MARCOS CLEISON SILVA SANTANA

Extração de Relações Semânticas e Grafos Contextuais a Partir de Imagens de Cardápios Digitalizados de Restaurantes Usando Redes Neurais Profundas

Criação de Ferramentas de Processamento de Cardápios envolvendo Detecção, OCR, e Aprendizagem Supervisionada

Relatório de Pós-Doutorado realizado na Universidade Estadual Paulista (UNESP), Faculdade de Ciências, Departamento de Computação, Bauru.

Supervisor(a): Prof. Dr. João Paulo Papa

Bauru

2025

Sumário

Resumo

Este relatório apresenta o desenvolvimento de um sistema para extração de informações hierárquicas de cardápios digitalizados, abrangendo desde a detecção de texto via OCR, até a classificação de grupos e itens de menu, utilizando técnicas de aprendizado supervisionado e não supervisionado, redes convolucionais e recorrentes, além de redes convolucionais em grafos (GCNs) para análise semântica. O estudo explorou também a detecção de regiões de texto através de imagens mascaradas, a clusterização por tamanho de fonte e diversas técnicas de redução de dimensionalidade (PCA, t-SNE e Isomap) para visualização dos dados. Um sistema de rotulação foi desenvolvido como produto mínimo viável, demonstrando o potencial da combinação das abordagens para uma extração de informações precisa e o auxílio na geração de dados rotulados para futuros aprimoramentos.

1 Introdução

Frequentemente, a necessidade de crescimento quanto ao volume de atendimento dos clientes de certos setores comerciais foi limitada pela natureza manual das ações que envolvem os processos de prestação de serviços. Diversos setores, tanto os de natureza business to customer quanto business to business, têm buscado desenvolver práticas e soluções que aceleram a evolução dos processos de atendimento aos cliente por meio de boas práticas, adquiridas ao longo dos anos, como também por meio do uso de ferramentas inteligentes que consigam solucionar as dores dos clientes e provedores de produtos e serviços.

Nos últimos anos, as empresas têm notado que ferramentas inteligentes baseadas em aprendizado de máquina podem ser um meio escalável de reduzir alguns problemas inerentes aos negócios. Alguns deles relacionados com o tempo de processamento das informações providas pelos clientes, outras pela falta de entendimento correto da carteira de clientes existente e dificuldade de verificar o status quo do mercado.

Empresas que provêm serviços de selling points de restaurantes e estabelecimentos similares não estão imunes às dificuldades relacionadas com obtenção e processamento dos dados, bem como o comportamento dos clientes. Em geral, as empresas que vendem dispositivos de selling points para restaurantes e outros estabelecimentos, necessitam receber do cliente o cardápio, as descrições, preços e outras informações necessárias para a completa prestação de serviços. Esta necessidade se enquadra nos problemas relacionados com atrasos nos processos, pois em geral, os clientes enviam uma imagem digitalizada dos menus, cardápios, lista etc.

A ação padrão nestes casos consiste em receber a imagem digitalizada e extrair as informações necessárias utilizando um agente humano. Apesar de efetiva e frequente,

esta prática não é otimizada por que consome tempo e recursos humanos. Desta forma, apresenta-se a necessidade de resolver minimamente este gargalo por meio de tecnologias de aprendizagem de máquina em que um processo inteligente é criado a partir dos dados e melhorado ao longo do tempo, criando um ciclo virtuoso de evolução.

O presente relatório visa principalmente apresentar os resultados parciais de pesquisa e técnicas desenvolvidas e implementadas para o processamento de imagem de cardápios, extração de informações fundamentais e criação de ferramentas de anotações.

2 Objetivos

Nesta seção, apresentaremos os objetivos gerais, bem como um detalhamento mais específicos acerca das atividades efetuadas.

- Estudar sistemas de Reconhecimento de Caracteres (OCR) para extração de informações a partir de imagens de cardápios digitalizados.
  - Criar classificador supervisionado de título de menus/itens de menu para classificação primária.
  - Estudar rede detectora de objetos como classificadora de grupo de menus.
  - Criar sistema de extração de características de fontes para caracterização não supervisionada.
  - Desenvolver sistema de anotação de cardápios digitalizados.
- Estudar técnicas de aprendizado de máquina para classificação de grupos e itens de menu.
  - Implementar classificador de grupos de menus.
  - Implementar classificador de itens de menu.
  - Explorar projeções de embeddings para análise semântica a partir de redes convolucionais em grafos.

3 Metodologia

A percepção da existência de problemas em processos de negócios nem sempre são evidentes, especialmente em negócios como os serviços de selling points, objeto de

estudo do presente relatório. Para evidenciar os problemas e propor soluções, é necessário se utilizar de uma metodologia que envolve multidisciplinaridade no entendimento do problema e ciclos iterativos entre pesquisa e desenvolvimento. A metodologia consiste em duas fases: a coleta de dados e ciclos de pesquisa-desenvolvimento.

A primeira fase consiste em obter os dados da prestadora de serviço de selling point, interagindo com os times de negócios, contabilidade e vendas. Esta interação visa obter os dados relevantes e entender as necessidades e problemas que os processos apresentam no atendimento, conservação ou captação dos clientes. Nesta fase, os cientistas de dados compõem a base de dados por meio de atividades que envolvem limpeza de dados inválidos, preenchimento de dados faltantes pré-processamento e em alguns casos rotulação.

Enquanto a primeira fase foca na obtenção dos dados e entendimento dos problemas, a segunda fase consiste em executar processos de interação pesquisa-desenvolvimento. Esta fase tem como foco, utilizar métodos de aprendizagem de máquina na busca por soluções a partir da base de dados formada. A partir dos experimentos efetuados, a proposta de solução é aplicada na formação de um produto mínimo viável, fechando o ciclo pesquisa desenvolvimento. Note que o termo produto minimamente viável consiste em um conjunto de aplicações ou visualizações que permitam os times de estratégia e desenvolvimento evoluir, integrando a uma solução. A Figura 1 apresenta a estrutura básica da metodologia utilizada nesta fase do projeto.

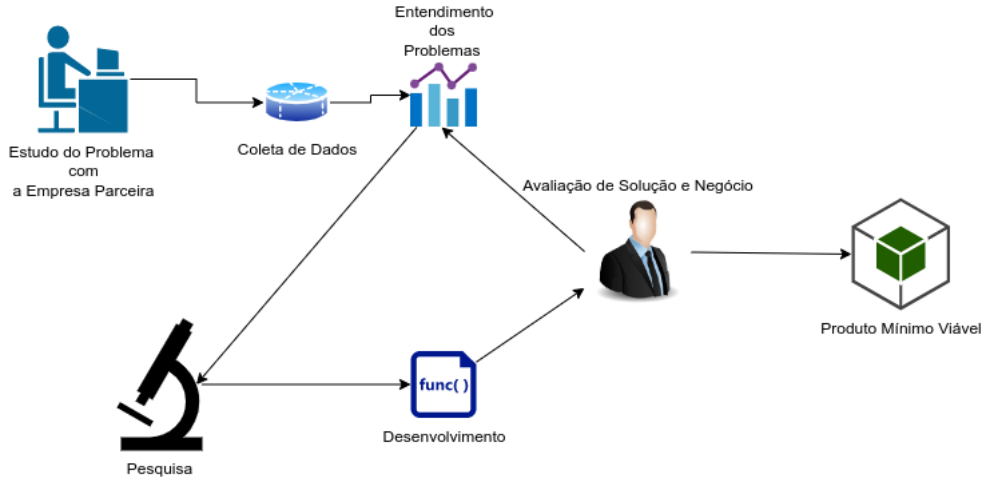

Figura 1: Metodologia de processos utilizada no desenvolvimento das atividades do projeto.

3.1 Materiais e Métodos

Para desenvolvimento das propostas do projeto, utiliza-se de ferramentas de ciência de dados e de programação. As análises de dados e processamento são feitas principalmente utilizando a linguagem Python 3.8 [29]. Os dados estruturados são carregados e processados principalmente utilizando a biblioteca Pandas [33] e SQL (Postgres) [19] em casos específicos. As técnicas envolvendo aprendizagem de máquina são providas em sua maioria pela biblioteca Scikit Learn [18] enquanto ferramentas de processamento de linguagem natural advém das bibliotecas Gensim [22], Fasttext [4] e NLTK [3]. As tarefas de visualização são providas por ferramentas de ciências de dados como Jupyter notebooks [14], Matplotlib [12] e Seaborn [32]. Nas fases de desenvolvimento, os sistemas de backend foram criados em Python utilizando o framework Flask [10] visando prover API. A construção do frontend e interfaces de interação são desenvolvidas em Typescript [2] utilizando o framework Vuejs 3.0 [31] e framework CSS Tailwind [24].

Alguns serviços providos por terceiros foram utilizados, em especial serviços da Amazon AWS [27] para extração de caracteres OCR a partir de imagens detalhadas.

4 Aspectos Teóricos

As necessidades do presente envolvem tarefas relacionadas com processamento de imagem, reconhecimento de padrões, extração semântica de características, extração de caracteres, processamento de textos e agrupamento de dados. Nesta seção, será dado um vislumbre dos aspectos teóricos básicos utilizados durante a execução do processo.

4.1 Extração de Caracteres

Extração de Caracteres (Optical Character Recognition OCR [16]) consiste em detectar e reconhecer caracteres a partir de imagens. No princípio, as técnicas de OCR utilizaram métodos de extração de características confeccionadas *ab-initio*. Tais técnicas, exemplo das features Viola-Jones [25], conseguiam até certo grau detectar a extração de informações das regiões de caracteres. Tais regiões eram segmentadas e enviadas para um classificador do tipo *Support Vector Machine* (SVM) [5]. Apesar de ser uma evolução para a época, o classificador apresentava dificuldade em reconhecer caracteres com fontes diferentes. Tal dificuldade era mais evidente em caracteres providos de textos cursivos em que a variabilidade era mais acentuada.

Yann LeCun, proveu uma destacada contribuição com a criação das redes convolucionais neurais (*Convolutional Neural Network - CNN*) [15]. Diferentemente das redes

perceptron de múltiplas camadas, as redes convolucionais neurais são formadas por filtros convolucionais que extraem características relevantes para a classificação dos caracteres. A aprendizagem se dá pelo algoritmo de retropropagação padrão. Esta técnica permite que as camadas convolucionais filtrem as características que apresentam mais separabilidade entre as classes de caracteres de interesse, facilitando a classificação para os classificadores de saída.

O problema da dificuldade de classificação com diferentes fontes e estilos cursivos foi grandemente mitigado com as redes convolucionais após a criação da arquitetura *LeNet* [15] para reconhecimento de dígitos. Nascia aí não somente uma revolução não somente para reconhecimento de caracteres, mas também de diversas áreas passando por visão computacional, processamento de linguagem natural etc.

A criação das camadas convolucionais em conjunto com outras camadas permitiram o desenvolvimento de arquiteturas mais profundas (com mais camadas) e maior capacidade de extração de características e melhoria acentuada da acuracidade de reconhecimento de padrões. Tal melhoria é verificada em certas tarefas de classificação alcançando marcas em torno de 99% para certas tarefas [1].

Além das camadas convolucionais que aprendem a extrair as características, as arquiteturas e CNN modernas têm a camadas de agregação (*pooling*) que agregam as features vizinhas reduzindo a dimensionalidade do tensor de entrada da camada. Esta agregação comumente se dá por meio da obtenção do valor máximo (*max pooling*) da região agregada, ou do valor médio (*average pooling*) [8]. Esta agregação é importante para fomentar a extração de características em um nível cada vez mais alto à medida que os dados de características são enviados para camadas mais profundas. Assim, enquanto as primeiras camadas tendem a destacar elementos de baixo nível como bordas e pontas, as camadas mais profundas realçam elementos de nível mais alto, destacando partes completas, curvas e estilos. A extração de características hierárquicas é um dos aspectos mais poderosos das redes convolucionais profundas. Além das camadas de agregação, as redes convolucionais contém camadas de função de ativação.

A exemplo das redes neurais perceptron comuns, as redes CNN utilizam se de camadas com funções de ativação. Estas funções apresentam a característica da não-linearidade. Sendo as funções de ativação não-linear, permite à rede encontrar bordas de separabilidade entre classes mesmo que estas bordas apresentam uma topologia complexa, algo que uma função de ativação linear não seria capaz de lidar com eficácia. A função de ativação mais simples é a ReLU (*Rectifier Linear Unit*). Esta função apresenta comportamento linear para valores positivos, mas retificação dos valores negativos induzindo não linearidade. Esta função é comumente utilizada por causa da simplicidade e eficiência durante as fases de avaliação da rede. Apesar

de padrão, há outras funções de ativações que são utilizadas em diferentes tarefas [6].

Em geral, as redes convolucionais são excelentes para extração de características relevantes. A partir destas características, técnicas são usadas para fazer a tarefa. Em classificação, pode se utilizar as features extraídas pela CNN como entrada de um classificador SVM e utilizá-la como classificação. Contudo, SVM não é diferenciável, o que implica no treinamento da CNN separado da SVM.

Para resolver esta dificuldade, as redes convolucionais de classificação usam uma camada *perceptron* em sua saída. A camada perception é diferenciável, permitindo a aprendizagem de ponta a ponta por meio do algoritmo de retropropagação. A Figura 2 apresenta uma arquitetura típica da CNN.

