INSTITUTO
UNIVERSITÁRIO
DE LISBOA

# Extração automática de informação de faturas digitalizadas

José Eduardo da Conceição Correia

Mestrado em Engenharia Informática

Orientador:
Doutor Ricardo Daniel Santos Faro Marques Ribeiro, Professor Associado, Iscte – Instituto Universitário de Lisboa

Co-Orientador:
Doutor Tomás Gomes da Silva Serpa Brandão, Professor Auxiliar, Iscte – Instituto Universitário de Lisboa

Outubro, 2024

TECHNOLOGY
AND ARCHITECTURE

Departamento de Ciências e Tecnologias da Informação

# Extração automática de informação de faturas digitalizadas

José Eduardo da Conceição Correia

Mestrado em Engenharia Informática

Orientador:
Doutor Ricardo Daniel Santos Faro Marques Ribeiro, Professor Associado, Iscte – Instituto Universitário de Lisboa

Co-Orientador:
Doutor Tomás Gomes da Silva Serpa Brandão, Professor Auxiliar, Iscte – Instituto Universitário de Lisboa

Outubro, 2024

# Agradecimentos

Gostaria de expressar a minha gratidão a todos que tornaram este percurso possível. Em primeiro lugar, aos meus professores, Tomás Brandão e Ricardo Ribeiro, que não desistiram de mim principalmente quando o tempo era escasso e pelo apoio e disponibilidade para ajudar. Sempre pacientes, mantiveram o meu foco em alcançar este meu objetivo.

À Inês Saraiva, colaboradora da Axians Portugal, um sincero obrigado pelo apoio indispensável para o desenvolvimento deste trabalho.

Por fim, à minha família, agradeço por cada gesto de apoio desde o início e por toda a compreensão ao longo deste percurso.

Obrigado a todos.

# Resumo

Atualmente, com a importância dos Fundos europeus (FE) torna-se especialmente relevante a otimização de processos com vista a conseguir fluxos de trabalho mais rápidos e eficientes. Para enfrentar este desafio, a utilização de ferramentas avançadas que integram Inteligência artificial (IA) tem o potencial de executar tarefas realizadas por humanos de forma mais rápida, com maior precisão e menor margem de erro.

Neste trabalho foi realizado um estudo comparativo entre um modelo que usa a ferramenta *LayoutLMv2* e um modelo que usa *LayoutLMv3* para extração automática de dados de faturas. Para tal, o modelo aplicado que utiliza a ferramenta *LayoutLMv3* foi submetido a inúmeras experiências, tendo-se descoberto melhores hiperparâmetros do modelo para se conseguir um melhor desempenho. Posteriormente, realizou-se uma avaliação comparativa de resultados entre o modelo aplicado com a ferramenta *LayoutLMv3*, que apresenta uma arquitetura com base *Vision transformer* (VIT) e um modelo já desenvolvido que se chama Intelligent Document Automation (IDA) e que usa a ferramenta *LayoutLMv2* que se baseia em Redes neuronais convolucionais (CNN).

Os resultados finais para o modelo aplicado *LayoutLMv3* apresentam um melhor desempenho nos campos mais genéricos como número da fatura (Pontuação F1 de 90%) ou data da fatura (Pontuação F1 de 91%) ou nome do fornecedor (Pontuação F1 de 91%). Enquanto que o IDA apresenta melhores resultados para campos de maior detalhe como descrição do produto (Pontuação F1 de 91%) ou quantidade (Pontuação F1 de 94%).

Palavras Chave: *Transformadores*, *Visão e linguagem*, *Aprendizagem automática*, *LayoutLM*, *Faturas*, *Extração de informação*, *IA para documentos*

# Abstract

Nowadays, with the importance of european funds, optimizing processes to achieve faster and more efficient workflows becomes especially relevant. To tackle this challenge, the use of advanced tools that integrate AI has the potential to perform tasks carried out by humans more quickly, with greater accuracy and a lower margin of error.

In this work, a comparative study was conducted between a model that uses the LayoutLMv2 tool and a model that uses LayoutLMv3 for automatic data extraction from invoices.

To this end, the applied model using the LayoutLMv3 tool was subjected to numerous experiments to discover the best hyperparameters for achieving the best possible performance. Subsequently, a comparative evaluation of results was carried out between the applied model using the LayoutLMv3 tool, which features a ViT-based architecture, and an already developed model called IDA that uses the LayoutLMv2tool, whose architecture focuses more on CNN.

The final results for the applied model using the LayoutLMv3 tool show better performance for more generic fields such as invoice number (F1 Score of 90%), invoice date (F1 Score of 91%), or supplier name (F1 Score of 91%). Meanwhile, IDA shows better results for more detailed fields such as product description (F1 Score of 91%) or quantity (F1 Score of 94%).



KEYWORDS: *Transformers*, *Vision and language*, *Machine learning*, *LayoutLM*, *Invoices*, *Information extraction*, *IA for documents*

# Conteúdo

# Lista de Figuras

# Lista de Tabelas

# Lista de acrónimos

**ADAM:** *Adaptive moment estimation*

**AG:** Autoridades de gestão

**BERT:** *Bidirectional encoder representations from transformers*

**CNN:** Redes neuronais convolucionais

**FE:** Fundos europeus

**FEADER:** Fundo europeu agrícola de desenvolvimento rural

**FEAMP:** Fundo europeu dos assuntos marítimos e das pescas

**FEEI:** Fundos europeus estruturais e de investimento

**FEDER:** Fundo europeu de desenvolvimento regional

**FN:** Falso negativo

**FP:** Falso positivo

**FPN:** *Feature pyramid network*

**FSE:** Fundo social europeu

**FC:** Fundo de coesão

**IBAN:** Número da conta bancária internacional

**IA:** Inteligência artificial

**IDA:** Intelligent Document Automation

**MIM:** *Masked image modeling*

**MLM:** *Masked language modeling*

**NIF:** Número de identificação fiscal

**PIB:** Produto interno bruto

**PLN:** Processamento de linguagem natural

**OCR:** Reconhecimento ótico de caracteres

**SGD:** *Stochastic gradient descent*

**SVTR:** *Scene text recognition with a single visual mode*

**UE:** União europeia

**VN:** Verdadeiro negativo

**VP:** Verdadeiro positivo

**VIT:** *Vision transformer*

**WPA:** *Word-patch alignment*

CAPÍTULO 1

# Introdução

Os Fundos europeus (FE) têm um papel verdadeiramente crucial no desenvolvimento dos diferentes países, notando-se, evidentemente, um maior impacto nos países na União europeia (UE) com Produto interno bruto (PIB) per capita abaixo da média da UE. Estes países não têm, na sua generalidade, verdadeira capacidade de realizar grandes investimentos. Consequentemente, estes fundos procuram promover um crescimento económico mais equilibrado e sustentável em toda a região da UE, visando reduzir as inevitáveis disparidades de desenvolvimento entre as regiões mais ricas e as mais pobres [1]. Para além dos fatores económicos, também constituem um pilar fundamental da política de coesão da UE, pois promovem o crescimento económico, a coesão social e territorial entre os Estados-Membros da UE, mitigando as disparidades económicas e sociais existentes, garantindo que todas as regiões e cidadãos da UE tenham acesso a oportunidades e serviços de maior qualidade [1].

## 1.1. Contexto e motivação

Nesta secção, será abordado o propósito dos FE, as razões que motivaram a sua criação e uma breve análise da sua evolução até à atualidade. Em seguida, será abordado o seu impacto no contexto de Portugal, explorando como têm sido aplicados para promover o desenvolvimento económico e social do país.

### 1.1.1. Fundos europeus

Desde sempre, o mundo tem estado em constante mudança com um ritmo de crescimento exponencial e para auxiliar os países menos desenvolvidos a acompanharem esta constante mudança e terem a capacidade de inovar em vários setores, foram criados os FE.

Em 1975, o primeiro Fundo Europeu de Desenvolvimento Regional teve como objetivo específico reduzir as disparidades regionais e promover o desenvolvimento das regiões menos desenvolvidas para melhorar a qualidade de vida [2]. O programa consistiu num grande e efetivo investimento no crescimento, no emprego e na cooperação territorial europeia através do desenvolvimento e ajuste estrutural das regiões menos desenvolvidas e na reconversão das regiões industriais que se encontravam em declínio.

Posteriormente, outros fundos foram estabelecidos, como o Fundo Social Europeu, o Fundo de Coesão e o Fundo Europeu Agrícola de Desenvolvimento Rural, cada um com foco em diferentes áreas específicas de intervenção na respetiva atividade económica.

Cerca de metade das verbas do orçamento de UE do período de 2014 a 2020 foi direcionado para cinco Fundos Europeus Estruturais e de Investimento Fundos europeus estruturais e de investimento (FEEI), Fundo europeu de desenvolvimento regional (FEDER),

Fundo social europeu (FSE), Fundo de coesão (FC), Fundo europeu agrícola de desenvolvimento rural (FEADER) e Fundo europeu dos assuntos marítimos e das pescas (FEAMP). Todos estes fundos foram geridos em parceria pela Comissão Europeia e pelos países membros da UE. Os Estados-Membros obtiveram autorização para a distribuição de recursos dos FEEI até o final de 2023 [3].

Todos os fundos criados tentaram impulsionar a competitividade das regiões europeias, apoiando investimentos em infraestruturas, inovação, educação e formação, bem como no desenvolvimento de Pequenas e Médias Empresas. Isso ajudou a criar empregos, estimular o empreendedorismo e promover um desenvolvimento económico mais dinâmico e inclusivo em toda a UE.

### 1.1.2. Fundos europeus em Portugal

Entre 2013 e 2020, foram injetados na economia portuguesa cerca 34 milhões de euros em fundos, dos quais cerca de 29 milhões de euros vieram de FE e o restante valor foi financiado diretamente pelo país com o objetivo primordial de estabelecer um co-financiamento [4].

Como se pode analisar na Figura 1.1, Portugal não conseguiu da maneira mais eficiente aproveitar os fundos europeus, sendo esta situação explicada pormenorizadamente na Secção 1.2.

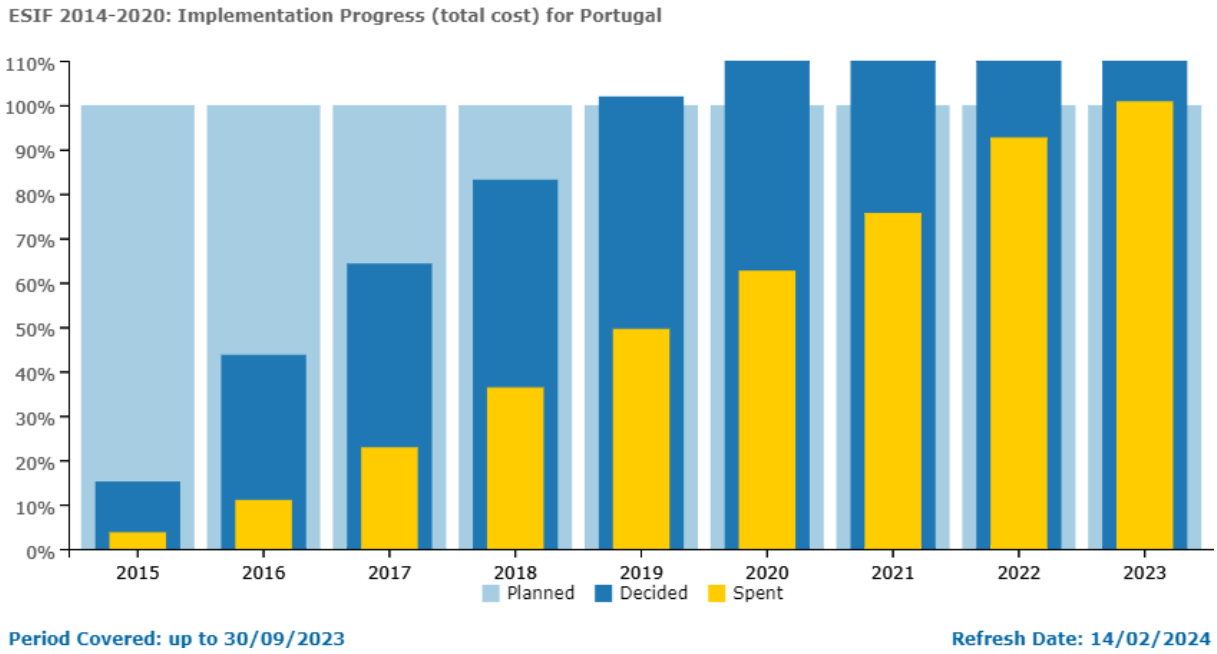



FIGURA 1.1. Progresso da Implementação dos FE em Portugal 2014-2023 [4]

Em 2015, a UE tinha orçamentado/disponibilizado para Portugal 32 milhões de euros, o que representa 100% *Planned*, cor azul clara, que significa o valor que UE disponibilizou no seu orçamento, porém, Portugal só conseguiu que aprovassem 15% dos 32 milhões de euros que a UE disponibilizava, representado a cor azul escura (*Decided*). Destes 15% do valor aprovado que Portugal ia receber, só conseguiram receber 4% do orçamento

planeado, representado pela cor amarela (*Spent*). Consequentemente, em 2015, Portugal não aproveitou os 96% do que estava orçamentado representando cerca de 30 milhões euros, dinheiro que não foi alocado e inviabilizou o crescimento económico [5].

A partir de 2019, dá-se um volte-face efetivo, pois começam a ser aprovados mais projetos do que estavam disponibilizados no orçamento da UE, existindo uma diferença em dois pontos percentuais de 100% para 102% nesse mesmo ano. Concretamente, até 2023 houve uma evolução notável, tendo-se conseguido aproveitar cabalmente os fundos que nos foram atribuídos, uma vez que Portugal implementou várias reformas estruturais, permitindo utilizar de forma mais eficaz os fundos europeus.

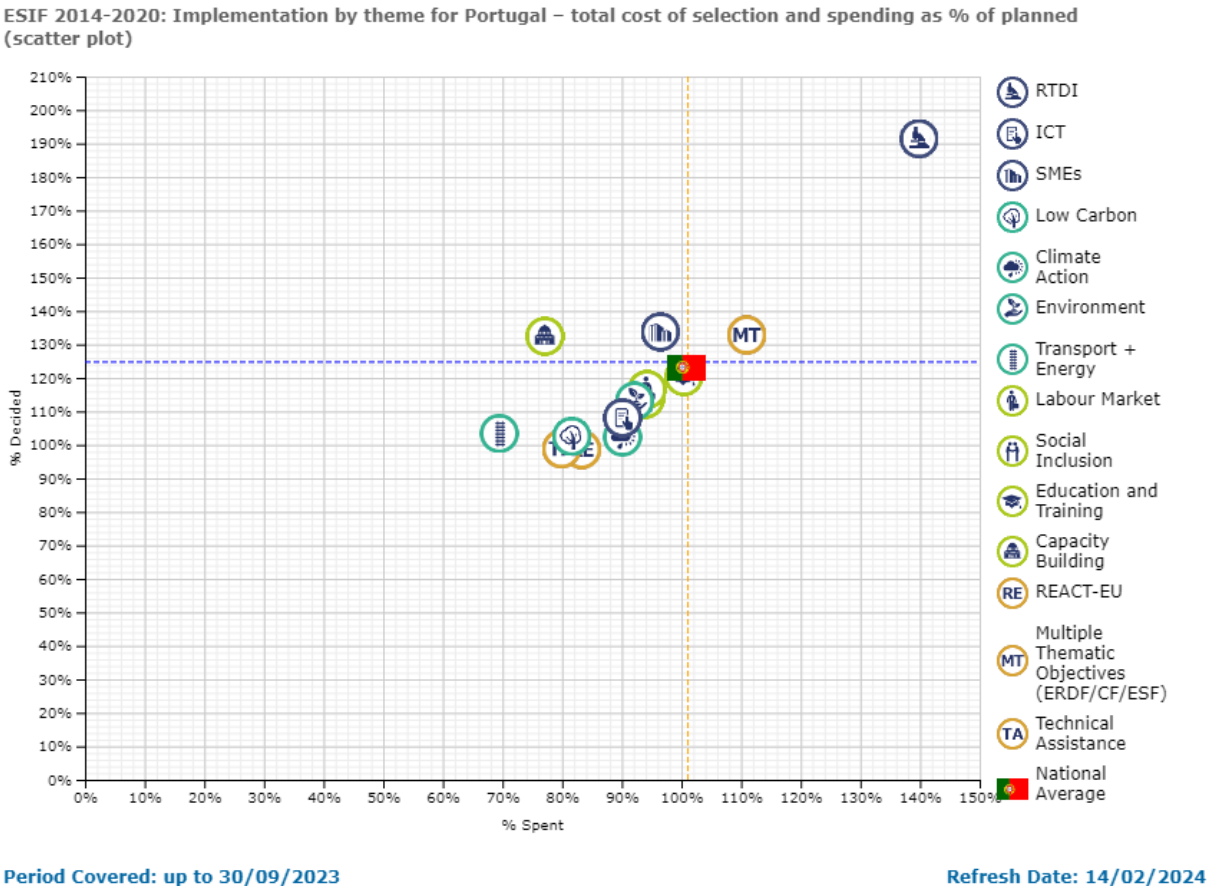



FIGURA 1.2. Implementação dos FE por tema em Portugal 2023 [4]

Na Figura 1.2 é possível analisar a distribuição dos fundos europeus em Portugal, por tema, em 2023, tendo sido o melhor ano para Portugal, quanto ao seu aproveitamento. Todos os temas, que estão representados no gráfico, tiveram 100% ou mais na aprovação para receber FE. A maioria dos temas está entre os 100% e os 120% planeado, porém, na maioria dos casos, só recebeu 80% do valor pretendido. Apesar disto, verifica-se uma grande evolução comparando com o ano de 2015.

O fundo que mais se notabilizou, foi o tema **RTDI**, *RESEARCH AND INNOVATION*, o qual teve o melhor aproveitamento nos FE, pois obteve 190% decidido para receber, ou seja, o FE tinha uma verba cerca de 3,5 milhões de euros para distribuir neste tema para Portugal e foi conseguido que a UE disponibilizasse mais 90% do que tinha sido planeado,

o que presenta um valor de 6,7 milhões de euros, mas o que foi realmente recebido foi 140% do orçamento que é 4,9 milhões de euros.

### 1.2. Desafios

Como se pode verificar na Secção 1.1.2, inicialmente em 2015, não existia um grande aproveitamento dos FE que a UE disponibilizava. Apesar de em 2023, o valor recebido ter sido superior em apenas 1 ponto percentual em relação ao valor orçamentado, ainda assim não foi possível chegar à eficiência máxima.

Após a candidatura entrar em estado de execução, é necessário analisar os pedidos de pagamento para que o valor apresentado nas faturas possa ser reembolsado. Para isso, é fundamental verificar se a fatura apresenta indícios de fraude, se a descrição dos produtos está dentro do âmbito do programa definido pelo aviso da candidatura, se o Número de identificação fiscal (NIF) fornecido é válido, entre outros. Existem projetos que podem conter milhares de pedidos de pagamento, o que representa milhares de faturas para analisar, se o processo for analisado por um indivíduo, consome muito tempo e o processo está sujeito a múltiplos erros.

Para automatizar a análise e a validação das faturas de uma forma automática sem intervenção humana, o governo português implementou uma medida que torna obrigatória a inclusão de um código de barras bidimensional (código QR) nas faturas [6]. O código QR permite a leitura de todas as informações relevantes da fatura. No entanto, existem desafios na abordagem do código QR que são os seguintes:

- Nem todos os países incluem como obrigatoriedade a implementação do código QR nas faturas, e, quando o fazem, podem utilizar mecanismos de integração e leitura de dados diferentes do sistema português.
- Frequentemente, a leitura do código QR pode falhar devido a ruídos na imagem da fatura ou problemas técnicos, o que impede a extração dos dados.

Para simplificar o processo, podemos usar recursos tecnológicos de inovação, neste caso Inteligência artificial (IA), que permitam substituir tarefas rotineiras, monótonas para o ser humano, permitindo, desta forma, uma maior velocidade de execução. Através de tecnologias de Processamento de linguagem natural (PLN) pode-se evitar este trabalho humano propício a erros e de uma maneira muito mais eficiente.

Neste momento, existe uma solução desenvolvida pela equipa *CO-INOVATION* da empresa Axians Portugal que se chama Intelligent Document Automation (IDA) que tem como finalidade o reconhecimento digital de faturas e a extração de informação. Esta ferramenta baseada no *LayoutLMv2* permite extrair informações da fatura através de uma imagem como por exemplo o NIF e os respetivos valores dos produtos da fatura, entre outros. Efetivamente, a ferramenta encontra-se operacional, porém existem, ainda, fatores a melhorar. Os três principais problemas identificados no IDA são os seguintes:

**Limite de entradas permitido pela ferramenta:** Existe um limite de extração de entradas e consequentemente é necessário dividir o documento e analisar de forma

independente cada uma das partes para extrair toda informação do documento. Dividir o documento vai gerar dificuldade, pois pode existir informação duplicada ou informação que não é extraída. Cada extração não tem o contexto da outra extração.

**Problemas com ligação de entidades:** Constituintes que deveriam estar totalmente ligados e aparecer como um único elemento, mas que se apresentam como três elementos separados, como ilustrado na Figura 1.3.


FIGURA 1.3. Exemplo de ligação de entidades

**Aperfeiçoamento no modelo para faturas com estruturas diferentes:** Existem faturas que têm uma estrutura diferente em relação a outras, o que acresce a dificuldade para um modelo pré-treinado para extração de informação.

