UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE DO CAMPUS ARARANGUÁ
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

Rodrigo Bueno Guedes

Recomendação ordenada de classes de patentes por meio da técnica de *Embedding*

Araranguá
2023

Rodrigo Bueno Guedes

Recomendação ordenada de classes de patentes por meio da técnica de *Embedding*

Trabalho de Conclusão de Curso submetido ao Curso de Graduação em Engenharia de Computação do Centro de Ciências, Tecnologias e Saúde do Campus Araranguá da Universidade Federal de Santa Catarina como requisito parcial para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Alexandre Leopoldo Gonçalves, Dr.

Araranguá
2023

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Bueno Guedes, Rodrigo
Recomendação ordenada de classes de patentes por meio da técnica de Embedding / Rodrigo Bueno Guedes ; orientador, Alexandre Leopoldo Gonçalves, 2023.
38 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Campus Araranguá, Graduação em Engenharia de Computação, Araranguá, 2023.

Inclui referências.

1. Engenharia de Computação. 2. Classificação de Patentes. 3. Classificação multi-saída. 4. Incorporação de Palavras. 5. Aprendizado Profundo. I. Leopoldo Gonçalves, Alexandre. II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Computação. III. Título.

Rodrigo Bueno Guedes

Recomendação ordenada de classes de patentes por meio da técnica de *Embedding*

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Engenharia de Computação e aprovado em sua forma final pelo Curso de Graduação em Engenharia de Computação.

Araranguá, 10 de Julho de 2023.

Documento assinado digitalmente
Jim Lau
Data: 10/07/2023 10:28:18-0300
CPF: ***.464.702-**
Verifique as assinaturas em https://v.ufsc.br

Prof. Jim Lau, Dr.
Coordenador do Curso

Banca Examinadora:

Documento assinado digitalmente
Alexandre Leopoldo Goncalves
Data: 10/07/2023 10:23:50-0300
CPF: ***.740.349-**
Verifique as assinaturas em https://v.ufsc.br

Prof. Alexandre Leopoldo Gonçalves, Dr.
Orientador
Universidade Federal de Santa Catarina

Documento assinado digitalmente
Andrea Sabedra Bordin
Data: 10/07/2023 09:47:12-0300
CPF: ***.193.220-**
Verifique as assinaturas em https://v.ufsc.br

Profª. Andréa Sabedra Bordin, Draª.
Avaliador(a)
Universidade Federal de Santa Catarina

Documento assinado digitalmente
Olga Yevseyeva
Data: 10/07/2023 09:47:54-0300
CPF: ***.003.889-**
Verifique as assinaturas em https://v.ufsc.br

Profª. Olga Yevseyeva, Draª.
Avaliador(a)
Universidade Federal de Santa Catarina

Recomendação ordenada de classes de patentes por meio da técnica de Embedding

Ranking-based patent class recommendation using Embedding technique

Rodrigo Bueno Guedes[1] Alexandre Leopoldo Gonçalves[2]

2023, Julho

Resumo

Patentes representam uma fonte de informação valiosa quanto às invenções tecnológicas e possibilidades de análises em variados cenários com o objetivo de auxiliar na tomada de decisão estratégica, principalmente àquelas voltadas à pesquisa, desenvolvimento e inovação. Ademais, fornecem insumos para uma ampla gama de tarefas, entre elas a tarefa de classificação de patentes. Todavia, devido a quantidade de categorias envolvidas no processo de avaliação por examinadores, classificar uma patente constitui-se em uma atividade dispendiosa e de difícil realização. Neste contexto, o presente trabalho propõe o desenvolvimento de um método de classificação baseado em Aprendizado Profundo voltado à recomendação de subclasses de patentes de maneira ordenada pela sua relevância, ou seja, um *ranking*. Para tal, um modelo pré-treinado de Processamento de Linguagem Natural foi considerado com o intuito de representar patentes na forma de vetores densos (*embeddings*). Visando cumprir este objetivo utilizou-se um conjunto de patentes presentes no *dataset* USPTO-2M® referentes aos anos de 2014 e 2015, sendo utilizados para as fases de treinamento e teste do método, respectivamente. Para a análise do método foram estabelecidos três cenários específicos e um cenário geral. No primeiro caso, apresentou-se de maneira detalhada como a recomendação de subclasses na forma de *ranking* funciona, além de analisar exemplos pontuais com o intuito de clarificar os resultados gerais obtidos. Por outro lado, o cenário geral apresentou os resultados com diferentes configurações na quantidade de subclasses recomendadas (k) para diferentes níveis de recuperação de documentos de patentes (n). Os resultados se mostram consistentes, atingindo uma acurácia em torno de 80% para valores de k entre 5 e 6, levando em conta um n de 50 documentos retornados para a análise e composição do *ranking* de subclasses. Diante dos resultados, o método proposto de recomendação ordenada de subclasses mostrou-se viável, com potencial para auxiliar examinadores na tarefa de classificação de patentes.

Palavras-chave: Análise de patentes. Classificação de patentes. Classificação multi-saída. Incorporação de palavras.