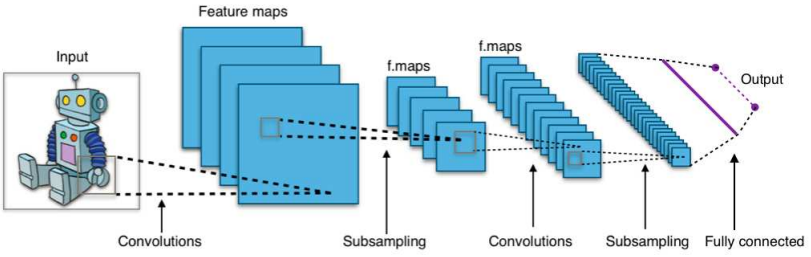

Figura 2: Arquitetura típica de uma rede neural convolucional. Cada camada extrai características que são utilizadas pela camada seguinte como entrada. Por Aphex34 - Own work, CC BY-SA 4.0,

4.2 Redes Convolucionais e Recorrentes no Reconhecimento de Palavras

As redes convolucionais se apresentam eficazes na classificação e reconhecimento de caracteres individuais, contudo, a tarefa de reconhecimento de caracteres se estende também ao reconhecimento de palavras. Neste aspecto, as redes convolucionais necessitam ser adaptadas para capturar não somente o caractere em si, mas também a sequência de caracteres em sua vizinhança. Esta adaptação pode ser entendida considerando a sequência de caracteres como uma série temporal, pois uma palavra é formada pela composição sequencial e ordenada de caracteres.

Em geral, o processamento de dados temporais com técnicas de aprendizagem de máquina pode ser efetuado por redes neurais recorrentes. As redes neurais recorrentes recebem a sequência de dados de maneira recursiva, extraindo não somente a característica do dado correto mas também a relação com os dados apresentados anteriormente na sequência.

As redes neurais recorrentes, porém, apresentam dificuldade de convergência durante o treinamento com retropropagação devido à sua natureza recorrente. Durante os ciclos de treinamento por vezes informações sequenciais que são importantes para a tarefa são perdidas, prejudicando a qualidade da aprendizagem do modelo. Para mitigar estes efeitos da perda de informação por causa da recursividade, novas arquiteturas recorrentes são propostas. *Long Short Term Memory Network* (LSTM) [11] é uma rede recorrente que apresenta portas internas que chaveiam o fluxo de dados bloqueando as informações desnecessárias de passar para próxima interação da sequência e permitindo informações relevantes passarem para o próximo passo, além de fornecer um vetor latente que sumariza as relações temporais das sequências. A Figura 3 apresenta o esquema da LSTM.

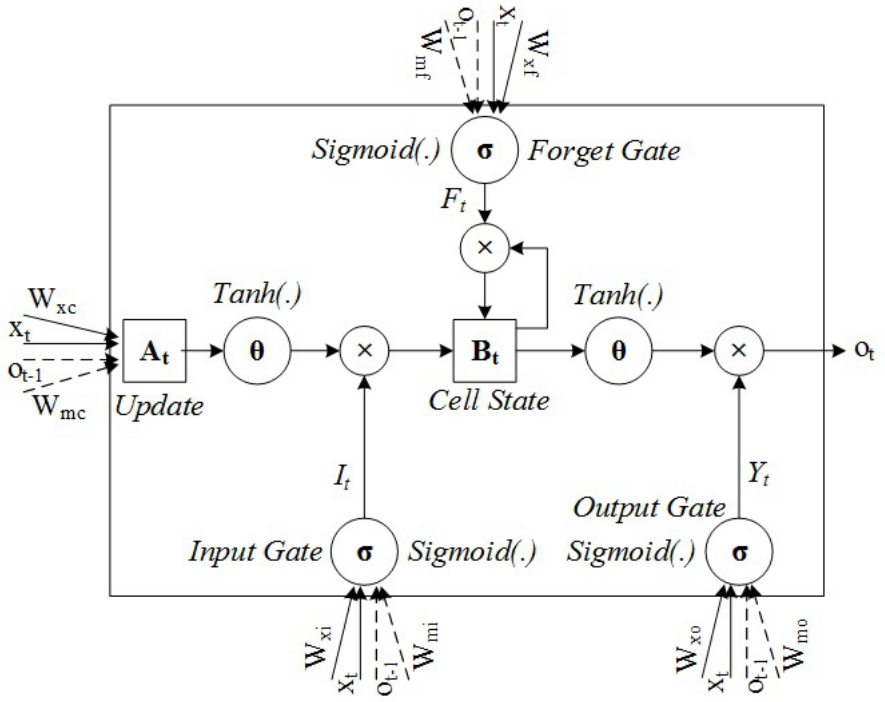

Figura 3: Célula básica *Long Short Term Memory* - LSTM. As diversas portas de entrada controlam quais características devem ser passadas para a próxima iteração e quais devem ser eliminadas. Por [30]

No contexto de reconhecimento de caracteres e palavras, enquanto a rede CNN é utilizada para extração de features, esta é acoplada com uma rede LSTM que classifica sequencialmente os caracteres observados pela CNN. Assim, uma imagem de entrada com uma palavra é convertida para um vetor de características, enquanto a rede LSTM decodifica este em caracteres formando sequência da palavra.

A natureza dos dados de entrada (imagem) e de saída (sequência) são diferentes, portanto elas não apresentam alinhamento natural de sequência. Esta dificuldade pode ser resolvida utilizando a Classificação Temporal Caneccionista (CTC) [9]. Esta técnica pode ser usada na função de erro durante o treinamento. O CTC permite que a saída da LSTM seja comparada com o rótulo de treinamento mesmo que este tenha uma sequência diferente e não alinhada em relação à saída da LSTM.

4.3 Detecção de Textos

Além da classificação de caracteres e da decodificação de sequências de palavras, o processamento de caracteres em imagens apresenta o desafio de detectar as regiões onde os textos são posicionados. Esta detecção, envolve não somente caracteres individuais, mas composições hierárquicas mais altas como palavras, parágrafos e páginas etc. A tarefa de detecção destes artefatos têm se aproveitado dos avanços de aprendizagem profunda relacionada advinda das redes convolucionais de detecção.

A tarefa de detecção é formada de sub tarefas auxiliares de regressão e classificação. A tarefa de regressão consiste em estimar a posição dos objetos de interesse, enquanto a classificação identifica a classe do objeto (caractere, palavra, parágrafo, tabela etc.) bem como se o objeto é ou não de interesse.

Duas arquiteturas são mais utilizadas na detecção de objetos: Detector de ponta-a-ponta e detector com rede auxiliar de região.

A rede detectora de ponta a ponta apresenta arquitetura similar à YOLO [20, 21], em que a rede CNN é utilizada para extrair características e estimar a posição e relevância de âncoras de detecção de objetos para cada região.

As redes com detectores de região auxiliares utilizam se de redes *Region Proposal Network* (RPN) que efetuam a regressão das regiões a partir de agregações (*pooling*) das regiões de interesse [23]. A agregação permite não somente a regressão, mas também classificação do tipo de objeto detectado. A Figura 4 apresenta a arquitetura básica da rede YOLO [21, 20] e enquanto a Figura 5 ilustra a estrutura básica da *Faster RCNN* [23].

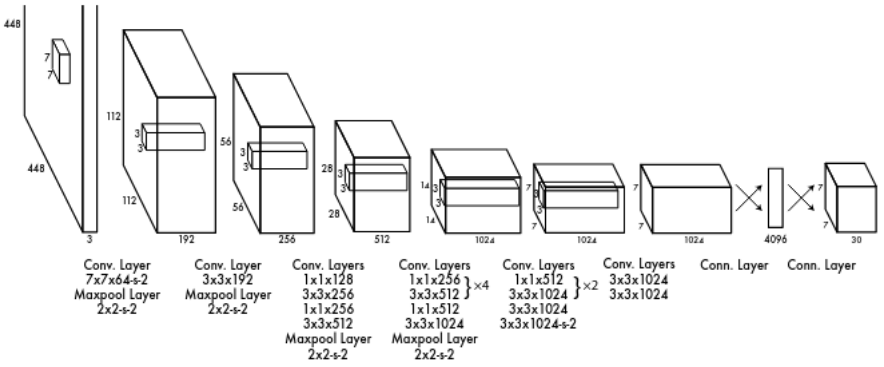

Figura 4: A rede YOLO possui 24 camadas convolucionais seguidas por 2 camadas totalmente conectadas. Por [20]

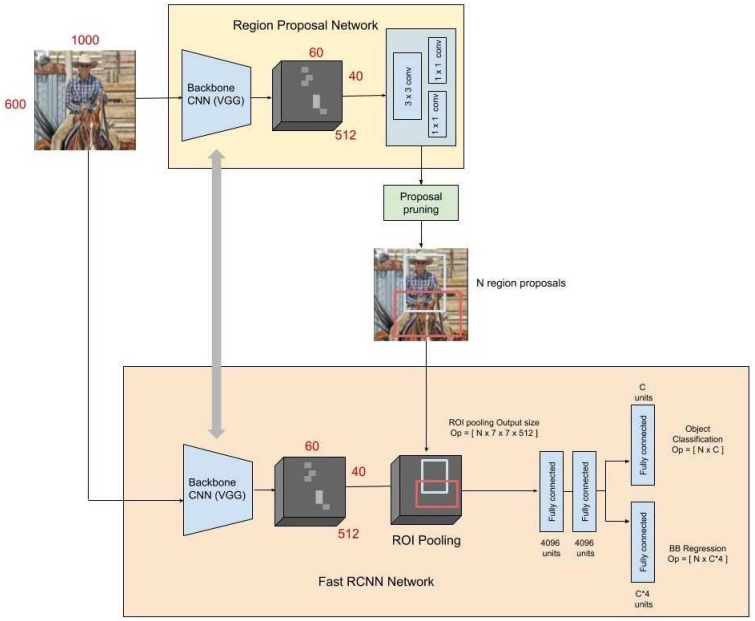

Figura 5: *Region Proposal Network* (RPN) é a parte principal da *Faster R-CNN*. A RPN recebe como entrada as características obtidas da imagem e estima a posição classe dos objetos detectados.

4.4 Representação Vetorial das Palavras

Técnicas de OCR que detectam palavras a as extraem de imagem por si só não fornece o quadro completo quando se necessita extrair conteúdos textuais de imagens, em especial em casos que a tarefa é processar informações de cardápios, pois, além do conteúdo extraído, necessita-se também de sua classificação de entidade. Tal classificação permite diferenciar título de menu, grupos e descrição dos pratos oferecidos. Para tal tarefa, faz-se necessário técnicas de aprendizagem de máquina que possibilitem representar as palavras de maneira vetorial de forma que representações vetoriais de palavras com significado semântico, ficam posicionadas a outras representações vetoriais que têm significado semelhante.

De maneira simplificada, há duas formas principais de se obter as representações vetoriais das palavras: *Skip-gram* e *Continuous Bag of Word* [17]. Dado um texto, em uma rede *Skip-gram*, uma palavra é selecionada como entrada de um modelo de rede neural que predizer quais palavras estão na vizinhança da palavra escolhida como entrada. Este tipo de treinamento força a rede neural escolher melhores representações vetoriais dependendo do contexto em que as palavras são escolhidas.

A técnica *Continuous Bag of Word*, por outro lado, recebe como entrada o conjunto de palavras vizinhas e tenta prever qual palavra se adequa mais neste contexto. De maneira diametral ao *Skip-gram*, na técnica de *Continuous Bag of Word*, o contexto é fornecido para prever qual melhor palavra se adequa a ele. Tal método força a rede a buscar representações semânticas tanto para as palavras do contexto como as escolhidas durante cada passo de treinamento. Após a aprendizagem, as representações podem ser utilizadas como entrada em modelos neurais para classificação e busca semântica. A Figura 6 ilustra as técnicas de representação vetorial de palavras.

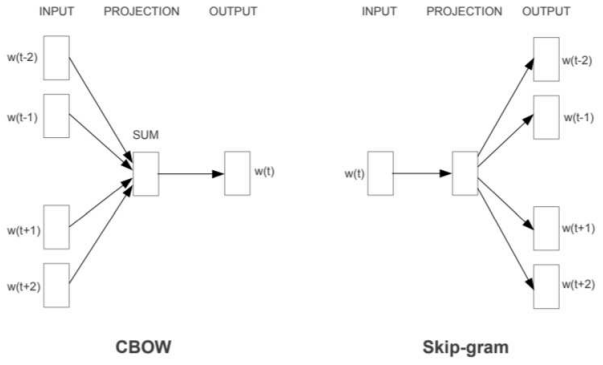

Figura 6: Esquemas de representação de palavras em formato vetorial.

4.5 Redes Convolutional em Grafos

As redes convolucionais em grafos (GCNs) [13] são uma extensão das redes neurais convolucionais (CNNs) para dados que possuem uma estrutura de grafo. Enquanto as CNNs são eficazes para dados estruturados em grades regulares, como imagens e vídeos, as GCNs são projetadas para lidar com dados que possuem relações complexas e irregulares, como redes sociais, moléculas químicas e sistemas de recomendação.

A ideia central das GCNs é generalizar a operação de convolução para grafos, permitindo a agregação de informações dos nós vizinhos para cada nó no grafo. Isso é feito utilizando uma matriz de adjacência que representa as conexões entre os nós e uma matriz de características que representa as características dos nós.

