JOÃO PAULO PAIVA LIMA

UMA NOVA CODIFICAÇÃO POSICIONAL PARA RECONHECIMENTO DE TABELAS COM MODELOS TRANSFORMERS IMAGEM-PARA-SEQUÊNCIA

LAVRAS – MG

2023

JOÃO PAULO PAIVA LIMA

UMA NOVA CODIFICAÇÃO POSICIONAL PARA RECONHECIMENTO DE TABELAS COM MODELOS TRANSFORMERS IMAGEM-PARA-SEQUÊNCIA

Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras para obtenção do título de Bacharel em "Ciência da Computação"

Prof. DSc. Denilson Alves Pereira
Orientador

LAVRAS – MG

2023

JOÃO PAULO PAIVA LIMA

UMA NOVA CODIFICAÇÃO POSICIONAL PARA RECONHECIMENTO DE TABELAS COM MODELOS TRANSFORMERS IMAGEM-PARA-SEQUÊNCIA
A NEW POSITIONAL ENCODING FOR TABLE RECOGNITION WITH IMAGE-TO-SEQUENCE TRANSFORMERS MODELS

Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras para obtenção do título de Bacharel em "Ciência da Computação"

APROVADA em 13 de Dezembro de 2023.

Prof. DSc. Denilson Alves Pereira	UFLA
Profa. DSc. Marluce Rodrigues Pereira	UFLA
Profa. DSc. Paula Christina Figueira Cardoso	UFLA

gov.br
Documento assinado digitalmente
DENILSON ALVES PEREIRA
Data: 19/12/2023 21:03:55-0300
Verifique em https://validar.iti.gov.br

Prof. DSc. Denilson Alves Pereira
Orientador

LAVRAS – MG
2023

Dedico este trabalho aos meus queridos pais, familiares e amigos que me ajudaram e ajudam nessa caminhada.

AGRADECIMENTOS

Agradeço aos meus pais pelo amor, carinho e incentivo todos estes anos.
Ao meu irmão Caio e os amigos do coração Arthur, Enzo e Thiago.
Ao meu orientador, Denilson.
Aos parceiros de Laboratório.
E ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pelo apoio financeiro.
Muito obrigado!

"O que um modelo probalístico de linguagem disse para o outro? the the the the the the the the the the the the the the the the the the the the the the the the the the the..."
(autora desconhecida)

RESUMO

Neste trabalho, é apresentada a utilização de um modelo imagem-para-sequência pré-treinado para Entendimento de Documentos Visuais na tarefa de Reconhecimento de Tabela. Também, visando obter o máximo da arquitetura Transformer, foi criada uma codificação intermediária que, em conjunto com embeddings posicionais de três dimensões, reduz o tamanho de sequência, que expande a capacidade de generalização do modelo e que pode ser convertida de e para HTML sem perda de dados. Com a nova codificação, conseguiu-se com que o acerto para estrutura de tabelas complexas após uma única época de treinamento passasse de 88,9% para 91,6% TEDS (Tree Edit Distance Score).

Palavras-chave: Extração de Dados. Extração de Tabelas. Processamento de Linguagem Natural

ABSTRACT

In this work, the use of a pre-trained image-to-sequence model for Visual Document Understanding in the Table Recognition task is presented. Also, aiming to get the most out of the Transformer architecture, an intermediate encoding was created which, together with three-dimensional positional embeddings, reduces the sequence size, which expands the generalization capacity of the model and which can be converted to and from HTML without data loss. With the new coding, the accuracy for the structure of complex tables after a single training period increased from 88.9% to 91.6% TEDS (Tree Edit Distance Score).

Keywords: Data Extraction. Table Extraction. Natural Language Processing.

SUMÁRIO

PRIMEIRA PARTE - INTRODUÇÃO GERAL

1 INTRODUÇÃO

As tabelas são elementos visuais que permitem organizar e apresentar dados de forma estruturada e eficiente. No entanto, extrair e interpretar as informações contidas nas tabelas não é uma tarefa trivial, especialmente quando as tabelas possuem uma estrutura complexa, com múltiplas linhas e colunas, células mescladas, cabeçalhos aninhados, entre outros recursos. Por isso, o Reconhecimento de Tabela é uma importante subárea do Entendimento de Documentos Visuais, que visa converter tabelas em um formato estruturado e legível por máquina, como HTML ou XML.

Uma das abordagens mais promissoras para o Reconhecimento de Tabela é o uso de modelos de aprendizado profundo baseados em imagem-para-sequência, que recebem uma imagem de uma tabela como entrada e geram uma sequência de tokens que representa o formato estruturado da tabela como saída. Esses modelos podem ser pré-treinados em grandes conjuntos de dados de imagens e textos, e depois ajustados para a tarefa específica de Reconhecimento de Tabela. No entanto, esses modelos ainda enfrentam alguns desafios, como a limitação do tamanho de sequência, a complexidade da codificação da estrutura da tabela, e a perda de informações durante a conversão entre diferentes formatos.

Neste trabalho, propomos uma solução inovadora para esses desafios, utilizando um modelo imagem-para-sequência pré-treinado para Entendimento de Documentos Visuais na tarefa de Reconhecimento de Tabela. O nosso modelo utiliza a arquitetura Transformer, que é composta por blocos de atenção multi-cabeça, que permitem capturar as relações entre os pixels da imagem e os tokens da sequência. Além disso, introduzimos uma nova codificação intermediária que, em conjunto com embeddings posicionais de três dimensões, reduz o tamanho de sequência, expande a capacidade de generalização do modelo, e permite a conversão de e para HTML sem perda de dados. A nossa codificação consiste em uma representação da estrutura da tabela, que utiliza símbolos especiais para indicar o

início e o fim de cada linha, coluna, célula e cabeçalho. Com a nossa codificação, conseguimos melhorar o desempenho do modelo para estrutura de tabelas complexas, alcançando um acerto de 91,6% TEDS (Tree Edit Distance Score) após uma única época de treinamento.