[1] rodrigobuenoguedes@gmail.com
[2] a.l.goncalves@ufsc.br

Recomendação ordenada de classes de patentes por meio da técnica de Embedding

Ranking-based patent class recommendation using Embedding technique

Rodrigo Bueno Guedes[3] Alexandre Leopoldo Gonçalves[4]

2023, Julho

Abstract

Patents are a valuable information source regarding technological inventions and possibilities of analysis in various scenarios in order to assist strategic decision making, especially those focused on research, development and innovation. Furthermore, they provide inputs for a wide range of tasks, among them the task of patent classification. However, due to the number of categories involved in the evaluation process by examiners, patent classification is a costly and difficult activity. In this context, the present work proposes the development of a classification method based on Deep Learning aiming to recommend patent subclasses in an orderly manner by their relevance, i.e., a ranking. To this end, a pre-trained Natural Language Processing model was considered in order to represent patents in the form of dense vectors (embeddings). In order to fulfill this goal, a set of patents from the USPTO-2M® dataset referring to the years 2014 and 2015 was used for the training and testing phases of the method, respectively. For the analysis of the method, three specific scenarios and a general scenario were established. In the first case, it was presented in detail how the recommendation of subclasses in the form of ranking works, in addition to analyzing specific examples aiming to clarify the overall achieved results. On the other hand, the general scenario presents the results with different configurations in the number of recommended subclasses (k) for different levels of patent document retrieval (n). The results are consistent, reaching an accuracy around 80% for k values between 5 and 6, taking into account an n of 50 documents returned for the analysis and composition of the subclass ranking. Given the results, the proposed method of subclass ranking recommendation proved to be feasible, with potential to assist examiners in the task of patent classification.

***Keywords*:** Patent analysis. Patent classification. Multi-output classification. Embedding.

[3] rodrigobuenoguedes@gmail.com
[4] a.l.goncalves@ufsc.br

1. INTRODUÇÃO

Não é de hoje que patentes são importantes no mundo moderno desempenhando um papel crucial na promoção da inovação em diversos setores, incentivando a divulgação de conhecimentos e a colaboração entre empresas e instituições de pesquisa, bem como impulsionando o avanço científico e tecnológico. Segundo Yun e Geum (2020), as patentes têm sido consideradas um ativo intangível importante, por protegerem os direitos sobre a inovação tecnológica e invenção, tornando-as especialmente importantes no ambiente empresarial, visto que a maioria das inovações provém do desenvolvimento tecnológico.

Tendo em vista a importância das patentes, faz-se necessário meios que auxiliem na sua análise permitindo a tomada de decisão em diferentes níveis e organizações, tanto nos chamados Escritórios de Patentes, por exemplo, o Escritório de Patentes e Marcas dos Estados Unidos (do inglês - *United States Patent and Trademark Office* - USPTO®) (ZAINI; LAI; LIM, 2022), quanto em departamentos de pesquisa e desenvolvimento de empresas. O termo Análise de Patentes (do inglês *Patent Analysis* - PA) refere-se a revisão sistemática e detalhada de documentos patentários para extrair informações relevantes sobre tecnologia, inovação e Propriedade Intelectual (do inglês *Intellectual Property* - IP). Essa análise é realizada por profissionais especializados e, segundo Krestel *et al.* (2021), é possível classificá-las em oito tarefas, onde cada uma delas tem uma forma diferente de ser abordada.

Dentre as tarefas de PA a serem automatizadas, a classificação de patentes, onde os documentos de patentes são categorizados hierarquicamente baseados no seu campo de invenção, está entre as mais proeminentes (KRESTEL *et al.*, 2021). Todavia, apesar de tal relevância, os documentos ainda passam por análises manuais e o veredito final da classificação é atribuído por examinadores, acarretando em uma demora significativa para na realização desta tarefa, produzindo uma sobrecarga de trabalho em futuras avaliações de novas patentes (DORAN; WEBSTER, 2019).

Anualmente, o número de patentes tem aumentado de maneira substancial. Segundo a Organização Mundial da Propriedade Intelectual (do inglês *World Intellectual Property Organization* - WIPO®), houve um aumento de 10% no registro de patentes concedidas no ano de 2021, a maior taxa de crescimento registrada desde 2012 (WIPO, 2022). Com este aumento significativo, elaborar estratégias eficientes e com bom desempenho em relação ao tempo de processamento para classificar novas patentes, ainda constitui-se como um desafio, visto que a demora no processo de classificação pode impactar no valor e até mesmo na própria aplicação de determinada patente (RUIJIE *et al.*, 2021).

A WIPO® também é responsável por fornecer a Classificação Internacional de Patentes (do inglês - *International Patent Classification* - IPC), que prevê um sistema hierárquico de símbolos independente da linguagem utilizada para classificar patentes e modelos de utilidade de acordo com diferentes áreas (WIPO, 2023). A taxonomia[5] IPC possui uma estrutura complexa distribuída em cinco níveis em que cada patente pode ser classificada em uma ou mais categorias. É possível ainda destacar duas vantagens com o uso dessa taxonomia. A primeira reside no fato das classificações serem baseadas em conceitos ao invés de palavras e, a segunda, devido ao fato da taxonomia ser publicada em diferentes línguas, isto proporciona ao examinador um nível mais acurado de entendimento semântico (MEGURO; OSABE, 2019).

Neste sentido, diferentes técnicas têm sido utilizadas na Análise de Patentes, principalmente àquelas relacionadas ao Aprendizado Profundo (do inglês *Deep Learning* - DL). O DL vem desempenhando um papel fundamental nos últimos anos neste contexto

[5] Taxonomia: Ciência que se dedica à classificação; técnica de classificação, ou de distribuição sistemática em categorias: taxonomia gramatical. https://www.dicio.com.br/taxonomia/

através de Redes Neurais Artificiais (do inglês *Artificial Neural Networks* - ANN), aplicadas principalmente na tarefa de classificação de patentes. Entre as arquiteturas de ANNs podem-se citar as: *Recursive Neural Networks* (RNN) (HE; SCHOMAKER, 2021), *Convolutional Neural Networks* (CNN) (LIN *et al*. 2018), *Long-Short Term Memory* (LSTM) (CHEN *et al*. 2022) e *Bidirectional Encoder Representation from Transformers* (BERT) (DEVLIN *et al*. 2019), com destaque para esta última, pois serve de base para o modelo utilizado neste trabalho.