## 1.3. Objetivos e questões de investigação

O objetivo da dissertação é estudar a aplicação de um modelo de aprendizagem profunda através da tecnologia PLN com a mesma finalidade do IDA, mas utilizando uma ferramenta mais recente do que o IDA, o *LayoutLMv3*, recorrendo ao mesmo conjunto dados.

Considerando o contexto abordado anteriormente, este trabalho aborda as seguintes questões de investigação:

- Quais os métodos alternativos ao código QR que podem ser adotados para a digitalização e validação de faturas e como é que eles se comparam em termos de eficiência e fiabilidade?
- Como se comporta o *LayoutLMv3* quando aplicado para análise de faturas em relação às métricas de avaliação de desempenho? As métricas são Precisão, Cobertura, Pontuação F1 e Taxa de Acerto.
- Quais são as diferenças significativas entre os resultados obtidos pelo novo modelo aplicado e os resultados anteriormente gerados pelo IDA e quais as melhorias e diferenças significativas observáveis?

## 1.4. Estrutura da dissertação

Após a apresentação da motivação, do contexto dos FE e das questões de investigação, o trabalho está estruturado em seis capítulos, no qual se inclui a introdução.

**Capítulo 2:** Aborda conceitos fundamentais para a exploração das tecnologias utilizadas na aplicação do problema, explicando desde o funcionamento de redes neuronais utilizando transformadores até métricas de avaliação e desempenho.

**Capítulo 3:** Apresenta uma revisão da literatura, abordando trabalhos relacionados e discutindo o processo de revisão sistemática da literatura para descobrir estes estudos.

Aborda também a ferramenta *LayoutLMv3* e as suas diferenças em relação ao modelo anterior, *LayoutLMv2*.

**Capítulo 4:** Este Capítulo aborda a descrição do conjunto de dados utilizado para o treino do modelo, detalhando suas características e relevância. Além disso, são descritas diversas experiências conduzidas para otimizar os hiperparâmetros do modelo, com o objetivo de alcançar o melhor desempenho possível no processo de treino.

**Capítulo 5:** Apresenta-se, neste Capítulo, os resultados obtidos a partir do treino do modelo, seguidos da sua respetiva avaliação. Posteriormente, é realizada uma comparação detalhada entre o desempenho do modelo aplicado e o modelo IDA, destacando as diferenças a fim de determinar qual o modelo que oferece os melhores resultados para o contexto analisado.

**Capítulo 6:** Por fim, este Capítulo sumariza as principais conclusões extraídas dos resultados apresentados no Capítulo 5, destacando a justificação do desempenho do respetivo modelo. Além disso, são sugeridas possíveis direções para o trabalho futuro.

CAPÍTULO 2

# Conceitos fundamentais

Este capítulo aborda os conceitos teóricos fundamentais que permitem compreender as bases do modelo aplicado. Além disso, são discutidas as principais métricas utilizadas para avaliar o desempenho do modelo.

## 2.1. Aprendizagem automática

A aprendizagem automática é uma tecnologia que tem vindo a ganhar destaque nos últimos anos, principalmente devido ao avanço na capacidade de processar grandes volumes de dados. O surgimento das GPUs (unidades de processamento gráfico), que permitem o processamento paralelo, facilitou significativamente a manipulação desses dados em grande escala.

A aprendizagem automática possibilita a identificação de objetos em tempo real, tanto em imagens quanto em vídeos, extração de informação e classificação através do reconhecimento de padrões, entre outros. Isso ocorre através de modelos previamente treinados com grandes conjuntos de dados, que adquirem a capacidade de reconhecer padrões.

Existem diversos tipos de algoritmos de aprendizagem automática, sendo os principais a aprendizagem supervisionada, a aprendizagem não supervisionada e a aprendizagem por reforço. Na aprendizagem supervisionada, o modelo é treinado com um conjunto de dados previamente anotados, permitindo classificar corretamente novos exemplos. Enquanto na aprendizagem não supervisionada, o modelo organiza os dados agrupamentos, onde cada um irá conter dados semelhantes (*clustering*). A aprendizagem por reforço tem a finalidade de através de um sistema de recompensas e penalizações, permitir que o modelo aprenda a tomar decisões com base na maximização das recompensas ao longo do tempo.

## 2.2. Introdução à classificação de texto

A classificação de textos é uma tarefa fundamental em PLN que envolve tratamento e anotação dos dados tendo como objetivo a atribuição de categorias/classes para os classificar [7].

Para a classificação de texto, é necessário o uso de algoritmos de aprendizagem automática, que são treinados com dados anotados para, posteriormente, classificar novas informações. No contexto da extração de dados de faturas, esses algoritmos são fundamentais para reconhecer padrões em textos nunca antes visualizados.

Em geral a classificação de texto baseada em algoritmos de aprendizagem automática funciona por cinco etapas, como se pode observar na Figura 2.1.

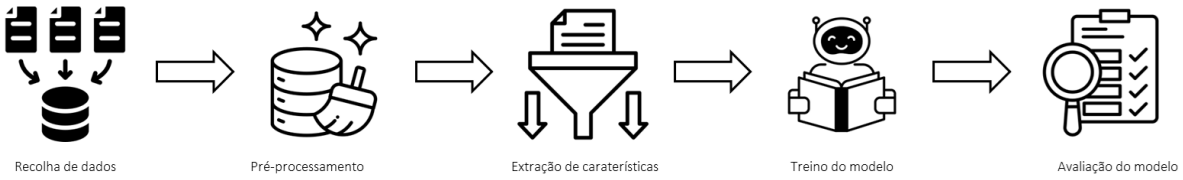



FIGURA 2.1. Processo da classificação de texto

**Recolha de dados:** É necessário criar um conjunto dados variado e numeroso com dados bem anotados, bem como com classes bem definidas para o contexto de negócio definido. Quanto maior e mais diversificado for o conjunto de exemplos, mais o modelo se aproximará da realidade.

**Pré-processamento:** Responsável por realizar um tratamento prévio aos dados do conjunto de dados, antes da fase de aprendizagem do modelo, para reduzir ao máximo o ruído dos dados.

**Extração de caraterísticas:** Existe um processo de tokenização em que todos inputs como as palavras, posições espacial entre outros, são representadas por *tokens*. Para cada *token* é gerado um vetor com n-dimensões de tamanho fixo que vai ser input para o modelo treinar. Este processo designa-se criação de *embeddings* e existem vários algoritmos para gerarem o vetor:

- *Word2Vec*
- *BERT (Bidirectional Encoder Representations from Transformers)*
- *FastText*

Além da tokenização, é comum realizar um tratamento dos dados para remover características desnecessárias e normalizar os *tokens*. Este tratamento pode consistir em remover as características que não são importantes, por exemplo, remover caracteres especiais como o "ç" ou o "º", entre outros; substituir todas as palavras com letras minúsculas; remover palavras genéricas que não acrescentam valor, por exemplo, "e" e "a", entre outros; também a utilização de técnicas como *stemming* e *lemmatization* que têm como finalidade reduzir palavras à sua forma base, sendo que o *stemming* tem como objetivo remover os afixos das palavras, enquanto a *lemmatization* é um processo mais complexo que reduz a palavra para a sua forma canónica, como por exemplo, substituir a formal verbal "correndo" pelo verbo na sua forma infinitiva "correr". O objetivo destas técnicas é normalizar as palavras, reduzindo as variações, o que consequentemente reduz o número de dimensões [7].

**Treino do modelo:** Uma vez que os dados sejam pré-processados e representados como características numéricas ou vetores (extração de caraterísticas), o modelo é treinado com um conjunto de dados anotados. Durante o treino, o algoritmo aprende a identificar padrões e relações entre os *tokens* e suas respetivas classes. Podem ser utilizados vários algoritmos para a classificação de textos, como o

*Naive Bayes*, que é baseado em probabilidades, ou o *Random Forest*, que constrói múltiplas árvores de decisão, entre outros.

- ***Naive Bayes***: é uma técnica baseada no teorema *Bayesian*, que calcula a probabilidade de cada *token* [8].
- ***Random Forest*** É um algoritmo para as tarefas de classificação e regressão. Funciona através da construção de múltiplas árvores de decisão de maneira aleatória [9].

**Avaliação do modelo:** Para terminar, o modelo é avaliado utilizando um conjunto de dados diferente daquele usado no processo de aprendizagem, conhecido como conjunto de testes, para garantir que ele generalize bem para dados não vistos. Esta avaliação é crucial para determinar o desempenho real do modelo em situações do mundo real e evitar problemas como *overfitting*, isto é, onde o modelo fica sobreajustado aos dados de treino, o modelo só memorizou exemplos específicos do conjunto dados de treino, consequentemente, o modelo não consegue aprender os padrões dos dados.

As métricas de avaliação mais comuns incluem a taxa de acerto, que avalia a proporção de previsões corretas em relação ao total de previsões; a precisão, que avalia a proporção de verdadeiros positivos em relação aos positivos previstos; a cobertura, que avalia a capacidade do modelo de identificar corretamente todos os exemplos positivos; e o pontuação F1, que representa a média harmónica entre a precisão e a cobertura, oferecendo um equilíbrio entre essas duas métricas. As métricas de avaliação vão ser explicadas com mais profundidade na Secção 2.6.

## 2.3. Reconhecimento ótico de caracteres (OCR)

O OCR, mais conhecido como *Optical Character Recognition*, tem como finalidade extrair informações de caracteres, como letras, números e símbolos, a partir de imagens, documentos e outros formatos visuais [10].

O processo de OCR inicia-se pela renderização dos dados de entrada em um *bitmap*, permitindo a deteção dos caracteres. Em seguida, aplica-se um algoritmo para identificar o conteúdo. Os algoritmos tradicionais mais utilizados são:

***Pattern Matching*:** Através de um vasto conjunto de dados de caracteres, o algoritmo compara os pixels da imagem com padrões pré-definidos, para depois classificar o carácter conforme as correspondências encontradas.

***Feature Analysis*:** Em vez de comparar diretamente a imagem do caracteres com modelos pré-definidos, este algoritmo identifica características chave, como linhas, curvas, interseções e pontos específicos e utiliza essas características para reconhecer os caracteres.

A principal diferença entre esses dois algoritmos é que o *Feature Analysis* destaca no reconhecimento caracteres de fontes variadas e textos manuscritos, pois não depende de

exemplos de treino específicos para cada caracteres. Em vez disso, ele aprende a partir de características chave, como linhas, curvas entre outros.

A combinação entre o OCR e a IA tem permitido a implementação de métodos mais avançados de reconhecimento, aumentando significativamente a eficiência e precisão do processo. Com o avanço da IA, especialmente com o uso de redes neuronais, destacam-se as principais técnicas de IA aplicadas no OCR:

**Redes neuronais convolucionais (CNN):** Permite identificar padrões visuais nos caracteres, como formas, bordas, linhas e curvas, essenciais para o reconhecimento do texto. Sendo robusto em situações onde existe variação de estilo e ruído ou distorções na imagem. Isso é especialmente útil em contextos onde o texto não está perfeitamente formatado.

**Transformadores:** Para o processamento de documentos com *layouts* complexos, por exemplo tabelas e formulários, esses modelos consideram não apenas o texto, mas também as relações espaciais do que rodeia, relação do texto com a imagem ou outro elemento da tabela. Este mecanismo, Auto atenção (*Self-attention*), permite analisar o contexto global e é capaz de compreender essas relações espaciais.

**Processamento de linguagem natural (PLN):** Após a extração do texto pelo OCR, o PLN é utilizado para entender o contexto, identificar possíveis erros e sugerir correções. Por exemplo, se uma palavra for reconhecida incorretamente, o PLN pode utilizar o contexto ao redor para sugerir a palavra correta.

## 2.4. Transformador

O artigo "*Attention is All You Need*" [11], publicado em 2017, revolucionou o campo do PLN ao introduzir um modelo capaz de processar grandes volumes de texto com alta eficiência. Além disso, a abordagem permite o processamento paralelo dos dados, o que acelera significativamente o tempo de treino.

O processo de auto-atenção inicia-se com a receção de um *input*, como uma frase em que cada palavra é representada por um *token* associado a um vetor, sendo representado por $\vec{E}_n$, obtido pelo processo *Embedding*. O conjunto dos vetores do *input* formam uma matriz que se chama bloco de atenção, permitindo que os vetores comuniquem e passem informação entre si. Dessa forma, cada *token* consegue saber o contexto em que está inserido. Compreender o contexto é fundamental, pois a mesma palavra pode ter significados diferentes dependendo de como é utilizada. Por exemplo, a palavra "banco" pode se referir tanto a uma instituição financeira quanto a um objeto para sentar.

De seguida, para cada palavra é criado um novo vetor que faz uma espécie de questionamento sobre a existência de uma relação com as outras palavras. Esse vetor é chamado de *query* e é representado da seguinte forma: $W_Q$.

Assim para verificar a relação entre as palavras, é efetuada a multiplicação entre o *token* ($\vec{E}_n$) pela *query* da seguinte forma:

$$\vec{Q}_n = \vec{E}_n \cdot W_Q. \tag{2.1}$$

Para cada *token*, também é criado outro vetor chamado *Key*, que pode ser entendido como uma possível resposta à pergunta representada pela *Query*. O objetivo é verificar se existe uma forte relação entre a *Key* e a *Query*, avaliando o grau de correspondência entre ambos. A *Key* pode ser representada da seguinte forma:

$$\vec{k}_n = \vec{E}_n \cdot W_k \tag{2.2}$$

Uma vez criadas as *Queries* e *Keys*, o valor obtido a partir da multiplicação entre os vetores *Query* e *Key* indica o nível de relação entre essas palavras. Se o resultado for um número positivo e alto, significa que há uma forte relação entre si. Por outro lado, se o valor for baixo ou negativo, isso indica que a relação entre as palavras é fraca ou inexistente.

Como o modelo aprende a tentar prever a seguinte palavra, na matriz de atenção não se pretende calcular a relação entre uma palavra e as palavras que aparecem depois, pois forneceria a resposta antecipadamente. Ou seja, na aprendizagem, as palavras futuras podem influenciar as palavras anteriores, portanto, atribuí-se valores de $-\infty$ para garantir que essas relações não sejam consideradas. Isso ajuda o modelo a aprender de forma mais eficaz. Este processo chama-se *masking* e é usado para garantir que certas informações (como palavras futuras) não influenciem o cálculo da atenção e é aplicado antes da normalização.

Os valores calculados na matriz de atenção podem variar entre $-\infty$ e $\infty$. Para transformar esses valores em uma distribuição de probabilidade, é necessário aplicar uma normalização que restrinja os valores entre 0 e 1, garantindo que a soma de todos os valores associados a um *token* seja igual a 1. Para realizar a normalização, pode utilizar-se a função *softmax*. A fórmula para aplicar a normalização com *softmax* é a seguinte:

$$\text{Atenção}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.3}$$

TABELA 2.1. Padrões de atenção sem normalização e com normalização

| Sem normalização | | | | |
|---|---|---|---|---|
| +3.53 | +0.80 | +1.96 | +4.48 | -3.74 |
| -0.30 | -0.21 | +0.82 | +0.29 | -1.95 |
| -0.00 | -0.59 | +0.67 | +2.99 | -0.41 |
| -2.00 | -1.00 | +1.00 | +3.00 | -1.48 |
| +2.00 | -3.00 | +3.07 | +2.94 | -0.31 |

$\xrightarrow{softmax}$

| Com normalização | | | | |
|---|---|---|---|---|
| 0.25 | 0.25 | 0.25 | 0.25 | 0.00 |
| 0.00 | 0.25 | 0.25 | 0.25 | 0.25 |
| 0.00 | 0.00 | 0.11 | 0.12 | 0.77 |
| 0.00 | 0.00 | 0.00 | 0.48 | 0.52 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.25 |

Por fim, é necessário calcular o *value* que é obtido através da *value matrix* $W_v$ multiplicado pelo *token* assim:

$$\vec{V_n} = \vec{E_n} \cdot W_v \tag{2.4}$$

Através do somatório, $\sum \vec{E_n}$ é possível refinar o próprio *token* com o *value*, concluindo o processo que se chama chefe de atenção (*Head of attention*) aplicando a seguinte formula:

$$\vec{E'_n} = \vec{E_n} + \sum \vec{E_n} \tag{2.5}$$

Para finalizar, o contexto de uma palavra pode influenciar de várias maneiras diferentes. Portanto, é necessário repetir esse processo várias vezes para capturar diferentes associações. Esse procedimento é conhecido como *Multi-Headed Attention*, que permite processar múltiplas associações em paralelo com distintas *keys, values* e *querys*.

## 2.5. Aperfeiçoamento do modelo (*Fine-tuning*)

O aperfeiçoamento do modelo é uma técnica usada em aprendizagem automática, a qual permite aproveitar um modelo pré-treinado e ajustá-lo para um cenário específico, em vez de criar um novo modelo do zero. Essa abordagem é baseada no conceito de transferência de conhecimento, onde o conhecimento adquirido por um modelo anterior é transferido e refinado conhecimento para uma nova tarefa.

O aperfeiçoamento do modelo é utilizado por diversos motivos, sendo uma abordagem eficiente em várias situações.

- Redução no tempo e recursos computacionais associados ao treino do modelo, pois não será necessário treiná-lo de raíz.
- Redução de dados para o treino, ou seja, em vez de criar um modelo de raiz com novo conjunto extenso de dados, com o aperfeiçoamento do modelo, é possível alcançar bons resultados com uma substancialmente menor de dados, pois o modelo já tem conhecimento adquirido de um conjunto dados robusto e assim, só necessita de realizar ajustes para o novo conjunto de dados.
- Desempenho superior em nova tarefa, o que permite personalizar o modelo para um novo caso de uso, melhorando o desempenho em tarefas específicas.

## 2.6. Métricas de avaliação e desempenho

Nesta secção, descrevem-se as principais métricas de avaliação utilizadas em classificação de texto, como a análise da matriz de confusão, Taxa de acerto, Precisão, Cobertura e Pontuação F1. Através dessas práticas, garantimos que o modelo final esteja pronto para lidar com novos textos de maneira robusta e eficaz.

### 2.6.1. Matriz de confusão

A matriz de confusão é uma tabela que apresenta os valores reais e os previstos pelo modelo, permitindo uma análise detalhada do seu desempenho. Através dela, é possível avaliar

a desempenho do modelo em termos de classificações, fornecendo métricas importantes para a compreensão da sua eficácia [12].

A definição de classe positiva refere-se à classe que o modelo está a identificar como verdadeira para um determinado registo, por exemplo, ao identificar que um valor corresponde ao Número da conta bancária internacional (IBAN). Enquanto, a classe negativa corresponde aos casos que o modelo tenta identificar como não pertencentes à classe positiva, como ao determinar que um valor não corresponde ao IBAN.

A matriz é composta por quatro quadrantes que seguem mais abaixo e respetivamente na Figura 2.2:

- **Verdadeiro positivo (VP)**: O modelo previu corretamente que o registo pertence à classe positiva;
- **Verdadeiro negativo (VN)**: O modelo previu corretamente que o registo não pertence à classe positiva;
- **Falso positivo (FP)**: O modelo previu que o registo pertence à classe positiva, mas na verdade este pertence à classe negativa;
- **Falso negativo (FN)**: O modelo previu que o registo pertence à classe negativa, mas na verdade este pertence à classe positiva;

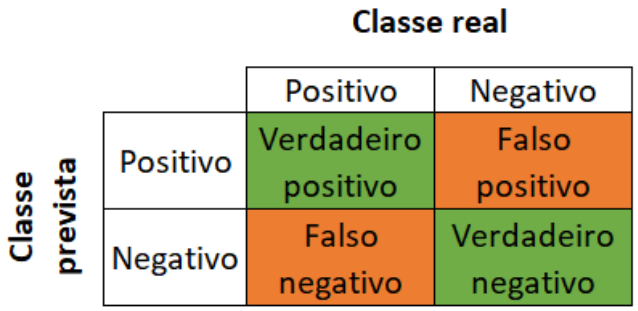



FIGURA 2.2. Matriz de confusão

Na Figura 2.2, a matriz de confusão em que as dimensões correspondem à "classe verdadeira" e à "classe prevista", cada linha/coluna representa o número de classes que o modelo pretende classificar. No Capítulo 5, Procedimentos e Avaliação dos Resultados Experimentais, os resultados são compostos por 20 classes, consequentemente, vamos ter uma matriz de confusão terá uma dimensão de 20 por 20.

Os valores apresentados nas caixas verdes representam a quantidade de vezes que o modelo acertou, enquanto as vermelhas representam as que o algoritmo falhou.

Por fim, a matriz de confusão é uma ferramenta de análise útil para comparar os resultados do modelo com diferentes algoritmos na fase de avaliação, pois permite analisar o desempenho do modelo consoante a classe.