SEGUNDA PARTE - ARTIGO

Uma Nova Codificação Posicional para Reconhecimento de Tabelas com Modelos Transformers Imagem-para-sequência

João Paulo P. Lima[1]

[1]Departamento de Ciência da Computação – Universidade Federal de Lavras (UFLA)
Caixa Postal 3037 – 37.203-202 – Lavras – MG – Brazil

joao.lima1@estudante.ufla.br

Abstract. *In this work, the use of a pre-trained image-to-sequence model for Visual Document Understanding in the Table Recognition task is presented. Also, aiming to get the most out of the Transformer architecture, an intermediate encoding was created which, together with three-dimensional positional embeddings, reduces the sequence size, which expands the generalization capacity of the model and which can be converted to and from HTML without data loss. With the new coding, the accuracy for the structure of complex tables after a single training period increased from 88.9% to 91.6% TEDS (Tree Edit Distance Score).*

Resumo. *Neste trabalho, é apresentada a utilização de um modelo imagem-para-sequência pré-treinado para Entendimento de Documentos Visuais na tarefa de Reconhecimento de Tabela. Também, visando obter o máximo da arquitetura Transformer, foi criada uma codificação intermediária que, em conjunto com embeddings posicionais de três dimensões, reduz o tamanho de sequência, que expande a capacidade de generalização do modelo e que pode ser convertida de e para HTML sem perda de dados. Com a nova codificação, conseguiu-se com que o acerto para estrutura de tabelas complexas após uma única época de treinamento passasse de 88,9% para 91,6% TEDS (Tree Edit Distance Score).*

1. Introdução

Tabelas são recursos visuais que permitem apresentar uma grande quantidade de informação de maneira organizada e densa, facilitando a comparação e a análise dos dados. Por essa razão, elas são frequentemente utilizadas na elaboração de documentos técnicos, financeiros e científicos, nos quais se busca transmitir informações de maneira interconectada, precisa e objetiva.

Entretanto, um desafio que se impõe na utilização computacional das tabelas em documentos visuais é que elas são, em sua maioria, projetadas estritamente para a leitura humana e exigem do leitor certa capacidade interpretativa, envolvendo o reconhecimento de símbolos, convenções e relações entre os dados. Essa tarefa não é trivial para sistemas computadorizados tradicionais, que apresentam dificuldades para compreender os elementos tabulares devido à grande variedade de estilos e topologias possíveis, o que afeta a sua padronização e dificulta a extração de informação. Esse desafio se mostra especialmente importante em casos nos quais a interpretação automatizada dos dados é

Instância	S							
	Com busca local				Sem busca local			
	A	R	B	K	A	R	B	K
1	-5,25	3	-2,25	3	-2,25	-2,25	3	3
2	-35	25	5	-15	-25	15	35	-5
3	-66,5	57	-38	19	-9,5	-38	57	19

Figura 1. Parte de uma tabela de exemplo. Produção original.

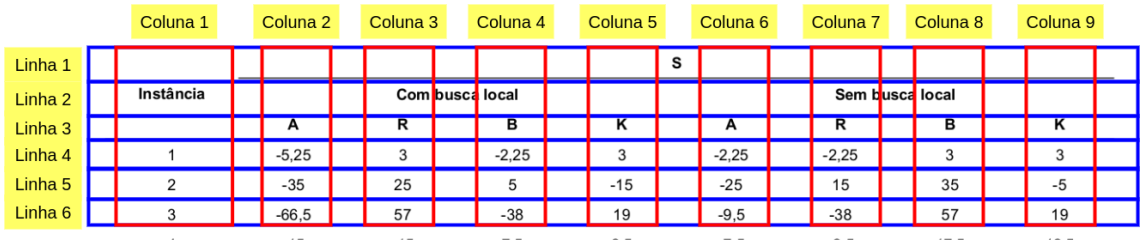


Figura 2. Exemplo do reconhecimento das linhas e colunas de uma tabela

imprescindível, como na recuperação de informação em grandes volumes de dados e em ferramentas de acessibilidade para deficientes visuais.impractical jokers

Apesar de modelos baseados em Deep Learning terem impactado o campo do Entendimento de Documentos e, por consequência, o processamento de tabelas, os principais modelos disponíveis ainda se mostram dependentes de grande quantidade de pré ou pós processamentos, seja na utilização das peculiaridades do formato dos dados ou na utilização de motores de Reconhecimento de Caracteres Visuais (Ferramentas para a extração de texto em imagens cuja sigla mais comum é OCR) [Kasem et al. 2022].

Neste artigo, foi treinado um novo modelo Transformer para a tarefa de Reconhecimento de Tabela que toma amplo proveito da disposição bidimensional das células durante a geração da sequência de saída. Para isso, criamos uma nova codificação para sequência a ser predita, que uniformiza a topologia da tabela, reduz o tamanho de sequência e nos permite utilizar *embeddings* de posição tridimensionais durante geração auto-regressiva. Além disso, nosso modelo dispensa o uso de caixas delimitadoras ou motores OCR, facilitando o processo de treinamento.