Visto que existem diversas técnicas e abordagens capazes de auxiliar na tarefa de classificação de patentes, o presente trabalho propõe um método voltado à recomendação de subclasses de patentes de maneira ordenada por meio de diferentes técnicas, entre elas as ANNs e *embeddings*. O resultado final do método tem, portanto, o objetivo de subsidiar examinadores na tarefa de análise e classificação de patentes. Ademais os objetivos específicos são: a) identificar possíveis métodos e técnicas de classificação de patentes; b) identificar modelos de linguagem natural capazes de transformar texto em representações vetoriais densas (*embeddings*); e c) desenvolver um protótipo que permita avaliar o método proposto por meio de cenários de estudo.

Este documento é composto por seis sessões, sendo a primeira a introdução. A segunda seção apresenta a fundamentação teórica sumarizando os conceitos relevantes para o entendimento do tema abordado. Na terceira seção são apresentados os trabalhos correlatos à classificação de patentes com abordagens semelhantes às propostas neste trabalho. Na quarta seção é detalhado o método proposto e, em seguida, na quinta seção, são discutidos os resultados obtidos pelo método. Por fim, na última seção, são apresentadas as considerações finais e possíveis trabalhos a serem realizados no futuro.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. ANÁLISE DE PATENTES

Com o aumento da globalização e evolução das Tecnologias da Informação e Comunicação (TICS) ficou muito mais fácil e rápido o acesso e disponibilização de informações, ideias e tecnologias. Entre essas informações encontram-se as patentes, vistas como uma referência confiável de inovação e tecnologia para as empresas devido a sua característica de proteger os direitos relativos à inovação tecnológica (YUM; GEUM, 2020). Segundo Abbas, Zhang e Khan (2014), o volume de patentes cresce significativamente a cada ano, ou seja, aumenta cada vez mais a quantidade de dados técnicos pertinentes à invenções, dificultando a análise de patentes por examinadores, principalmente devido à complexidade das taxonomias. Neste contexto, a área de Análise de Patentes (PA) possui expressiva relevância podendo ser definida como uma disciplina especializada que envolve a investigação e avaliação de documentos de patentes para obter informações relevantes sobre o estado da arte, a proteção legal e a viabilidade técnica de uma determinada invenção (CLARKE, 2018; KAHRAMAN; DERELI; DURMUŞOĞLU, 2023).

No intuito de promover o gerenciamento da Propriedade Intelectual (do inglês *Intellectual Property* - IP) a WIPO®, fundada em 1967, é responsável por incentivar a inovação e a criatividade, bem como facilitar a cooperação entre os países membros no campo da IP. Conforme os relatórios presentes na WIPO®, os países que mais solicitaram registro de patentes no ano de 2020 foram China (CNIPA®), Estados Unidos (USPTO®) e Japão (JPO®) e, segundo a WIPO (2021), cerca de dois-terços das atividades de registro de patente global se encontram na Ásia.

Mundialmente existem diversas agências ou escritórios que realizam a análise de patentes e, a partir disso, disponibilizam repositórios públicos visando facilitar o acesso e o

estudo dentro do domínio de patentes. Entre as principais agências destacam-se a *China National Intellectual Property Administration* (CNIPA®), a USPTO®, citada anteriormente, e o *Japan Patent Office* (JPO®) (MEGURO; OSABE, 2019). Juntos estes repositórios fornecem uma fonte de inestimável valor para as mais diferentes tarefas na área de análise de patentes.

Para um melhor entendimento da estrutura de uma patente é apresentada a Figura 1 retirada do repositório USPTO®. É possível verificar alguns itens importantes, tais como: (12) o repositório de origem da patente; (10) o número da patente; (45) a data de publicação; (54) o título da patente; (75) o nome do inventor; (51) as subclasses atribuídas à patente; e (57) o resumo da patente.

Figura 1: Exemplo de uma patente


US008022732B2

(12) **United States Patent**
Nugent

(10) **Patent No.: US 8,022,732 B2**
(45) **Date of Patent: Sep. 20, 2011**

(54) **UNIVERSAL LOGIC GATE UTILIZING NANOTECHNOLOGY**

(75) Inventor: **Alex Nugent**, Santa Fe, NM (US)

(73) Assignee: **Knowmtech, LLC**, Albuquerque, NM (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 718 days.

(21) Appl. No.: **12/143,803**

(22) Filed: **Jun. 22, 2008**

(65) **Prior Publication Data**

US 2008/0258773 A1 Oct. 23, 2008

Related U.S. Application Data

(62) Division of application No. 11/449,321, filed on Jun. 8, 2006, now Pat. No. 7,420,396.

(60) Provisional application No. 60/692,109, filed on Jun. 17, 2005.

(51) **Int. Cl.**
H03K 19/20 (2006.01)
H03K 19/00 (2006.01)

(52) **U.S. Cl.** **326/104**; 326/136

(58) **Field of Classification Search** 326/104, 326/136; 977/773, 882, 932, 933, 936, 938, 977/940

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

7,219,018	B2 *	5/2007	Vitaliano et al.	702/19
2001/0044114	A1 *	11/2001	Connolly	435/6
2005/0059167	A1 *	3/2005	Vitaliano et al.	436/518

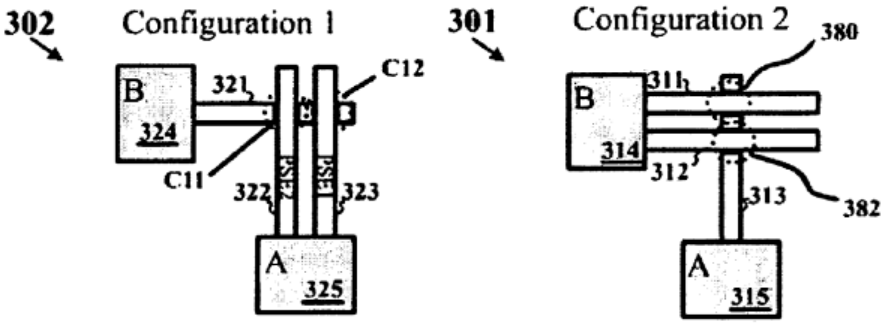
* cited by examiner

Primary Examiner — Shawki S Ismail
Assistant Examiner — Thienvu V Tran
(74) *Attorney, Agent, or Firm* — Kermit D. Lopez; Luis M. Ortiz; Ortiz & Lopez, PLLC

