Uma abordagem exploratória para o mapeamento de distribuições de probabilidade de modelos de linguagem

Felipe Zan Coelho
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Email: felipezan@ufmg.br

Orientador: Prof. Adriano Alonso Veloso
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Email: adrianov@dcc.ufmg.br

***Abstract*—Este trabalho apresenta uma investigação sobre a viabilidade de aproximar a distribuição de saída de um modelo de linguagem sofisticado utilizando um modelo menor e técnicas correlatas de mapeamento estatístico. Baseando-nos, principalmente, em arquiteturas do tipo transformer, n-gramas e métodos inspirados no problema de transporte ótimo, nosso objetivo foi explorar a transição de níveis de complexidade na predição de tokens, bem como discutir a eficiência de modelos reduzidos no cenário de inferência. Resultados envolvendo perplexidade, BLEU e ROUGE mostram o impacto de nossa abordagem, a qual recebe a alcunha de T_BRIDGE, em comparação a um modelo baseline estilo GPT2 e a um modelo n-grama de ordem 4. Por fim, discutimos os desafios e perspectivas futuras na área, ressaltando sua relevância no ecossistema de Processamento de Linguagem Natural (PLN).**

***Index Terms*—Modelos de linguagem, modelos n-grama, transformadores, transporte ótimo, aproximação de distribuições, eficiência em inferência.**

I. Introdução

A habilidade de atribuir probabilidades a sequências de tokens, usualmente expressa pelo chamado *modelo de linguagem*, constitui pilar básico em inúmeros aplicativos de processamento de linguagem natural. Nos últimos anos, arquiteturas de redes neurais, como *Transformers* ou grandes modelos de linguagem (LLMs), expandiram significativamente a fronteira do que consideramos possível em termos de *compreensão semântica* e *geração de texto*.

Por outro lado, embora tais modelos avancem o estado da arte, ainda enfrentamos os gargalos de custo computacional (particularmente durante a fase de inferência) e a dificuldade de interpretabilidade. Isso sugere que, em larga escala, parte das informações previsíveis do texto (por exemplo, padrões puramente estatísticos de baixa ordem) poderia ser isolada e confiada a arquiteturas de menor complexidade. Nesse contexto, este trabalho propõe estudar um arcabouço em que um modelo de linguagem mais simples (especificamente, um modelo n-grama) auxilia na construção de uma distribuição de probabilidade preliminar, sendo depois refinada por uma rede de menor porte em direção à distribuição de um modelo mais robusto, estilo GPT2.

Nossa hipótese-chave é a possibilidade de uma *aproximação* efetiva que possibilite reduzir custos de inferência, ao mesmo tempo preservando qualidades de modelagem associadas a arquiteturas maiores. Seguindo recomendações do âmbito de trabalhos científicos [1], [2], estruturamos este texto em cinco capítulos:

- **Capítulo Introdutório:** motivação, objetivo e escopo;
- **Capítulo Referencial:** principais referências teóricas e correlatos;
- **Capítulo de Contribuição:** detalhes do procedimento metodológico e resultados experimentais;
- **Capítulo de Fechamento:** conclusões e possíveis extensões;
- **Referências Bibliográficas:** listagem das fontes principais e correlatas.

II. Referencial Teórico

A evolução de modelos de linguagem até arquiteturas transformacionais e a exploração de técnicas de distilação e transporte ótimo ofereceu a base para este projeto. A seguir, revisamos as noções principais e trabalhos correlatos.

A. Modelos de Linguagem: de n-gramas a Transformers

Modelos de linguagem iniciaram-se de modo simples, baseados em *n-gramas*, nos quais a probabilidade de uma palavra dependeria de suas $n-1$ anteriores [3]. No entanto, a escalabilidade desses modelos pode se tornar inviável para n altos ou vocabulários extensos, além de falhar em capturar longas dependências. Com o advento das redes neurais recorrentes e a arquitetura Transformer [4], surgiram os grandes modelos de linguagem (LLMs), capazes de incorporar conhecimentos de forma mais rica e contextual.

B. Distilação de Conhecimento

Os trabalhos de Hinton *et al.* sobre *Distilling the Knowledge in a Neural Network* [1] inspiraram inúmeras abordagens para treinar modelos menores que reproduzem, em alguma medida, o poder preditivo de modelos maiores. A motivação é clara: reduzir custos de inferência ao mesmo tempo mantendo, tanto quanto possível, a capacidade de predição de uma rede massiva.