1.1. Reconhecimento de Tabela

A tarefa do processamento de tabelas de formatos pouco-estruturados para formatos que sejam facilmente utilizados por sistemas automatizados é comumente referida como Reconhecimento de Tabela [Zhong et al. 2020]. Nessa tarefa, tem-se como objetivos reconhecer a estrutura da tabela em linhas e colunas, identificar células expandidas (células que ocupam mais de uma coordenada) e seus tamanhos, identificar células que sejam cabeçalhos e efetuar a transcrição do conteúdo textual de cada célula para formatos facilmente computáveis. As Figuras 2, 3 e 4 são exemplos de identificação de estrutura, células expandidas e de cabeçalhos sendo aplicadas à tabela original, contida na Figura 1.

1.2. O Problema

O Reconhecimento de Tabelas é um problema complexo e, como um todo, é profundamente dependente do formato utilizado para codificação das informações a serem proces-

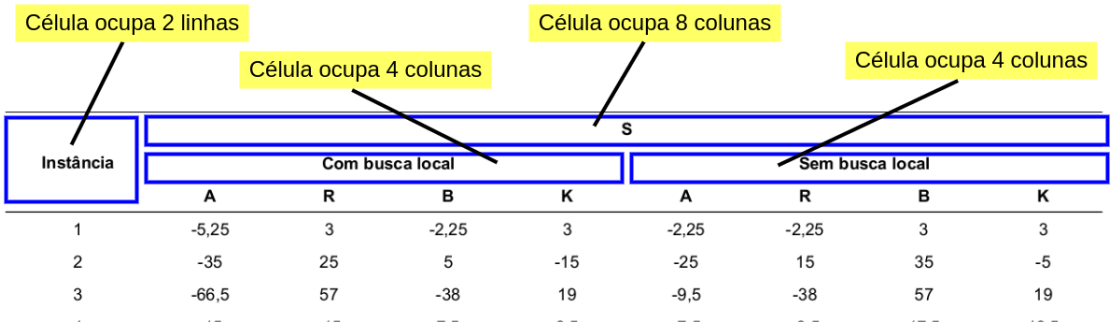


Figura 3. Exemplo da identificação de células expandidas

Figura 4. Exemplo da identificação cabeçalhos

sadas [Göbel et al. 2013]. A identificação, reconhecimento e extração podem ser mais ou menos complexas dependendo da quantidade de dados estruturais e textuais preservados em cada formatação.

Tabelas em formato HTML são geralmente uma opção com estrutura bem definida e de mais fácil extração, permitindo organizar os dados em linhas e colunas, usando tags como <table>, <tr>, <td> e <th> para explicitar sua estrutura. Erros de extração neste caso se limitam a documentos mal formatados [Cafarella et al. 2018]. Entretanto, certos tipos de documentos, como artigos científicos, documentos governamentais e financeiros, raramente são publicados em formato HTML.

Até formatos mais bem definidos, como os gerados por editores de planilhas e texto, sofrem com imprecisões na representação dos dados tabulares. Nestes, as imprecisões são geradas majoritariamente por diferenças no estilo de cada autor, que podem usar cores, bordas, alinhamentos e outros recursos visuais para destacar ou agrupar informações. Além disso, muitos autores assumem que o interlocutor é capaz de interpretar o significado e a relação entre as células da tabela, sem fornecer padrões claros ou cabeçalhos adequados. Esses fatores dificultam a extração automatizada dos dados ao ponto que determinados corpus não contêm nem ao menos 3% das tabelas normalizadas o bastante para que a extração nestes formatos seja trivial [Dong et al. 2019].

O formato PDF, um dos formatos digitais mais populares na atualidade [Nassar et al. 2022], conta com uma forte compressão que resulta uma perda quase total da informação estrutural dos documentos, dificultando severamente a extração. PDF foi criado para preservar a aparência visual dos documentos, independentemente do dispositivo ou plataforma usados para visualizá-los. No entanto, isso implica que os elementos do documento, como tabelas, gráficos, imagens e texto, são armazenados como objetos gráficos, sem grandes informações sobre o sua estrutura ou semântica

[Hassan and Baumgartner 2007].

Documentos escaneados ou em formato de imagem representam o extremo oposto ao HTML em termos de preservação de informação. Toda a informação estrutural e textual que originalmente compunha o documento é perdida, já que as páginas são convertidas para agrupamentos de píxels.

2. Trabalhos Relacionados

Com as distinções descritas na Seção 1.2, pode-se perceber a complexidade do problema. Na literatura, nota-se que diversas soluções foram propostas para diferentes formatos de dados.

[Cafarella et al. 2018] desenvolveram ferramentas eficazes para extrair informações de páginas web em formato HTML. Aproveitando-se do código fonte disponível e da sintaxe bem-definida da linguagem, o trabalho possibilitou diversos avanços e ferramentas para indexar informações web, sendo até utilizado em ferramentas de busca como Google Tables.

Ao utilizar regiões de layout e codificar as inter-relações espaciais entre elas como um grafo, [Koci et al. 2018] propuseram o algoritmo Remove and Conquer (RAC), que aplica uma lista de regras para remover conexões específicas do grafo e identificar as tabelas e células. Entretanto, o algoritmo requer que as informações estejam em formato de planilhas.

Modelos com componentes visuais são comuns [Huang et al. 2019, Sun et al. 2019, Agarwal et al. 2020, Paliwal et al. 2019]. Geralmente, eles utilizam um módulo para os reconhecimentos de tabela, estrutura e função, encontrando linhas de separação ou caixas delimitadoras, e outro para a extração do conteúdo em si.