### 2.6.2. Taxa de acerto

A Taxa de acerto é uma das métricas mais comuns para avaliar o desempenho dos modelos, especialmente na classificação de texto. Basicamente, a Taxa de acerto avalia a fração correta de todas as previsões realizadas, sendo calculada através da seguinte forma [13]:

$$\text{Taxa de acerto} = \frac{VP + VN}{VP + VN + FP + FN}$$

É fundamental prestar atenção à métrica de Taxa de acerto, para não se limitar a analisá-la isoladamente. Outras métricas devem ser consideradas, pois mesmo que a Taxa de acerto apresente bons resultados, não garante que o modelo esteja a apresentar um bom desempenho. Isso pode ocorrer, se o conjunto de dados for desbalanceado, se não estiver diversificado com o modelo acertando predominantemente na classe com maior representatividade. Por exemplo, a classe "Outros", que representa 90% do conjunto de treino, o modelo pode acertar na maioria dos casos desta classe, mas falhar nos restantes. Assim, o resultado da Taxa de acerto será bastante elevado, já que o modelo acerta na maioria dos cenários, mas o desempenho nas outras classes, que compõem 10% do conjunto dados, será péssimo. E essas classes podem ser até mais importantes que a classe "Outros". Em suma, neste cenário a Taxa de acerto apresenta um bom resultado, porém o modelo não teve um bom desempenho, falhando nas classes críticas para o contexto do negócio.

### 2.6.3. Precisão

A Precisão indica a proporção de previsões positivas corretas em relação ao total de previsões positivas feitas pelo modelo ($VP + FP$). Ou seja, de todos os positivos, os que são realmente positivos. Segue a sua fórmula [13]:

$$\text{Precisão} = \frac{VP}{VP + FP}$$

Através desta métrica, é possível avaliar se o modelo consegue classificar bem a classe.

### 2.6.4. Cobertura

A Cobertura pretende identificar corretamente as instâncias corretas de uma classe. Ou seja, avalia a proporção de verdadeiros positivos em relação ao total de instâncias da classe. Segue a sua fórmula [13]:

$$\text{Cobertura} = \frac{VP}{VP + FN}$$

A diferença entre Cobertura e Precisão é que a Cobertura pretende identificar todos os casos positivos que pertencem à classe positiva, enquanto a Precisão avalia quantos dos casos classificados como positivos pelo modelo são realmente positivos. Essas duas métricas estão relacionadas, mas, em muitos casos, um alto valor de Cobertura pode resultar em uma precisão mais baixa, e vice-versa, devido ao balanço entre verdadeiros positivos e falsos positivos.

Para encontrar um equilíbrio adequado entre essas duas métricas, utiliza-se a Pontuação F1, que combina Cobertura e Precisão, proporcionando uma avaliação mais equilibrada do desempenho do modelo.

### 2.6.5. Pontuação F1

Como foi referido, o Pontuação F1 permite encontrar um equilíbrio entre as métricas da Precisão e Cobertura que é a média harmónica entre as duas métricas que se pode representar na seguinte [13]:

$$\text{Pontuação F1} = 2 \times \frac{\text{Precisão} \times \text{Cobertura}}{\text{Precisão} + \text{Cobertura}}$$

### 2.6.6. Curva de aprendizagem e Paragem antecipada

Através do gráfico das curva de aprendizagem, pretende-se analisar se o modelo adquiriu realmente conhecimento, em vez de apenas memorizar o conjunto dados de treino, permitindo avaliar a sua capacidade de generalização para novos dados e identificar se está a ocorrer *overfitting* [14].

Também existe o cenário de *underfitting*, que ocorre quando o modelo não está aprender adequadamente os padrões dos dados, resultando num desempenho insatisfatório tanto no conjunto de treino quanto no conjunto de teste. Isto acontece quando a perda no conjunto de treino tem valores elevados.

Durante o processo de treino, o modelo vai ajustando os seus parâmetros de forma minimizar a função de perda. Por outro lado, a perda no conjunto de validação é calculada em um conjunto de dados que não foi utilizado para treinar o modelo, fornecendo uma avaliação imparcial do desempenho do modelo com dados nunca vistos.

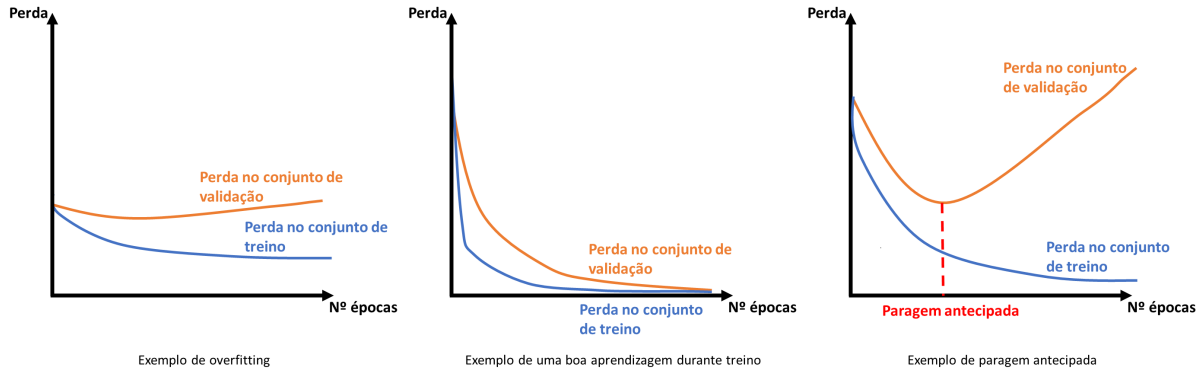



FIGURA 2.3. Exemplos de curvas de aprendizagens

Quanto maior for a confiança e a correção das predições feitas pelo modelo, menor será a perda no conjunto de treino, mas isso por si só não significa que o modelo esteja a generalizar bem o que aprendeu ao realizar predições para novos dados.

Neste cenário, no primeiro gráfico da Figura 2.3 é possível verificar que está acontecer *overfitting*, a perda no conjunto de treino encontra-se a diminuir, enquanto a perda no conjunto de validação está a aumentar, isto significa que não está a generalizar bem para novos dados.

Quando maior for a distância entre a perda no conjunto de validação e a perda no conjunto de treino, menor robustez apresentará o conjunto dados de treino. É imprescindível que o conjunto dados seja, realmente, robusto com uma com dimensão suficiente para que o modelo obtenha bons resultados através do conjunto dados de validação. Quando a diferença é pequena, isso, geralmente, significa que o modelo está apresentar um bom desempenho tanto nos dados de treino quanto nos de validação.

Por outro lado, no segundo gráfico da Figura 2.3, o modelo apresenta um desempenho razoável, com a diferença entre a perda no conjunto de validação e a perda no conjunto de treino a diminuir ao longo das épocas, até se tornar mínima.

O eixo das abcissas representa épocas. Uma época corresponde a um ciclo completo de treino do modelo, no qual todos os dados do conjunto de treino foram processados. Assim, na fase de aprendizagem são realizadas múltiplas épocas para o modelo para que este aprenda corretamente os padrões dos dados. Assim sendo, a quantidade ideal de épocas pode variar dependendo da complexidade do modelo e dos dados.

Um outro conceito importante é a paragem antecipada. A paragem antecipada é uma técnica que se baseia na monitorização do desempenho do modelo com o conjunto de dados de validação durante o treino e interrompe o treino quando não existem melhorias significativas nesse conjunto. Assim é necessário definir um número de épocas para que o modelo possa aprender. Mas quantos ciclos serão necessários? Se o número de épocas for insuficiente, o modelo pode sofrer de *underfitting*. Por outro lado, se o número de épocas for muito elevado, pode ocorrer *overfitting*.

A avaliação do desempenho do modelo é feita a cada época calculando a perda. Se a perda no conjunto dados de validação começar a aumentar, enquanto a perda no conjunto dados de treino continua a diminuir, isso indica que o modelo pode estar aproximando-se do ponto de paragem antecipada. Quando o número de épocas configurado é atingido ou quando o ponto de paragem antecipada é detetado, o treino é interrompido para evitar o *overfitting*, e o modelo retornado é aquele obtido antes do aumento da perda no conjunto dados de validação [15].

Como se pode observar na Figura 2.3, existem duas curvas que representam a perda do conjunto de treino e do conjunto de validação. Quando estas curvas começam a seguir direções opostas, é um indicativo de que o *overfitting* está a ocorrer, assim o ponto ideal para interromper o treino do modelo é no momento do paragem antecipada, onde a perda de validação ainda está a diminuir antes do erro aumentar devido ao *overfitting*.

CAPÍTULO 3

# Revisão da literatura

Neste capítulo, apresenta-se uma análise à literatura relacionada com o tema abordado. Para tal, foi aplicada a metodologia PRISMA [16], com uma recolha de avaliações, análises e sínteses técnicas a partir de fontes académicas, artigos, livros e conferências sobre o tema em específico, estabelecendo o contexto para o novo estudo e projetos que se assemelham.

## 3.1. Processo de revisão sistemática da literatura

A seguinte revisão da literatura baseia-se na metodologia PRISMA [16]. A estrutura compreende várias fases-chave, incluindo identificação, triagem, avaliação da elegibilidade, extração de dados e síntese, sendo importante garantir a adesão de critérios e diretrizes predefinidos, minimizando assim o enviesamento e melhorando a qualidade da evidência sintetizada a partir da literatura. Neste contexto, a questão a que esta revisão pretende responder é "Qual é o estado da arte dos modelos que extraem informação de faturas"?

Na aplicação da metodologia Prisma, durante a fase de extração de informação, foi utilizada a base da dados do *Scopus*[1].

Como se pode verificar na Figura 3.1, a *query* de pesquisa foi dividida segundo três categorias: o conceito que abrange as tecnologias abordadas no tema da dissertação, aplicação que representava o objetivo da aplicação e, por fim, o contexto que significava o contexto do tema em si. A *query* tinha o objetivo de procurar a interceção das três categorias, ou seja, (Conceito *AND* Aplicação *AND* Contexto), posteriormente era filtrada pelo ano da publicação do artigo e pelo tipo de documento.

| Conceito | Aplicação | Contexto | Limitações |
|---|---|---|---|
| "deep learning" or "neural network" or "computer vision" or "LayoutLM" or "machine learning" or "Natural Language Processing" or "text mining" | "information extraction" or "document understanding" or "understanding" or "extraction" | "invoice" or "bill" | ano: 2017-2024<br>tipo de documento: artigo ou revisão |

FIGURA 3.1. Seleção de palavras chaves

Embora não se tenha encontrado nenhum trabalho especificamente relacionado com fundos europeus, muitos estudos abordam a extração de informação de faturas, utilizando tanto as tecnologias exploradas neste trabalho, como transformadores, bem como outras abordagens semelhantes, como as CNN, entre outros.

No processo de aplicação da metodologia PRISMA, ilustrado na Figura 3.2, já eram conhecidos quatro trabalhos que dois abordam a extração de informação de faturas e os

[1] `https://www.scopus.com/`

outros dois abordam a arquitetura da ferramentas *LayoutLMv2* e *LayoutLMv3* conforme detalhados na Secção 3.3 e na Secção 3.2.

Na fase de identificação, foram encontrados 233 registos únicos. Após a aplicação dos filtros de idioma (inglês), data (2017 a 2024), tipo de documento (artigos e revisões) e área temática (Ciências Informáticas), o número foi reduzido para 156 registos. Em seguida, realizou-se um filtro adicional, analisando-se se o título dos trabalhos estavam adequados ao contexto, o que resultou na exclusão de 92 registos, restando 64 para análise.

Depois, uma análise mais detalhada foi conduzida, examinando os resumos dos trabalhos e, em alguns casos, aprofundando a revisão do conteúdo completo. Dos 64 registos iniciais, 24 registos foram considerados elegíveis. Desses, 8 registos foram excluídos por não abordarem o tema de faturas, 19 registos por abordarem tecnologias fora do âmbito deste trabalho e 13 registos por estarem descontextualizados, não se enquadrando no objetivo da revisão sistemática.

Por fim, foram removidos 18 trabalhos por não serem de acesso gratuito e por não apresentarem valor contributivo relevante, uma vez que abordavam informações semelhantes aos trabalhos já incluídos. Assim, restaram 10 trabalhos, incluindo os 4 identificados por meio de outros métodos, discutidos no Capítulo 3.

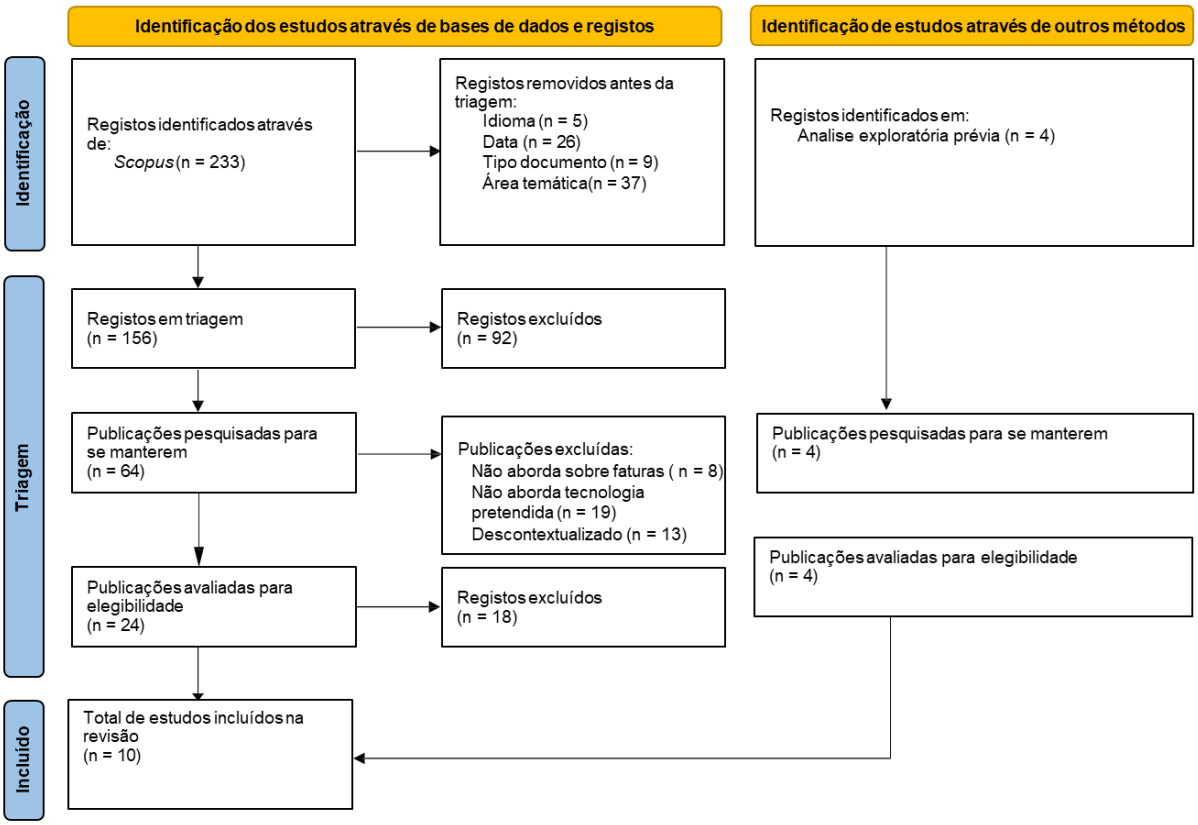



FIGURA 3.2. PRISMA 2020 Fluxograma para revisões sistemáticas

### 3.2. Ferramenta *LayoutLM*

A primeira arquitetura da ferramenta *LayoutLM* foi proposta em 2019 por Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou através do trabalho "*LayoutLM: Pre-training of Text and Layout for Document Image Understanding*" [17]. A ferramenta tinha a finalidade, através de um modelo pré-treinado de texto e por *layout* de extrair informação de documentos de imagens focando-se especificamente na relação entre o texto com o *layout* da imagem onde foi pré-treinado com um conjunto dados com 6 milhões de documentos sendo no total 11 milhões de imagens composto por emails, faturas, orçamentos, publicidade, artigos, apresentações, relatórios, entre outros [18]. Para a fase de avaliação do modelo usaram outros conjunto dados distintos.

A arquitetura da ferramenta *LayoutLM* utiliza o modelo *Bidirectional encoder representations from transformers* (BERT) como o transformador. O BERT foi desenhado para entender o contexto bidirecional de uma palavra numa frase, ou seja, considerando tanto as palavras anteriores como as subsequentes ao processar o texto. Também foi adicionado posição 2-D *embedding* e *image embedding*, técnicas usadas para representar espaço espacial e a imagem para vetores numéricos.

O que revolucionou neste trabalho foram as funcionalidades *Document Layout Information* e *Visual Information* [17]. A *Document Layout Information* consegue relacionar um *token* numa posição da imagem e, assim, consegue criar um padrão nas classes que estavam em posições específicas no documentos, como, por exemplo, o número do Passaporte, que se encontra sempre a um dos cantos da imagem. Também a *Visual Information*, consegue relacionar a importância ou prioridade do conteúdo, como, por exemplo, o texto encontrar-se em sombreado ou itálico.

Posteriormente, foram melhorando a ferramenta para *LayoutLMv2* e recentemente para *LayoutLMv3* que é a que vai ser utilizada no âmbito desta dissertação.

#### 3.2.1. Ferramenta *LayoutLMv2*

O *LayoutLMv2* apresenta uma estrutura semelhante à do *LayoutLM*, mas com algumas melhorias que resultam em um desempenho significativamente superior em relação ao *LayoutLM*. Por exemplo através do aperfeiçoamento do modelo com o conjunto dados *FUNDS* [19], um conjunto dados composto por faturas, para ambas as ferramentas os resultados na métrica *Pontuação F1* foram de 0,7895 vs 0,8420 sendo a diferença de 0,0525 [20].

A diferença entre o *LayoutLMv* e o *LayoutLMv2* é que o visual *embedding* é combinado com o aperfeiçoamento do modelo, aplicando uma arquitetura de transformador para aprender de uma forma *cross-modality*, isto é, a interação entre visual e o texto informação. Também se aplica mecanismo de auto atenção espacial, este mecanismo foi desenhado para ser mais eficiente na deteção de relação entre *tokens* e *visually-rich documents*. Ao contrário do mecanismo de auto atenção, que utiliza a incorporação de posições absolutas, o mecanismo de auto atenção com consciência espacial introduz representações de posições relativas 2-D para *tokens*. Esta abordagem proporciona uma visão mais alargada da

modelação espacial contextual, permitindo que o modelo compreenda melhor a disposição e a estrutura dos documentos através da incorporação explícita de informações espaciais no mecanismo de atenção.

Por último, também tem como nova funcionalidade a *Masked language modeling* (MLM) que seleciona *tokens* aleatórios, "mascarando-os". Isto é, o modelo não sabe o conteúdo do *token* e tem de tentar adivinhar o respetivo conteúdo através do contexto dos outros *tokens*.

A arquitetura do *LayoutLMv2*, que está apresentada na Figura 3.3, consiste numa estrutura *Multi-modal Transformer*, representando a base da ferramenta, que recebe o *input* to texto, visual e *layout* do documento que posteriormente faz o respetivo *embedding*, *Text, visual e Layout embedding* [20].

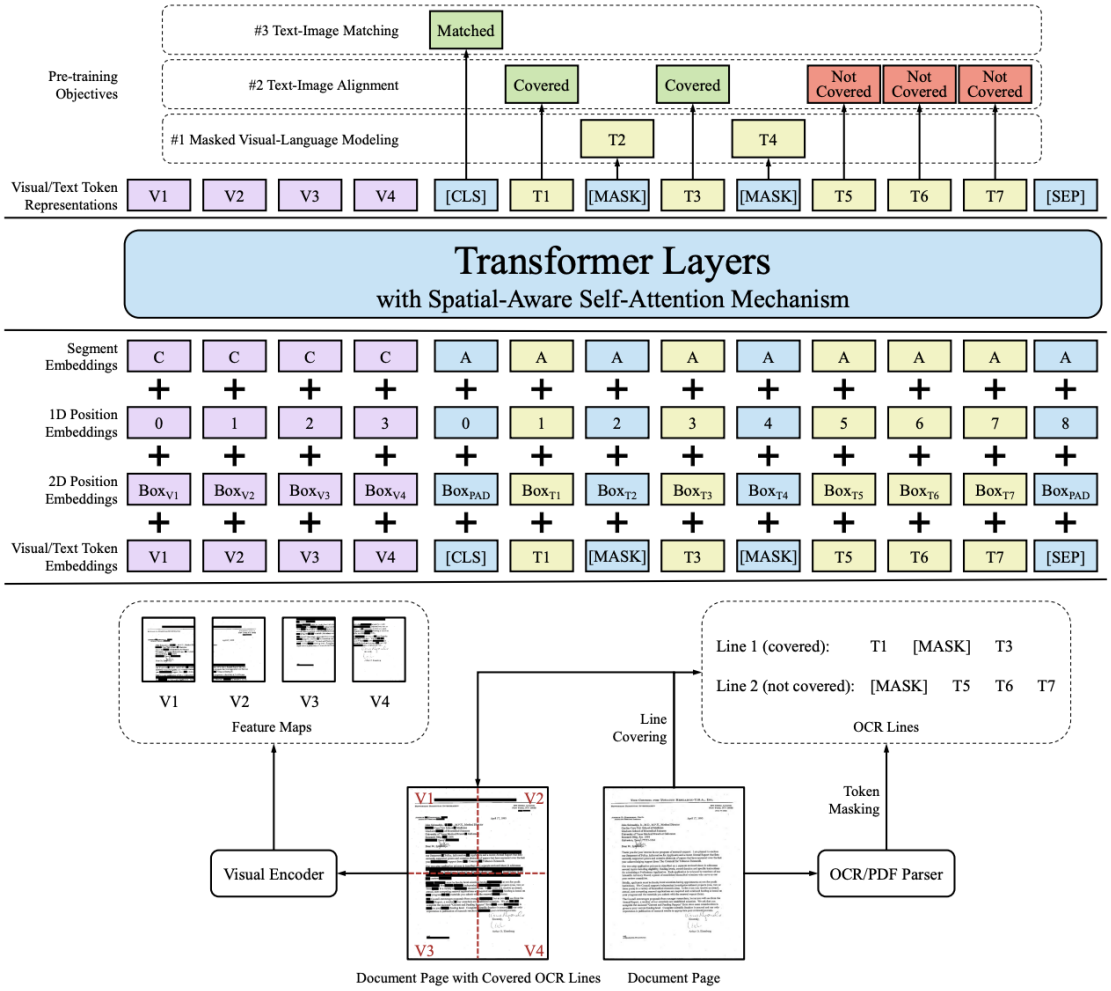



FIGURA 3.3. Arquitetura do *LayoutLMv2* [20]

O algoritmo de tokenização utilizado no *Text embedding* foi o *WordPiece* é uma técnica que divide as palavras em palavras mais pequenas [21], posteriormente adiciona na sequência dos *tokens* os caracteres "[ CLS ]" no *token* inicial e no final o "[ SEP ]", permitindo assim identificar origem e o fim da sequência de *tokens* [20].