C. Transporte Ótimo

O conceito de transporte ótimo, na formulação de Kantorovich, estuda maneiras de transportar *massa* entre distribuições de probabilidade, minimizando certo custo [5]. A regularização entrópica apresentada por Cuturi [6] viabilizou aplicações computacionalmente eficientes via *Sinkhorn-Knopp*. Em paralelo, a separação da estrutura de *dependência* das *distribuições marginais* por meio de cópulas [7] inspirou a idéia de tratar modelos de linguagem por distintos estágios, mesclando princípios de transporte de probabilidade e técnicas neurais.

III. Contribuição: Atividades Conduzidas e Resultados

Seguindo a estrutura clássica de projeto, conduzimos nossas atividades em torno de três eixos: construção dos conjuntos de dados, definição de modelos de linguagem (baseline, n-grama e T_BRIDGE) e avaliação baseada em métricas quantitativas e exemplos qualitativos. A seguir, descrevemos cada um desses pontos.

A. Construção dos Conjuntos de Dados

Para avaliar o desempenho dos diferentes modelos, selecionamos um *dataset* extenso, composto de 765.700.721 tokens espalhados em 3.866.634 histórias para treino do modelo *baseline* e do *n-grama*, além de um conjunto de teste com 58.697.662 tokens (283.513 histórias). Esse material, combinado a histórias adicionais geradas sinteticamente, permitiu cobrir uma variedade razoável de tópicos e estilos textuais.

Para treinar a arquitetura T_BRIDGE em sua tarefa de aproximação da distribuição, recorremos a um *dataset de pares* contendo 20.000 pares de treino e 2.000 pares de teste. Cada par corresponde a $(A(x), B(x))$, em que $A(x)$ advém de um modelo n-grama de ordem 4 e $B(x)$ provém de um modelo Transformer. Dessa forma, o modelo T_BRIDGE aprende a transformar a distribuição simples em algo mais sofisticado.

B. Modelos Base e Arquitetura T_BRIDGE

1) n-grama de Ordem 4: Para o modelo tradicional, treinou-se um n-grama de ordem 4, o qual resultou num arquivo final de aproximadamente 289,418 KB (armazenado como *final_model.pkl.gz*). Esse modelo realiza predições baseadas unicamente em frequências locais de tokens, ignorando dependências de maior escala.

2) Baseline do Tipo GPT2: Nosso modelo baseline segue o estilo GPT2, porém configurado para ter 2.559.488 parâmetros. Especificamente:

- *vocab_size* = 8192
- *block_size* = 512
- *n_layer* = 2
- *n_head* = 4
- *n_embd* = 128
- *dropout* = 0.1

O tamanho final do arquivo deste modelo *checkpoint* (*model_checkpoint.pt*) é de 12,060 KB.

3) Arquitetura T_BRIDGE: Nossa proposta central, T_BRIDGE, caracteriza-se por possuir dois encoders e um decoder, totalizando cerca de 130 mil parâmetros. De forma sintética, trata-se de uma arquitetura em que um *EncoderContexto* extrai informações do contexto textual, enquanto um *encoderDist* processa a distribuição oriunda do n-grama (convertida para log-distribuição). Finalmente, um decoder combina ambas as representações, gerando a distribuição ajustada final. O seguinte pseudocódigo (adaptado de nossa implementação) ilustra o fluxo:

Descrição: encoder dual + decoder
Pseudocódigo:

```
H_contexto = EncoderContexto(tokens_contexto)
log_distribuicao_1 =
Log(distribuicao_1)
sequencia_distribuicao_1 =
EmbedDistribuicao(log_distribuicao_1)
H_distribuicao_1 =
encoderDist(sequencia_distribuicao_1)
decoder_input = [<START>]
H_decoder = decoder(decoder_input,
crossKeyValor1=H_contexto,
crossKeyValor2=H_distribuicao_1)
logits_intermediarios = W_out *
H_decoder + b_out
logits_finais = log_distribuicao_1
+ logits_intermediarios
distribuicao_2 =
Softmax(logits_finais)
```

Em essência, o T_BRIDGE soma uma *correção* aos logits do n-grama, aproximando-os daqueles típicos de um modelo mais sofisticado.

C. Métricas e Avaliação Quantitativa

Avaliamo-nos em termos de *Perplexity*, *BLEU* e *ROUGE* sobre o conjunto de teste. A Tabela I apresenta os resultados do baseline estilo GPT2:

TABLE I
Métricas do Baseline no *dataset* de teste

Métrica	Valor
Perplexity	15.1434
BLEU	0.1798
ROUGE-1 F	0.5818
ROUGE-2 F	0.2113
ROUGE-L F	0.4233

Já na Tabela II, exibimos os valores registrados para o n-grama (ordem 4) e para o T_BRIDGE. Observamos que o n-grama apresenta maior perplexidade, com métricas inferiores de BLEU e ROUGE. O T_BRIDGE, por sua vez, revela-se promissor, alcançando resultados intermediários entre o n-grama e o baseline.