Utilizando DETR, um modelo de detecção de objetos baseado em Transformers, e um dataset com quase 1 milhão de exemplos, [Smock et al. 2021] alcançaram o estado da arte em Detecção de Tabela e Reconhecimento de Estrutura. Ao detectar linhas, colunas, células expandidas e cabeçalhos, os autores puderam encontrar as caixas delimitadoras e as coordenadas topológicas de cada célula. Com essas informações, então, podem fazer a referência cruzada desses dados para a extração textual. A extração textual usada no artigo foi feita diretamente dos arquivos PDF, o que nem sempre é viável, seja por codificações específicas ou pela representação dos dados em outros formatos, esses casos requeririam a aplicação de um motor de OCR.

Outro modelo relevante, não só pela originalidade, mas também pela proximidade com nossa proposta, foi publicado por [Zheng et al. 2020]. Nele os autores utilizaram um codificador convolucional para entradas em imagem e dois decodificadores LSTM (*Long Short-term Memory*) baseados em mecanismos de atenção. Um para extração de estrutura e outro para extração de conteúdo, produzindo uma saída completa já em formato HTML. No entanto, apesar de seu grande potencial, o modelo ainda apresenta algumas importantes desvantagens: o treinamento do modelo foi efetuado num limite baixo para tamanho de sequência (somente 300 *tokens* de estrutura) e teve longa duração (16 dias em duas GPUs V100).

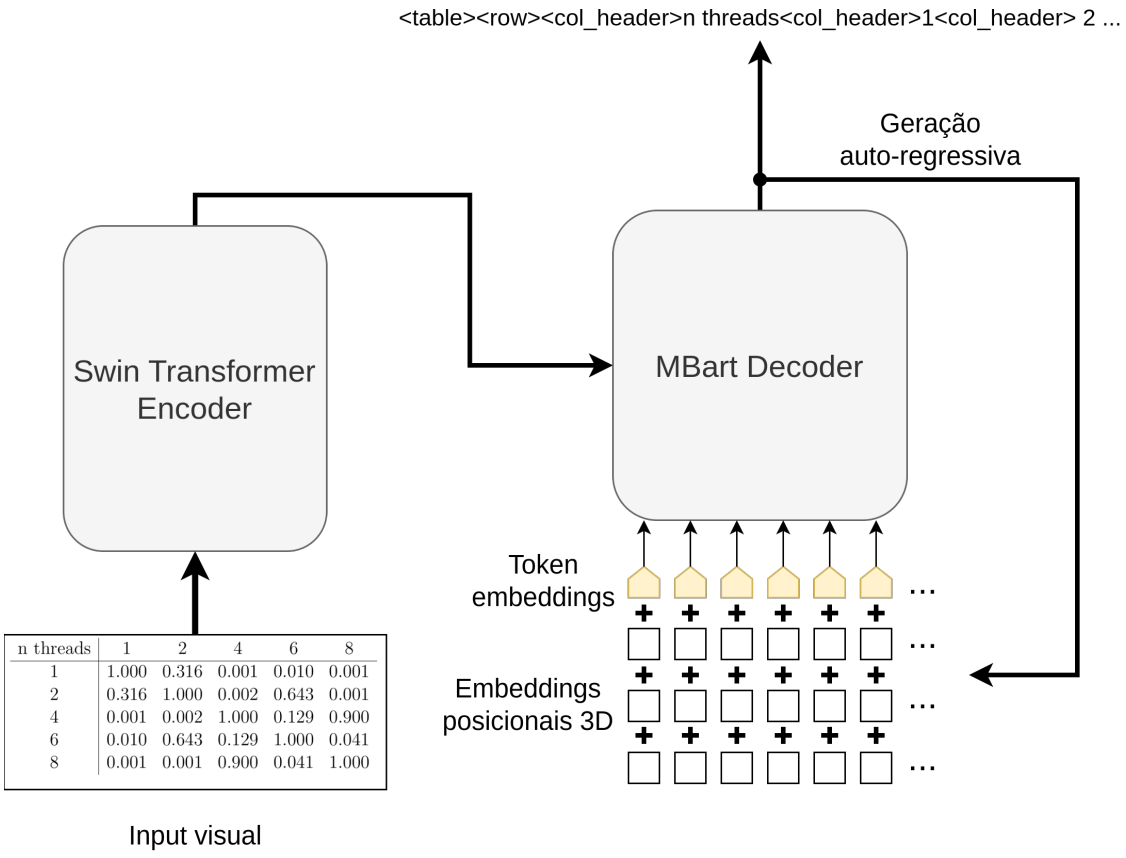
Apesar das diversas soluções específicas propostas, o reconhecimento generalizado ainda se mostra como um problema a ser solucionado [Kasem et al. 2022]. Uma

solução geral para o reconhecimento topológico, textual e semântico implica na utilização da única constante entre todos os formatos apresentados: as informações visuais. Este é um complicador importante, já que geralmente implica na utilização de ferramentas externas, como motores de OCR, os quais nem sempre apresentam melhor velocidade de processamento e qualidade no reconhecimento tabular. Nosso modelo utiliza somente informações visuais para todas as etapas de reconhecimento, e, portanto, pode ser aplicado a qualquer tabela que possa ser representada por uma imagem.

3. Modelo Proposto

Ao analisar as limitações dos métodos estudados, observa-se a lacuna que um modelo que consiga extrair informações estruturais, funcionais e textuais simultaneamente poderia preencher. Assim, tomando proveito dos recentes avanços da capacidade de processamento e na disponibilidade de dados, foi avaliada a viabilidade de um modelo que atenda esses requisitos.