O processo de convolução em grafos pode ser descrito em três etapas principais:

1. **Agregação de Vizinhos**: Para cada nó no grafo, as características dos nós vizinhos são agregadas utilizando uma função de agregação, como a soma, média ou máximo. Essa etapa permite que cada nó reúna informações de seus vizinhos.
2. **Combinação de Características**: As características agregadas dos vizinhos são combinadas com as características do próprio nó utilizando uma função de combinação, como uma transformação linear seguida por uma função de ativação não linear (por exemplo, ReLU).
3. **Atualização das Características**: As características combinadas são atualizadas para cada nó, resultando em novas representações de características que incorporam informações dos vizinhos.

A fórmula matemática para uma camada de convolução em grafos pode ser expressa como:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)}\right)$$

onde: - $H^{(l)}$ é a matriz de características na camada l, - $\tilde{A}$ é a matriz de adjacência com auto-conexões adicionadas, - $\tilde{D}$ é a matriz diagonal de graus dos nós, - $W^{(l)}$ é a matriz de pesos treináveis na camada l, - σ é a função de ativação não linear (por exemplo, ReLU).

As GCNs têm sido aplicadas com sucesso em várias tarefas, como classificação de nós, previsão de links e segmentação de grafos. Elas são particularmente eficazes em cenários onde a estrutura dos dados é complexa e não pode ser representada adequadamente por métodos tradicionais de aprendizado de máquina.

4.6 Clusterização

Frequentemente, os dados fornecidos não são corretamente tratados, muito menos rotulados. Nestas situações, utiliza-se de técnicas de aprendizagem não supervisionada visando extrair informações e padrões de um grande contingente de dados não rotulados. A clusterização (ou agrupamento) consiste em um conjunto de técnicas que se utilizam das semelhanças/distância entre features na segmentação e formação dos grupos. Tais técnicas são aplicadas para categorizar características que apresentam padrões definidos, evitando o custo de rotular. A clusterização é aplicada na descoberta e segmentação de dados a partir de comportamento, e outras características de clientes de uma determinada empresa. Assim, a partir de dados não rotulados dos clientes, é possível descobrir comportamentos e oportunidades para ações de marketing, retenção e ações promocionais.

Apesar da existência de várias técnicas de clusterização não supervisionada, a técnica k-means certamente é uma das mais utilizadas. Este fato se deve à sua facilidade de entendimento e implementação, bem como apresenta bons resultados na maioria dos estudos de caso.

O algoritmo *K-means* [34] consiste em escolher K pontos no espaço de característica que se posicione no centróides dos clusters de dados. Isso se dá por meio de dois passos principais que envolvem minimização de distância e atualização da posição dos centróides.

Inicialmente, os centróides são posicionados aleatoriamente ao longo do espaço de características. No segundo passo, os pontos da base de dados são associados ao centróides mais próximo do ponto. Finalmente, os centróides têm sua posição atualizada a partir da média das posições de todos os pontos que pertencem ao cluster em questão. Estes passos são repetidos iterativamente até que não haja mudanças efetivas das posições dos centróides alcançando o critério de convergência.

Apesar de sua simplicidade, o algoritmo *K-means* apresenta resultados satisfatórios na segmentação de vários tipos de dados, tornando-se um método basilar forte.

4.7 Redução de Dimensionalidade

A redução de dimensionalidade é uma técnica essencial em aprendizado de máquina e análise de dados, que visa simplificar conjuntos de dados complexos e de alta dimensionalidade, preservando ao máximo as informações relevantes. Em muitos casos, os dados originais podem conter redundâncias ou ruídos que dificultam a análise e a visualização. A redução de dimensionalidade ajuda a mitigar esses problemas, facilitando a visualização, melhorando a eficiência computacional e, muitas vezes, aumentando a performance de algoritmos de aprendizado de máquina.

Existem várias técnicas para redução de dimensionalidade, sendo as mais comuns a Análise de Componentes Principais (PCA) [7], o t-SNE (t-Distributed Stochastic Neighbor Embedding) [28] e o Isomap [26].

A PCA transforma os dados originais em um novo conjunto de variáveis não correlacionadas, chamadas componentes principais, ordenadas de forma que as primeiras componentes retêm a maior parte da variância dos dados. O t-SNE é particularmente eficaz para a visualização de dados em duas ou três dimensões, preservando as relações de proximidade entre os pontos de dados. As etapas do PCA são as seguintes:

1. **Centralização dos Dados**: Subtrair a média de cada variável dos dados originais para centralizá-los em torno da origem. Isso resulta em um conjunto de dados com média zero.
2. **Cálculo da Matriz de Covariância**: Calcular a matriz de covariância dos dados centralizados. A matriz de covariância captura as relações de variância e covariância entre as variáveis.
3. **Cálculo dos Autovalores e Autovetores**: Calcular os autovalores e autovetores da matriz de covariância. Os autovetores representam as direções das componentes principais, enquanto os autovalores indicam a quantidade de variância explicada por cada componente principal.
4. **Ordenação dos Autovalores e Seleção dos Componentes Principais**: Ordenar os autovalores em ordem decrescente e selecionar os autovetores correspondentes aos maiores autovalores. Esses autovetores formam a base das componentes principais.
5. **Projeção dos Dados**: Projetar os dados originais nos autovetores selecionados para obter as novas variáveis (componentes principais). As primeiras componentes principais retêm a maior parte da variância dos dados originais.

Essas etapas permitem transformar os dados originais em um espaço de menor dimensão, preservando ao máximo a variância e as informações relevantes.

Já o Isomap (Isometric Mapping) é uma técnica de redução de dimensionalidade que busca preservar as distâncias geodésicas entre os pontos de dados em um espaço de menor dimensão. A ideia central do Isomap é capturar a estrutura não linear dos dados, mapeando-os para um espaço de menor dimensão onde as relações de proximidade são mantidas.

O processo do Isomap pode ser dividido em três etapas principais:

1. **Construção do grafo de vizinhança**: Inicialmente, um grafo de vizinhança

é construído conectando cada ponto de dados aos seus vizinhos mais próximos. Isso pode ser feito utilizando a distância euclidiana ou outras métricas de distância. O número de vizinhos pode ser definido por um parâmetro k ou por um raio ϵ.

2. **Cálculo das distâncias geodésicas**: Em seguida, as distâncias geodésicas entre todos os pares de pontos são calculadas. As distâncias geodésicas são as menores distâncias ao longo do grafo de vizinhança, que podem ser encontradas utilizando algoritmos de caminho mínimo, como o algoritmo de Dijkstra ou o algoritmo de Floyd-Warshall.

3. **Redução de dimensionalidade**: Finalmente, uma técnica de redução de dimensionalidade, como a Decomposição em Valores Singulares (SVD), é aplicada às distâncias geodésicas para mapear os dados para um espaço de menor dimensão. O objetivo é encontrar uma representação de menor dimensão que preserve as distâncias geodésicas originais.

O Isomap é particularmente útil em situações onde os dados possuem uma estrutura não linear complexa, como em problemas de reconhecimento de padrões, visualização de dados e análise de imagens. Ao preservar as distâncias geodésicas, o Isomap consegue capturar a estrutura intrínseca dos dados, facilitando a interpretação e a análise em um espaço de menor dimensão.

O t-Distributed Stochastic Neighbor Embedding (t-SNE) é uma técnica de redução de dimensionalidade particularmente eficaz para a visualização de dados de alta dimensão em duas ou três dimensões. Desenvolvido por Laurens van der Maaten e Geoffrey Hinton em 2008, o t-SNE é amplamente utilizado para explorar e visualizar padrões em dados complexos, como imagens, textos e dados genômicos.

O t-SNE funciona ao converter as distâncias euclidianas entre pontos de dados em probabilidades que representam similaridades. Em seguida, ele tenta minimizar a divergência de Kullback-Leibler entre essas probabilidades no espaço de alta dimensão e no espaço de baixa dimensão. O processo pode ser dividido em três etapas principais:

1. **Cálculo das Similaridades no Espaço de Alta Dimensão**: Para cada par de pontos de dados, o t-SNE calcula a probabilidade condicional de que um ponto escolheria o outro como seu vizinho, com base em uma distribuição gaussiana centrada no ponto. Isso resulta em uma matriz de similaridade simétrica.

2. **Cálculo das Similaridades no Espaço de Baixa Dimensão**: No espaço de baixa dimensão, o t-SNE calcula as probabilidades de similaridade de maneira semelhante, mas utilizando uma distribuição t de Student com um grau

de liberdade (distribuição t de Student de uma cauda longa). Isso ajuda a capturar melhor as relações de proximidade entre os pontos.

3. **Minimização da Divergência de Kullback-Leibler**: O t-SNE ajusta as posições dos pontos no espaço de baixa dimensão para minimizar a divergência de Kullback-Leibler entre as distribuições de similaridade no espaço de alta e baixa dimensão. Isso é feito utilizando um algoritmo de gradiente descendente.

Uma das principais vantagens do t-SNE é sua capacidade de preservar tanto as relações locais quanto as globais entre os pontos de dados, o que o torna ideal para a visualização de clusters e padrões em dados complexos. No entanto, o t-SNE pode ser computacionalmente intensivo e requer a escolha cuidadosa de hiperparâmetros, como a perplexidade, para obter resultados ótimos.

5 Resultados e Discussões

A tarefa de obtenção de informações dos cardápios digitalizados exige uma estruturação hierárquica como requisito fundamental. Assim, o desafio é receber como entrada uma imagem de cardápio e como saída uma hierarquia contemplando minimamente o grupo de menu, item de menu, preço e descrição.

Um grupo de menu consiste em um conjunto de pratos que compartilham características, diferenciando em sua variação. Um exemplo de grupo de menu é Pizza, em que há várias variedades de pratos pizza com diferentes temperos e recheios, etc. Já os itens de menus referem-se aos pratos fornecidos pelo estabelecimento. Com poucas exceções, o item de menu é a menor unidade relacionada com o prato. Além da nomeação dos pratos especificada pelo item de menu, é comum haver uma descrição de tais itens, indicando os ingredientes envolvidos em sua formação. Finalmente, o preço é uma informação fundamental. A Figura 7 exemplifica a estrutura hierárquica dos menus.

Figura 7: Estrutura dos Cardápios de Menus. A região com borda vermelha representa o grupo de menu. Em azul, os itens de menus. Em verde, as descrições. Em preto, o preço.

A estruturação da informação na forma hierárquica se deve não somente à necessidade de negócio, mas também à estrutura geométrica que os textos são posicionados. Como ilustrado na Figura acima, o grupo de menu é frequentemente posicionado como um título, com cabeçalho destacado, seja por alinhamento, tamanho de fonte, estilo e cores. Os itens de menus, por outro lado, são posicionados frequentemente como lista vertical abaixo dos grupos de menus, pode apresentar destaque quanto ao estilo de sua fonte ou mesmo cor, contudo, sua dimensão é maior que os grupos de menus. As descrições dos pratos são caracterizadas por apresentar listagem de

ingredientes, conter mais palavras e formar pequenos parágrafos. Frequentemente, são encontrados logo abaixo dos itens de menus ou em seu lado direito. A fonte é destacadamente menor ou com estilo mais discreto, com estilo itálico ou normal, mas com tamanho de fonte reduzida. Os preços, por sua vez, são, na maioria das vezes, alinhados com o item de menu, se dispondo no lado direito, caracterizando, às vezes, uma formação tabular.

Dado a descrição, o desafio é associar os conteúdos detectados por ferramentas de OCR com a estrutura hierárquica. Para isso, duas possíveis abordagens podem ser utilizadas.

- Classificação dos Grupos de Menus/ Item de Menus.
- Classificação por aprendizagem de máquina supervisionada.
- Classificação a partir dos títulos e descrições dos itens.
- Detecção das regiões por meio das imagens mascaradas de itens.
- Classificação por aprendizagem de máquina não-supervisionada
- Clusterização de características de fonte.
- Pré-processamento de características de fonte.
- Identificação dos grupos de menus por heurística.

5.1 Classificação dos Grupos de Menus / Item de Menus

A primeira abordagem para solucionar o problema consiste em identificar quais dos textos detectados pela ferramenta de OCR são referentes a descrições de Grupos de Menus Itens de Menu. Para isso, foi derivado de uma base de dados de uma empresa parceira, um corpus contendo 956.475 entradas, sendo 64.995 entradas associadas a Grupo de menus, enquanto 891.480 são entradas rotuladas como Item de Menu.

Além da base de treinamento, uma base de validação foi criada. Esta é composta de 409.915 entradas, correspondendo a 42% do número de entradas do conjunto de treinamento. O modelo escolhido para a tarefa de classificação foi o *Fasttext*, porque apresenta excelente *baseline* e eficiência de treinamento com tempo reduzido. O treinamento se deu utilizando os valores de hiperparâmetros padrão do *Fasttext*, outras configurações de hiperparâmetros foram executadas verificando-se que a configuração padrão apresenta os melhores resultados.

Durante o treinamento, o modelo alcançou 97% de precisão e 97.3% de revocação. O conjunto de validação apresentou resultados semelhantes, com 96% de precisão e

96, 4% de revocação. Estes resultados são considerados bons considerando o conjunto de validação.

5.2 Detecção das regiões por meio das imagens mascaradas de itens.

Enquanto na abordagem anterior visava detectar os grupos de menus por meio de classificação dos textos detectados, a abordagem de detecção por imagens mascaradas consiste em utilizar uma rede detectora de objetos com a Faster RCNN para detectar grupos de texto que possam fazer parte do grupo de menus.