(57) **ABSTRACT**

A universal logic gate apparatus is disclosed, which include a plurality of self-assembling chains of nanoparticles having a plurality of resistive connections, wherein the plurality of self-assembling chains of nanoparticles comprise resistive connects utilized to create A plasticity mechanism is also provided, which is based on a plasticity rule for creating stable connections from the plurality of self-assembling chains of nanoparticles for use with the universal, reconfigurable logic gate. The plasticity mechanism can be based, for example, on a 2-dimensional binary input data stream, depending upon design considerations. A circuit is also associated with the plurality of self-assembling chains of nanoparticles, wherein the circuit provides a logic bypass that implements a flip-cycle for second-level logic. Additionally, an extractor logic gate is associated with the plurality of self-assembling chains of nanoparticles, wherein the extractor logic gate provides logic functionalities.

12 Claims, 21 Drawing Sheets



Fonte: USPTO[6].

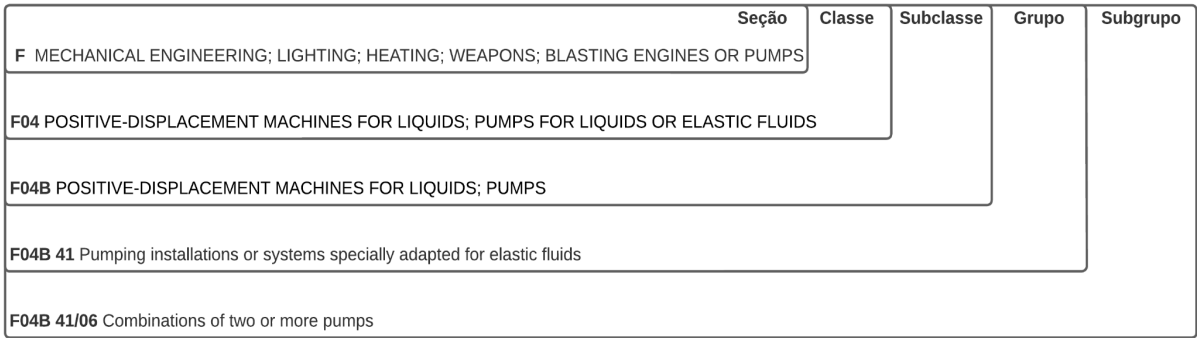
Entre as diferentes tarefas no âmbito da PA a classificação de patentes possui destaque. Para tal, existem dois esquemas populares para classificar esses conjuntos de dados

[6] https://patents.google.com/patent/US8022732B2/en?q=(nanotechnology)&oq=nanotechnology

sendo a Classificação Cooperativa de Patentes[7] (do inglês *Cooperative Patent Classification* - CPC) e a organização IPC (DEGROOTE; HELD, 2018), que será utilizada neste trabalho.

O IPC[8] é o mais popular a nível mundial, sendo que mais de 100 países utilizam-na para classificar seus pedidos nacionais de patentes. A IPC foi estabelecida em 1971 baseado no Tratado de Cooperação em Matéria de Patentes (do inglês *Patent Cooperation Treaty* - PCT) e é composto por cerca de 80000 categorias que cobrem os mais diferentes ramos de invenção (SOFEAN, 2021). Especificamente, a taxonomia IPC é constituída de seções, classes, subclasses, grupos principais e subgrupos. Na taxonomia IPC, a parte da seção é representada pelas letras maiúsculas de A a H, onde cada letra abrange uma categoria de tecnologia diferente. Já a camada de classes é representada por números que variam de 0 a 99 com o intuito de promover uma maior especificação da tecnologia apresentada na seção, sendo ainda dividida em subclasses que vão de A a Z. Por fim, vêm as camadas inferiores compostas pelo grupo principal e subgrupo (WIPO, 2023). A Figura 2 exemplifica o modelo da taxonomia IPC.

Figura 2: Modelo da taxonomia IPC



Fonte: Elaborado pelo autor (2023).

No âmbito da análise de patentes, a classificação é de longe a tarefa mais popular. Uma das razões se deve pela disponibilidade de grandes quantidades de dados, mais especificamente de documentos de patentes que podem possuir um ou mais rótulos (classes[9]) (KRESTEL *et al.*, 2021). De maneira geral, pode-se entender a classificação de patentes como uma tarefa de classificação de documentos/textos considerando múltiplos marcadores. Desta forma, constitui-se um desafio visto que o número desses marcadores podem ser expressivos, chegando a ter mais de 630 a níveis de subclasse e cerca 80000 subgrupos (LEE; HSIANG, 2020; SOFEAN, 2021).

Segundo Krestel *et al.* (2021) e Kalip, Erzurumlu e Gün (2022), uma boa parte dos estudos de patentes ainda requer que especialistas realizem um serviço trabalhoso sobre um numeroso volume de dados. Todavia, a automatização de tarefas na análise de patentes por meio de técnicas de Inteligência Artificial (do inglês *Artificial Intelligence* - AI) e Aprendizado de Máquina (do inglês *Machine Learning* - ML) pode reduzir o esforço despendido por especialistas acelerando o processo como um todo. Krestel *et al.* (2021) identificam as tarefas que mais se destacam e as divide em oito principais grupos, sendo: i) tarefas de suporte que ajudam no pré-processamento e extração de informações para análises futuras; ii) a recuperação de patentes, que é dividida em pesquisa de estado da arte, pesquisa automatizada