Para isso, utilizou-se um modelo imagem-para-sequência pré-treinado baseado num *encoder* Visual e um *decoder* textual cujos *embeddings* posicionais foram alterados a fim se adaptar melhor à disposição bidimensional dos elementos tabulares. A Figura 5 mostra a arquitetura usada para o modelo.



n threads	1	2	4	6	8
1	1.000	0.316	0.001	0.010	0.001
2	0.316	1.000	0.002	0.643	0.001
4	0.001	0.002	1.000	0.129	0.900
6	0.010	0.643	0.129	1.000	0.041
8	0.001	0.001	0.900	0.041	1.000

Figura 5. Arquitetura geral do modelo.

3.1. Donut

O modelo escolhido foi o proposto e pré-treinado por [Kim et al. 2022], que utiliza um *encoder* visual Swin Transformer [Liu et al. 2021] para a as entradas e um *decoder* mBart [Liu et al. 2020] para geração auto-regressiva da sequência de saída. A proposta dos autores é um modelo imagem-para-sequência que consiga predizer anotações JSON tendo somente informações visuais como entrada.

O modelo base disponibilizado pelos autores foi pré-treinado para tarefas de Entendimento de Documento e já compreende a extração de texto em diversos idiomas. Os autores também efetuaram o ajuste fino para problemas como Pergunta e Resposta em Documentos Visuais e Entendimento de Nota Fiscal, mas a aplicação para tarefas de Reconhecimento de Tabela ainda é inédita na literatura.

3.2. Conjunto de Dados

Existem diversos conjuntos de dados diferentes para o processamento de tabelas [Smock et al. 2021, Chi et al. 2019, Li et al. 2020], cada um com características especificas. Para nosso ajuste fino, utilizou-se o dataset Pubtabnet [Zheng et al. 2020]. Escolhemos esse dataset por, principalmente, três vantagens apresentadas sobre os demais: sua extensão, com pouco mais de 500 mil tabelas de exemplos, é um dos maiores dataset para este fim já publicados; a presença de anotações completas específicas para modelos imagem-para-sequência e, também, pela resolução dos exemplos, que é o bastante para o reconhecimento de conteúdo sem consumo excessivo de armazenamento.

3.3. HTML e Nova Codificação

As anotações providas pelo conjunto de dados são em formato HTML [Zheng et al. 2020, Ye et al. 2021]. Porém, esta forma de representação demonstra algumas notáveis desvantagens:

1. As sequências são excessivamente longas. O uso de *tags* de abertura e fechamento (ex: <td> e </td>) são úteis para elementos aninhados e para melhor legibilidade, entretanto legibilidade não é um conceito importante na codificação para máquinas e a utilização das *tags* de fechamento se faz desnecessária em determinadas hierarquias. Especialmente em transformers de atenção completa, cuja complexidade espacial para tamanho de sequência é $\Theta(n^2)$, sequências longas podem ser um problema.
2. A forma com que células expandidas são comumente representadas [Zheng et al. 2020, Ye et al. 2021], utilizando dois tipos de *tags* (*colspan*='n' e *rowspan*='m', com m e n indo de 2 até n_{max} e m_{max}, sendo n_{max} e m_{max} as expansões máximas contidas nos dados das colunas e das linhas respectivamente), dificulta o treinamento e a extrapolação dos dados.
   Esta forma de codificação geralmente implica na criação de, no mínimo, $n_{\text{max}} + m_{\text{max}}$ novos *embeddings* e, em virtude da maneira com que os modelos de linguagem geralmente operam, a relação linear entre cada um dos novos *embeddings* de *colspan*='2' até *colspan*='n_{max}' (ou 2 até m_{max} para *rowspans*) precisaria ser descoberta nos dados de treinamento.
   Isso, além de aumentar o que é exigido do modelo, implica que o reconhecimento de células expandias está profundamente ligado à variedade dos dados de treino:

para valores de n ou m raros, a precisão tenderá a ser inferior; para valores de n ou m ausentes nos dados de treino, o modelo será incapaz de realizar a decodificação correta.
Uma possível forma de extrapolação é o utilizar modelos que já entendam a relação linear entre os números inteiros e, então, utilizar *embeddings* que já sejam conhecidos para a representação de n e m. Entretanto, extrapolar a relação dos números inteiros com o tamanho de célula não é trivial e não é garantido que o modelo consiga fazê-lo para dados fora do treinamento.

3. Células expandidas alteram a uniformidade da estrutura codificada. Uma vez que uma célula expandida é declarada no código HTML, ela passa a ocupar o espaço de células vizinhas. Uma célula expandida de *colspan* n e *rowspan* m alteraria a topologia das próximas m linhas, ao reduzir a quantidade de células em cada linha em n elementos. Então, somado ao complicador citado no Item 2, seria esperado do modelo prever os $n \times m$ impactos topológicos de cada célula expandida baseado em somente dois *tokens* específicos da sequência. A complexidade pode ser ainda maior caso ocorra a interseção dos impactos de duas ou mais células expandidas.

Desta maneira, para nosso treinamento, propomos e utilizou-se uma nova forma de codificação que é uma representação um-para-um do formato HTML, mantendo as vantagens do formato porém aliviando os problemas descritos acima.