Devido à diversidade artística dos cardápios, as imagens apresentam muitos elementos visuais que podem confundir o detector de objetos. Visando mitigar esta dificuldade, foi elaborada uma abordagem de mascaramento das imagens. O processo de mascaramento consiste em mascarar a imagem com as regiões de texto, deixando de fora as regiões de fundo.

Assim, a imagem de um dado cardápio tem seus textos detectados por um sistema de OCR. As máscaras são confeccionadas a partir das caixas de detecção (bounding boxes) dos textos. Os pixels fora da caixa de detecção dos textos são removidos, permanecendo somente as regiões de texto de interesse. Este tipo de pré-processamento elimina elementos desnecessários na detecção. A Figura 8 exemplifica uma imagem com mascaramento de fundo.

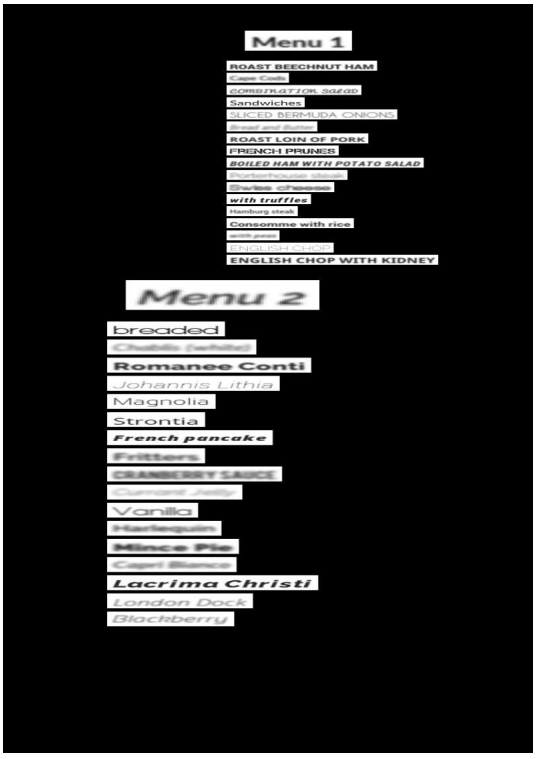

Figura 8: Exemplo de menu com mascaramento de fundo. Esta imagem é um exemplo de menu sintético gerado com grupos e itens de menus aleatórios.

O raciocínio por detrás do detector de objetos consiste em considerar um grupo de menus como um objeto a ser detectado. De fato, este raciocínio pode ser válido porque um grupo de menu tem partes e características essenciais, como título do grupo no topo ou na lateral, lista de itens de menus, descrições, preços etc.

Apesar de a utilização de detectores para este fim ser razoável, um problema comum é a falta de imagens de menus com as regiões de grupo de menu rotuladas. Para mitigar tal limitação, criamos um dataset de imagens sintéticas com grupos de menus associados e treinamos uma Faster RCNN por 100 épocas e taxa de aprendizagem 10^{-3} com otimizador Adam.

As imagens sintéticas foram criadas a partir de grupos de textos aleatórios posicionados 1-3 colunas por página. Após o treinamento, o melhor métrica dentro do domínio foi de cerca de 70% de Intersection over Union (IoU). Por outro lado, em imagens de teste fora do domínio (imagens de menus não sintéticos) a métrica cai para cerca de 48% de IoU.

Apesar da concepção da arquitetura ter valor prático, a falta de um conjunto de

dados expressivo anotado com grupos de menus reais prejudicou a generalização do modelo e seu uso prático. Esta barreira poderá ser passada se houver um dataset devidamente anotado e com volume suficiente.

5.3 Categorização por aprendizagem não-supervisionada

As abordagens desenvolvidas até agora eram baseadas em técnicas de aprendizagem de máquina supervisionada, que dependiam de uma base de dados rotulada e de qualidade. Em geral este tipo de dado é custoso de se obter porque necessita de anotação em massa de especialistas no tema. Visando mitigar o efeito da falta do dataset, foram executados experimentos com métodos não-supervisionados.

Uma análise das imagens dos cardápios percebe se que o título do grupo de menus apresenta uma fonte maior que os itens de menus e descrições. Com base nesta suposição, o tamanho da fonte pode ser uma característica marcante que possa ser utilizada na identificação do grupo de menu/item de menu.

Inferir o tamanho da fonte em pontos a partir de imagem é difícil porque a detecção de OCR dá a altura da caixa de detecção dos textos em pixels e a conversão de pontos para pixel depende da densidade de pontos em que o cardápio foi digitalizado ou impresso. Assim, fica-se restrito à altura em pixels. para evitar variação desnecessária, normalizamos a altura de todos os textos em relação à altura da página do menu, ficando com unidades dimensionais.

Levando em conta a suposição que os grupos de menus apresentam textos com mesma altura entre si e altura diferente dos itens de menus, aplicamos o algoritmo *k-means* para particionar.

Tipicamente, o algoritmo é aplicado em toda base de dados, contudo, no contexto atual, deseja-se fazer a separação entre textos com diferentes alturas (correspondentes a fontes de tamanhos diferentes) dentro do documento em análise.

Assim, ao se obter os dados de OCR, estes são utilizados para treinar o *k-means* e fazer a avaliação. Desta forma, cada linha de texto é associada a um cluster. Espera-se que textos com mesmo tamanho de fonte sejam categorizados no mesmo cluster.

Como uma abordagem não-supervisionada, a clusterização dos títulos apresenta a vantagem de não serem supervisionados. Ela pode ser utilizada como ferramenta auxiliar do sistema de classificação por texto descrito na seção anterior, de tal forma que um grupo de menu que não foi corretamente como tal pode ser corrigido, verificando se o cluster que ele está categorizado pertence ao grupo e menu ou não.

5.3.1 Pré-processamento de características de fonte.

Após executar a análise dos OCR de cerca de pouco mais de 9.000 imagens, percebeu-se que a qualidade de detecção do texto é de extrema importância para clusterizar corretamente os títulos de acordo com o tamanho da fonte. Foi detectado que há uma ocorrência expressiva de textos que a área detectada não está ajustada ao caractere do texto. Este tipo de erro de detecção quebra a presunção de que textos tenham o mesmo tamanho de fonte, apresentem mesma altura de detecção. Como consequência, não há garantias fortes que os títulos referentes aos grupos de menus, por exemplo, pertencerão ao mesmo cluster que outros títulos de mesma classe. Infelizmente este problema reduz a eficácia do algoritmo em auxiliar encontrar títulos de menus, como mostra a Figura 9.

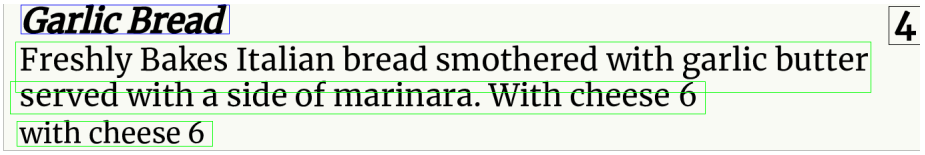

Figura 9: Por vezes, o detector de linhas detecta uma região com altura maior que a altura real da fonte causando dispersão e confundindo a clusterização. Vide o erro de deteção da descrição em verde.

A observação dos casos em que a região de detecção não se ajusta bem como tamanho dos caracteres dos textos, indicam que os erros advém de artefatos na imagem que confundem o detector. Elementos como ruído/partes da imagem de fundo, textos estilizados com caractere inicial com tamanho de fonte diferente do restante dos caracteres da palavra e textos sublinhados.

Para mitigar o efeito, desenvolveu-se um módulo de pré-processamento cuja finalidade é obter a altura mais ajustada possível de um texto detectado, fornecendo características mais robustas para detecção. O módulo é composto por alguns passos essenciais: Recorte da região de interesse, análise estatística e detecção de pontos de interesse.

Primeiramente, o módulo cria um recorte das imagens a partir das detecções do OCR. De cada palavra detectada, criamos uma imagem com o recorte da região. A imagem passa por um filtro gaussiano para reduzir os ruídos inerentes da imagem é convertida para escala de cinzas.

Após a conversão para escala de cinza, é necessário binarizar a imagem de forma que o texto fique com cor branca e o fundo com cor preta. Para isso, toma-se a média da imagem em escala de cinza. Se o valor for acima de 127, então o fundo é provavelmente branco e os caracteres pretos, então se inverte as intensidades dos

pixels. A binarização é feita por meio do algoritmo de Otsu.

O perfil do texto é obtido a partir de uma análise estatística. Basicamente, faz-se um histograma projetivo no eixo vertical. Assim dada uma imagem $I_{i,j}$ de largura W e altura H, a projeção é calculada pela seguinte equação:

$$P_i = \frac{1}{H} \sum_{j}^{H} I_{i,j}, \tag{1}$$

em que P_i representa o valor da projeção (histograma) da i-ésima linha da imagem $I_{i,j}$.

O histograma informa os pontos em que dá-se o início e o final ao longo do eixo vertical dos caracteres. Esta projeção permite obter a altura mais ajustada com o texto fornecendo melhores características para a clusterização. A Figura ilustra o fluxo de processamento.

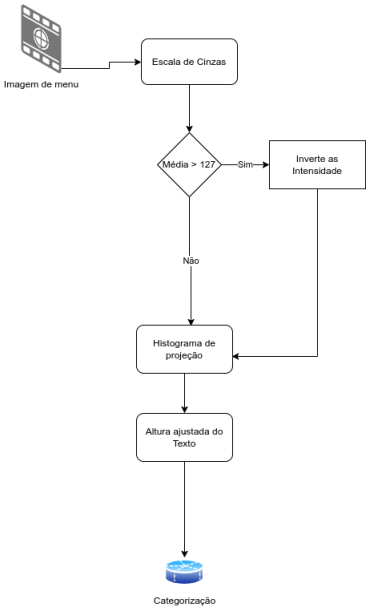

Figura 10: Fluxograma da abordagem de obtenção da altura dos textos.

5.3.2 Identificação dos grupos de menus por heurística.

Até então, com exceção do detector de grupo de menus utilizando Faster R-CNN, os outros módulos e abordagens visavam identificar os textos que eram potencialmente referentes ao grupo de menus. Além do desafio de encontrar o título do grupo de menus, há o desafio de encontrar a região, contendo itens de menus que pertencem

a este grupo. Tal desafio não é simples, contudo, com a correta classificação dos títulos dos grupos de menus, podemos utilizar de estratégias heurísticas que possam delimitar, em uma primeira aproximação, as regiões de cada grupo de menu.

O módulo utiliza-se da abordagem dividir para conquistar e se fundamenta no fato em que a maioria dos títulos de grupos de menus estão posicionados no topo enquanto os itens de menus associados a ele fica logo abaixo alinhado de forma columnar. Assumindo esta suposição, a partir das posições dos títulos dos grupos de menus, a região é conquistada e estendida seguindo as seguintes regras: Uma dada região é estendida até alcançar outra região. A região é estendida primeiramente na direção direita e depois à esquerda. As regiões são ajustadas para abarcar todas as regiões de itens de menus. Para seguir as regras supracitadas, o módulo se utiliza de algoritmos geométricos de detecção de sobreposição e colisão de regiões com extensão dinâmica. A Figura 11 ilustra o resultado final da detecção.

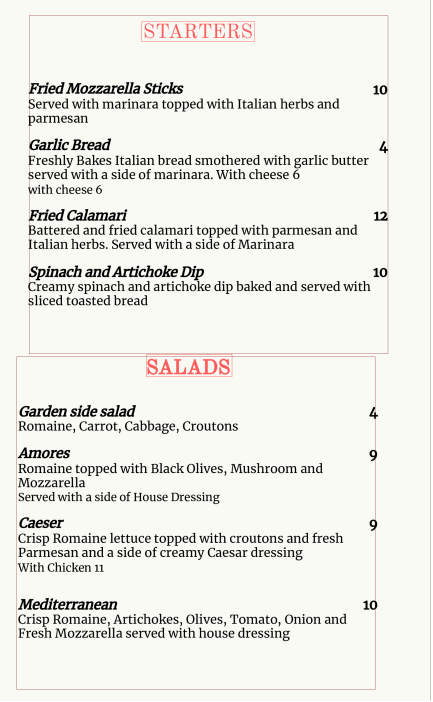

Figura 11: Os títulos de menus, em vermelho, são utilizados como ponto inicial para o algoritmo heurístico de detecção de grupos de menus.

5.4 Análise de Embeddings de Conteúdo de Menu por Utilizando Redes Convolucionais em Grafos

Este estudo teve como objetivo analisar os embeddings de nós obtidos a partir de uma Rede Neural Convolucional em Grafos (GCN), treinada por meio de aprendizado supervisionado em um conjunto de dados de grafos de menus. O objetivo principal foi avaliar a capacidade da GCN em classificar nós que representam itens de menu (Item de Menu) e categorias de menu (Grupo de Menu), e investigar a influência da estrutura do grafo nos embeddings resultantes.

O conjunto de dados era composto por 84 grafos, representando categorias (grupo de menus) e itens de menu. A arquitetura da GCN consiste em uma camada de entrada, uma camada GCN intermediária e uma camada de saída MLP (Perceptron Multicamadas). A camada GCN intermediária foi projetada para extrair os embeddings dos nós, que foram posteriormente utilizados para classificação pela MLP (multilayer perceptron). O treinamento da rede foi realizado ao longo de 50 épocas, utilizando uma taxa de aprendizado de 10^{-3} e um tamanho de lote de um grafo por iteração. O objetivo do treinamento foi minimizar a função de perda de entropia cruzada.