[7] https://www.cooperativepatentclassification.org
[8] https://www.wipo.int/classifications/ipc/en/faq/
[9] Aqui o termo "classes" serve de referência geral, visto que a tarefa de classificar uma patente configura-se na atribuição de um ou mais subgrupos. Dependendo do conjunto de dados categorização resino no nível de subclasses.

de patentes, pesquisa de infrações, pesquisa de liberdade de exploração de determinado nicho e recuperação de trechos específicos de patentes; iii) a avaliação de patentes visando uma predição de seu valor no mercado; iv) a predição de tecnologia, onde as patentes são usadas para avaliar um panorama tecnológico; v) a geração de texto de patente, que visa automatizar a escrita das reivindicações das patentes através da estrutura e estilo incorporado nos documentos publicados; vi) a análise de litígio, um processo legal onde as potenciais patentes são conduzidas a uma disputa ou litígio entre duas empresas proibindo o desenvolvimento de estratégias comerciais; vii) a visão computacional que trabalha com figuras e desenhos de documentos de patentes; e por último, viii) a classificação de patente, onde documentos da patente são categorizados hierarquicamente baseados em sua área de invenção. Aderente a esta última tarefa, este trabalho possui a classificação de texto (patente) como seu objetivo principal.

2.2. CLASSIFICAÇÃO DE TEXTO

A classificação de texto fundamentalmente é uma tarefa de Processamento de Linguagem Natural (do inglês *Natural Processing Language* - NLP) que lida com atribuições automáticas de um único ou múltiplos rótulos pré-definidos de um dado texto ou documento. Ao longo dos anos a classificação de textos tem sido aplicada a diferentes áreas a fim de promover uma resposta mais adequada à dispendiosa atividade da etiquetagem manual (ABDELGAWAD *et al.*, 2019). Conforme Conneau *et al.* (2017), o objetivo da NLP é processar textos através de computadores com o intuito de analisá-los, extrair informações e eventualmente representar essa mesma informação de forma diferente. Afirmam ainda que pode-se querer associar categorias a partes do texto, estruturar o texto de diferentes formas, ou converter para uma outra forma que preserve todas as partes do conteúdo ou o sentido, por exemplo, a tradução de máquina ou a sumarização.

Dentro do contexto de ML, a classificação de texto pode ser dividida em quatro tipos diferentes, sendo eles a classificação binária, multiclasse, multi-rótulos e multi-saídas, sendo esta o foco deste trabalho.

A classificação binária consiste essencialmente na distinção entre duas classes diferentes, podendo ser elas 0 ou 1, ou qualquer outro conjunto de valores binários. A principal característica desta classificação consiste na escolha por determinado algoritmo entre duas possibilidades distintas de classes após a análises de atributos que representam determinada instância. Um exemplo deste caso seria o diagnóstico médico para uma única condição médica, sendo a resposta positiva ou negativa, baseada em um conjunto de características.

Enquanto a classificação binária consegue distinguir entre duas classes, na classificação multiclasse o texto é atribuído a uma de várias categorias exclusivas. Pode-se levar em conta o trabalho de Guleria *et al.* (2022) que busca detectar e classificar o hipotireoidismo em quatro classes diferentes através de diversos classificadores baseados em ML visando um melhor estudo e análise do problema.

A classificação de multi-rótulos sai do padrão das duas anteriores em que cada instância de entrada é atribuída para somente uma classe. Segundo Xinzheng *et al.* (2019), esta abordagem de classificação fornece um cenário onde o classificador pode indicar múltiplas classes para cada instância. Um exemplo desta classificação são os gêneros de filmes, onde esses podem ser classificados em mais de uma categoria, ou seja, um filme pode ser considerado como Horror, Sci-fi e Suspense, por exemplo.

Por último, tem-se a classificação multi-saída, sendo considerada uma aprimoramento da classificação de multi-rótulos, mas cada um do(s) múltiplo(s) rótulos associados a uma instância em particular possui uma relevância, uma probabilidade ou um intervalo de valor. Segundo Géron (2019), uma maneira de ilustrar este tipo de classificação é através de um

sistema que remove o ruído de imagens, onde a entrada é uma imagem de dígito ruidoso e a saída é uma imagem de dígito limpo. É possível perceber que a saída do classificador é considerada multi-rótulos (um rótulo por pixel), e cada um desses pixels rotulados contêm valores variando sua intensidade de 0 a 255. Tratando-se, portanto, de um exemplo de sistema de classificação multi-saída. Vale destacar que o próprio cenário de estudo deste trabalho caracteriza-se como multi-saída, visto que uma patente pode ser atribuída para várias classes em que cada uma dessas classes possui uma relevância associada.

A diferença nas definições de tarefas, sejam elas classificações mono-categóricas ou multicategóricas, sendo usadas em diferentes níveis dentro da hierarquia de classificação junto com as escolhas dos conjuntos de dados, fazem o objetivo de avaliar e comparar os métodos existentes uma tarefa desafiante (LI *et al.,* 2018).

Adicionalmente, menciona-se que uma das principais etapas da classificação de texto é a representação do texto ou a extração de características, consideradas fundamentais para se atingir resultados relevantes nesta tarefa. Diferentes esquemas de representação de texto fornecem diferentes características no momento da extração, o que impacta na acurácia da classificação de texto. Entre as representações atualmente utilizadas e que promovem vantagens na classificação de texto, principalmente, ao evitar a perda semântica durante a transformação do texto original para uma forma computacional são os *embeddings*, discutidos a seguir.