A nova codificação foi denominada de GTML(Graph-based Table Machine Language) e as alterações que ela apresenta com relação ao HTML são as seguintes:

1. Dispensou-se o uso de *tags* de fechamento, utilizamos somente *tags* de inicio a fim de indicar o conteúdo textual da célula, de maneira semelhante a separadores em elemento em uma lista. Essa mudança pode reduzir o tamanho das sequências estruturais até pela metade.
2. Alterou-se as *tags* de cabeçalho que são utilizadas a cada grupo de linhas no formato HTML por *tags* de cabeçalho utilizadas a cada célula, substituindo a *tag* de início mencionada acima, já que reduz o tamanho de sequência e aumenta as possibilidades de representação.
3. Utilizou-se uma representação de células expandidas inédita para codificação tabela-para-sequência que mantém uma estrutura uniforme para tabela, utiliza somente 16 *tokens* especiais para a representação de qualquer célula expandida independente de tamanho e utiliza um padrão que é reforçado durante a decodificação. Além disso, os *tokens* são comuns nos dados, precisando de somente uma célula de dimensão 3×3, uma de 1×3 e uma de 3×1 para encontrar todos *tokens* de codificação possíveis, ao contrário de $n_{\text{max}} + m_{\text{max}}$ células necessárias para uma codificação em formato HTML, com n_{max} e m_{max} sendo os valores máximos de expansão suportado pelo modelo nas linhas e nas colunas, respectivamente.
Nessa nova representação, cada tabela é um grafo G, com N subgrafos desconexos, sendo N o número de células. Cada célula unitária (não expandida) é um subgrafo G_{i} com 1 vértice e 0 arestas; cada célula expandida é um subgrafo G_{i} com $n \times m$ vértices e todo vértice se conecta a seus vizinhos imediatos que também fazem parte do subgrafo, totalizando $(n-1) \times m + (m-1) \times n$ arestas, com n e m sendo o tamanho da célula expandida nas linhas e nas colunas, respectivamente.
Para definir as arestas de um vértice, utilizou-se uma codificação binária (conectado ou não conectado) com 4 posições, uma para cada vizinho, sendo a pri-

meira posição para conexão à direita; segunda, conexão abaixo; terceira, acima e a quarta, à esquerda. Para reduzir o tamanho de sequência e para aproximar ao máximo o valor decodificado da sua posição na imagem, o conteúdo de uma célula expandida é contido somente no vértice central do grafo. A Figura 6 é um exemplo da representação em grafo.

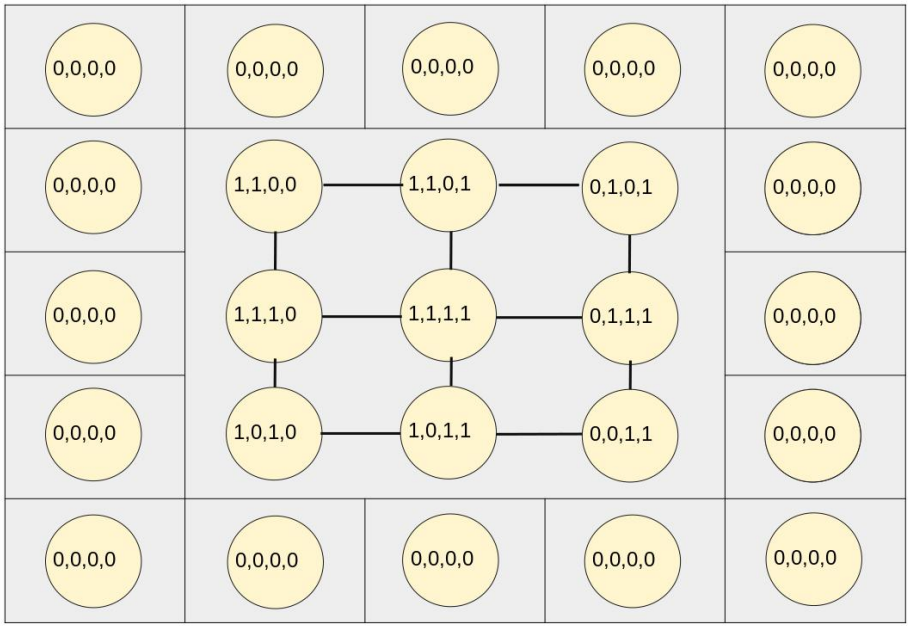


Figura 6. Exemplo do grafo de representação de uma tabela 5×5 com uma célula expandida 3×3 ao centro

3.4. Codificação Posicional

Até nos modelos mais recentes de Reconhecimento de Tabela [Ye et al. 2021, Ly and Takasu 2023], é padrão a utilização de codificações posicionais sequenciais e diretas, entretanto, apesar de serem ideais para o processamento de linguagem natural, a codificação posicional sequencial não se adéqua tão bem às tabelas, cujos conteúdos estão dispostos em duas dimensões (três, se contarmos a ordem dos valores de conteúdo das células).

Uma codificação posicional que leva em consideração diferenças de dimensão não é inédita [Huang et al. 2022], mas é ausente na tarefa de Reconhecimento de Tabelas, principalmente graças a dois importantes complicadores: A geração auto-regressiva não permite o pré-processamento das posições antes da decodificação e as células expandidas complicam a determinação das coordenadas, graças aos seus impactos topológicos.

Entretanto, utilizando codificação proposta, GTML, que uniformiza a topologia, pode-se facilmente definir o posicionamento de cada *token* em tempo constante, desde que se saiba o estado dos contadores de linhas, células e de conteúdo de célula durante a geração. Então, contando com a fácil definição garantida pela codificação, foi adicionado ao modelo de decodificação mBart um novo posicionamento tridimensional, composto por três listas de *embeddings* representando cada uma das três coordenada de cada *token* decodificado.

Ao encontrar qualquer *token* que não seja especial, soma-se um à primeira dimensão; ao encontrar *tokens* especiais de célula, zera-se o contador da dimensão anterior e aumenta-se a dimensão de células, o mesmo acontece quando encontra-se *tokens* especiais de linhas, zerando os contadores de todas as dimensões anteriores e somando um ao contador de linhas. A Figura 7 é um exemplo dos posicionamentos. No caso atual, é dispensável, mas o mesmo processo poderia ser facilmente extrapolado para ainda mais dimensões.