As matrizes de adjacência dos grafos foram construídas ponderando as arestas usando a similaridade de cosseno entre os embeddings de texto dos nós. Para as características iniciais dos nós, foram gerados embeddings usando o modelo CLIP, resultando em uma representação de dimensionalidade 512. Para avaliar a importância da informação estrutural, foram utilizados dois métodos de construção dos grafos: grafos informativos, nos quais as arestas na matriz de adjacência conectavam explicitamente os itens de menu às suas categorias (grupos de menus) correspondentes, e grafos não informativos, que formavam um grafo completo, desconsiderando a relação entre itens de menu e categorias.

Os resultados da classificação mostraram a eficácia da GCN em ambas as abordagens de estrutura de grafo. Nos grafos informativos, a classificação dos nós atingiu uma acurácia de 98,1% no conjunto de validação. Este alto desempenho indica que a GCN conseguiu utilizar a estrutura relacional explícita para aprender embeddings significativos. Em comparação, a acurácia da classificação com a estrutura de grafos não informativos foi de 93% no conjunto de validação. Apesar de ainda ser um bom resultado, essa pontuação indica que a ausência de informação estrutural explícita torna a tarefa mais desafiadora.

Uma análise detalhada dos embeddings revelou que a principal fonte de confusão durante a classificação ocorreu entre as categorias de menu e itens de menu com nomes semelhantes, como “Frangos” e “Frango”. Essa confusão é esperada devido à semântica similar, que resulta em embeddings mais próximos entre esses nós.

Os resultados deste experimento demonstram a capacidade da GCN em capturar efetivamente as características dos nós em grafos de menus. A alta acurácia obtida com a estrutura de grafos informativos sugere que a relação entre itens de menu e suas categorias correspondentes melhora a qualidade dos embeddings. A acurácia ligeiramente inferior obtida nos grafos não informativos destaca que aprender embeddings dos nós apenas pelo conteúdo é uma tarefa mais desafiadora. A confusão entre nomes semelhantes indica que, mesmo com o contexto do grafo, a similaridade dos nós precisa ser considerada, usando, por exemplo, informações sobre a semântica dos nós para discriminar termos semelhantes.

Os resultados dos experimentos com a Rede Neural Convolucional em Grafos (GCN) indicaram uma diferença mínima no desempenho da rede ao utilizar camadas intermediárias com 64 ou 128 neurônios. Este achado sugere que, para a tarefa em questão, o aumento da capacidade da camada intermediária não se traduziu em ganhos significativos no desempenho da classificação, indicando que talvez uma representação de menor dimensão seja suficiente para capturar as nuances dos dados. A análise aprofundada dos embeddings, obtidos através da camada GCN, empregando as técnicas de redução de dimensionalidade Isomap, PCA e t-SNE, revelou como a estrutura da variedade dos dados é capturada e como a informação é preservada, principalmente na variedade dos grafos completos, ou não informativos.

O Isomap, em particular, demonstrou a capacidade de capturar a estrutura da variedade dos embeddings, preservando as relações locais entre os dados. Especificamente, ao utilizar embeddings de dimensão 32 em grafos não informativos, o Isomap revelou a formação de três clusters alongados, com os itens de menu posicionados nas extremidades de cada um. Curiosamente, esses clusters apresentaram correlação com o tipo de restaurante, indicando que o espaço de embeddings está capturando características semânticas mais amplas relacionadas ao tema geral do menu. Ao elevar a dimensionalidade para 64, a projeção indicou a presença de um número maior de clusters, também alongados e formando curvas, com alguns clusters se mostrando mais isotrópicos. Já o aumento para 128 manteve as características da representação com 64 dimensões, mas com os itens de menu agrupados junto a grupos de menus semelhantes, espalhando-se ao longo da variedade e não apenas nas extremidades. Este resultado sugere que dimensionalidades maiores permitem que os nós se agrupem baseados em relações semânticas e contextuais do conteúdo e não apenas na relação com outros nós no grafo.

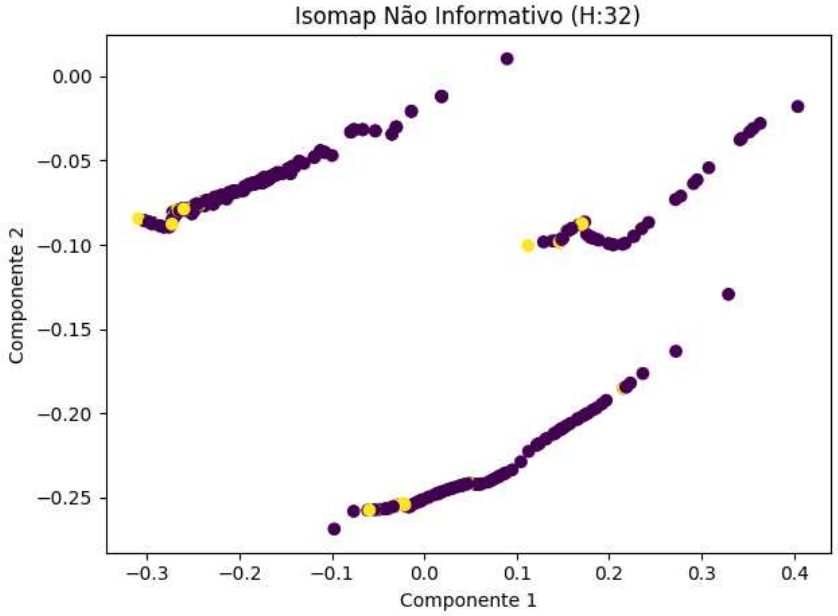

Figura 12: Projeção Isomap com dimensionalidade 32 para grafos não informativos. Em amarelo, os grupos de menus. Em azul, os itens de menus.

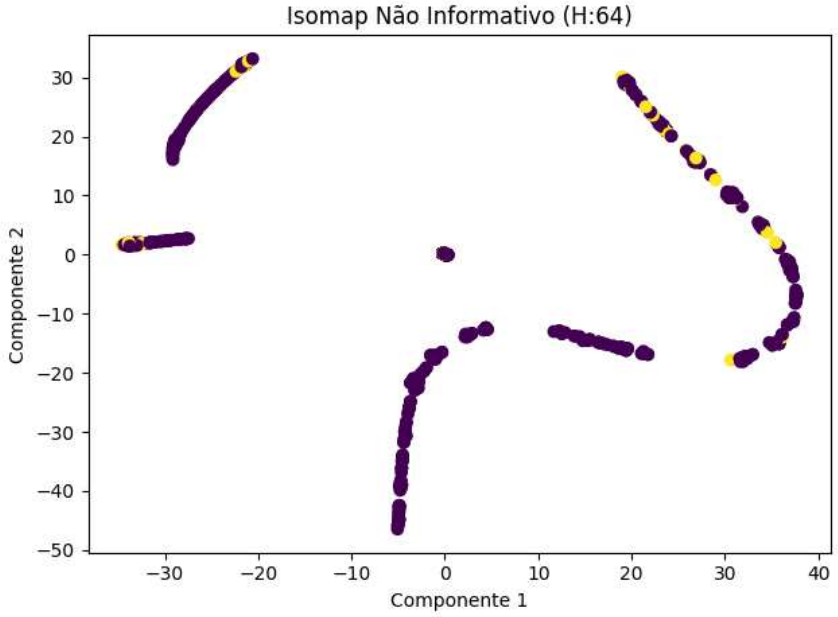

Figura 13: Projeção Isomap com dimensionalidade 64 para grafos não informativos. Em amarelo, os grupos de menus. Em azul, os itens de menus.

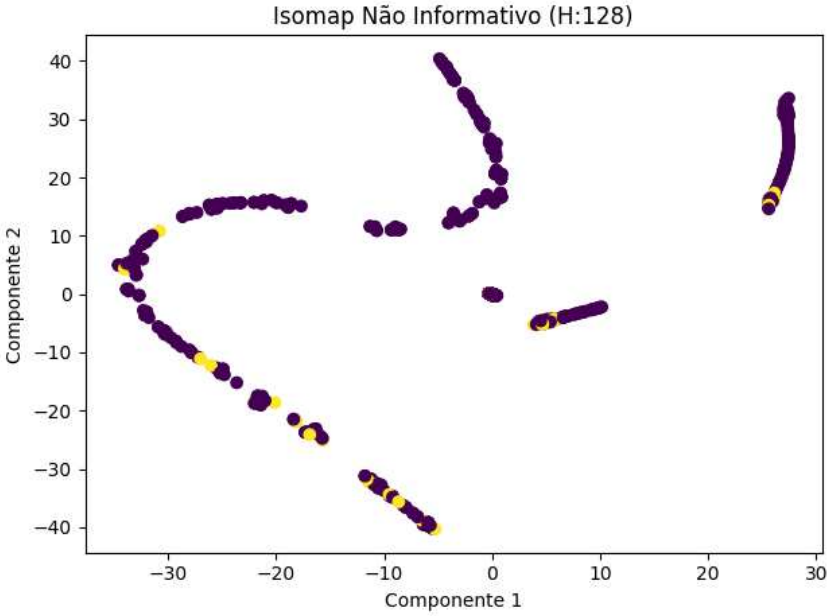

Figura 14: Projeção Isomap com dimensionalidade 128 para grafos não informativos. Em amarelo, os grupos de menus. Em azul, os itens de menus.

A projeção obtida através do PCA apresentou um comportamento similar com relação à distribuição dos Grupos de Menu e a formação de clusters, mas com uma projeção linear, o que era esperado para esta técnica. Ao utilizar dimensionalidade 128, o PCA mostrou a formação de duas hipersuperfícies claramente separadas, contudo, com clusters mais concentrados dentro dessas hipersuperfícies.

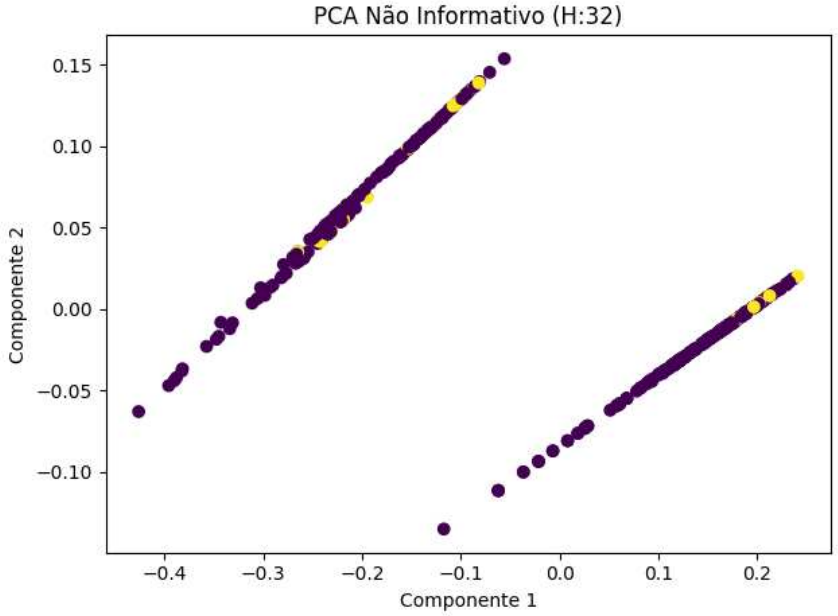

Figura 15: Projeção PCA com dimensionalidade 32 para grafos não informativos. Em amarelo, os grupos de menus. Em azul, os itens de menus.

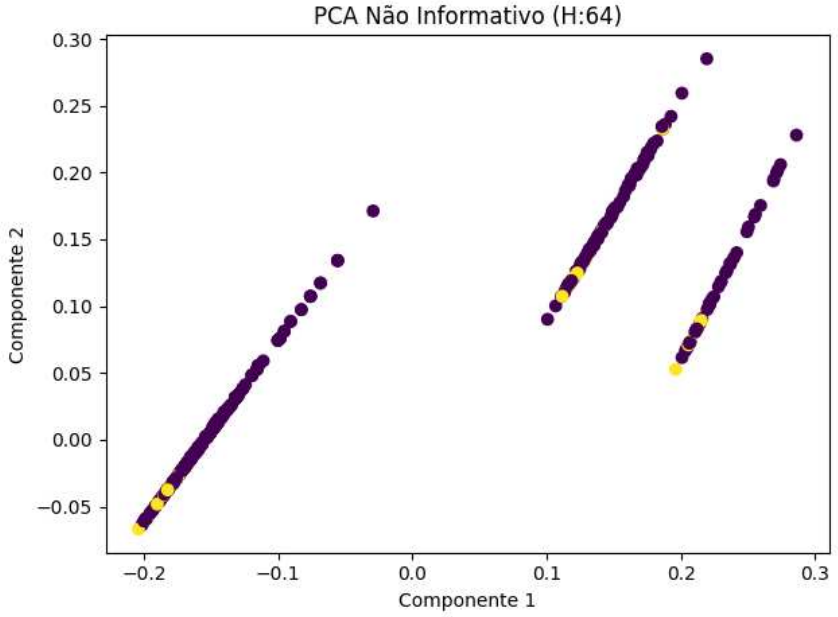

Figura 16: Projeção PCA com dimensionalidade 64 para grafos não informativos. Em amarelo, os grupos de menus. Em azul, os itens de menus.