2.2.1. EMBEDDINGS

A incorporação de palavras (do inglês - *Word Embeddings* - *WE*) pode ser entendida como a geração de um vetor denso que captura o significado semântico das palavras, sendo aplicável em diferentes tarefas de NLP (RISCH; KRESTEL, 2018). Todavia, este conceito pode ser generalizado para outras unidades de informação, por exemplo, para sentenças, parágrafos, textos, ou mesmo, imagens, áudios, vídeos e estruturas em grafos. Com a utilização de *embeddings* espera-se uma eficácia maior na aplicação de algoritmos de ML para grandes conjuntos de dados, visto que cada vetor denso de uma unidade de informação, em geral, possui uma representação reduzida em relação ao vetor original, ainda assim, expandindo o significado (semântica) da representação original. Segundo Géron (2019), *embeddings* conseguem executar diversas tarefas de NLP. Ademais, uma vez que determinado modelo de um domínio em particular tenha sido treinado para produzir os *embeddings,* este pode ser salvo e reutilizado de maneira bastante eficiente em diferentes contextos de NLP.

O conceito de incorporação de palavras segundo Incitti, Url e Snidaro (2023) é um dos mais poderosos dentro da ciência do aprendizado profundo relacionado à aplicações de NLP. Um conjunto de vetores de palavras para um vocabulário é capaz de capturar o seu significado e também a relação entre elas e o seu contexto. As representações vetoriais incorporam palavras dentro de um espaço de características, recebendo assim o nome de “incorporação de palavras”. A incorporação de palavras possui um número interessante de propriedades:

- Representação única ou múltiplas para cada palavra, dependendo do modelo de NLP utilizado;
- Vetores com algumas centenas de dimensões, considerados relativamente baixos se comparado com outros modelos que chegam aos milhares, por exemplo o *Bag of Words* (BoW) (LIU *et al.*, 2020).;
- Palavras com um significado similar possuem vetores com valores similares e estão assim próximas do espaço de características *n*-dimensionais;

- As regularidades semânticas (que se referem a padrões ou relações semânticas existentes entre as palavras, regularidades essas que são capturadas por modelos de incorporação de palavras) correspondem às propriedades geométricas.

Apesar de todas as vantagens apresentadas, ainda é importante ressaltar que a incorporação de palavras possui o inconveniente de não serem interpretáveis por humanos. Apesar desta limitação, a abordagem de *word embeddings* constitui-se em uma ferramenta amplamente usada e uma das mais revolucionárias dentro da NLP. A Tabela 1 demonstra através de dados ilustrativos o processo de transformação do texto em um vetor denso (*embedding)*. Neste sentido, um embedding que representa a palavra "London" indica que as características "Cidade", "Turismo" e "Idioma Inglês" são relevantes, caracterizando como uma cidade, com potencial de turismo em que o idioma inglês é o predominante. Isto determina a representação semântica da unidade, posicionando-a em um espaço *n*-dimensional próxima a outras cidades com características similares.

Tabela 1 - Representação de palavras e seus atributos

	Cidade	**País**	**Idioma Português**	**Turismo**	**Idioma Inglês**
London	0.88	0.06	0.05	0.70	0.95
Inglaterra	0.15	0.90	0.13	0.80	0.94
Brasília	0.83	0.11	0.95	0.35	0.10
Brasil	0.14	0.78	0.97	0.83	0.15

Fonte: Elaborado pelo autor (2023).

Adicionalmente, para uma melhor compreensão, existe um exemplo bem conhecido utilizando as palavras Rei, Homem e Mulher (GÉRON, 2019), onde ao se adicionar e subtrair seus vetores de incorporação, tem-se um resultado bem próximo do *embedding* da palavra Rainha. Ou seja, a incorporação de palavras codifica o conceito de gênero. Fazendo uma análise similar pode-se computar Madrid, Espanha e França e o resultado levaria próximo a Paris, o que indica que a noção de cidade capital também está codificada dentro dos *embeddings*.

O Quadro 1 apresenta um exemplo de patente contendo o número de identificação e o resumo, assim como o *embedding* gerado a partir do resumo. É importante observar que as posições do vetor denso gerado não possuem relação com as palavras ou outras unidades de informações presentes no texto de origem. Indicam sim, a posição no espaço *n*-dimensional, possibilitando comparar vetorialmente unidades de informações resultando em valores positivos (quanto mais positivo maior a similaridade, sendo o limite 1) ou negativos (quanto mais negativo maior a dissimilaridade, sendo o limite -1).

Quadro 1 - Exemplo de uma patente e seu respectivo embedding

Número da Patente	US08910406
Resumo	*An pressure reducing component for a firearm that includes one or more vent apertures formed through a side wall or the forward assist cover of an upper receiver the vent aperture s may be completely open to allow gases to be vented from within the upper receiver or may be adjusted to allow a desired amount of gas to be vented from within the upper receiver.*
***Embedding* (incluídos os cinco valores iniciais e finais do vetor)**	*[0.022737016901373863, 0.11122982203960419, 0.005626704078167677, 0.013364208862185478, 0.08874276280403137, ..., -0.042180228978395 46, -0.023644492030143738, 0.056213777512311935, -0.05300694331526756, 0.01516404934227 4666]*

Fonte: Elaborado pelo autor (2023).

Levando em conta o contexto de *embeddings*, o DL tem ganhado destaque como uma poderosa ferramenta para resolver problemas complexos de NLP. A próxima seção discute DL, assim como uma das arquiteturas mais promissoras de ANN atualmente, a arquitetura *transformer*.

2.3. DEEP LEARNING

As redes neurais são estruturas criadas com o intuito de simular um comportamento similar ao do cérebro humano, em que as arquiteturas dessas redes podem ser multi camada ou camada única. Na camada única, um conjunto de entradas é diretamente mapeado para uma saída, utilizando uma variação generalizada de uma função linear; esta simples instanciação de rede neural também pode ser referida como *perceptron*. Nas redes neurais multicamadas, os neurônios são arranjados em camadas, onde as camadas de entrada e saída são separadas por um grupo de camadas ocultas, geralmente estando totalmente interconectadas formando uma arquitetura em camadas do tipo *feedforward* (AGGARWAL, 2018).