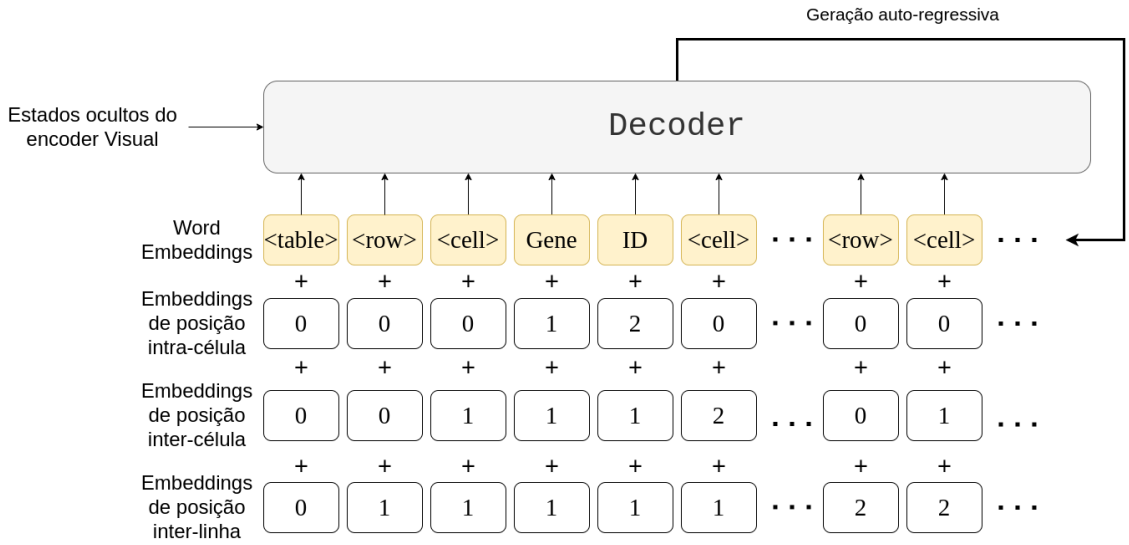


Figura 7. Posicionamento dimensional durante a geração

4. Treinamento

Realizou-se o ajuste fino em dois modelos, ambos iniciando pelo modelo pré-treinado Donut base e tendo sua codificação posicional alterada, mas utilizando duas codificações diferentes, HTML com *encoddings* sinusoidais (como propostos por [Vaswani et al. 2017]) e GTML com posicionamento tridimensional.

Os modelos foram treinados por uma única época, em uma GPU RXT3090. Para reduzir o consumo de memória durante o treinamento, fez-se o truncamento das sequências de treinamento para um máximo de 1500 *tokens*, o que engloba mais de 99% do total dos tokens de treinamento nos dados. A resolução do modelo foi mantida em 720×720 píxels, pelo mesmo motivo.

Utilizou-se os valores de taxa de aprendizado como 1e-4 e um tamanho de *batch* igual a 4, os quais geraram melhores resultados durante a etapa de busca de hiperparâmetros (valores de tamanho de *batch* maiores que 4 não foram avaliados devido ao limite de memória gráfica do sistema).

5. Resultados

Após o ajuste-fino dos modelos, mediu-se, então, a taxa de acerto baseado na *TEDS* (Tree Edit Distance Score), utilizando o código disponibilizado por [Zhong et al. 2020]. Esta métrica mede a quantidade mínima de edições necessárias para que a sequência predita seja igual à sequência verdadeira. A distância entre *tags* de estrutura é encontrada com

	Reconhecimento Completo			Estrutura		
	simples	complexas	todas	simples	complexas	todas
HTML + enc. sin.	83,9%	78,0%	81,0%	94,2%	88,9%	91,6%
GTML + pos. 3D	85,8%	80,5%	83,2%	96,2%	91,6%	93,9%

Tabela 1. Comparação em TEDS. Tabelas complexas são as que contêm ao menos uma célula expandida.

uma busca em árvore e a distância entre conteúdos é dada pela distância de Levenshtein de comparação entre strings. Os valores de distância, então, são convertidos para coeficientes de semelhança entre 0 e 100%.

Dividiu-se os resultados em reconhecimento completo (conteúdo e estrutura) e reconhecimento somente de estrutura. E, para cada resultado também analisou-se a diferença de performance para tabelas simples e para tabelas complexas, que possuem uma ou mais células expandidas.

A codificação GTML mostrou-se superior ao reconhecimento diretamente para a linguagem HTML em todas as subdivisões de teste, com a maior vantagem advinda de tabelas complexas. O resultado indica uma possível vantagem de convergência, além de maior capacidade de generalização ao utilizar a nova codificação. Os resultados são mostrados na Tabela 1.

6. Conclusão

Apesar das inúmeras vantagens da utilização do reconhecimento imagem-para-sequência estudado, como menor custo computacional, menor pré-processamento e maiores possibilidades de ampliação dos dados, o Reconhecimento de Tabelas ainda é um desafio. Especialmente para modelos que utilizam a arquitetura *transformer*, as tabelas no último decil dos dados são mais complexas, consomem mais recursos e são mais esparsas, o que acaba reduzindo a capacidade de treinamento e de inferência.