O *token* `[CLS]` significa "*Classification*" como abreviatura é inserido no início da sequência de entrada em certos modelos de linguagem, sendo crucial para tarefas de entendimento de sequência. O `[CLS]` tem a finalidade como um agregador da informação da sequência completa, onde captura a essência da relação entre todas as palavras e é usado como a representação global da entrada.

Como o conteúdo da imagem tem bastante informação, não é possível processar tantos *inputs*. Assim é necessário dividir a imagens em pequenas imagens com o mesmo tamanho para que o conteúdo da imagem como se pode analisar na Figura 3.3 Arquitetura do *LayoutLMv2*.

Na fase do *Visual embedding* é utilizada a tecnologia *ResNext-FPN* que é a combinação de duas arquiteturas de redes neuronais: *ResNeXt* e *Feature pyramid network* (FPN).

*ResNeXt* é uma melhoria do *ResNet* que agrega as transformações residuais, isso significa que, em vez de apenas adicionar camadas residuais como na *ResNet*, o *ResNeXt* agrupa várias transformações residuais em paralelo, o que melhora a eficiência e a precisão do modelo [22]. Já o FPN melhora o modelo a extrair recursos em múltiplas escalas, sendo especialmente útil em tarefas como a deteção de objetos, onde os objetos aparecem com tamanhos diferentes.

Por fim, *Layout embedding*, o que revolucionou o *LayoutLMv2*, através do Codificador multimodal com mecanismo de auto-atenção com consciência espacial permite juntar os *embeddings* do texto e o visual para relacionar entre eles garantido que o modelo compreende as relações espaciais entre os elementos, permitindo-lhe interpretar melhor a disposição e a estrutura do documento [20].

### 3.2.2. Ferramenta *LayoutLMv3*

Tal como as versões anteriores, o *LayoutLMv3* utiliza o modelo BERT para sua aprendizagem. O que torna este modelo favorável é a sua bidirecionalidade, que, ao analisar um *token*, considera o seu contexto. Ou seja, verifica se as palavras anteriores e posteriores impactam o *token* em questão.

Para além disto, tem uma abordagem MLM que mascara os *tokens* e tenta adivinhar o seu conteúdo. Isto tudo na fase de aprendizagem do texto [23].

Já na fase de aprendizagem da imagem, para o mascaramento de imagens, o *DocFormer* aprende a reconstruir imagens através *CNN decoder* [24].

O modelo utiliza uma abordagem unificada de mascaramento de texto e imagem durante o pré-treino. Isso significa que ele aplica técnicas de mascaramento tanto em *tokens* de texto quanto em *patches* de imagem para ajudar o modelo a aprender representações multimodais eficazes. Essa abordagem unificada visa melhorar a capacidade do modelo de entender e processar documentos que contêm informações textuais e visuais.

Também utiliza a técnica *Word-patch alignment* (WPA), que tem como finalidade prever o *pacth* da imagem correspondente a um *token* mascarado o que permite aprender a relação entre o *token* e o *patch* [23].

O *Vision transformer* (VIT) é responsável por subdividir a imagem em pequenas imagens, *patches*, como uma sequência de *tokens*, semelhante ao processamento de palavras em um texto. O VIT processa esses *tokens* usando camadas de auto-atenção para capturar relações espaciais e contextuais entre diferentes partes da imagem [25].

Basicamente o *LayoutLMv3* tem como novidade:

- Ser um modelo *Multimodal*, que consegue processar na íntegra todos os tipos de dados de texto e imagens como tabelas, gráficos, entre outros;
- O *LayoutLMv3* não necessita pré-treino de CNN ou *Faster R-CNN* para extrair informação, consequentemente como não utiliza arquiteturas complexas, consegue reduzir o número de parâmetros o que torna mais eficiente não gastando muitos recursos computacionais. Também não existe a necessidade de realizar anotações manuais, como caixas em torno de palavras ou elementos visuais,o que simplifica o treino e a utilização do modelo;
- Para tentar mitigar a discrepância na aprendizagem de texto e imagem, ambas as modalidades são aprendidas em conjunto aplicando as técnicas MLM e *Masked image modeling* (MIM) usadas de forma unificada para que o modelo aprenda de forma mais eficiente tanto com texto quanto com imagens. O MLM é uma técnica para mascarar algumas palavras de texto. Este tenta prever as palavras mascaradas com o objetivo de enetender o contexto das mesmas palavras. MIM onde partes das imagens são mascaradas e o modelo tem de tentar prever o conteúdo da imagem;
- Um novo método, WPA que associa as palavras com os patches onde estão;
- O transformador do *LayoutLMv3* processa texto e imagem ao mesmo tempo;

A arquitetura aplicada no *LayoutLMv3*, apresentada na Figura 3.4, consiste numa estrutura *unified text-image multimodal Transformer*, como foi referido no capítulo anterior para aprender representação de *cross-modal*. O transformador apresenta múltiplas camadas.

Na fase *Text Embedding* foi utilizado o modelo RoBERTA [26]. Cada *token* sabia a posição onde estava localizado na frase e também a posição 2D, coordenadas(x, y) na página (*Layout Position*), no final existe uma normalização nas coordenadas pelo tamanho da imagem (largura e altura).

Também enquanto o *LayoutLMv2* apresentava coordenadas para cada *token*, o *LayoutLMv3* adota uma abordagem onde existe uma coordenada para todos os *tokens* do mesmo nível de segmento, uma vez que as palavras exprimem normalmente o mesmo significado semântico.

Na fase *Image Embedding*, sem utilizar as abordagens clássicas de Redes Neurais Convolucional, utiliza-se o VIT que já foi explicado anteriormente.

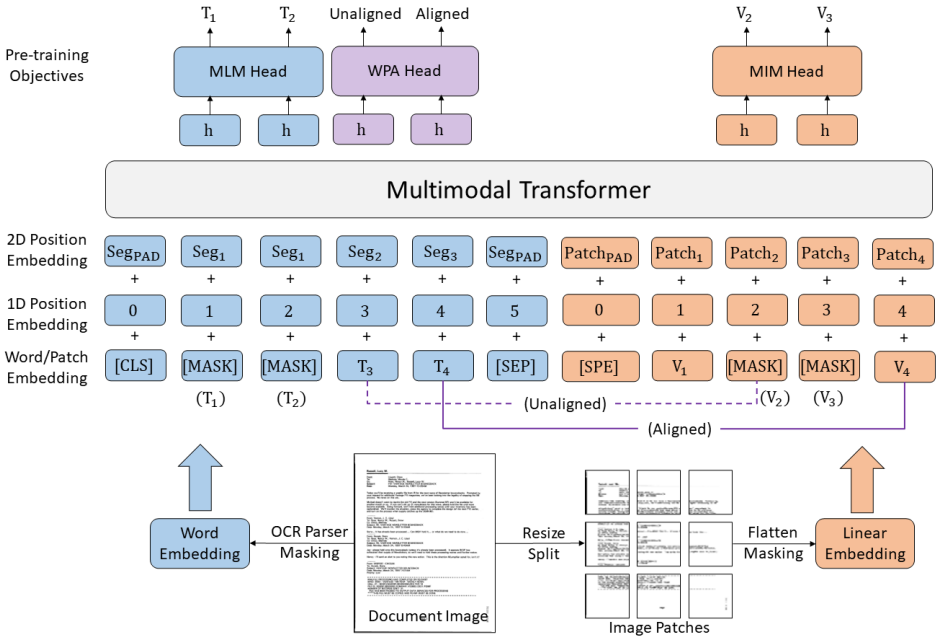



FIGURA 3.4. Arquitetura do *LayoutLMv3* [23]

## 3.3. Trabalhos que abordam extração de informação de faturas

Nesta secção, apresenta-se o projeto do IDA, contextualizando a solução proposta para automatizar a extração de informações e a tecnologia utilizada. Além disso, realiza-se uma análise de projetos que abordam a extração de informações de faturas, com tecnologias do *LayoutLM*.

### 3.3.1. *Intelligent Document Automation - IDA 2.0*

O Intelligent Document Automation (IDA), desenvolvido pela Axians Portugal na unidade de negócios "*Funds & Investment*", tem como principal objetivo extrair automaticamente as entidades-chave de documentos comerciais, como recibos, contratos, faturas, entre outros. Por exemplo, informações sobre o NIF do beneficiário, IBAN, preço, descrição do produto, entre outros elementos essenciais, permitindo assim a inserção automática desses dados no sistema ou em bases de dados. Este processo visa automatizar e validar documentos, facilitando o preenchimento automático de informações em formulários, reduzindo a burocracia e validando automaticamente dados como datas, nomes, moradas e entre outros.

A arquitetura subjacente ao IDA utiliza o *LayoutLMv2*, que treinou com uma base de dados contendo 600 imagens de faturas anotadas para a tarefa de classificação, com 20 classes. Do total de dados, 80% foi utilizado para treino, 10% para validação e os restantes 10% para testes.

Esta solução foi desenvolvida especificamente para aplicação no contexto da análise de fundos europeus em Portugal, porém pode ser generalizado para outro contexto.

### 3.3.2. Trabalhos que abordam extração de informação de faturas, comparando abordagens e tecnologias de Inteligência artificial (IA)

Nesta secção, serão discutidos trabalhos que aplicam a tecnologia Reconhecimento ótico de caracteres (OCR) ou a do *LayoutLM* no contexto de extração de informações em faturas ou documentos comerciais.

Os seguintes estudos abordam a tecnologia OCR como abordagem principal para a extração de informações.

No caso de faturas húngaras, o trabalho [27] aplicou um modelo que alcançou bons resultados, tanto com o algoritmo *Decision Tree* como *Random Forest* sendo os resultados bastante semelhantes. Porém, apresenta o mesmo problema com o IDA 2.0, onde existe um conjunto de dados desequilibrado, com a maioria dos dados anotados com a classe "Outros". Neste cenário, 93% das anotações do conjunto dados tinha o valor de "Outros".

Com um idioma diferente, o trabalho que aborda a extração de informação de faturas coreanas [28], apresenta um resultado de 87% na métrica Pontuação F1, porém foi no ambiente muito controlado com um conjunto dados de 375 imagens. Onde existem muitas faturas que se encontravam desfocadas e que foram removidas no processo de treino. A tecnologia utilizada para reconhecimento de texto foi *Scene Text Recognition with a Single Visual ModelScene text recognition with a single visual mode* (SVTR) que utiliza uma abordagem mais tradicional focando-se na divisão da imagem em pequenos *batchs* que aplicam o reconhecimento em separado e não uma lógica sequencial nas palavras, mas sim numa hierarquia. Os caracteres são reconhecidos por *linear prediction* [29].

Outro trabalho [30] que aborda a extração de faturas que apresentem o IVA através da tecnologia *Tesseract*, uma tecnologia de código aberto de OCR compatível com vários idiomas, apresentou resultados promissores. Com o treino de 300 imagens, conseguiu uma Taxa de acerto geral de 96,21% na extração de informações como nome do cliente, nome do banco, nome da rua e telemóvel, entre outros.

Neste trabalho [31], que aborda a extração de informações de faturas, pretende-se obter resultados com uma quantidade reduzida de dados, em comparação com sistemas que exigem grandes conjuntos de dados para aprender, como é o caso dos transformadores. Através de modelos de redes neuronais como NER (aperfeiçoamento do modelo com BERT) e o modelo *CloudScan* [32] pretende-se classificar 8 classes onde obteve bons resultados. Para as classes data e número do documento, ambas tiveram 96% de precisão avaliada pelo NER. Já para o *CloudScan*, a classe de deteção do câmbio teve 99% de precisão, enquanto no NER foi de 90%.

Por fim, este trabalho [33] desenvolveu uma rede multimodal baseada em grafos semânticos, chamada SGFNet, com o objetivo de melhorar a extração de informações em faturas financeiras para idiomas em Inglês e Chinês. Os resultados obtidos demonstraram uma pontuação F1 geral de 93,71% para o idioma inglês e de 96,27% para o idioma chinês.

Os seguintes estudos exploram uma arquitetura de transformadores que aplicam as versões um e dois da ferramenta *LayoutLM* para extração de informações de faturas.

O trabalho sobre o desempenho na extração e classificação de informações em faturas e ordens de pagamento utiliza uma arquitetura de transformadores, especificamente, com a ferramenta *LayoutLM* [34]. O conjunto de dados utilizado para aprendizagem do modelo combina tanto dados públicos quanto privados. Os conjuntos contêm uma grande quantidade de registos. O "*The Business Documents Collection*", um conjunto de dados privado, é composto por 100 mil faturas e 300 mil ordens de pagamento. Esses documentos, em inglês, estão distribuídos entre 70 mil emissores diferentes. No entanto, adotou-se a abordagem de limitar o número de documentos por emissor a um máximo de 50. Para o outro conjunto de dados, foi aplicada a mesma lógica, com 9 mil emissores diferentes, sempre divididos em conjuntos de treino, validação e teste de forma distinta.

A extração de informação concentrou-se em quatro classes com os seguintes valores na métrica Taxa de acerto: informação Fornecedor (93,72%), morada de entrega (91,41%) data de entrega (99,68%) e preço total(90,89%). Os resultados obtidos mostraram melhorias significativas, sem aumento de complexidade ou da quantidade de parâmetros do modelo.

Outro trabalho [35] que aborda a tecnologia do *LayoutLM* compara o desempenho de outros modelos, com um universo de faturas de idiomas variados sendo 955 de idioma em inglês, 76 em alemão e as restantes com outros idiomas, totalizando 1059 faturas. Avaliou-se o desempenho dos seguintes modelos: *BERTgrid*, *Chargrid*, *GAT+BiLSTM-CRF*, *GCN*, *LayoutLM* e *Random Forest*.

Entre todos os modelos, o *LayoutLM* foi aquele que apresentou os melhores resultados, alcançando uma Pontuação F1 macro-média de 88%. Ao analisar o desempenho por classe, o *LayoutLM* também obteve resultados bastante superiores em campos como número da fatura e a data correspondente com valores de 91% e 90%. Para as classes relacionadas com a descrição do produto, preço e quantidade, o *LayoutLM* destacou-se, com pontuações F1 variando entre 75% e 89%, demonstrando uma vantagem significativa em relação aos outros modelos.

Neste trabalho [36], foi realizado um estudo comparativo entre diferentes modelos para extração de informações de faturas, incluindo o *LayoutLM*, *LiLT*, *Donut*, *Yolov8x* e *Yolov5X*. Os resultados mostraram que o *Yolov8x* apresentou o melhor desempenho entre as 13 classes, alcançando uma Pontuação F1 entre 89% a 94%. O *LayoutLM* foi o segundo melhor modelo, obtendo, na maioria dos casos, Pontuações F1 elevadas entre 88% e 93%, no entanto, teve um desempenho inferior em algumas classes, como por exemplo o número do fornecedor, com uma Pontuação F1 de 79%.

CAPÍTULO 4

# Conjunto de dados e treino do modelos

Neste capítulo, serão explicados os conjuntos de dados utilizados para o treino e avaliação do modelo aplicado e do IDA. Além disso, será descrito o processo de treino com a ferramenta *LayoutLMv3*, descobrindo os melhores hiperparâmetros que apresentam os resultados mais adequados dentro do contexto.

## 4.1. Conjunto de dados

O conjunto de dados aplicado no treino do modelo foi desenvolvido pela equipa da Axians Portugal, com anotação manual de todas as palavras. Todos os elementos do conjunto de dados correspondem a faturas inseridas no contexto de fundos europeus. Essas faturas foram disponibilizadas por Autoridades de gestão (AG) de vários programas, ou seja, tratam-se de documentos reais aplicados no âmbito dos fundos europeus. Todas as faturas encontram-se, originalmente em formato `.pdf` e posteriormente convertidas para o formato de imagem, sendo as classes apresentadas na Tabela 4.1.

Todas as classes apresentadas acrescentam valor útil para a análise de candidaturas de fundos europeus, permitindo avançar para a fase de pagamentos aos beneficiários. As classes com maior impacto no negócio são as apresentadas em seguida:

***client_tax_id*:** Representa o número de contribuinte dos beneficiários que ajuda a garantir a integridade do processo de candidatura, evitando fraudes e garantindo que os fundos sejam destinados aos beneficiários corretos;

***invoice_date* e *invoice_no*:** Representa identificação da fatura;

***item_desc*:** Certifica o detalhe como a descrição dos produtos e é verificada em relação aos critérios do programa de fundos europeus, assegurando que os gastos sejam elegíveis e relevantes para o projeto financiado;

***total_price*:** Classes relacionadas ao valor gasto no produto, como *total_price_w_tax* (preço total com impostos) e *total_price_wo_tax* (preço total sem impostos), entre outras: tem importância registar os valores com e sem impostos, permitindo uma análise financeira precisa e a conformidade com as regulamentações fiscais;

***seller* e *seller_tax_id*:** Representa a informação do fornecedor para verificar se a atividade do fornecedor está dentro do âmbito do concurso para os fundos europeus;

O conjunto dados é constituído por 3 conjuntos: treino, validação e teste com um total de 597 imagens, das quais 477 são utilizadas para treino, representando aproximadamente 80% do total da amostra. Para validação, foram usadas 89 imagens, o que corresponde a cerca de 15%. As restantes 31 imagens são destinadas a testes que correspondem a 5%.

Na Tabela 4.1 apresentam-se todas as classes do conjunto dados e a sua respetiva designação.

TABELA 4.1. Designação das classes do conjunto dados

| Class | Designação |
|---|---|
| client | Nome do cliente |
| client_tax_id | Número de contribuinte do cliente |
| iban | International Bank Account Number |
| invoice_date | Data da fatura |
| invoice_no | Número da fatura |
| item_desc | Descrição do produto |
| item_perc_discount | Percentagem do desconto do produto |
| item_perc_vat | Percentagem do produto de IVA (Imposto sobre o Valor Acrescentado) |
| item_qty | Quantidade do produto |
| item_total_price_w_tax | Preço total do produto com imposto |
| item_total_price_wo_tax | Preço total do produto sem imposto |
| item_unitary_price_w_tax | Preço unitário do produto com imposto |
| item_unitary_price_wo_tax | Preço unitário do produto sem imposto |
| other | Outro |
| seller | Nome do fornecedor |
| seller_tax_id | Número de contribuinte do fornecedor |
| total_discount_value | Valor desconto total |
| total_price_w_tax | Valor total com imposto |
| total_price_wo_tax | Valor total sem imposto |
| total_tax_price | Valor total do imposto |

De seguida, pretende-se realizar uma análise das classes do conjunto dados que se foca no número de ocorrências das classes estando dividida pelo conjunto dados de treino, validação e de teste.

A tabela representa em cada linha uma classe, contabilizando o número de ocorrências que aquela classe teve no conjunto dados e consequentemente realizam-se cálculos para saber a percentagem de ocorrências da classe e o número médio da classe por cada fatura.

Como se pode verificar na Tabela 4.2, existem muitas palavras com a classe *other*, neste cenário com 70% das ocorrências. A classe *other* representa todos as palavras que não fazem parte das classes mencionadas na Tabela 4.2, por exemplo notas adicionais que se encontram no rodapé das faturas. Sendo este número bastante elevado, pode levar ao enviesamento dos resultados do modelo de forma geral, pois representa uma grande parte da amostra, que deveria estar equilibrada. Porém, tal não é possível fazer, pois cada fatura varia bastante consoante o fornecedor e existe muita informação que não é crítica para este contexto.

As restantes classes apresentam pouca ocorrência nas faturas com uma percentagem entre 1% a 2%, exceto a classe *item_ dec* que representa a descrição/designação do produto. Isto acontece, pois cada palavra representa uma ocorrência/*token* o que pode levar a um cenário onde uma fatura tenha um produto, que pode ter vários *tokens* na descrição desse produto. O mesmo acontece na classe *iban*, nas faturas por regra geral é apresentado com um IBAN, mas como existem espaços entre os números do IBAN, pode ser dividido em quatro *tokens*, assim pode ter por média quatro ocorrências por fatura.