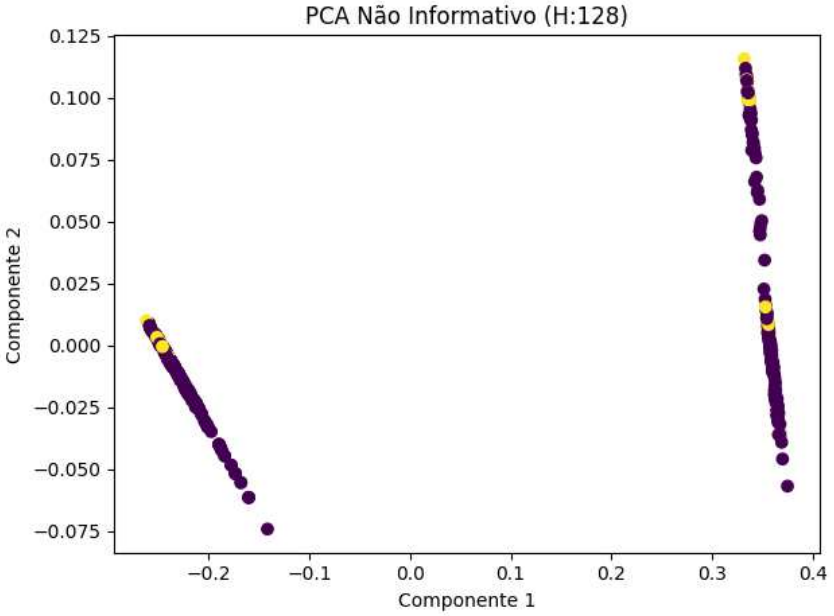

Figura 17: Projeção PCA com dimensionalidade 128 para grafos não informativos. Em amarelo, os grupos de menus. Em azul, os itens de menus.

A projeção t-SNE também apresentou resultados análogos aos anteriores para os grafos não informativos, delineando, entretanto, de forma mais clara, o perfil da variedade dos nós e a formação dos clusters. A análise sugere que, mesmo com o modelo sendo treinado de maneira supervisionada, os embeddings dos nós em grafos completos tendem a modelar a estrutura da variedade dos dados.

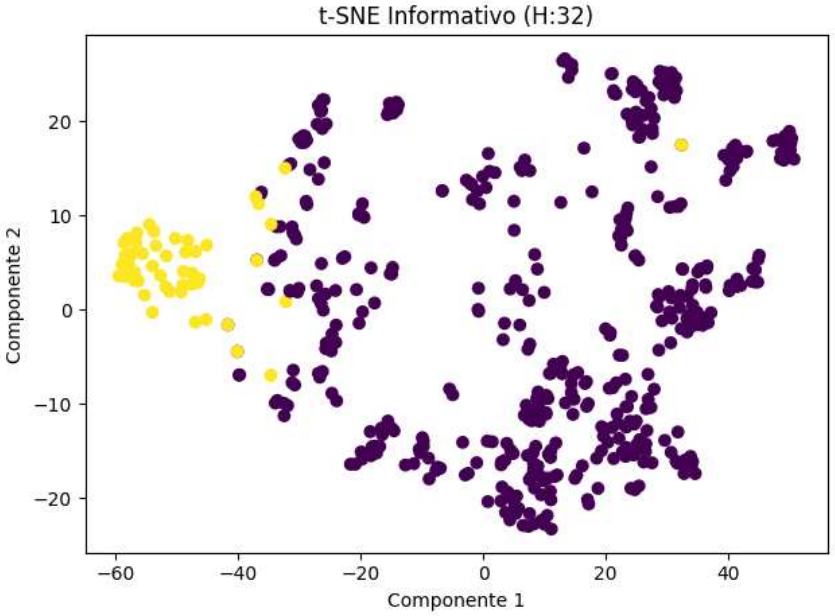

Figura 18: Projeção t-SNE com dimensionalidade 32 para grafos informativos. Em amarelo, os grupos de menus. Em azul, os itens de menus.

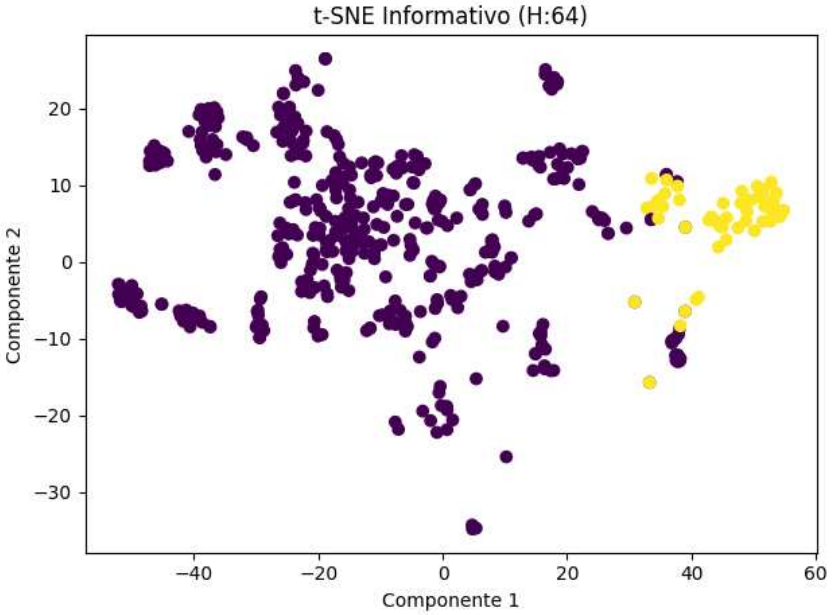

Figura 19: Projeção t-SNE com dimensionalidade 64 para grafos informativos. Em amarelo, os grupos de menus. Em azul, os itens de menus.

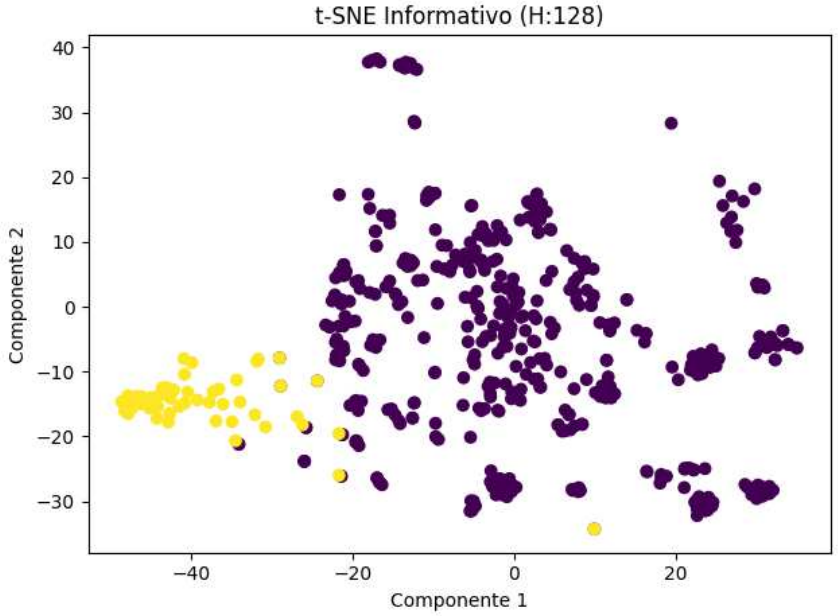

Figura 20: Projeção t-SNE com dimensionalidade 128 para grafos informativos. Em amarelo, os grupos de menus. Em azul, os itens de menus.

Em contraste, as projeções dos grafos informativos apresentaram uma estrutura diferente das dos não informativos. Ao utilizar embeddings de dimensão 32, a projeção Isomap apresentou o mesmo perfil de variedade observado na projeção t-SNE dos grafos não informativos. A redução dos graus de liberdade, neste caso, parece restringir os efeitos da informação e delineando uma variedade semelhante à obtida nos grafos não informativos. Apesar disso, a formação de clusters bem definidos de grupos de menus é marcante. Com uma dimensionalidade maior, a projeção mudou seu comportamento, enfatizando a formação de clusters isotrópicos de grupos de menus, indicando a distinção clara entre as classes. Como esperado, o grafo informativo favoreceu a clusterização dos grupos de menus, refletindo o efeito do aprendizado supervisionado. Com dimensionalidade 128, além da formação de aglomerados dos grupos de menus, independente de seu conteúdo semântico, destacou-se a separação entre classes.

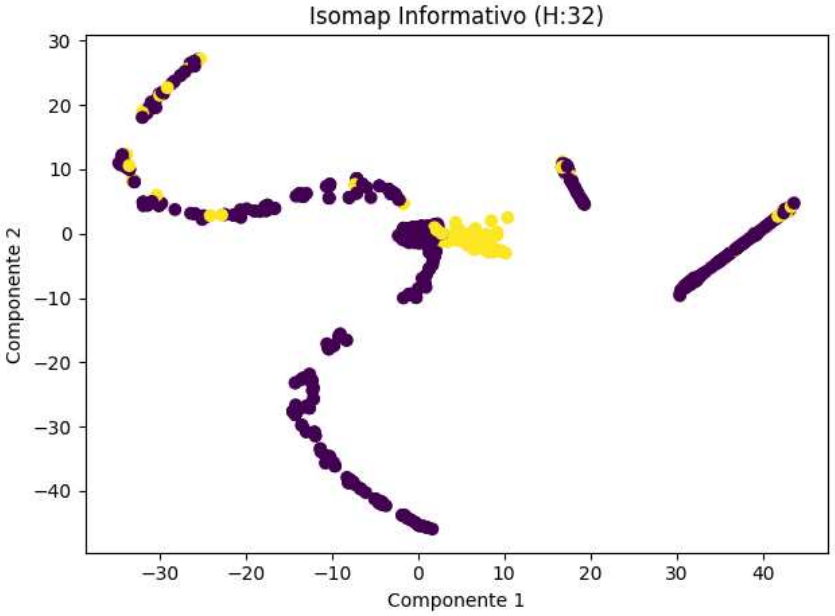

Figura 21: Projeção Isomap com dimensionalidade 32 para grafos informativos. Em amarelo, os grupos de menus. Em azul, os itens de menus.

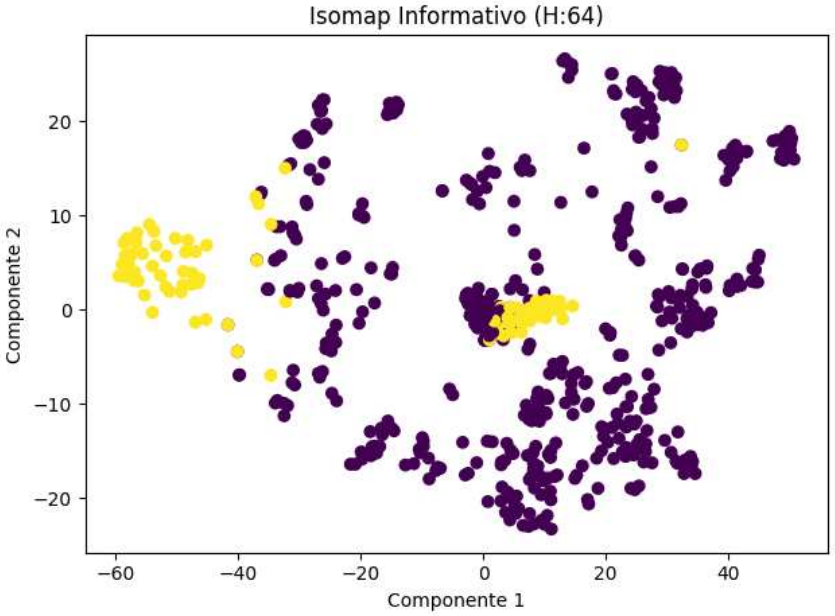

Figura 22: Projeção Isomap com dimensionalidade 64 para grafos informativos. Em amarelo, os grupos de menus. Em azul, os itens de menus.

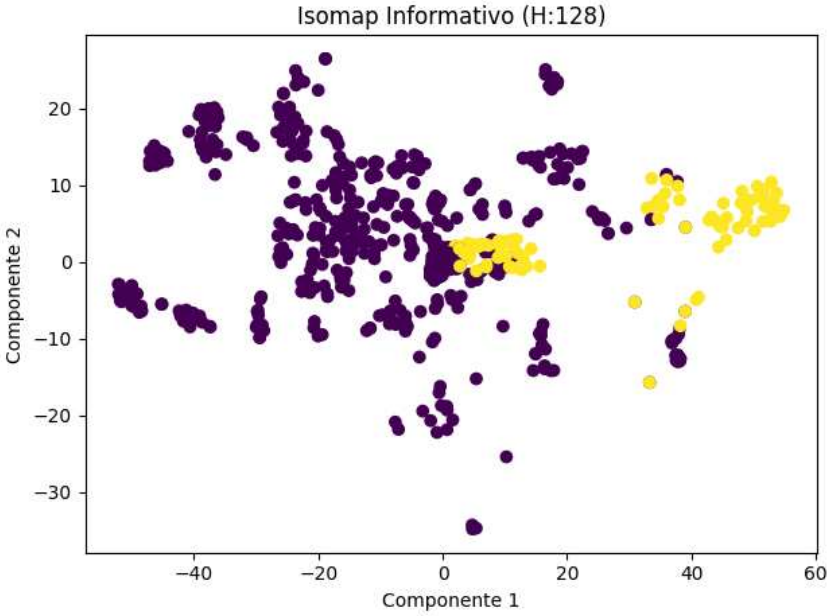

Figura 23: Projeção Isomap com dimensionalidade 128 para grafos informativos. Em amarelo, os grupos de menus. Em azul, os itens de menus.