Para modelos que utilizam as caixas delimitadoras de cada célula para extração textual, como [Nassar et al. 2022], a codificação GTML e o posicionamento tridimensional ainda podem ser utilizados, já que há a possibilidade de heuristicamente dividir cada caixa delimitadora de células expandidas em $n \times m$ sub-caixas, a fim de uniformizar a predição para o treinamento, da mesma maneira que mostrou-se para predição de estrutura. Para a inferência, pode-se, então, reconstruir as caixas completas, unindo as $n \times m$ sub-caixas preditas pelo modelo, e, assim, tirar proveito da estrutura em grade dos dados visuais para extração textual da mesma maneira com que faz-se para predição de estrutura.

A codificação proposta e a utilização do modelo Donut se mostraram eficientes em possibilitar a extração de mais dados com menos tempo de treinamento e são mais um passo em direção a um modelo baseado em *transformers* que consiga o reconhecimento livre de OCR independentemente de tamanho de tabela.

7. Trabalhos Futuros

Ainda seria de interesse analisar se os mesmos benefícios obtidos pelos métodos apresentados poderiam se estender para tarefas nas quais informações tabulares são os dados de

entrada, como as tarefas de Sumarização de Tabela [Liu et al. 2022] e Resposta a Perguntas em Tabelas [Jin et al. 2022].

Referências

Agarwal, M., Mondal, A., and Jawahar, C. V. (2020). Cdec-net: Composite deformable cascade network for table detection in document images. *25th International Conference on Pattern Recognition (ICPR)*, pages 9491–9498.

Cafarella, M., Halevy, A., Lee, H., Madhavan, J., Yu, C., Wang, D. Z., and Wu, E. (2018). Ten years of webtables. *Proceedings of the VLDB Endowment.*, 11(12):2140–2149.

Chi, Z., Huang, H., Xu, H.-D., Yu, H., Yin, W., and Mao, X.-L. (2019). Complicated table structure recognition. *arXiv preprint arXiv:1908.04729*.

Dong, H., Liu, S., Han, S., Fu, Z., and Zhang, D. (2019). TableSense: Spreadsheet table detection with convolutional neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):69–76.

Göbel, M., Hassan, T., Oro, E., and Orsi, G. (2013). Icdar 2013 table competition. In *12th International Conference on Document Analysis and Recognition*, pages 1449–1453. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

Hassan, T. and Baumgartner, R. (2007). Table recognition and understanding from pdf files. *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, 2:1143–1147.

Huang, Y., Lv, T., Cui, L., Lu, Y., and Wei, F. (2022). Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 4083–4091, New York, NY, USA. Association for Computing Machinery.

Huang, Y., Yan, Q., Li, Y., Chen, Y., Wang, X., Gao, L., and Tang, Z. (2019). A yolo-based table detection method. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 813–818.

Jin, N., Siebert, J., Li, D., and Chen, Q. (2022). A survey on table question answering: Recent advances. In *China Conference on Knowledge Graph and Semantic Computing*.

Kasem, M., Abdallah, A., Berendeyev, A., Elkady, E., Abdalla, M., Mahmoud, M., Hamada, M., Nurseitov, D., and Taj-Eddin, I. (2022). Deep learning for table detection and structure recognition: A survey. *arXiv preprint arXiv:2211.08469*.

Kim, G., Hong, T., Yim, M., Nam, J., Park, J., Yim, J., Hwang, W., Yun, S., Han, D., and Park, S. (2022). Ocr-free document understanding transformer. In *17th European Conference on Computer Vision – ECCV: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, page 498–517, Berlin, Heidelberg. Springer-Verlag.

Koci, E., Thiele, M., Lehner, W., and Romero, O. (2018). Table recognition in spreadsheets via a graph representation. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 139–144.

Li, M., Cui, L., Huang, S., Wei, F., Zhou, M., and Li, Z. (2020). TableBank: Table benchmark for image-based table detection and recognition. In *Proceedings of the*

Twelfth Language Resources and Evaluation Conference, pages 1918–1925, Marseille, France. European Language Resources Association.

Liu, S., Cao, J., Yang, R., and Wen, Z. (2022). Long text and multi-table summarization: Dataset and method. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1995–2010, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., and Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002.

Ly, N. T. and Takasu, A. (2023). An end-to-end multi-task learning model for image-based table recognition. In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP*, pages 626–634. SciTePress.

Nassar, A. S., Livathinos, N., Lysak, M., and Staar, P. W. J. (2022). Tableformer: Table structure understanding with transformers. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4604–4613.

Paliwal, S., D, V., Rahul, R., Sharma, M., and Vig, L. (2019). Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images. *International Conference on Document Analysis and Recognition (ICDAR)*, pages 128–133.

Smock, B., Pesala, R., and Abraham, R. (2021). Pubtables-1m: Towards comprehensive table extraction from unstructured documents. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4624–4632.

Sun, N., Zhu, Y., and Hu, X. (2019). Faster r-cnn based table detection combining corner locating. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1314–1319.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Ye, J., Qi, X., He, Y., Chen, Y., Gu, D., Gao, P., and Xiao, R. (2021). Pingan-vcgroup's solution for icdar 2021 competition on scientific literature parsing task b: Table recognition to html. *ArXiv*, abs/2105.01848.

Zheng, X., Burdick, D., Popa, L., and Wang, N. X. R. (2020). Global table extractor (gte): A framework for joint table identification and cell structure recognition using visual context. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 697–706.

Zhong, X., ShafieiBavani, E., and Jimeno Yepes, A. (2020). Image-based table recognition: Data, model, and evaluation. In *16th European Conference on Computer Vi-*

sion – ECCV 2020: Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI, page 564–580, Berlin, Heidelberg. Springer-Verlag.