TABLE II
MÉTRICAS COMPARATIVAS: N-GRAMA VS T_BRIDGE

Métrica	n-grama	T_BRIDGE
Perplexity	22.51	18.75
BLEU	0.1256	0.1623
ROUGE-1 F	0.5240	0.5540
ROUGE-2 F	0.1425	0.1850
ROUGE-L F	0.3527	0.3980

D. Gráfico de Loss do Treinamento

A Figura 1 compara as curvas de treinamento do baseline e do T_BRIDGE. Nota-se que o baseline atinge menores valores de loss mais rapidamente, dada sua maior capacidade de modelagem. O T_BRIDGE demonstra convergência estável, com uma trajetória de loss que reflete seu processo de ajuste incremental sobre a distribuição do n-grama.

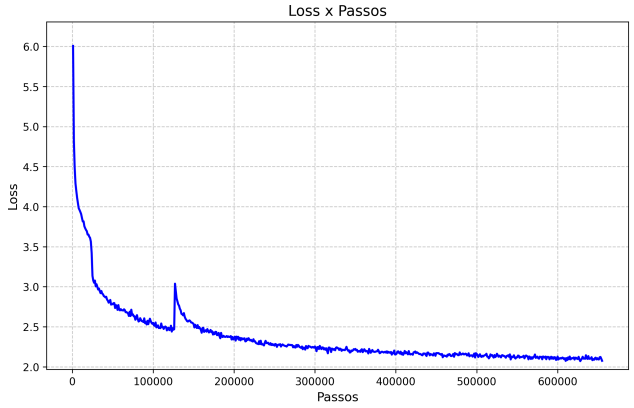

Fig. 1. Curvas de loss de treinamento: baseline (azul) vs T_BRIDGE (vermelho). O eixo x representa épocas de treino, enquanto o y mostra a loss em escala logarítmica.

E. Exemplos Qualitativos de Geração de Texto

Para ilustrar a diferença entre modelos, apresentamos alguns exemplos de geração condicionados a um breve contexto. Abaixo, listamos excertos característicos do *n-grama* e do *baseline*, notando o estilo repetitivo e, por vezes, desconexo do n-grama versus a maior coerência do baseline.

1) Saída do n-grama (ordem 4): **Contexto:** *"Once upon a time"*

Texto Gerado:

> " . . . there was a little seed . The seed started to grow bigger and taller than Anna . He ran to his friends , and the ball all day . It was home to many bugs using the special brush close , knowing it held more magic . . . "

Observa-se pouca consistência narrativa, com frases adicionadas quase aleatoriamente.

2) Saída do Baseline (GPT2-like): **Contexto:** *"Once upon a time"*

Texto Gerado:

> ". . . there was a little girl named Lucy . She was three years old and loved to explore . One day , she decided to go to the park with her mom . . . "

Aqui, a resposta mostra maior fluidez e coesão narrativa, embora ainda possamos notar características de geração simplificada, dado tratar-se de um GPT2 pequeno.

IV. CONCLUSÕES E TRABALHOS FUTUROS

Neste relatório, documentamos a proposta de aproximar distribuições de probabilidade de um modelo sofisticado usando informações de um modelo simples (n-grama) e uma rede neural ajustadora (T_BRIDGE). Os resultados nos encorajam a prosseguir na investigação, pois indicam que o T_BRIDGE é capaz de melhorar distribuições geradas apenas por estatísticas de baixa ordem.

Alguns pontos de aprimoramento emergem:

- **Completar avaliação do T_BRIDGE:** incorporar novas métricas e refinar hiperparâmetros para verificar se conseguimos reduzir perplexidade e elevar BLEU/ROUGE.
- **Escalabilidade:** avaliar comportamento em corpora ainda maiores, reforçando o ganho em cenários de inferência mais onerosos.

REFERENCES

[1] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," *arXiv preprint arXiv:1503.02531*, 2015.

[2] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," *OpenAI Research Paper*, 2018. [Online]. Available: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

[3] S. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, 1996, pp. 310–318.

[4] A. Vaswani *et al.*, "Attention Is All You Need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[5] C. Villani, *Optimal Transport: Old and New*. Springer, 2009.

[6] M. Cuturi, "Sinkhorn Distances: Lightspeed Computation of Optimal Transport," in *Advances in Neural Information Processing Systems*, 2013, pp. 2292–2300.

[7] R. B. Nelson, *An Introduction to Copulas*. Springer, 2006.