Esta separação foi evidente nas projeções PCA para todas as dimensionalidades exploradas. A projeção t-SNE compartilhou algumas características das projeções PCA, mas também destacou a formação de clusters isotrópicos dentro da classe de itens de menu, relacionados em parte com seu conteúdo semântico.

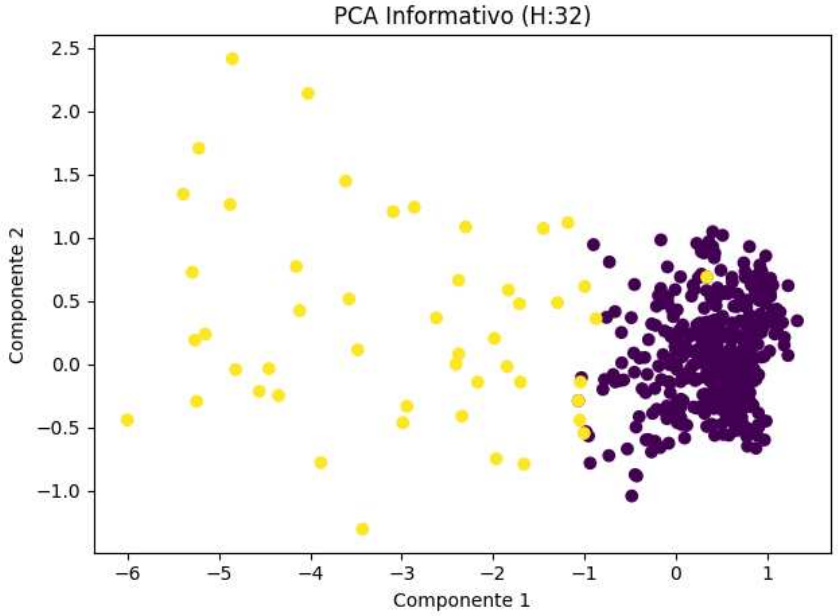

Figura 24: Projeção PCA com dimensionalidade 32 para grafos informativos. Em amarelo, os grupos de menus. Em azul, os itens de menus.

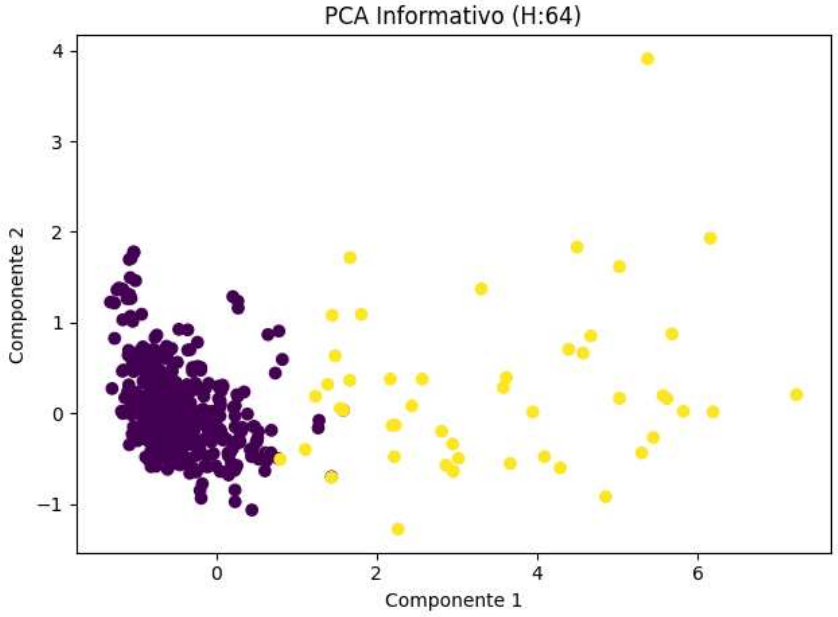

Figura 25: Projeção PCA com dimensionalidade 64 para grafos informativos. Em amarelo, os grupos de menus. Em azul, os itens de menus.

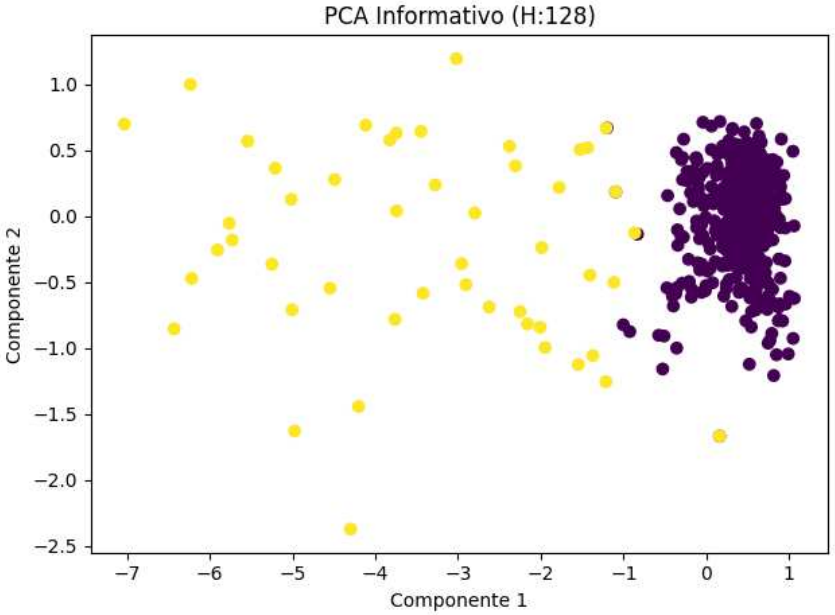

Figura 26: Projeção PCA com dimensionalidade 128 para grafos informativos. Em amarelo, os grupos de menus. Em azul, os itens de menus.

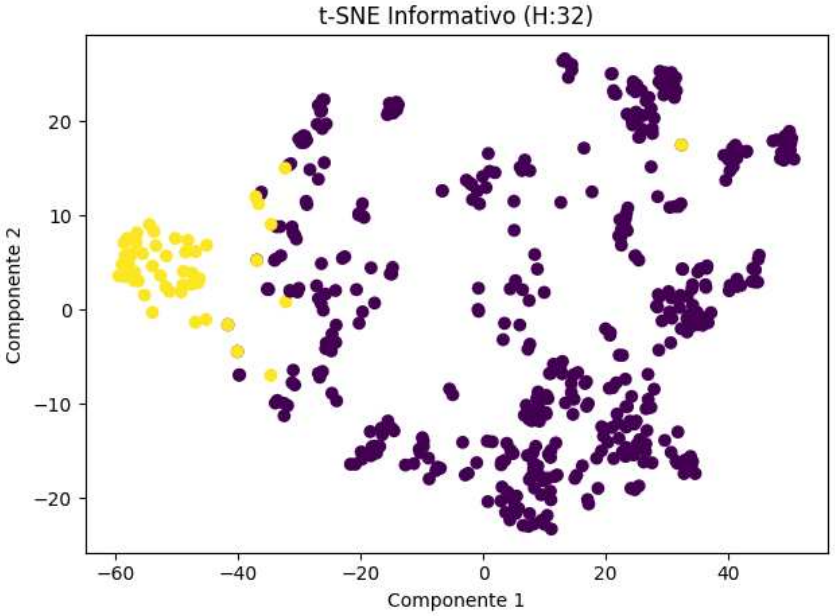

Figura 27: Projeção t-SNE com dimensionalidade 32 para grafos informativos. Em amarelo, os grupos de menus. Em azul, os itens de menus.

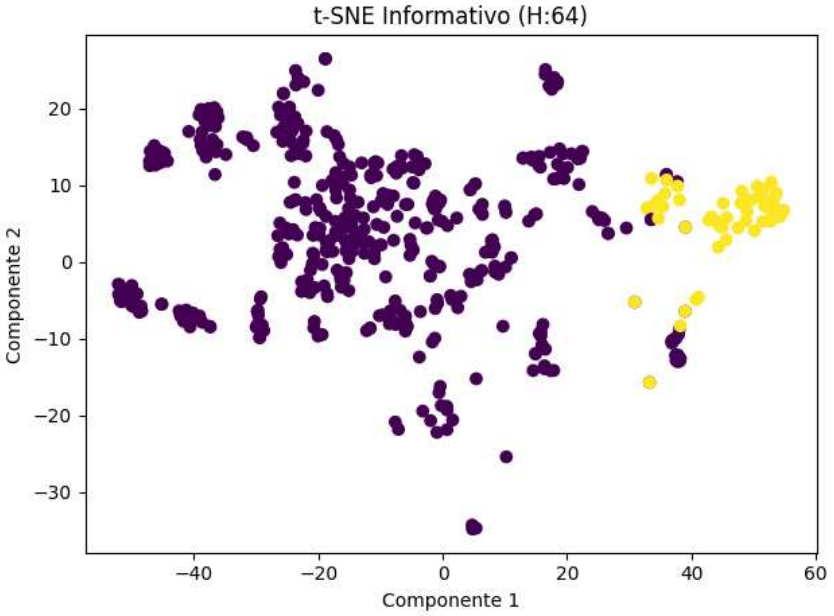

Figura 28: Projeção t-SNE com dimensionalidade 64 para grafos informativos. Em amarelo, os grupos de menus. Em azul, os itens de menus.

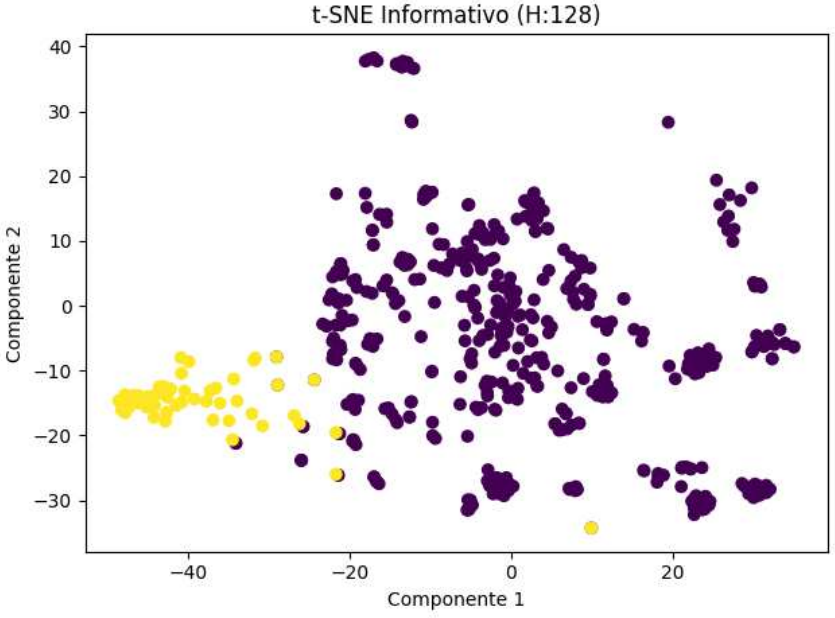

Figura 29: Projeção t-SNE com dimensionalidade 128 para grafos informativos. Em amarelo, os grupos de menus. Em azul, os itens de menus.

Trabalhos futuros devem explorar técnicas para diferenciar nós com alta similaridade semântica. Estratégias que utilizem informações de fontes externas, como modelos de linguagem, podem ser benéficas para melhorar a precisão da classificação e gerar embeddings mais significativos. Experimentos adicionais devem explorar variações no número de camadas, na taxa de aprendizado e no uso de outros modelos, como GAT, bem como o impacto de diferentes inicializações de pesos e funções de perda.

Em conclusão, a GCN demonstrou sua capacidade de utilizar a estrutura do grafo para gerar embeddings de nós que permitam uma classificação precisa. A diferença de desempenho entre grafos informativos e não informativos destaca que considerar as relações entre os nós ajuda no aprendizado, obtendo resultados mais precisos. Pesquisas futuras devem investigar o uso de conhecimento externo e métodos de representação de nós mais robustos para obter melhores resultados.

5.5 Sistema de Rotulação de Menus

Embora o sistema de classificação de grupos de menus por títulos e a abordagem metaheurística ser uma solução aceitável para estruturas de cardápios simples, ela ainda não apresenta o nível de qualidade necessária para inserção na cadeia de produção de alta demanda. Isso se deve por causa de algumas limitações práticas, dentre as quais pode-se elencar: a) O corpus de grupo de menu para classificação de texto não vem da mesma fonte de dados das imagens de menu disponíveis. A origem distinta entre as bases de dados resulta perceptivelmente em vieses que prejudicam a correta classificação dos itens. Mesmo com a abordagem da aprendizagem não-supervisionada na categorização dos grupos de menus, o corpus não abrange minimamente o domínio da base de dados de menus. b) A base de imagens de menus, apesar de ter um volume razoável (pouco mais de 9000 imagens) não são rotuladas, portanto, dificultando uso de técnicas de aprendizagem de máquina, bem como processos de avaliação quantitativo. c) As detecções de OCR apresenta alguns erros pontuais. Estes erros de detecção precisam ser removidos para evitar ruído na clusterização e outras tarefas essenciais.