Um dos principais ramos da ML é o DL, que assim como o próprio nome sugere, realiza uma aprendizagem mais profunda se utilizando de várias camadas de processamento. O termo *learning* significa que os parâmetros das redes, ou seja, o peso que compõem as camadas são modificados de maneira a minimizar a função de perda para um conjunto de instâncias utilizadas durante a fase de treinamento da rede. A função de perda precisa ser diferenciável para que a redução do gradiente do erro possa ser utilizada para achar um mínimo local na função (KRESTEL *et al*., 2021), permitindo que o aprendizado de determinado conjunto de dados ocorra.

A principal diferença entre o ML e o DL está em como as características são extraídas. A abordagem tradicional do ML conta com a utilização de uma engenharia através da aplicação de vários algoritmos de extração, e após isso é aplicado o algoritmo de aprendizagem. Por outro lado, no caso do DL, as características são aprendidas automaticamente e são representadas hierarquicamente em múltiplos níveis. Existem diferentes tipos de abordagens onde o DL pode vir a ser utilizado, sendo um deles o aprendizado supervisionado, onde são utilizados dados rotulados no processo, semi-supervisionado ou parcialmente supervisionado, que ocorre a utilização do conjunto de dados parcialmente rotulados para a aprendizagem e o não supervisionado em que não há a presença de dados rotulados (ALOM. *et al*., 2019).

Como citado no começo deste trabalho, o DL tem um papel crucial na classificação de patentes, fornecendo uma abordagem com potencial para lidar com grandes volumes de dados textuais e identificar características relevantes para a categorização precisa de patentes. A capacidade das redes neurais profundas de aprender representações semânticas, reconhecer padrões complexos, capturar relações hierárquicas e melhorar continuamente seu desempenho torna essa abordagem extremamente valiosa para o campo da Propriedade Intelectual (LO; CHU, 2021) e a área de Análise de Patentes. Para tal, neste trabalho, será utilizado um modelo de DL baseado na arquitetura *transformer*, sendo descrita a seguir.

2.3.1. TRANSFORMER

A arquitetura *Transformer* foi desenvolvida dentro do contexto de tradução automática e está baseada em uma estrutura que se utiliza do formato codificador-decodificador (*encoder-decoder)* divididos em diversos blocos de *encoder* e *decoder*. Tem se mostrado uma das abordagens mais poderosas e eficazes para uma variedade de tarefas de NLP e Visão Computacional. Possui como mecanismo fundamental o conceito de auto-atenção (*self-attention)* para computar representações de entrada e saída, evitando limitações de dependências sequenciais encontradas em outras arquiteturas de ANNs. De modo geral, este mecanismo processa relações entre partes diversas em uma sequência de entrada qualquer, como o relacionamento entre palavras em uma sentença ou pontos em uma imagem, permitindo a geração de contextos. Vaswani *et al*. (2017) estão entre os primeiros autores a propor o modelo *transformers* usando camadas empilhadas de *self-attention* para o *encoder* e o *decoder*, ambos compostos por uma pilha de seis camadas idênticas.

Em uma visão geral, e com a ajuda da Figura 3, analisando a arquitetura *transformer* tradicional é possível dizer que (RUAN; JIN, 2022; VASWANI *et al.*, 2017):

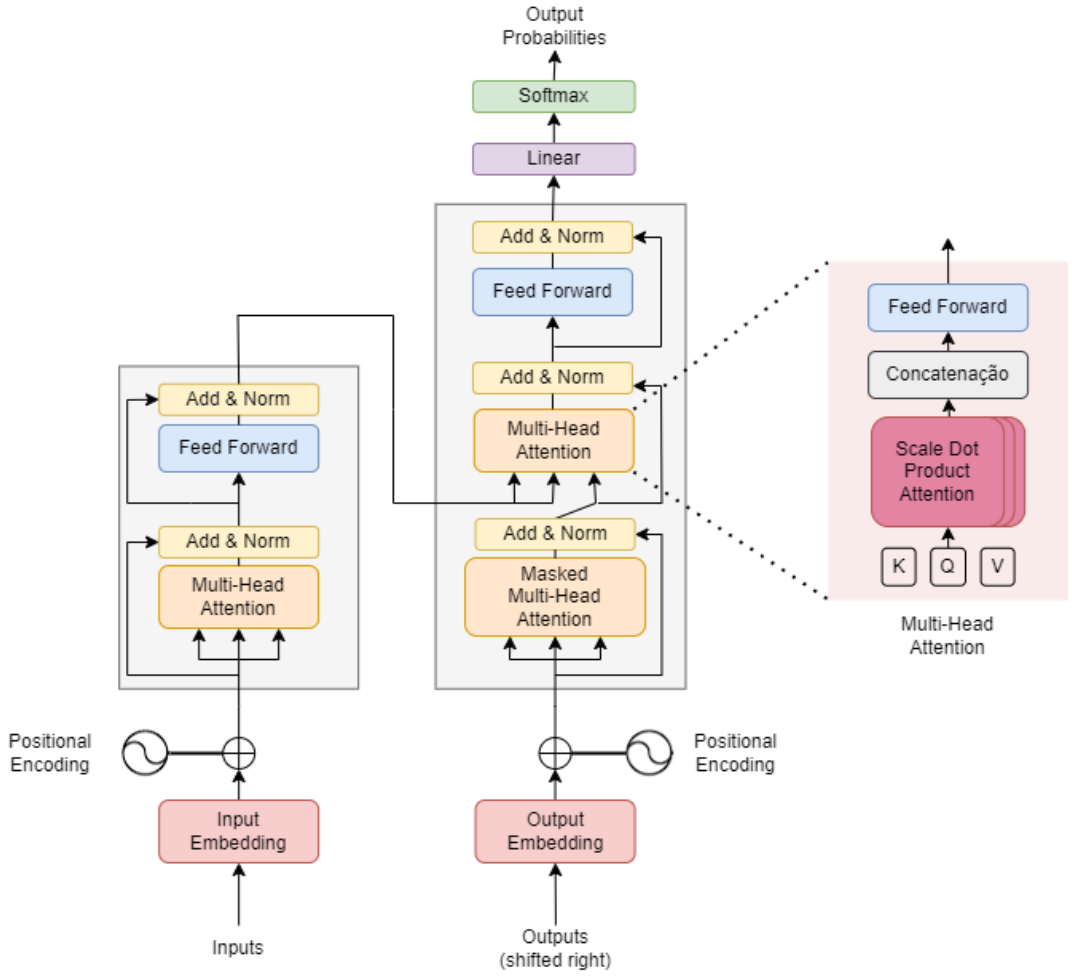
- O bloco *encoder* consiste de duas subcamadas, sendo a primeira um mecanismo de *self-attention* com várias cabeças (*multi-head attention*) e, a segunda, uma rede simples, totalmente conectada e com alimentação à frente (*feed forward)*. É empregada uma ligação residual em torno de cada uma das subcamadas, seguida por uma operação de normalização dos dados.
- O bloco *decoder* além de conter as duas subcamadas vistas no *encoder*, possui adicionalmente uma terceira subcamada de atenção do tipo *encoder-decoder* que executa a ação do mecanismo de *self-attention* com várias cabeças na saída da pilha do *encoder*. Similar ao *encoder* é realizada a ligação residual em torno de cada subcamada e a normalização da camada a fim de promover maior desempenho.