TABELA 4.2. Distribuição das classes no conjunto dados de treino

| Treino | | | |
|---|---|---|---|
| Classe | Nº de ocorrências | Percentagem ocorrências | Nº médio de ocorrências p/ página |
| client | 1 950 | 1,21% | 4,09 |
| client_tax_id | 649 | 0,40% | 1,36 |
| iban | 2 141 | 1,33% | 4,49 |
| invoice_date | 617 | 0,38% | 1,29 |
| invoice_no | 1 183 | 0,73% | 2,48 |
| item_desc | 20 099 | 12,45% | 42,14 |
| item_perc_discount | 1 381 | 0,86% | 2,90 |
| item_perc_vat | 3 776 | 2,34% | 7,92 |
| item_qty | 2 823 | 1,75% | 5,92 |
| item_total_price_w_tax | 235 | 0,15% | 0,49 |
| item_total_price_wo_tax | 3 261 | 2,02% | 6,84 |
| item_unitary_price_w_tax | 108 | 0,07% | 0,23 |
| item_unitary_price_wo_tax | 3 266 | 2,02% | 6,85 |
| other | 113 742 | 70,48% | 238,45 |
| seller | 2 590 | 1,60% | 5,43 |
| seller_tax_id | 625 | 0,39% | 1,31 |
| total_discount_value | 607 | 0,38% | 1,27 |
| total_price_w_tax | 799 | 0,50% | 1,68 |
| total_price_wo_tax | 895 | 0,55% | 1,88 |
| total_tax_price | 645 | 0,40% | 1,35 |
| **Total:** | **161 392** | **100%** | **338** |

A mesma lógica explicada para o conjunto dados de treino, pode ser aplicado no conjunto de dados da validação e de teste. Apesar de terem número de registos bastante inferiores em comparação com o conjunto dados de treino, a analise é a mesma, apresentando valores bastantes semelhantes sem existir uma diferença significativa na distribuição de classes dos três conjuntos de dados.

TABELA 4.3. Distribuição da classe no conjunto dados de validação

| Validação | | | |
|---|---|---|---|
| Classe | Nº de ocorrências | Percentagem ocorrências | Nº médio de ocorrências p/ página |
| client | 341 | 1,12% | 3,83 |
| client_tax_id | 103 | 0,34% | 1,16 |
| iban | 473 | 1,55% | 5,31 |
| invoice_date | 117 | 0,38% | 1,31 |
| invoice_no | 219 | 0,72% | 2,46 |
| item_desc | 3 934 | 12,91% | 44,20 |
| item_perc_discount | 264 | 0,87% | 2,97 |
| item_perc_vat | 581 | 1,91% | 6,53 |
| item_qty | 433 | 1,42% | 4,87 |
| item_total_price_w_tax | 23 | 0,08% | 0,26 |
| item_total_price_wo_tax | 522 | 1,71% | 5,87 |
| item_unitary_price_w_tax | 22 | 0,07% | 0,25 |
| item_unitary_price_wo_tax | 648 | 2,13% | 7,28 |
| other | 21 632 | 70,99% | 243,06 |
| seller | 465 | 1,53% | 5,22 |
| seller_tax_id | 154 | 0,51% | 1,73 |
| total_discount_value | 120 | 0,39% | 1,35 |
| total_price_w_tax | 147 | 0,48% | 1,65 |
| total_price_wo_tax | 166 | 0,54% | 1,87 |
| total_tax_price | 108 | 0,35% | 1,21 |
| **Total:** | **30 472** | **100%** | **342** |

Na Figura 4.1, apresenta-se um exemplo de fatura com dados mascarados, proveniente de um ambiente relacionado com os fundos europeus. É importante destacar que as

TABELA 4.4. Distribuição da classe no conjunto dados de teste

| Teste | | | |
|---|---|---|---|
| Classe | Nº de ocorrências | Percentagem ocorrências | Nº médio de ocorrências p/ página |
| client | 118 | 1,13% | 3,81 |
| client_tax_id | 38 | 0,36% | 1,23 |
| iban | 170 | 1,63% | 5,48 |
| invoice_date | 38 | 0,36% | 1,23 |
| invoice_no | 92 | 0,88% | 2,97 |
| item_desc | 1 282 | 12,31% | 41,35 |
| item_perc_discount | 36 | 0,35% | 1,16 |
| item_perc_vat | 170 | 1,63% | 5,48 |
| item_qty | 124 | 1,19% | 4,00 |
| item_total_price_w_tax | 1 | 0,01% | 0,03 |
| item_total_price_wo_tax | 151 | 1,45% | 4,87 |
| item_unitary_price_w_tax | 5 | 0,05% | 0,16 |
| item_unitary_price_wo_tax | 147 | 1,41% | 4,74 |
| other | 7 634 | 73,30% | 246,26 |
| seller | 186 | 1,79% | 6,00 |
| seller_tax_id | 36 | 0,35% | 1,16 |
| total_discount_value | 35 | 0,34% | 1,13 |
| total_price_w_tax | 53 | 0,51% | 1,71 |
| total_price_wo_tax | 57 | 0,55% | 1,84 |
| total_tax_price | 42 | 0,40% | 1,35 |
| **Total:** | **10 415** | **100%** | **336** |

faturas possuem inúmeras estruturas diferenciadas e essa diversidade é fundamental para garantir que o conjunto de dados seja representativo do universo das faturas. No entanto, quanto maior a variedade no conjunto de dados, mais complexo se torna o problema, pois é necessário um conjunto de dados extenso para alcançar essa representatividade. Dessa forma, embora um conjunto de dados mais diversificado tenha maior potencial de generalização poderá ser mais difícil aprender todos os padrões distintos.


FIGURA 4.1. Exemplo de uma fatura no contexto de FE

## 4.2. Processo de treino

Esta Secção aborda as configuração utilizadas no treino do modelo e a respetiva justificação através de inúmeras experiências que abordarão parâmetros essenciais, as técnicas de regularização e os métodos otimizados implementados bem como as limitações técnicas. Estas configurações são essenciais para garantir que o modelo tem o desempenho mais elevado possível, o que permitirá generalizar de forma adequada para cenários reais tentando-se mitigar eventuais problemas de *overfitting* ou *underfitting*.

Será abordada a arquitetura do modelo, os hiperparâmetros definidos desde a taxa de aprendizagem, tamanho do lote de processamento, épocas, função de ativação, métricas de avaliação do modelo, existindo igualmente uma análise da função de perda, qual é o melhor algoritmo de otimização, quais são as técnicas de regularização, tais como *dropout* e, por último, qual a estratégia para a paragem antecipada (*Early Stopping*).

Essas configurações foram ajustadas após testes preliminares, com o objetivo de maximizar a desempenho do modelo no conjunto de teste, mantendo sua capacidade de generalização para dados não vistos.

Todas as métricas gerais aplicadas nas experiências foram calculadas utilizando a média micro, incluindo a Pontuação F1, Cobertura, Precisão e Taxa de acerto. Essa abordagem considera cada ocorrência individual, independentemente da classe, somando os VP, FP e FN de todas as classes para calcular as métricas com base nesses totais agregados. Os benefícios da média micro são especialmente úteis em conjuntos de dados não balanceados, pois esse método permite uma avaliação mais representativa do desempenho geral do modelo.

Para as experiências foram usados dois modelos pré-treinados do *LayoutLmV3*, o $LayoutLMv3_{BASE}$ e o $LayoutLMv3_{LARGE}$. Enquanto o $LayoutLMv3_{BASE}$ é um modelo mais leve com menos parâmetros, composto por 12 camadas neuronais, o $LayoutLMv3_{LARGE}$ apresenta 24 camadas, com maior capacidade de capturar informações complexas apresentando uma aprendizagem mais robusta.

Na construção de um modelo de classificação, a escolha adequada dos hiperparâmetros desempenha um papel fundamental no desempenho final do modelo. Os hiperparâmetros são variáveis ajustadas antes do início do treino, uma vez que estão definidos fora do processo de aprendizagem. Eles controlam o comportamento dos algoritmos de aprendizagem de máquina, influenciando aspetos como a complexidade do modelo. Esses parâmetros, por exemplo, a taxa de aprendizagem, o número de épocas em redes neuronais, afetam diretamente o desempenho e a capacidade de generalização do modelo.

### 4.2.1. Taxa de aprendizagem

A primeira experiência tem a finalidade de descobrir a melhor taxa de aprendizagem. Realizaram-se múltiplas experiências com taxas de aprendizagem distintas tanto para o modelo $LayoutLMv3_{BASE}$ e $LayoutLMv3_{LARGE}$ e apresentam-se os resultados que se podem observar na Tabela 4.5.

TABELA 4.5. Experiências com taxas de aprendizagens para $LayoutLMv3_{BASE}$ e $LayoutLMv3_{LARGE}$

| | | $LayoutLMv3_{BASE}$ | $LayoutLMv3_{LARGE}$ |
|---|---|---|---|
| $3e{-}3$ | Pontuação F1 | 14,35% | 14,35% |
| | Cobertura | 8,43% | 8,43% |
| | Precisão | 48,31% | 48,31% |
| | Taxa de acerto | 90,31% | 90,31% |
| $1e{-}4$ | Pontuação F1 | 78,95% | 14,35% |
| | Cobertura | 82,74% | 8,43% |
| | Precisão | 75,49% | 48,31% |
| | Taxa de acerto | 98,04% | 90,31% |
| $3e{-}5$ | Pontuação F1 | **82,52%** | 81,38% |
| | Cobertura | **84,71%** | 83,13% |
| | Precisão | **80,44%** | 79,69% |
| | Taxa de acerto | **98,76%** | 98,84% |
| $5e{-}5$ | Pontuação F1 | 77,77% | 15,66% |
| | Cobertura | 78,39% | 10,10% |
| | Precisão | 77,43% | 47,20% |
| | Taxa de acerto | 99,00% | 90,22% |

Relativamente à Tabela 4.5 optou-se por escolher as quatro métricas para análise da melhor taxa de aprendizagem, Pontuação F1, Cobertura, Precisão e Taxa de acerto. A tabela foi calculada com dados do conjunto de dados de teste e as métricas são representadas pela média micro de todas as classes.

Como o conjunto dados é composto pela maioria dos dados representados numa só classe, *other*, que não é crítica para o contexto de análises de faturas de fundos europeus, a métrica Taxa de acerto que tem valores altos, não pode ser considerada. Já a métrica Pontuação F1 que é a média harmónica entre a Precisão e a Cobertura, permite visualizar um equilíbro na avaliação do modelo. A cobertura é uma boa métrica para determinar a melhor taxa de aprendizagem, pois avalia a proporção de exemplos positivos que foram corretamente classificados como positivos.

As hipóteses escolhidas para a melhor taxa de aprendizagem foram baseadas no trabalho "*LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking*" [23] que na fase de pré-treino do *LayoutLMv3*, para o $LayoutLMv3_{BASE}$ a taxa de aprendizagem foi de $1e{-}4$, sendo uma taxa relativamente baixa, o que ajuda o modelo a fazer ajustes pequenos e precisos. Já para o modelo $LayoutLMv3_{LARGE}$, aplicou uma taxa de aprendizagem de $5e{-}5$, indicando que, para o modelo maior, os ajustes nos parâmetros são feitos de forma mais gradual. Já as outras duas hipóteses de taxa de aprendizagem foram baseadas no modelo do IDA que eles aplicaram no seu treino.

As piores taxas de aprendizagem observadas foram de $3e{-}3$ para o $LayoutLMv3_{BASE}$ e para o $LayoutLMv3_{LARGE}$, além de $1e{-}4$ apenas para o $LayoutLMv3_{LARGE}$. Nesses três cenários, os resultados foram iguais, pois o modelo classificou todas as palavras como se pertencessem à classe *other*.

A melhor taxa de aprendizagem encontrada pelas experiências efetuadas foi a de $3e{-}5$, tanto para o $LayoutLMv3_{BASE}$ como o $LayoutLMv3_{LARGE}$ sendo o melhor resultado na métrica da Cobertura com 84,71% no$LayoutLMv3_{BASE}$ e 83,14% no $LayoutLMv3_{LARGE}$. Enquanto para $1e{-}4$ teve resultados semelhantes, mas foram inferiores onde para o Pontuação F1 teve um resultado de 78,95% contra 82,52% para a $3e{-}5$.

O modelo $LayoutLMv3_{LARGE}$ apresentou resultados inferiores para a taxa de aprendizagem $5e{-}5$ que utilizaram no trabalho "*LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking*" [23]. Este desempenho inferior pode ser atribuído ao fato de o conjunto dados ser pequeno, o que o modelo $LayoutLMv3_{LARGE}$ não estava preparado para aprender.

### 4.2.2. Paragem antecipada e número de épocas

Nesta fase, é fundamental determinar o número de épocas necessárias para que o modelo consiga aprender de forma eficaz. A cada época, o modelo ajusta os pesos do transformador. Se esses pesos forem alterados continuamente em todas as épocas, o desempenho do modelo pode ser prejudicado. Sendo importante identificar a última época ideal, para evitar que o modelo aprenda de menos, *underfitting*, ou aprenda em excesso, *overfitting*.

Inicialmente, não foi aplicada a técnica de paragem antecipada para analisar se está acontecer *overfitting* ou *underfitting*. Assim ao forçar o treino até 60 épocas, analisam-se as funções da perda no conjunto de treino e da validação, calculando-se a distância entre ambas. Diversas experiências foram realizadas, mantendo sempre resultados consistentes com o comportamento das funções, conforme pode ser observado na Figura 4.2.

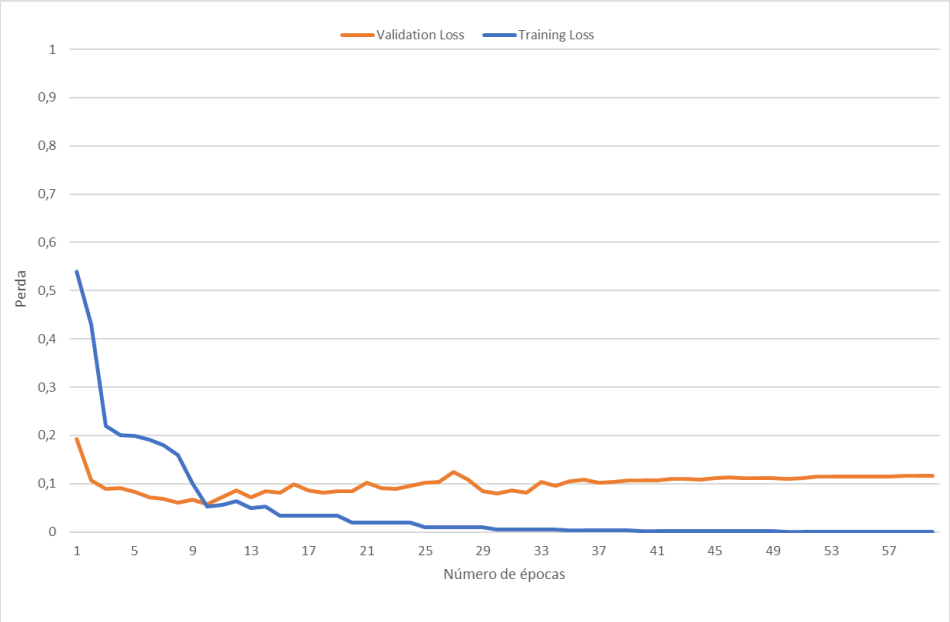



FIGURA 4.2. Função de perda no conjunto de treino e de validação

Na Figura 4.2 é representado um gráfico de perda ao longo do tempo da aprendizagem do modelo, com duas linhas:

- *Training Loss* (azul): A perda observada durante o treino do modelo.
- *Validation Loss* (laranja): A perda observada durante a validação do modelo.

A perda na validação começa a apresentar pequenas variações, atingindo seu valor mais baixo na trigésima época, com 0,080074 sendo a distância mais curta da perda no conjunto de treino com validação que foi de 0,074374. No entanto, é percetível a possibilidade de existir *overfitting*, pois a perda no conjunto de treino está muito baixa, enquanto a perda no conjunto de validação não mostra melhorias significativas, permanecendo entre 0,08 e 0,09. Ou seja, como a perda no conjunto de treino estabilizou em valores bastante reduzidos quase a chegar ao zero, o que sugere que o modelo se ajustou bem aos dados de treino, talvez até excessivamente, pois depois não consegue generalizar bem para novos dados.

Neste momento, a técnica de paragem antecipada entra em ação para avaliar se faz sentido interromper o treino mais cedo, seja na nona, vigésima ou trigésima época, quando a perda no conjunto de validação atinge seu valor mais baixo. O objetivo é entender se o modelo está a sofrer de *overfitting* ou se permitir mais épocas de treino poderia trazer bons resultados, ao prolongar o ajuste sem comprometer o desempenho.

Foi aplicada a técnica de paragem antecipada para evitar o *overfitting* e otimizar o tempo de treino do modelo. No entanto, a implementação não obteve os resultados esperados, pois o treino estava a parar de forma prematura, por volta da oitava e nona época. Esse comportamento indicava que, apesar da validação inicial sugerir uma estagnação no desempenho do modelo, podia existir ainda espaço para melhorar os parâmetros nas épocas subsequentes.

Assim, para saber se a paragem antecipada traria melhorias para o desempenho do modelo foi realizada uma experiência onde em 30 épocas de treino, se detetou o modelo em que a função de perda teve o resultado mais inferior na validação e comparou-se com o modelo na última época, na época 30, com o conjunto dados de testes para validar que modelo apresentou melhor resultado e tentar perceber se uma paragem antecipada traz bons resultados em relação ao da última época, parecendo que está a acontecer *overfitting*.

Seguem os resultados através do conjunto de dados de teste aplicando média micro na Tabela 4.6 onde se compara o modelo da sétima época foi a que apresentou a perda mais baixa na validação com 0,0518 em relação ao modelo da última época, época 30.

TABELA 4.6. Experiências entre a época com a perda mais baixa Vs com a ultima época

| Tamanho batch treino | Pontuação F1 | Cobertura | Precisão | Taxa de acerto |
|---|---|---|---|---|
| Época 7 (menor perda) | 74,73% | 75,68% | 73,80% | 98,94% |
| Época 30 | **81,76%** | **82,54%** | **81,74%** | **98,72%** |

Como se pode verificar na Tabela 4.6, apesar da perda ser inferior ainda é demasiado prematuro parar o treino, pois os resultados apresentados nos testes são inferiores relativamente à última época sendo uma diferença ainda substancial, por exemplo na métrica

Pontuação F1 o resultado para a sétima época era de 74,73% enquanto para a trigésima época foi de 81,76% apresentando uma diferença de 7,03 pontos percentuais.

Já em relação à função da perda sobre esta experiência, no final das épocas, ambas se estabilizam. A perda no conjunto de treino continua quase nula, enquanto a perda no conjunto de validação permanece entre 0,08 e 0,10, sem mostrar melhorias.

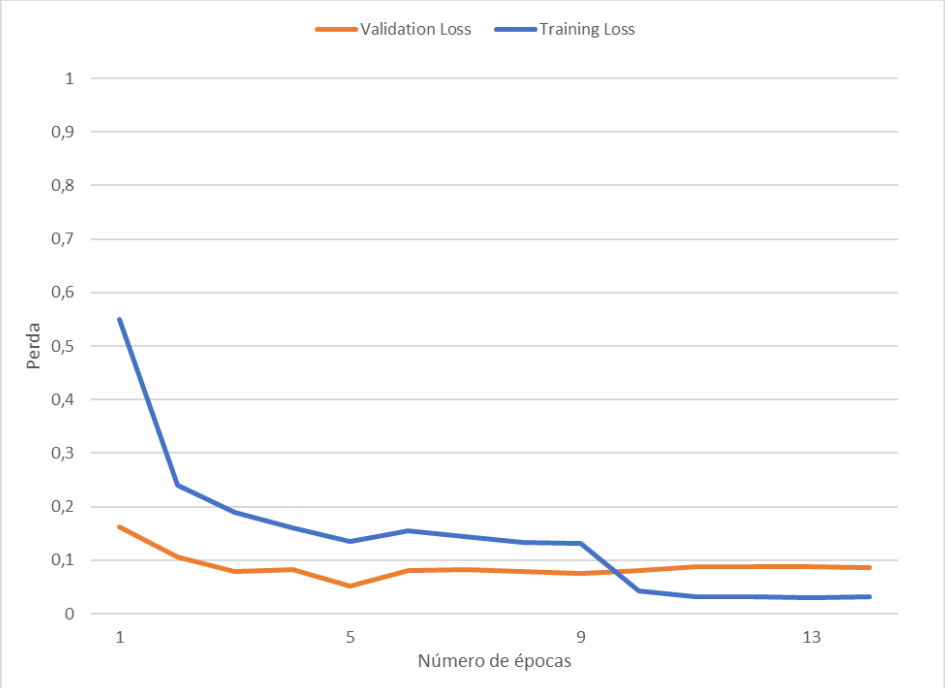



FIGURA 4.3. Função da perda durante a experiência entre a época com menor perda Vs com a ultima época

Podemos concluir que, que apesar de não existir muita diferença entre o modelo com a menor perda ou o da última época, os resultados são ambos positivos. No contexto, as faturas para os três conjuntos dados apresentam a mesma estrutura com pouca variação, seguindo um padrão fixo e repetitivo. A consistência na estrutura das faturas significa que o modelo consegue capturar bem as características importantes do conjunto de dados, sem a necessidade de se ajustar a detalhes excecionais. Dessa forma, mesmo que o modelo apresente sinais de *overfitting*, ele não tem um impacto tão significativo, pois a pouca diversidade nas faturas reduz o risco de o modelo perder a capacidade de generalização para novos dados.

