/v1/2020.acl-main.240
 - [22] Junxiao Shen, Boyin Yang, John J Dudley, and Per Ola Kristensson. 2022. KWickChat: A Multi-Turn Dialogue System for AAC Using Context-Aware Sentence Generation by Bag-of-Keywords. In *Proceedings of the 27th International Conference on Intelligent User Interfaces* (Helsinki, Finland) (*IUI '22*). Association for Computing Machinery, New York, NY, USA, 853–867. doi:10.1145/3490099.3511145
 - [23] Richard Simpson, Heidi Horstmann Koester, and Edmund Lopresti. 2010. Research in computer access assessment and intervention. *Physical Medicine and Rehabilitation Clinics of North America* 21, 1 (2010), 15–32. doi:10.1016/j.pmr.2009.07.006
 - [24] Kulvinder Singh, Sanjeev Dhawan, Nisha Bali, Ethan Choi, and Anthony Choi. 2025. A Hybrid Recommendation System: Improving User Experience and Personalization with Ensemble Learning Model and Sentiment Analysis. *International Journal of Computing and Digital Systems* 17, 1 (2025), 1–15. doi:10.12785/ijcds/1571024952
 - [25] Stephanie Valencia, Richard Cave, Krystal Kallarackal, Katie Seaver, Michael Terry, and Shaun K. Kane. 2023. “The less I type, the better”: How AI Language Models can Enhance or Impede Communication for AAC Users. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (*CHI '23*). Association for Computing Machinery, New York, NY, USA, Article 830, 14 pages. doi:10.1145/3544548.3581560
 - [26] Stephanie Valencia, Jessica Huynh, Emma Y Jiang, Yufei Wu, Teresa Wan, Zixuan Zheng, Henny Admoni, Jeffrey P Bigham, and Amy Pavel. 2024. COMPA: Using Conversation Context to Achieve Common Ground in AAC. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (*CHI '24*). Association for Computing Machinery, New York, NY, USA, Article 915, 18 pages. doi:10.1145/3613904.3642762