Diante destas limitações, a melhor estratégia, tanto de vista de negócio como de pesquisa e desenvolvimento, decidiu-se desenvolver um produto mínimo viável que possibilite a rotulação hierárquica das imagens de cardápios.

O sistema é composto por duas partes principais: 1) *Back-end* e 2) *Front-end*.

O *Back-end* é parte do sistema que tem como responsabilidade processar as informações enviadas pelos usuários, aplicando as regras de negócios da aplicação e integrando com banco de dados e outros serviços auxiliares.

No contexto do produto mínimo viável, o *Back-end* tem como função principal forne-

cer uma API para receber as imagens de cardápios enviadas pelos usuários e alimentar os algoritmos de OCR, detecção de regiões de interesse, bom como, integração com base de dados.

O *Front-end*, por outro lado, tem a responsabilidade de fornecer a experiência de usuário durante o processo de rotulação hierárquica das imagens. O *Front-end* fornece interface gráfica e interação com o usuário. Por este motivo, ele apresenta maior complexidade.

Ao ser enviada uma imagem para o sistema, o *Back-end* aplica o OCR, e algoritmos supracitados para detecção inicial de regiões de grupo de menus, itens de menus , descrição e preços. Assim, a imagem já é carregada na interface com uma rotulação inicial, ainda que provavelmente não seja perfeita.

O *Front-end* por vez possibilita corrigir, inserir e remover regiões por meio da ferramenta de interação com o usuário. A interface possibilita remover, movimentar e criar caixas de anotações, associando as classes: Grupo de Menu, Item de Menu, Descrição e Preço. Ao selecionar uma região como Grupo de Menu, o sistema inclui todos os Itens de Menus que estão posicionados dentro da região.

Heuristicamente, o sistema busca associar os Itens de Menus com preços à sua direita. Esta associação inicial é geralmente eficaz na maioria dos cardápios porque os preços estão posicionados à esquerda dos itens de menus. O sistema ainda permite rotular os textos relacionados com descrição de itens e outros textos presentes no documento.

Devido à complexidade da interface gráfica, o *Front-end* foi desenvolvido, no padrão MVC em que o modelo é representado por um gerenciamento de estado centralizado. Esta arquitetura permite à interface gerenciar as complexas interações entre os objetos e elementos da interface. No produto mínimo viável, o gerenciamento de estado centralizado da aplicação permite atualizar o estado da aplicação utilizando padrões de programação funcional redux com transições de estado em imutabilidade temporal. Para este fim, a aplicação utilizou-se a biblioteca Pinia integrada com o framework VueJS. A Figura 30 apresenta a estrutura esquemática do sistema de anotações, enquanto a Figura 31 mostra a tela principal do sistema de anotações desenvolvido.

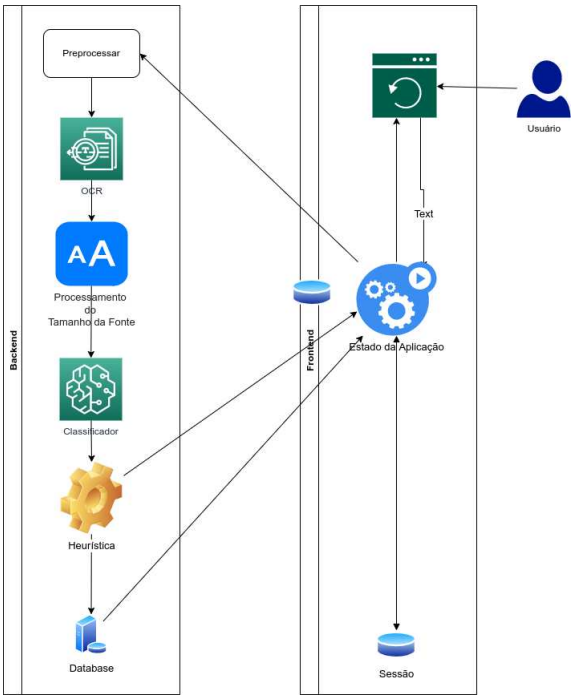

Figura 30: Fluxograma do sistema de anotação. O sistema envolve os módulos desenvolvidos durante a pesquisa, contudo, visa principalmente fornecer uma ferramenta de anotação de cardápio especializada.

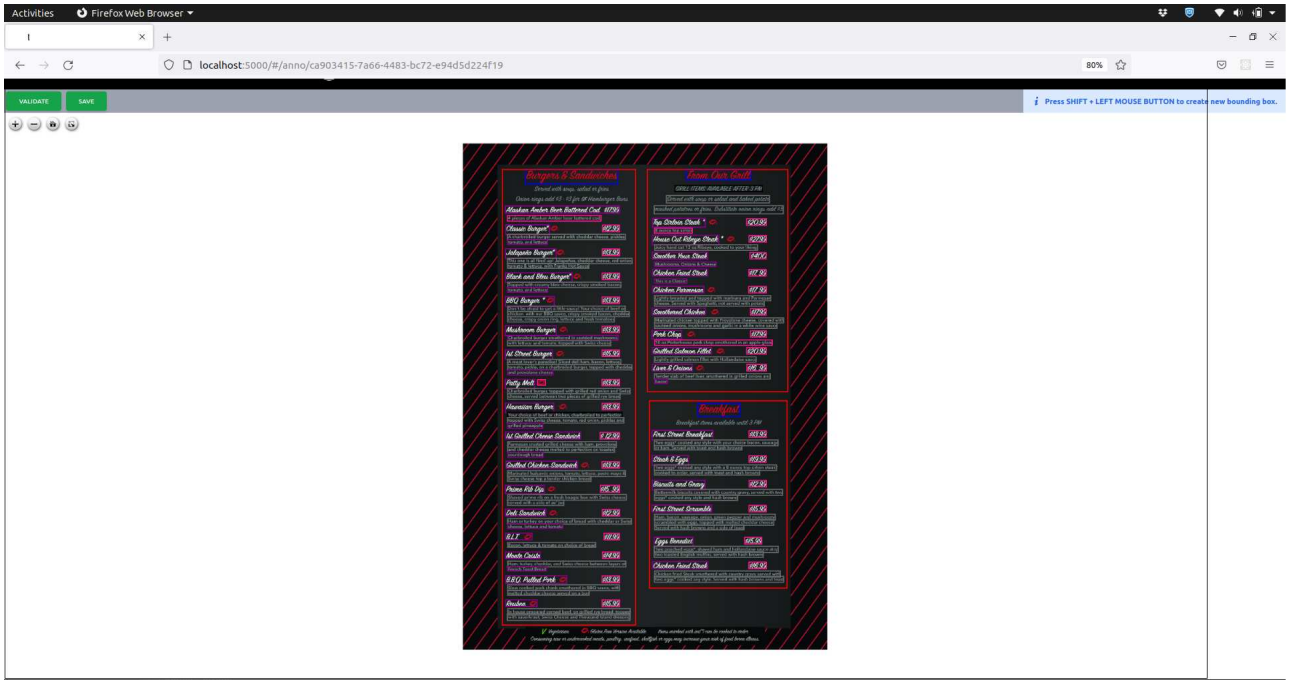

Figura 31: Interface principal do produto mínimo viável. Nesta interface, os usuário podem classificar as regiões de texto detectadas e adicionar novas regiões como grupo de menus. A cor de cada retângulo representa o tipo de classe que foi escolhido para a região de interesse.

6 Conclusões

O processamento eficiente de dados de clientes é essencial para os negócios, especialmente para empresas que fornecem sistemas de pontos de venda (*Point of Sale* – POS) para restaurantes e estabelecimentos de venda direta de alimentos. O principal gargalo nesse processamento é a extração de informações dos cardápios enviados pelos clientes. Para buscar soluções que minimizem esse obstáculo, adotou-se uma abordagem cíclica de pesquisa e desenvolvimento, com o objetivo de entender as dificuldades inerentes ao problema, tanto na obtenção dos dados quanto na aplicação das soluções.

Durante as fases de pesquisa, foram aplicadas técnicas de aprendizagem de máquina supervisionadas e não supervisionadas aos dados disponibilizados pela instituição parceira, com o intuito de compreender e classificar hierarquicamente as informações extraídas por OCR. As investigações revelaram que a extração de dados não estruturados, como imagens de cardápios, apresenta desafios significativos, não apenas em relação aos modelos de aprendizagem de máquina, mas também devido à

escassez de dados rotulados de qualidade. Diante dessas descobertas, estabeleceu-se que a necessidade imediata na fase de desenvolvimento seria a criação de um protótipo que auxiliasse na anotação hierárquica das regiões de interesse nos menus.

Os experimentos conduzidos com redes convolucionais em grafos (*Graph Convolutional Networks* – GCNs) forneceram insights valiosos sobre como a estrutura do grafo influencia os *embeddings* gerados. A discrepância observada no comportamento das projeções de grafos informativos e não informativos realça a importância da informação relacional entre itens e grupos de menus e da forma como esses dados são estruturados. A análise integrada dos resultados provenientes de diferentes técnicas de redução de dimensionalidade evidenciou o potencial das GCNs em capturar informações tanto estruturais quanto semânticas, abrindo caminho para futuras pesquisas em representação e classificação de dados hierárquicos.

Adicionalmente, foram exploradas técnicas de detecção, classificação e métodos não supervisionados, levando à compreensão de que uma única técnica isolada não é suficiente para lidar com dados não rotulados e layouts com estruturas complexas. A combinação de técnicas de aprendizagem de máquina e processamento de imagens, como redes neurais convolucionais e GCNs, mostrou-se promissora para a extração de informações de cardápios, pois cada abordagem captura diferentes aspectos dos dados, facilitando esse processo complexo.

O acesso a dados públicos permanece um desafio, visto que os dados existentes são fechados e não estão disponíveis para a comunidade científica. A falta de dados rotulados de qualidade é um dos principais obstáculos para a aplicação eficaz de técnicas de aprendizagem de máquina em problemas de extração de informações de cardápios. Neste estudo, o acesso a dados de uma empresa parceira permitiu explorar técnicas de aprendizagem de máquina para essa finalidade. Contudo, a escassez de dados rotulados de qualidade e os aspectos de licenciamento limitaram a produção de contribuições mais significativas.

A fase de desenvolvimento culminou em uma aplicação mínima viável que permite a anotação de menus processados por OCR e algoritmos heurísticos, servindo como fonte para a captura de dados rotulados de qualidade e para a criação de bases de dados para modelos mais elaborados. Em suma, este trabalho evidenciou os desafios e as potenciais soluções na extração de informações de cardápios, destacando a importância da combinação de técnicas e do desenvolvimento de ferramentas que auxiliem na geração de dados de qualidade, contribuindo para avanços futuros na área.

Referências

[1] Sanghyeon An, Minjun Lee, Sanglee Park, Heerin Yang, and Jungmin So. An ensemble of simple convolutional neural network models for mnist digit recognition. *arXiv preprint arXiv:2008.10400*, 2020.

[2] Gavin Bierman, Martín Abadi, and Mads Torgersen. Understanding typescript. In *European Conference on Object-Oriented Programming*, pages 257–281. Springer, 2014.

[3] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. "O'Reilly Media, Inc.", 2009.

[4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[5] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[6] Shiv Ram Dubey, Satish Kumar Singh, and Bidyut Baran Chaudhuri. Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, 2022.

[7] Felipe L Gewers, Gustavo R Ferreira, Henrique F De Arruda, Filipi N Silva, Cesar H Comin, Diego R Amancio, and Luciano da F Costa. Principal component analysis: A natural approach to data exploration. *ACM Computing Surveys (CSUR)*, 54(4):1–34, 2021.

[8] Hossein Gholamalinezhad and Hossein Khosravi. Pooling methods in deep neural networks, a review. *arXiv preprint arXiv:2009.07485*, 2020.

[9] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

[10] Miguel Grinberg. *Flask web development: developing web applications with python*. "O'Reilly Media, Inc.", 2018.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[12] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[13] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2017.

[14] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, and Carol Willing. Jupyter notebooks – a publishing format for reproducible computational workflows. In F. Loizides and B. Schmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87 – 90. IOS Press, 2016.

[15] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[16] Jamshed Memon, Maira Sami, Rizwan Ahmed Khan, and Mueen Uddin. Handwritten optical character recognition (ocr): A comprehensive systematic literature review (slr). *IEEE Access*, 8:142642–142668, 2020.

[17] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[19] PostgreSQL. `https://www.postgresql.org/`.

[20] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[21] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.

[22] Radim Rehurek and Petr Sojka. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2), 2011.

[23] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[24] Tailwind CSS. `https://tailwindcss.com/`.

[25] PV Vishnu Teja and Araddhana Arvind Deshmukh. A survey on identification of vehicle number plate using viola jones. 2019.

[26] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[27] textract. `https://aws.amazon.com/pt/textract/`.

[28] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[29] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.

[30] Siva R Venna, Amirhossein Tavanaei, Raju N Gottumukkala, Vijay V Raghavan, Anthony S Maida, and Stephen Nichols. A novel data-driven model for real-time influenza forecasting. *Ieee Access*, 7:7691–7701, 2018.

[31] VujeJS. `https://vuejs.org/`.

[32] Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021.

[33] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010.

[34] Dongkuan Xu and Yingjie Tian. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193, 2015.