### 4.2.3. *Dropout*

Como foram observados indícios de *overfitting* nas funções de perda tanto no conjunto treino quanto na avaliação, conforme discutido anteriormente, decidiu-se aplicar a técnica de *dropout*. O *dropout* é uma abordagem eficaz para prevenir o *overfitting*, que ocorre quando o modelo se ajusta excessivamente aos dados de treino, mas não generaliza bem para novos dados.

A aplicação do *dropout* visa melhorar a capacidade do modelo de generalizar para novos dados, reduzindo a diferença de desempenho entre os conjuntos de treino e teste. Isso é alcançado por meio da desativação aleatória de neurónios durante o treino, impedindo que esses neurónios contribuam para as ativações das camadas subsequentes e para o processo de *backpropagation*.

Ao desativar neurónios de forma aleatória, o *dropout* força a rede a não depender excessivamente de neurónios específicos, promovendo uma aprendizagem mais robusta e distribuída, o que contribui para uma melhor generalização do modelo.

Foram realizadas quatro experiências: a primeira sem técnica *dropout*, seguido de testes com *dropout* de 10%, 30% e, por fim, 50% sendo o valor da percentagem indicativo da proporção de neurónios que serão desativados. Os resultados dessas experiências com o conjunto de dados de testes podem ser analisados na Tabela 4.7.

TABELA 4.7. Experiências utilizando a técnica *dropout* com o modelo pré-treinado $LayoutLMv3_{BASE}$

| | Pontuação F1 | Cobertura | Precisão | Taxa de acerto |
|---|---|---|---|---|
| Sem dropout | 81,81% | 82,35% | 80,72% | 98,62% |
| Dropout de 10% | **82,82%** | **85,09%** | **80,06%** | **99,06%** |
| Dropout de 30% | 81,59% | 82,10% | 81,04% | 98,75% |
| Dropout de 50% | 29,17% | 33,72% | 25,71% | 96,42% |

O modelo sem qualquer técnica de *dropout* apresentou uma Cobertura de 82,35% sendo já bastante satisfatório. Entretanto já com 10% de *dropout*, a Cobertura aumentou para 85,09%, indicando uma melhoria no desempenho do modelo em 2,74 pontos percentuais.

Já com o *dropout* de 30% e 50% apresentaram resultados inferiores, sugerindo que este nível de *dropout* pode não ser ideal para este modelo específico, apesar de com o *dropout* de 30% se ter obtido resultados semelhantes com o *dropout* de 10%.

A Tabela 4.7 mostra que a aplicação de *dropout* de 10% é a mais eficaz, proporcionando uma maior Cobertura com 85,09%.

Como verificamos anteriormente, a função de perda sem o uso da técnica *dropout*, é mais suscetível ao *overfitting*, enquanto utilizando abordagem do *dropout* de com 10%, conforme apresentado na Figura 4.4, teve melhores resultado na função da perda em relação à função anterior.

Através da Figura 4.4 é possível analisar que o modelo está aprender de forma eficaz durante o treino, com as perdas no conjunto de treino e validação diminuindo progressivamente, sendo inferiores em relação às ultimas experiências. Esse comportamento é um indicativo positivo de que o modelo está a ajustar-se bem aos dados de treino e conseguindo generalizar adequadamente para os dados de validação.

A perda de validação apresenta valores mais baixos, com uma diferença mínima em relação à perda de treino, o que também é um sinal positivo.

O uso de um dropout de 10% parece contribuir para uma curva de perda mais estável e consistente, sem grandes flutuações na perda no conjunto de validação, o que pode indicar

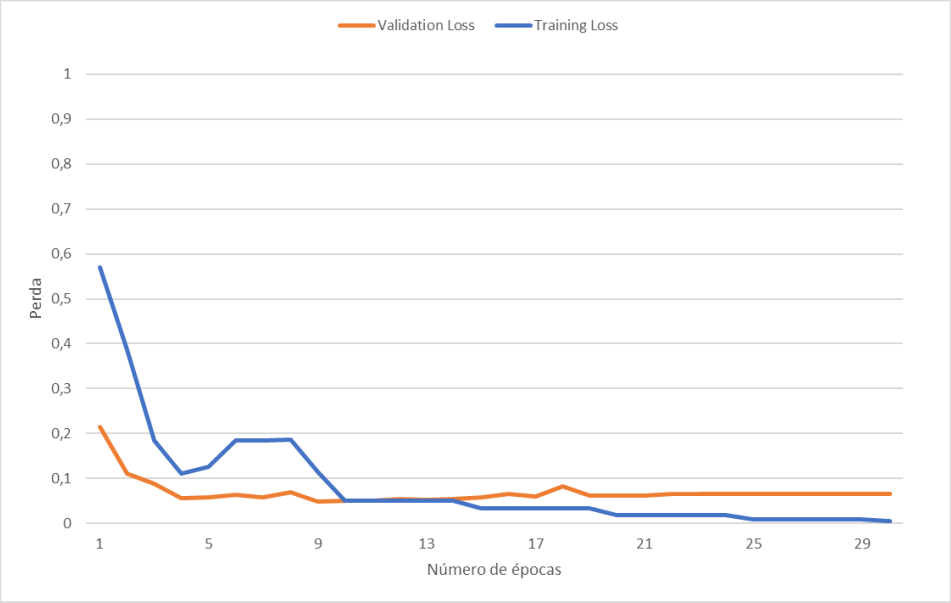



FIGURA 4.4. Função de perda no conjunto de treino e de validação com *dropout* de 10%

uma melhor regularização do modelo. Por outro lado, em comparação com a Figura 4.4 sem aplicação da técnica do *dropout* na Figura 4.2, mostra um aumento acentuado na perda no conjunto de validação, sugerindo possíveis problemas de *overfitting* ou outras anomalias durante o treino.

### 4.2.4. Métrica de avaliação do modelo

Durante o processo de treino é crucial avaliar o desempenho do modelo em dados não vistos, ou seja, dados que não foram utilizados durante o treino. Essa avaliação é frequentemente realizada num conjunto de validação, que serve para medir a capacidade do modelo de generalizar as suas previsões para novos dados. É importante escolher a melhor métrica para fazer a avaliação que pode variar de diferentes contextos e conjuntos de dados.

Esta configuração permite especificar qual a métrica que deve ser monitorizada para determinar o melhor modelo durante o treino através da métrica definida e, ao final do treino o modelo que apresentou o melhor valor para essa métrica será salvo como o melhor modelo.

Como temos um conjunto dados muito específico e com desequilíbrio, onde existe uma classe com uma grande percentagem de ocorrência, a classe *other*. Das 4 métricas, Pontuação F1, Cobertura, Precisão e Taxa de acerto, não pretendemos a que tenha o valor mais elevado no geral, pois a classe *other* está presente em muitas ocorrências e não tem importância para este contexto, Mas sim pretendemos analisar para as classes interessantes a que teve melhor resultado.

Segue a Tabela 4.8 que representa a média das métricas na fase de treino do modelo.

TABELA 4.8. Experiências de métricas de avaliação com $LayoutLMv3_{BASE}$ e $LayoutLMv3_{LARGE}$

| | Pontuação F1 | Cobertura | Precisão | Taxa de acerto |
|---|---|---|---|---|
| $LayoutLMv3_{BASE}$ | 82,82% | 85,09& | 80,06% | **99,06%** |
| $LayoutLMv3_{LARGE}$ | **84,25%** | **85,49%** | **83,04%** | 98,86% |

Apesar de os resultados não demonstrarem muita diferença entre o $LayoutLMv3_{BASE}$ para o $LayoutLMv3_{LARGE}$, podemos verificar que a maior diferença é de na métrica Precisão com 2,98 pontos percentuais.

Das 4 métricas a que teve melhor resultado a nível geral foi, sem sombra de dúvida, a Taxa de acerto com 99,06% de resultado no $LayoutLMv3_{BASE}$, porém não é a melhor escolha para a métrica de avaliação do modelo, mas sim a Cobertura, pois para tanto os dois modelos pré-treinados, teve resultados superiores nas métricas importantes para o negócio como se pode analisar na seguinte Tabela 4.9.

TABELA 4.9. Resultado por classe por cada métrica de avaliação do modelo

| | Pontuação F1 | Cobertura | Precisão | Taxa de acerto |
|---|---|---|---|---|
| iban | 81,25% | 90,32% | 84,84% | 98,55% |
| total_discount_value | 66,66% | 66,66% | 63,63% | 83,33% |
| total_price_w_tax | 54,54% | 66,66% | 63,63% | 60,00% |
| total_price_wo_tax | 41,46% | 69,55% | 53,84% | 48,27% |
| total_tax_price | 77,77% | 80,00% | 77,77% | 77,77% |
| item_qty | 84,21% | 72,13% | 85,71% | 69,81% |
| other | 94,50% | 92,81% | 95,02% | 97,75% |

O modelo para a classe *iban*, apresenta com um Cobertura elevada (90,32%), indicando que consegue identificar corretamente a maioria das instâncias dessa classe. A Pontuação F1 e a Precisão são também altos, o que demonstra um bom equilíbrio entre falsos positivos e falsos negativos.

A classe *total_price_wo_tax* apresenta o desempenho mais baixo entre todas, com uma Pontuação F1 reduzida e uma Taxa de acerto muito baixa (48,27%). A Cobertura elevada indica que o modelo consegue identificar muitas instâncias dessa classe, mas a precisão mais baixa sugere um elevado número de falsos positivos.

Como esperado, a classe *other* apresenta o melhor desempenho geral, dado o seu domínio no conjunto dados. O modelo é altamente preciso e tem um excelente nas quatro métricas para esta classe. No entanto, como esta classe não é a de maior interesse no contexto específico, não deve ser o foco principal da avaliação.

O modelo apresenta bons resultados nas classes de maior interesse, como *iban*, *item_qty* e *total_tax_price*, com Pontuação F1 superiores a 75%. No entanto, há oportunidades de melhoria, especialmente para as classes *total_price_w_tax* e *total_price_wo_tax*.

### 4.2.5. Optimizador

Um optimizador é um algoritmo fundamental no processo de ajuste dos parâmetros de um modelo com o objetivo de minimizar uma função da perda, sendo o optimizador responsável por encontrar os melhores valores de parâmetros que minimizem essa diferença para a função da perda. Dessa forma, a finalidade é garantir que o modelo aprenda de maneira eficiente a partir dos dados, ajustando os seus parâmetros para reduzir a perda ao longo do processo de treino.

Dado o papel crítico do optimizador na convergência e no desempenho final do modelo, é fundamental experimentar diferentes algoritmos para determinar qual deles se adapta melhor às características específicas do problema. Neste secção, aplicou-se testes para *Stochastic gradient descent* (SGD), *Adaptive moment estimation* (ADAM) e AdamW para avaliar qual desses optimizador apresenta o melhor equilíbrio no treino do modelo. A expectativa é que, dependendo das características dos dados e da arquitetura do modelo, o desempenho possa variar, e, portanto, uma análise cuidadosa será realizada para comparar os resultados de cada um, onde os resultados estão apresentados na Tabela 4.10.

Tabela 4.10. Experiências por optimizador

| | Pontuação F1 | Cobertura | Precisão | Taxa de acerto |
|---|---|---|---|---|
| SGD | 27,08% | 23,29% | 31,02% | 96,64% |
| Adam | 79,88% | 82,15% | 77,73% | 98,59% |
| AdamW | **82,25%** | **84,70%** | **80,44%** | **98,76%** |

O SGD é um optimizador básico e amplamente comum onde atualiza os parâmetros do modelo calculando o gradiente da função da perda em pequenos *patch* de dados, o que pode tornar o treino mais rápido em comparação com o *gradient descent* tradicional. No entanto, o SGD pode ter dificuldades em convergir de forma eficiente para estruturas complexas como neste caso, apresentando um resultado bastante inferior de 31,02% na Cobertura.

O ADAM é bastante popular, pois consegue lidar melhor com *gradients* ruidosos e é geralmente uma escolha eficaz para uma ampla gama de problemas.

O AdamW é uma variação do ADAM, foi introduzido para melhorar a forma como a regularização que consegue generalizar o modelo com bastante eficiência. Este optimizador foi utilizado no trabalho "*LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking*" [23], sendo uma escolha estratégica para lidar com a complexidade de tarefas envolvendo modelos pré-treinados com textos e imagens. O AdamW foi o que apresentou melhor resultado, com 84,70% na métrica da Cobertura em comparação com o ADAM que obteve 75,33%.

### 4.2.6. Outros Hiperparâmetros

Outros parâmetros que foram aplicados, mas não têm tanto impacto na aprendizagem do modelo foram:

- Tamanho *batch* de treino: o número de amostras que serão processadas por cada dispositivo, neste caso por uma GPU, durante o treino. Falta capacidade computacional, pelo que não foi possível utilizar 16 *batch*, porém através de experiências utilizando tamanhos de 2, 4 e 8, obtiveram-se os seguintes resultados que estão apresentados na Tabela 4.11.

TABELA 4.11. Experiências por tamanho do *batch*

| Tamanho *batch* treino | Pontuação F1 | Cobertura | Precisão | Taxa de acerto |
|---|---|---|---|---|
| 2 | 82,25% | 84,70% | 80,44% | 98,76% |
| 4 | **84,37%** | **84,70%** | **84,04%** | **98,82%** |
| 8 | 79,00% | 80,78% | 77,29% | 98,34% |
| 16 | - | - | - | - |

O *Batch* de tamanho 4 parece ser o mais eficaz, pois oferece uma combinação superior de Pontuação F1, Precisão e Taxa de acerto, sem queda de desempenho significativa. O desempenho cai quando o *batch* aumenta para 8, o que indica que há um limite no qual aumentar o *batch* pode começar a afetar negativamente o modelo. Portanto, o melhor resultado é com o *batch* de tamanho 4, garantindo uma bom desempenho tanto em precisão quanto em cobertura.

- Tamanho *batch* de avaliação: Número de amostras por dispositivo durante a fase de avaliação. Similar ao tamanho *batch* de treino, que teve o mesmo valor.
- *weight decay*: A taxa de decaimento de peso, usada para regularização e evitar *overfitting*, penalizando pesos muito grandes. O valor 0.01 indica uma penalização suave.

CAPÍTULO 5

# Teste dos modelos

Neste capítulo, abordar-se-á a análise dos resultados do conjunto de dados de teste, avaliando o desempenho do modelo tanto de forma geral quanto por classe através das métricas de avaliação, por fim, realizou-se uma análise comparativa com o modelo do IDA a fim de determinar qual apresentou o melhor desempenho e identificar que classes apresentaram melhor desempenho.

## 5.1. Avaliação dos resultados

Após realizar uma análise para determinar os melhores hiperparâmetros, optou-se pela utilização do modelo pré-treinado $LayoutLMv3_{BASE}$. Embora outros modelos tenham apresentado resultados similares, este mostrou-se consistentemente superior, mesmo que a diferença estivesse em poucas casas decimais. Essa vantagem, ainda que subtil, foi considerada relevante, principalmente devido à sua consistência em diferentes conjuntos de dados de testes.

Na sequência, apresenta-se uma explicação detalhada dos resultados obtidos com o modelo implementado. A avaliação será conduzida com base em métricas Pontuação F1, Cobertura, Precisão, Taxa de acerto e a função de perda. Estas métricas fornecerão uma visão abrangente do desempenho geral do modelo.

Além disso, apresenta-se uma análise detalhada pelas classes individualmente, utilizando a matriz de confusão para observar as taxas de erros em diferentes categorias. Isso permitirá entender de forma pormenorizada como o modelo se comporta em cada classe e identificar eventuais áreas de melhoria. Também será avaliada o desempenho específica em termos de Pontuação F1, Cobertura, Precisão, Taxa de acerto por classe, pois esses detalhes podem revelar tendências que não são tão evidentes numa análise mais superficial das métricas globais.

O objetivo final é verificar se o modelo desenvolvido é capaz de apresentar um bom desempenho, trazendo resultados positivos e demonstrando a sua viabilidade no contexto da análise de faturas relacionadas aos fundos europeus.

### 5.1.1. Resultados do modelo aplicado

Com base no conjunto dados apresentado no início do Capítulo 4 e após a aplicação das configurações dos hiperparâmetros previamente discutidos, obtiveram-se os seguintes resultados aplicando média micro no conjunto de dados de testes na Tabela 5.1.

Os resultados apresentados na Tabela 5.1 são satisfatórios, com todas as métricas acima dos 80%. Isto sugere que as escolhas dos hiperparâmetros foram acertadas e que o conjunto dados está bem anotado, proporcionando uma base sólida para o treino do

Tabela 5.1. Resultado do modelo aplicado

| Pontuação F1 | Cobertura | Precisão | Taxa de acerto |
|---|---|---|---|
| 85,05% | 86,47 | 83,68% | 98,93% |

modelo. No geral, os dados indicam que o modelo é fiável, com um bom equilíbrio entre Precisão e Cobertura, além de uma excelente Taxa de acerto.

- A cobertura de 86,47% indica que o modelo está a capturar a maioria dos casos relevantes. Isso é essencial para garantir que poucos casos importantes estejam errados.
- Com uma precisão de 83,68%, o modelo demonstra que, dos casos que classifica como positivos, 83,68% são realmente corretos. Sendo esta métrica fundamental para analisar o número de falsos positivos.
- A Taxa de acerto do modelo é extremamente alta, indicando que ele faz previsões corretas na grande maioria dos casos, tanto para as classes positivas quanto para as negativas. No entanto, é importante considerar que a classe *other* representa cerca de 70% do conjunto dados, o que pode influenciar essa métrica de maneira significativa.

### 5.1.2. Resultados das classes

A Figura 5.1 exibe a matriz de confusão normalizada, que foi utilizada para avaliar o desempenho de um modelo de classificação. Nessa matriz, os valores variam de 0 a 100, representando a percentagem de previsões corretas ou incorretas feitas pelo modelo em relação a cada classe.

O cálculo da matriz de confusão apresenta valores arredondados para 0 casas decimais para maior legibilidade. A normalização foi aplicada utilizando uma proporção simples para converter os valores absolutos em percentagens, considerando a proporção de previsões corretas e incorretas em relação ao total de amostras reais de cada classe.

A diagonal principal da matriz mostra as previsões corretas em cada classe. Valores mais próximos de 100 na diagonal indicam que o modelo obteve um bom desempenho ao classificar essas classes. Por outro lado, os valores fora da diagonal correspondem a previsões incorretas, ou seja, instâncias onde o modelo confundiu uma classe com outra.

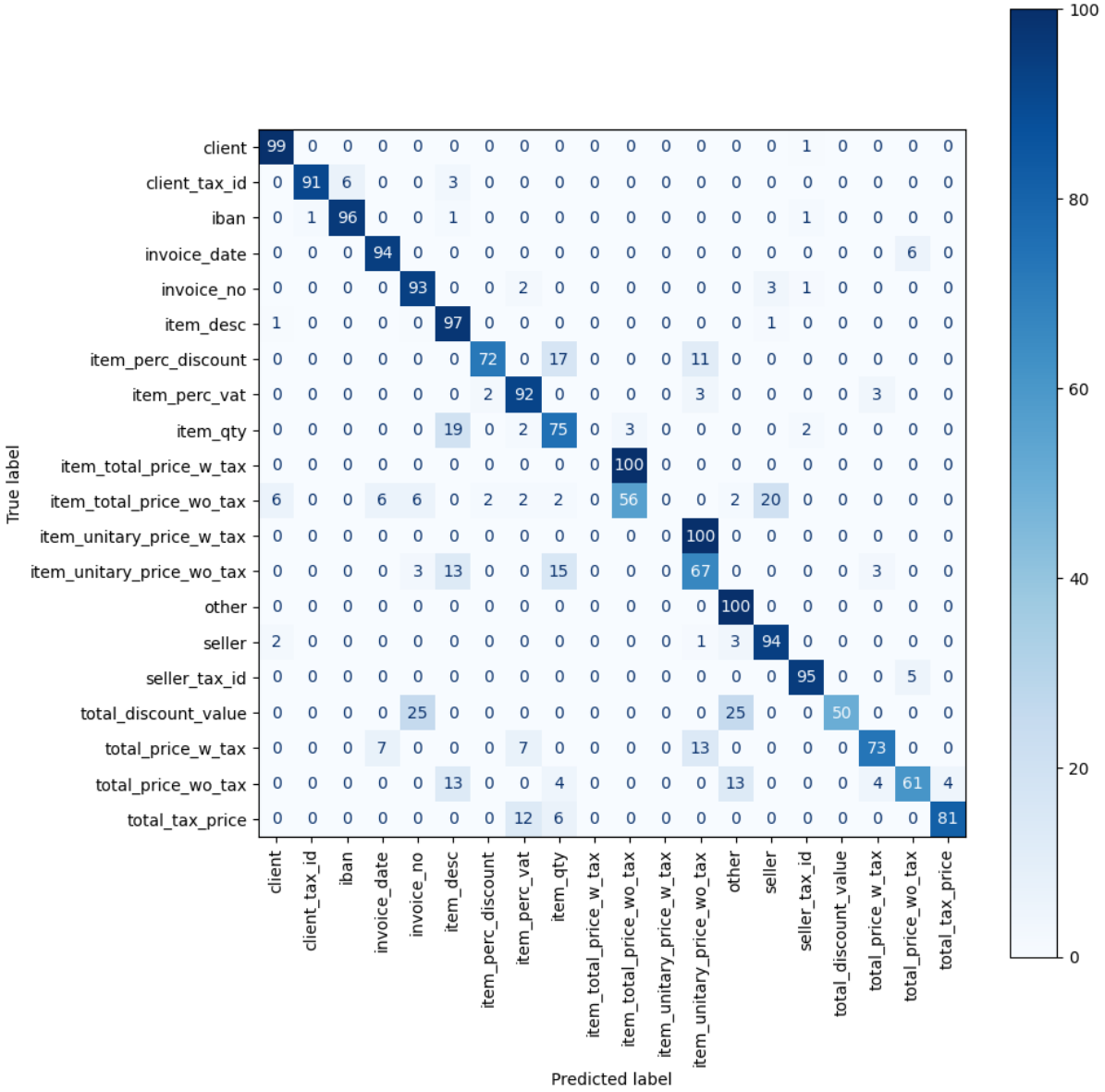



FIGURA 5.1. Matriz de confusão do conjunto de dados de teste

No ponto de vista geral, o modelo apresentou resultados para alguma classes muito bons, para outras classes um desempenho mediano e ainda existem exceções de resultado não tão positivos , não tendo sequer acertado um caso, nestes últimos.

Para os resultados muito bons que teve acima de 90% de previsões corretas foram as classes:

- *client*
- *client_tax_id*
- *iban*
- *invoce_date*
- *item_desc*
- *item_perc_vat*
- *other*

- *seller*
- *seller_tax_id*

Excluindo a classe other, as restantes classes que apresentaram resultados bastantes positivos tem um papel critico para o contexto inserido no trabalho que pode demonstrar um bom nível de confiança para o modelo prever estes campos críticos.

Para os resultados medianos, acima da média, apresentam um aumento da granularidade da análise da fatura que são classes com mais pormenor são as seguintes classes que apresentarem com uma percentagem prevista entre de 50% até 81%:

- *item_perc_discount*
- *item_qty*
- *total_discount_value*
- *total_price_w_text*
- *total_price_wo_text*
- *total_taxt_price*

As classes que apresentam resultados bastante inferiores onde em certas situações não conseguiram acertar nenhuma corretamente, foram:

- *item_total_price_w_tax*
- *item_total_price_wo_tax*
- *item_unitary_price_w_tax*
- *item_unitary_price_wo_tax*

Neste cenário, o modelo não conseguiu distinguir o preço com e sem imposto, pois podemos verificar que não existe no conjunto dados, suficientemente dados anotados nestas classes. Como se pode verificar na Tabela 4.2 a percentagem de ocorrências desta classe é bastante baixa, para as classes item_*total_price_w_tax* e *item_unitary_price_w_tax* tem uma percentagem de ocorrência de 0,15% e de 0,07% e no conjunto dados de teste ainda mais baixo é o valor. Assim está explicado o mau resultado para estas classes. Talvez aumentando o número de ocorrências no conjunto dados para o modelo aprender com mais exemplo, poderia melhorar os resultados destas classes.

Apesar do modelo apresentar resultados baixos para estas quatro classes, se agregarmos as classes sem fazer a distinção do valor com ou sem imposto os resultados já são bem diferentes. Assim, foi criada uma nova matriz de confusão (Tabela 5.2) agregando os resultados *item_total_price_w_tax* com *item_total_price_wo_tax* na classe *item_total_price* e *item_unitary_price_w_tax* com *item_unitary_price_wo_tax* na classe *item_unitary_price*.

Como se pode verificar na Tabela 5.2, sem a distinção do valores com ou sem imposto, já consegui apresentar um resultado mais positivo. Porém, se ainda existissem mais dados para estas classes, os valores teriam sido ainda mais positivos.

Por último, segue a Tabela 5.3 que apresenta os resultados das três métricas, Pontuação F1, Cobertura e Precisão na fase de testes.

TABELA 5.2. Matriz de confusão agregando as classes para *item_total_price* e *item_unitary_price*

| True label | | | |
|---|---|---|---|
| | item_total_price | 77% | 0% |
| | item_unitary_price | 0% | 74% |
| | | item_total_price | item_unitary_price |
| | | Predicted label | |

TABELA 5.3. Resultado dos testes do modelo aplicado

| Classe | Pontuação F1 | Cobertura | Precisão |
|---|---|---|---|
| client | 91,42% | 96,96% | 86,48% |
| client_tax_id | **91,80%** | 93,33% | 90,32% |
| iban | 90,32% | 87,50% | 93,33% |
| invoice_date | 91,17% | 93,93% | 88,57% |
| invoice_no | 90,41% | 91,67% | 89,19% |
| item_desc | 82,10% | 88,63% | 76,47% |
| item_perc_discount | 64,00% | 57,14% | 72,72% |
| item_perc_vat | 82,35% | 84,00% | 80,76% |
| item_qty | 67,79% | 71,42% | 64,51% |
| item_total_price_w_tax | 0,00% | 0,00% | 0,00% |
| item_total_price_wo_tax | 75,57% | 70,00% | 82,35% |
| item_unitary_price_w_tax | 0,00% | 0,00% | 0,00% |
| item_unitary_price_wo_tax | 69,09% | 70,37% | 67,85% |
| other | 92,39% | 95,50% | 92,00% |
| seller | 91,13% | 92,30% | 90,00% |
| seller_tax_id | 91,67% | **97,05%** | 86,84% |
| total_discount_value | 80,00% | 66,67% | **100,00%** |
| total_price_w_tax | 70,00% | 70,00% | 70,00% |
| total_price_wo_tax | 69,23% | 64,28% | 75,00% |
| total_tax_price | 84,21% | 80,00% | 88,89% |

A Tabela 5.3 está relacionada com a matriz na Figura 5.1 apresentando um desempenho satisfatório.

Excluindo a classe *other* , a classe que teve o melhor desempenho para a Pontuação F1 foi a classe *client_tax_id* com 91,80%, porém para a métrica Cobertura foi a classe *seller_tax_id* com 97,05%. Para a métrica Precisão, a classe que apresentou o valor mais elevado foi a *total_discount_value* com 100%, porém esta tem características que merecem atenção. Com a precisão a 100%, significa que acertou todas as instâncias previstas como *total_discount_value* pelo modelo estão corretas, porém apresenta uma cobertura de 66,7% que só está a identificar 66,7% dos casos reais.

Como foi referido anteriormente, o modelo apresenta um resultado muito bom nas classes *client*, *client_tax_id*, *iban*, *invoce_date*, *item_desc*, *item_perc_vat*, *other*, *seller* e *seller_tax_id*. No entanto, existe um desempenho insatisfatório nas classes relacionadas a preços unitários e totais com e sem impostos. O que o modelo precisa é de mais ajustes para melhorar a distinção entre essas classes.

### 5.1.3. Discussão

Nesta secção, o objetivo foi avaliar a capacidade de um modelo de classificação de inferir corretamente os campos de uma fatura a partir de um conjunto de dados de teste. O modelo foi treinado para identificar e classificar automaticamente os diferentes atributos presentes numa fatura, como o número da fatura, o cliente, o valor total, os impostos, entre outros. Para medir a qualidade dessa inferência, os resultados foram comparados com os dados reais, previamente anotados manualmente.

A Figura 5.2 representa as classes previstas pelo modelo, enquanto a Figura 5.3 mostra as classes reais anotadas no conjunto de dados.

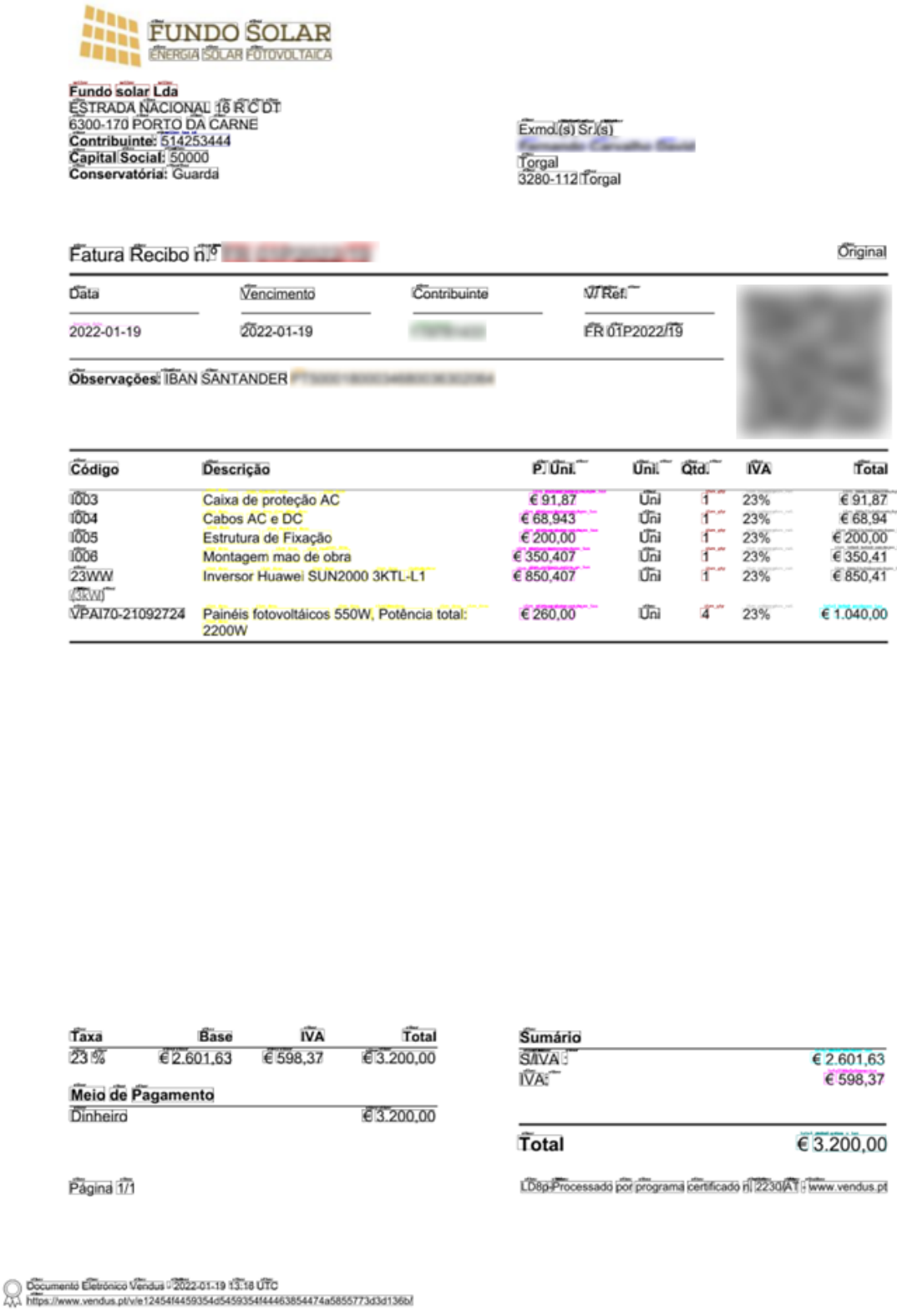
As faturas encontram-se com alguns dados mascarados como informação do cliente, número da fatura ou o respetivo código QR como também o IBAN, pois são faturais reais no âmbito do contexto abordado.

Inicialmente, pode verificar-se que todas as palavras estão anotadas. Todas as palavras que estão anotadas a preto são consideradas como classe *other*, assim é possível verificar o desequilíbrio de dados no conjunto dados.

Como já é prevista na apresentação dos resultados do modelo, a maioria das classes previstas presentes nas figuras apresentam estar corretas como se pode analisar na classe *seller* com a cor vermelha, que foi corretamente prevista como também a classe *iban* ou *client* entre outros.

Também consegue identificar bem a descrição dos produtos das faturas e ignorando o código do produto, classificando como *other*.

Também previu corretamente a quantidade do produto, porém no último produto, o produto "Paneis fotovoltaicos 550W, Potência total: 2200W", considerou o seu valor como o valor total, classe *total_price_wo_tax* onde só devia existir uma classe por fatura, que está apresentada mais abaixo na secção do sumário.



FUNDO SOLAR
ENERGIA SOLAR FOTOVOLTAICA

**Fundo solar Lda**
ESTRADA NACIONAL 16 R C DT
6300-170 PORTO DA CARNE
**Contribuinte:** 514253444
**Capital Social:** 50000
**Conservatória:** Guarda

Exmo.(s) Sr.(s)
[redacted]
Torgal
3280-112 Torgal

Fatura Recibo n.º [redacted] Original

| Data | Vencimento | Contribuinte | O/ Ref. |
|---|---|---|---|
| 2022-01-19 | 2022-01-19 | [redacted] | FR 01P2022/19 |

**Observações:** IBAN SANTANDER [redacted]

| Código | Descrição | P. Uni. | Uni. | Qtd. | IVA | Total |
|---|---|---|---|---|---|---|
| 1003 | Caixa de proteção AC | € 91,87 | Uni | 1 | 23% | € 91,87 |
| 1004 | Cabos AC e DC | € 68,943 | Uni | 1 | 23% | € 68,94 |
| 1005 | Estrutura de Fixação | € 200,00 | Uni | 1 | 23% | € 200,00 |
| 1006 | Montagem mao de obra | € 350,407 | Uni | 1 | 23% | € 350,41 |
| 23WW | Inversor Huawei SUN2000 3KTL-L1 | € 850,407 | Uni | 1 | 23% | € 850,41 |
| (3kW) | | | | | | |
| VPAI70-21092724 | Painéis fotovoltáicos 550W, Potência total: 2200W | € 260,00 | Uni | 4 | 23% | € 1.040,00 |

| Taxa | Base | IVA | Total |
|---|---|---|---|
| 23 % | € 2.601,63 | € 598,37 | € 3.200,00 |

**Meio de Pagamento**

| | |
|---|---|
| Dinheiro | € 3.200,00 |

**Sumário**

| | |
|---|---|
| S/IVA: | € 2.601,63 |
| IVA: | € 598,37 |
| **Total** | € 3.200,00 |

Página 1/1

LD8p Processado por programa certificado nº 2230/AT - www.vendus.pt

Documento Eletrónico Vendus - 2022-01-19 13:18 UTC
https://www.vendus.pt/v/e12454f4459354d5459354f4446385447 4a5855773d3d136b

FIGURA 5.2. Exemplo de uma fatura classificada pelo modelo

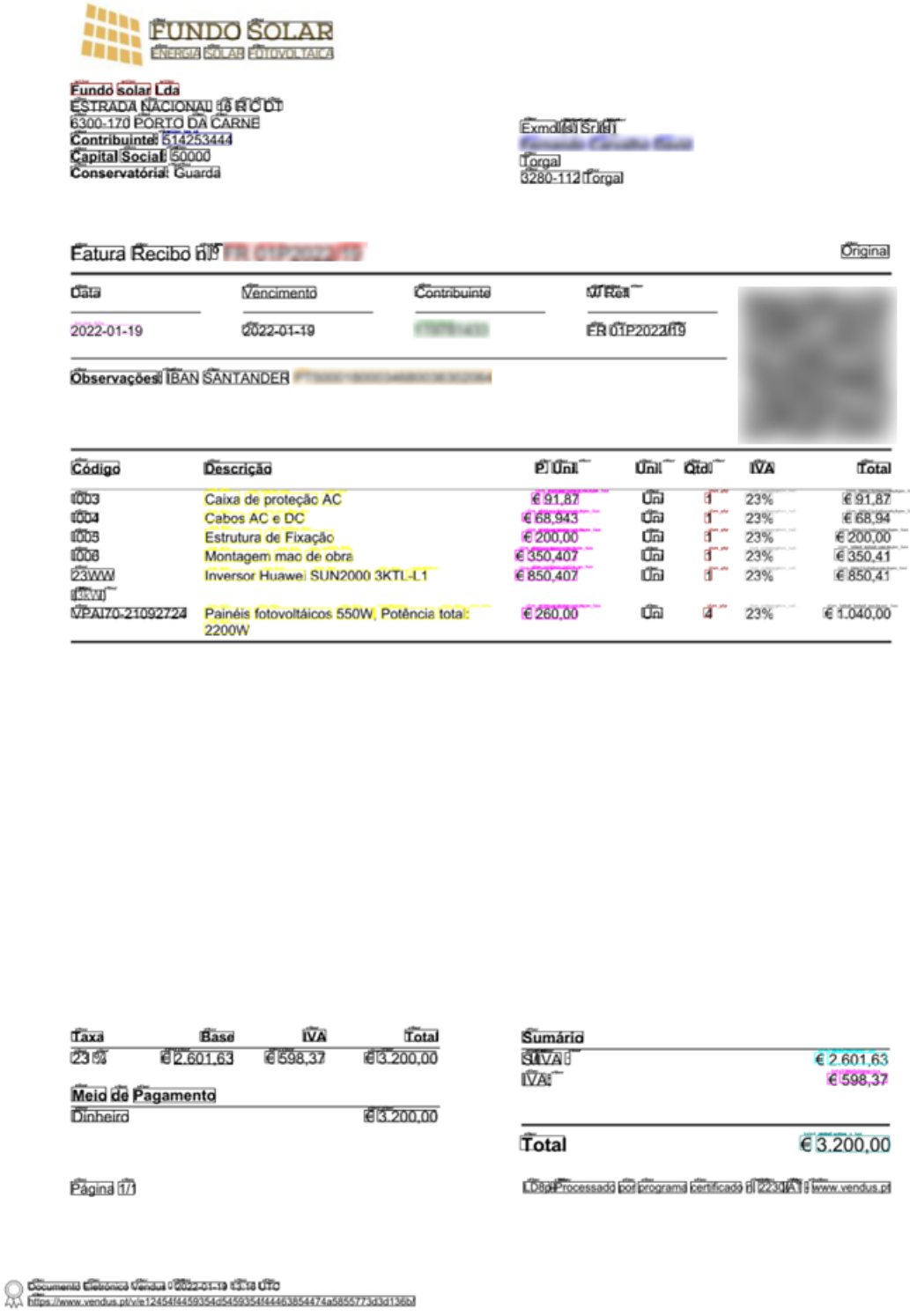



FIGURA 5.3. Exemplo de uma fatura anotada manualmente

## 5.2. Análise comparativa com o modelo IDA

Nesta secção abordam-se as configurações aplicadas pelo IDA comparando com o modelo aplicado e analisando o seu desempenho relativamente ao contexto do trabalho.

O IDA aplica a ferramenta *LayoutLMv2* orientado mais a redes neuronais convolucionais, CNN, enquanto o modelo aplicado com a ferramenta *LayoutLMv3* já tem presente a *Vision transformer* (VIT).

Como se pode verificar na Tabela 5.4, estão presentes as configurações dos hiperparâmetros aplicados do IDA e do modelo aplicado (*LayoutLMv3* ). Apresentam todos as mesmas configurações sem nenhuma diferença, exceto no modelo pré-treinado que o IDA usou o modelo apropriado para a sua ferramenta, "*microsoft/layoutlmv2-base-uncased*", enquanto no *LayoutLMv3* utilizou-se o "*microsoft/layoutlmv3-base*".

Tabela 5.4. Tabela comparativa entre *LayoutLMv2* e *LayoutLMv3*

| Modelo e Optimizador | | |
|---|---|---|
| | LayoutLMv2 (IDA) | LayoutLMv3 |
| Modelo Pré-Treinado | *microsoft/layoutlmv2-base-uncased* | *microsoft/layoutlmv3-base* |
| Revisão | no_ocr | no_ocr |
| Nome Optimizador | AdamW | AdamW |
| Learning rate | 3e-5 | 3e-5 |
| Parâmetros de Otimizador (betas) | (0.9, 0.999) | (0.9, 0.999) |
| weight decay | 0.01 | 0,01 |

No que diz respeito às configurações de treino, as diferenças entre o IDA e o modelo aplicado foram o número de épocas: o IDA aplicou 60 épocas enquanto o modelo aplicado foram 30 épocas, sendo explicada a razão no Capítulo 5. Também para o lote de processamento tanto de treino como validação existiram diferenças: no modelo aplicado utilizou-se o tamanho quatro, pois levou a melhores resultados, ainda que esta opção também é motivada por limitação de recursos computacionais.

Tabela 5.5. Configurações de treino para *LayoutLMv2* e *LayoutLMv3*

| Treino | | |
|---|---|---|
| | LayoutLMv2 (IDA) | LayoutLMv3 |
| Número Máximo de épocas | 60 | 30 |
| Tamanho do Lote de Treino | 8 | 4 |
| Tamanho do Lote de Validação | 16 | 4 |
| Intervalo de Verificação de Validação | 0,4 | 0,4 |
| Tipo Acelerador | GPU | GPU |

Após o treino de ambos os modelos, os resultados por classe na fase de testes estão apresentados na Tabela 5.6. Os modelos foram avaliados utilizando o mesmo conjunto de dados para treino e teste.

Para cada classe foi calculado a Pontuação F1, a Cobertura e a Precisão. Do modo geral, ambos os modelos apresentaram bons resultados com valores elevados. Porém, existem diferenças no desempenho apresentado entre os modelos algumas menores e outras mais significativas.

TABELA 5.6. Métricas para *LayoutLMv2* e *LayoutLMv3*

| Classes | Pontuação F1 | | Cobertura | | Precisão | |
|---|---|---|---|---|---|---|
| | LayoutLMv2 (IDA) | LayoutLMv3 | LayoutLMv2 (IDA) | LayoutLMv3 | LayoutLMv2 (IDA) | LayoutLMv3 |
| client | 93% | 91% | 93% | 97% | 92% | 86% |
| client_tax_id | 94% | 92% | 97% | 93% | 91% | 90% |
| iban | 96% | 90% | 96% | 88% | 95% | 93% |
| invoice_date | 76% | 91% | 78% | 94% | 75% | 89% |
| invoice_no | 79% | 90% | 82% | 92% | 77% | 89% |
| item_desc | 97% | 82% | 96% | 89% | 97% | 76% |
| item_perc_discount | 91% | 64% | 84% | 57% | 100% | 73% |
| item_perc_vat | 92% | 82% | 90% | 84% | 95% | 81% |
| item_qty | 94% | 68% | 97% | 71% | 90% | 65% |
| item_total_price_w_tax | 33% | 0% | 20% | 0% | 100% | 0% |
| item_total_price_wo_tax | 95% | 76% | 95% | 70% | 95% | 82% |
| item_unitary_price_w_tax | 0% | 0% | 0% | 0% | 0% | 0% |
| item_unitary_price_wo_tax | 93% | 69% | 88% | 70% | 98% | 68% |
| other | 98% | 92% | 98% | 96% | 98% | 92% |
| seller | 79% | 91% | 84% | 92% | 74% | 90% |
| seller_tax_id | 64% | 92% | 59% | 97% | 70% | 87% |
| total_discount_value | 76% | 80% | 71% | 67% | 81% | 100% |
| total_price_w_tax | 95% | 70% | 95% | 70% | 95% | 70% |
| total_price_wo_tax | 81% | 69% | 86% | 64% | 76% | 75% |
| total_tax_price | 86% | 84% | 89% | 80% | 83% | 89% |

O modelo da ferramenta *LayoutLMv3*, baseando na métrica Pontuação F1, apresentou resultados superiores em relação ao *LayoutLMv2* do IDA nas seguintes classes.

- *invoice_ date*
- *invoice_ no*
- *seller*
- *seller_ tax_ id*
- *total_ discount_ value*

Destacam-se as maiores diferenças verificadas nas classes *invoice_ no* e *invoice_ date* que teve uma diferença de 15 e 11 pontos percentuais, respetivamente. Além disso, nas classes relacionadas com o fornecedor, a diferença é ainda maior, com a classe *seller* teve valor de 79% em relação a 91% e na classe *seller_ tax_ id* foi de 64% para 92% sendo uma grande diferença de 28 pontos percentuais.

Para as restantes classes, o modelo do IDA apresentou melhores resultados nas seguintes classes, porém as diferenças são reduzidas:

- *client*
- *client_ tax_ id*
- *iban*
- *other*
- *total_ tax_ price*

As diferenças rondam os 2 e 3 pontos percentuais: por exemplo, para a classe *client* o IDA teve 93% e o modelo aplicado teve 91%, não sendo a diferença significativa.

Para as restantes classes o modelo IDA apresentou resultados bastante melhores que o modelo do *LayoutLMv3*:

- *item_ desc*
- *item_ perc_ discount*
- *item_ perc_ vat*

- *item_qty*
- *item_total_price_wo_tax*
- *item_unitary_price_wo_tax*
- *total_price_w_tax*
- *total_price_wo_tax*

Por exemplo, na classe *item_perc_discount* o IDA teve 91% enquanto o modelo aplicado teve 64%.

Outro aspeto interessante é que ambos os modelos tiveram maus resultados para as classes *item_total_price_w_tax*, *item_total_price_wo_tax*, *item_unitary_price_w_tax* e *item_unitary_price_wo_tax*. Como foi abordado anteriormente, a razão mais propícia para isto acontecer é a falta de diversidade no conjunto dados para estas classes que não permite ao modelo aprender adequadamente a diferenciação dos valores com ou sem imposto.

Em suma, existem classes em que o *LayoutLMv2* apresenta melhor desempenho, noutras é o *LayoutLMv3* que tem melhor desempenho e as restantes têm valores muito aproximadas. Assim, é possível verificar para informação mais gerais como detetar informação sobre o fornecedor, informação da fatura como o número ou a data e também informação sobre o cliente ou IBAN (que apesar do *LayoutLMv2* teve melhor resultado, mas por muito pouco) o *LayoutLMv3* apresenta melhor desempenho para estes campos gerais que não dependem muito de processamento de imagem. Porém, para campos mais específicos que necessitam mais pormenor na análise com maior granularidade como informação do produto que normalmente estão presente em tabelas, requer um melhor processamento de imagem, como os respetivos valores unitários e valores totais, o *LayoutLMv2* do IDA apresenta melhor desempenho.

CAPÍTULO 6

# Conclusão

Neste capítulo, são abordadas as conclusões retiradas do modelo aplicado, respondendo às questões de investigação. Além disso, será discutida a relevância dos resultados obtidos. Por fim, serão sugeridos possíveis trabalhos futuros que poderão dar continuidade a este tema, explorando novas abordagens e metodologias para aprofundar o conhecimento na área.

## 6.1. Principais conclusões

Para concluir, o modelo aplicado com a ferramenta *LayoutLMv3* pode ser uma alternativa ao uso de códigos QR. O modelo obteve bons resultados para classes como data da fatura (Pontuação F1 de 91%) e nome do fornecedor (Pontuação F1 de 91%), demonstrando fiabilidade para esses campos genéricos. No entanto, para campos mais específicos que exigem maior precisão, como percentagem de desconto do produto (Pontuação F1 de 64%) e quantidade do produto (Pontuação F1 de 68%), o desempenho ainda é limitado, o que pode resultar em classificações incorretas nessas classes. Assim, o modelo pode ser uma alternativa fiável em relação ao código QR quando o objetivo é extrair informações mais genéricas, como número ou data da fatura e dados do fornecedor ou cliente.

Além disso, com base na comparação de resultados apresentada, a ferramenta *LayoutLMv2* do IDA apresentou um desempenho superior ao *LayoutLMv3* no contexto de extração de faturas ligadas a fundos europeus em Portugal. Como ambas as ferramentas apresentam arquiteturas diferentes consequentemente devolveram resultados diferentes sendo que a *LayoutLMv2* apresentou melhores resultados nas maioria das classes, principalmente nas classes que têm mais pormenor em relação aos produtos com mais processamento de imagem.

A arquitetura do *LayoutLMv2* é mais orientada para processamento de imagem através do algoritmo ResNetX-FTT, uma CNN, altamente eficaz, para extrair características visuais relevantes, antes de as integrar com as informações textuais e de *layout*. Isso permite que o *LayoutLMv2* capte informações sobre a estrutura e a posição do texto na imagem, o que é crucial para documentos com *layout* complexo, como as faturas.

Por outro lado, o *LayoutLMv3* processa o texto e as imagens de forma integrada, em vez de os separar como no *LayoutLMv2*. A diferença mais significativa de arquitetura que pode ter influenciado o desempenho é que o *LayoutLMv3* usa uma abordagem baseada no VIT, uma abordagem mais sofisticada. Esta abordagem é mais exigente em termos de recursos e dados devido à complexidade de integrar texto e imagem no mesmo espaço.

A razão de a ferramenta *LayoutLMv2* ter apresentado melhores resultados em relação ao *LayoutLMv3* foi pelo tamanho do conjunto dados. Ou seja, as abordagens baseadas no VIT na maioria dos cenários são as que apresentam melhores resultados, porém precisam de ter um extenso conjunto de dados a representar o universo das faturas. Neste cenário, o conjunto dados do treino não tinha grande dimensão de amostras nem muita ocorrência das classes, o que dificultou a aprendizagem ao *LayoutLMv3*. Uma abordagem baseada em CNN, neste cenário usando o algoritmo ResNetx, não precisa de existir um grande conjunto de dados para apresentar bons resultados.

As classes que estão presentes na tabela dos produtos das faturas, como o nome ou código do produto, ou o preço com ou sem iva, entre outros, são campos dependentes de processamento de imagens onde é mais propício a que CNN consigam melhores resultados do que VIT, pois tem mais a ver com extração de imagem do que texto. Estes pressupostos estão justificados nos seguintes trabalhos:

- No trabalho "*A comparative study between vision transformers and CNNs in digital pathology*" [37], foi demonstrado que a arquitetura VIT tem grande potencial na classificação de imagem , pois consegue perceber relação entre os dados. O estudo realizou uma comparação entre a arquitetura CNN, ResNet18, e a VIT, concluindo que, embora ambas apresentassem desempenhos muito similares, a VIT obteve resultados ligeiramente superiores. Esse desempenho foi particularmente evidente num conjunto de dados composto por 100.000 segmentos de imagens altamente diversificados, destacando a eficácia da VIT em cenários com maior variabilidade nos dados.
- Já no trabalho "*Comparison of the potential between transformer and CNN in image classification*" [38] foi apresentada a mesma conclusão. Para um conjunto dados de imagens com 10270 imagens e 23 classes para classificar, o modelo que usa CNN apresentou um resultado superior de 68,99% na Taxa de acerto da validação, enquanto o VIT obteve o resultado de 45,57%. Porém, ao duplicar o conjunto dados para o dobro, 20540 imagens, ambos tiveram resultados semelhantes, no qual o CNN obteve um resultado de 75,71% e o VIT obteve um resultado de 75,46% o que exigiu significativamente mais tempo.
- Por último, no trabalho "*Comparing Vision Transformers and Convolutional Neural Networks for Image Classification: A Literature Review*" [39], as arquiteturas que usam VIT têm tendência a serem mais fiáveis do que CNN para imagens que têm mais ruído e que têm melhor desempenho, mas novamente se tiver um conjunto dados maior. O CNN generaliza melhor com conjunto dados mais pequenos sendo o tempo de treino bastante reduzido em relação aos VIT.

Sendo o *LayoutLMv3* mais complexo que o *LayoutLMv2*, são necessários mais recursos computacionais e um maior volume de dados para alcançar todo o seu potencial. Em contrapartida, o *LayoutLMv2* tende a ser mais robusto em cenários onde os dados são limitados.

Para finalizar, o *LayoutLMv2* pode apresentar desempenho superior em alguns cenários devido à sua simplicidade e menor exigência de recursos. Já o *LayoutLMv3*, com sua integração mais profunda entre texto e imagem, pode precisar de um conjunto dados mais composto e variado para alcançar resultados. No caso deste conjunto dados reduzido, o *LayoutLMv2* obteve melhor desempenho por ser mais eficiente e menos dependente de interações complexas entre texto e imagem.

## 6.2. Trabalho futuro

A investigação realizada proporcionou uma comparação entre as ferramentas *LayoutLMv2* e *LayoutLMv3*. Esta secção apresenta sugestões para futuras pesquisas e desenvolvimentos que podem contribuir para o avanço deste tema.

Sugestões para dar continuidade ao tema:

- Utilizar ou criar um conjunto de dados com mais instâncias e onde o número de ocorrências das classes seja superior ao apresentado no conjunto dados atual. Consequentemente permitirá ao modelo ter mais variedade no treino que proporcionará melhor desempenho. Assim com um conjunto dados maior e variado podemos novamente realizar uma nova comparação entre a ferramenta *LayoutLMv2* que utiliza redes neuronais, CNN, com a ferramenta *LayoutLMv3* que utiliza transformadores, assim podemos verificar com base na conclusão se existir um maior conjunto dados variado as arquiteturas que abordam transformadores têm ou não melhores desempenhos em relação a CNN.
- Experimentar outras ferramentas ou arquiteturas, como o Donut [40]. O Donut é uma arquitetura que se destaca por sua capacidade de lidar com tarefas complexas de reconhecimento de padrões e processamento de linguagem natural, oferecendo uma abordagem inovadora que pode melhorar os resultados.
- Desenvolver estratégias para detetar preços com e sem impostos de forma mais eficaz. Isso pode incluir a criação ou descoberta de algoritmos baseados em cálculos matemáticos.
- Abordar o mesmo tema, mas para contextos diferentes como por exemplo com leitura de cartas de condução de veículos agrícolas, leitura de contratos, documentos jurídicos, declarações fiscais, entre outros.

# Bibliografia


[1] European Parliament, "Coesão, crescimento e emprego," European Parliament. [Online]. Available: https://www.europarl.europa.eu/erpl-app-public/factsheets/pdf-chapter/pt/pt-chapter-3.pdf.

[2] E. Parliament, *Fundo europeu de desenvolvimento regional (feder)*. [Online]. Available: https://www.europarl.europa.eu/factsheets/pt/sheet/95/el-fondo-europeo-de-desarrollo-regional-feder (visited on 10/01/2023).

[3] E. Commission, *2014-2020 european structural and investment funds*. [Online]. Available: https://commission.europa.eu/funding-tenders/find-funding/funding-management-mode/2014-2020-european-structural-and-investment-funds_en.

[4] E. Commission, *European commission cohesion policy data for portugal (2021-2027) - https://cohesiondata.ec.europa.eu/countries/pt/21-27*, European Commission. [Online]. Available: https://cohesiondata.ec.europa.eu/countries/PT/21-27.

[5] "Portugal's 2014-2020 cohesion data," European Commission. (), [Online]. Available: https://cohesiondata.ec.europa.eu/countries/PT/14-20.

[6] Finanças, *Portaria n.º 195/2020*, Diário da República n.º 157/2020, Série I de 2020-08-13, páginas 13-15, Acesso em: 19 out. 2024, 2020. [Online]. Available: https://diariodarepublica.pt/dr/detalhe/portaria/195-2020-140210523.

[7] Kowsari, J. Meimandi, Heidarysafa, Mendu, Barnes e Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, Apr. 2019, ISSN: 2078-2489. DOI: 10.3390/info10040150. [Online]. Available: http://dx.doi.org/10.3390/info10040150.

[8] Vikramkumar, V. B e Trilochan, *Bayes and naive bayes classifier*, 2014. arXiv: 1404.0933 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1404.0933.

[9] G. Louppe, *Understanding random forests: From theory to practice*, 2015. arXiv: 1407.7502 [stat.ML]. [Online]. Available: https://arxiv.org/abs/1407.7502.

[10] S. Mori, C. Suen e K. Yamamoto, "Historical review of ocr research and development," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1029–1058, 1992. DOI: 10.1109/5.156468.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser e I. Polosukhin, *Attention is all you need*, 2023. arXiv: 1706.03762 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1706.03762.

[12] I. Düntsch e G. Gediga, "Confusion matrices and rough set data analysis," *Journal of Physics: Conference Series*, vol. 1229, no. 1, p. 012 055, May 2019, ISSN: 1742-6596. DOI: 10.1088/1742-6596/1229/1/012055. [Online]. Available: http://dx.doi.org/10.1088/1742-6596/1229/1/012055.

[13] M. Vakili, M. Ghamsari e M. Rezaei, *Performance analysis and comparison of machine and deep learning algorithms for iot data classification*, 2020. arXiv: 2001.09636 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2001.09636.

[14] M. J. Anzanello e F. S. Fogliatto, "Learning curve models and applications: Literature review and research directions," *International Journal of Industrial Ergonomics*, vol. 41, no. 5, pp. 573–583, 2011, ISSN: 0169-8141. DOI: https://doi.org/10.1016/j.ergon.2011.05.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016981411100062X.

[15] G. Raskutti, M. J. Wainwright e B. Yu, *Early stopping and non-parametric regression: An optimal data-dependent stopping rule*, 2013. arXiv: 1306.3574 [stat.ML]. [Online]. Available: https://arxiv.org/abs/1306.3574.

[16] M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan e et al., "The prisma 2020 statement: An updated guideline for reporting systematic reviews," *International Journal of Surgery*, vol. 88, p. 105 906, 2021. DOI: 10.1016/j.ijsu.2021.105906.

[17] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei e M. Zhou, "LayoutLM: Pre-training of text and layout for document image understanding," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, ser. KDD '20, ACM, Aug. 2020. DOI: 10.1145/3394486.3403172. [Online]. Available: http://dx.doi.org/10.1145/3394486.3403172.

[18] D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman e J. Heard, "Building a test collection for complex document information processing," in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '06, Seattle, Washington, USA: Association for Computing Machinery, 2006, ISBN: 1595933697. DOI: 10.1145/1148170.1148307. [Online]. Available: https://doi.org/10.1145/1148170.1148307.

[19] G. Jaume, H. K. Ekenel e J.-P. Thiran, "Funsd: A dataset for form understanding in noisy scanned documents," in *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, IEEE, vol. 2, 2019, pp. 1–6.

[20] Y. Xu, Y. Xu, T. Lv, L. Cui, F. Wei, G. Wang, Y. Lu, D. Florencio, C. Zhang, W. Che, M. Zhang e L. Zhou, *LayoutLMv2: Multi-modal pre-training for visually-rich document understanding*, 2022. arXiv: 2012.14740 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2012.14740.

[21] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Ł. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang,

C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes e J. Dean, *Google's neural machine translation system: Bridging the gap between human and machine translation*, 2016. arXiv: 1609.08144 [cs.CL].

[22] S. Xie, R. Girshick, P. Dollár, Z. Tu e K. He, *Aggregated residual transformations for deep neural networks*, 2017. arXiv: 1611.05431 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1611.05431.

[23] Y. Huang, T. Lv, L. Cui, Y. Lu e F. Wei, *LayoutLMv3: Pre-training for document ai with unified text and image masking*, 2022. arXiv: 2204.08387 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2204.08387.

[24] S. Appalaraju, B. Jasani, B. U. Kota, Y. Xie e R. Manmatha, *Docformer: End-to-end transformer for document understanding*, 2021. arXiv: 2106.11539 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2106.11539.

[25] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit e N. Houlsby, *An image is worth 16x16 words: Transformers for image recognition at scale*, 2021. arXiv: 2010.11929 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2010.11929.

[26] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer e V. Stoyanov, *Roberta: A robustly optimized bert pretraining approach*, 2019. arXiv: 1907.11692 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1907.11692.

[27] G. Szegedi, D. B. Veres, I. Lendák e T. Horváth, "Context-based information classification on hungarian invoices," in *CEUR Workshop Proceedings*, vol. 2718, 2020, pp. 147–151. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85095977829.

[28] X. Hoangvan, P. Tranquang, M. Dinhbao e T. Vuhuu, "Developing an ocr model for extracting information from invoices with korean language," 2023, pp. 84–89. DOI: 10.1109/ATC58710.2023.10318877. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85179140563&doi=10.1109%2fATC58710.2023.10318877&partnerID=40&md5=b01fe8c3c5bca12c2d482dde16c62dc2.

[29] Y. Du, Z. Chen, C. Jia, X. Yin, T. Zheng, C. Li, Y. Du e Y.-G. Jiang, *Svtr: Scene text recognition with a single visual model*, 2022. arXiv: 2205.00159 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2205.00159.

[30] E. Commission, *Vat invoicing*, https://taxation-customs.ec.europa.eu/taxation/vat/vat-businesses/vat-invoicing_en, [Online; Accessed: Dec. 15, 2024], 2024.

[31] A. Hamdi, E. Carel, A. Joseph, M. Coustaty e A. Doucet, "Information extraction from invoices," in *International Conference on Document Analysis and Recognition (ICDAR 2021)*, ser. Lecture Notes in Computer Science, vol. 12822, Lausanne, Switzerland: Springer International Publishing, Sep. 2021, pp. 699–714. DOI: 10.1007/978-3-030-86331-9_45.

[32] R. B. Palm, O. Winther e F. Laws, *Cloudscan - a configuration-free invoice analysis system using recurrent neural networks*, 2017. arXiv: 1708.07403 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1708.07403.

[33] S. Luo e J. Yu, "SGFNet: A semantic graph-based multimodal network for financial invoice information extraction," *Expert Systems with Applications*, vol. 258, 2024, ISSN: 0957-4174. DOI: 10.1016/j.eswa.2024.125156. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417424020232.

[34] T. Douzon, S. Duffner, C. Garcia e J. Espinas, "Improving information extraction on business documents with specific pre-training tasks," in *Document Analysis Systems*. Springer International Publishing, 2022, pp. 111–125, ISBN: 9783031065552. DOI: 10.1007/978-3-031-06555-2_8. [Online]. Available: http://dx.doi.org/10.1007/978-3-031-06555-2_8.

[35] F. Krieger, P. Drews e B. Funk, "Automated invoice processing: Machine learning-based information extraction for long tail suppliers," *Intelligent Systems with Applications*, vol. 20, p. 200 285, 2023, ISSN: 2667-3053. DOI: https://doi.org/10.1016/j.iswa.2023.200285. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2667305323001102.

[36] M. Bajrami, N. Ackovska, B. Stojkoska e P. Lameski, "Deep dive into invoice intelligence: A benchmark study of leading models," in *Proceedings of Ninth International Congress on Information and Communication Technology: ICICT 2024, London, Volume 5*, Springer Nature, vol. 5, 2024, p. 177.

[37] L. Deininger, B. Stimpel, A. Yuce, S. Abbasi-Sureshjani, S. Schönenberger, P. Ocampo, K. Korski e F. Gaire, "A comparative study between vision transformers and cnns in digital pathology," 2024.

[38] K. Lu, Y. Xu e Y. Yang, "Comparison of the potential between transformer and cnn in image classification," in *ICMLCA 2021; 2nd International Conference on Machine Learning and Computer Application*, pp. 1–6.

[39] J. Maurício, I. Domingues e J. Bernardino, "Comparing vision transformers and convolutional neural networks for image classification: A literature review," *Applied Sciences*, vol. 13, no. 9, ISSN: 2076-3417. DOI: 10.3390/app13095521. [Online]. Available: https://www.mdpi.com/2076-3417/13/9/5521.

[40] G. Kim, T. Hong, M. Yim, J. Nam, J. Park, J. Yim, W. Hwang, S. Yun, D. Han e S. Park, *Ocr-free document understanding transformer*. arXiv: 2111.15664 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2111.15664.