Analisando em mais detalhes a Figura 3 verifica-se que a arquitetura tem como entrada *embeddings*, onde o módulo Positional Encoding é responsável por adicionar, dentro do *embedding* de cada token[10] a informação da posição relativa ou absoluta. Após isso, o *embedding* de entrada passa pelo *encoder*, que recebe como entrada o *embedding* de cada um dos *tokens*, e dentro do *encoder* passa pelos dois mecanismos, o de atenção e o *feed forward.* O primeiro auxilia na geração dos *encodings* de saída através da criação de pesos de relevância para cada entrada, e o segundo processa cada *encoding* de saída individualmente enviado-o tanto para a entrada do próximo *encoder,* quanto para os *decoders*. O primeiro *decoder* recebe como entrada o *embedding* de cada *token* da sequência de saída, junto com a informação relativa ou absoluta do *token* na sequência, adicionada pelo *Positional Encoding*.

[10] Um *token* representa a menor unidade de análise em um texto, podendo ser uma palavra ou outro elemento indivisível no contexto de determinado domínio, por exemplo, uma data ou um URL.

O último *decoder* é seguido por uma transformação linear e uma camada *softmax* para produzir a saída de probabilidades sobre o vocabulário.

Figura 3: Modelo de uma arquitetura *transformer*



Fonte: Vaswani *et al*. (2017).

Por fim, após o treino de uma rede de arquitetura *transformers* os pesos que representam determinada unidade de informação, por exemplo, uma palavra, e o relacionamento com outras palavras, são armazenados na forma de *embeddings*, promovendo no contexto de NLP um LLM (do inglês *Large Language Model*), ou seja, um modelo pré-treinado. Atualmente, existem diversos LLMs, sendo o SBERT um deles (REIMERS; GUREVYCH, 2019). Desta forma, utilizando um LLM pode-se obter a representação de uma unidade de informação diferente, por exemplo, de uma sentença ou texto, na forma de um vetor denso (*embedding*). Esta abordagem facilita e agiliza o processamento de diversas tarefas de NLP, uma vez que evita que determinado modelo de aprendizado seja gerado para cada novo domínio de interesse. Ou seja, desta forma tem-se um modelo geral que, ao receber determinada entrada, produz um vetor denso que pode ser armazenado em uma estrutura que possibilite a realização de buscas vetoriais aproximadas. Ademais, este tipo de estratégia promove um ganho de desempenho, visto que o custo de processamento de gerar um novo vetor e armazená-lo em determinada estrutura de pesquisa terá um custo fixo. Em oposição, modelos tradicionais de redes neurais exigem o treinamento e ajuste fino do modelo constantemente, aumentando em muito os requisitos computacionais para se atingir os mesmos resultados obtidos por meio de um LLM.

2.3.1.1. BERT/SBERT

Devlin *et al.* (2019) introduziram uma arquitetura de pré-treinamento de modelos de linguagem baseada em redes neurais do tipo *transformers* chamada BERT (do inglês - *Bidirectional Encoder Representation from Transformers*). A arquitetura é baseada em um modelo de linguagem pré-treinado em grandes conjuntos de dados, também chamados de LLMs, que pode ser ajustado para tarefas específicas, por exemplo, a classificação de texto. O BERT é um modelo de linguagem que se utiliza de duas estratégias-chave para desempenhar tarefas de NLP: pré-treinamento bidirecional e máscara de *tokens*. Um dos principais benefícios do BERT é que ele não requer a criação de recursos específicos para cada tarefa de NLP. Em vez disso, utiliza o mesmo modelo de linguagem pré-treinado para várias tarefas, o que o torna altamente eficiente e escalável em vários cenários.

Segundo os estudos de Srebrovic e Yonamine (2020), o BERT é um modelo que marcou um grande avanço em NLP superando drasticamente as outras estruturas existentes através de uma série de tarefas de modelagem linguística. BERT é a primeira representação linguística profundamente bidirecional, sem supervisão, pré-treinada usando somente um simples corpo de texto (*corpus*). A estrutura do BERT permite captar o fato de que, dependendo do contexto em que a palavra é utilizada, ela pode obter significados bem diferentes, mesmo dentro do mesmo documento ou sentença.

Com a notoriedade do modelo BERT, outros estudos surgiram com diferentes abordagens usando este modelo como base, um deles é o SBERT[11] (*Sentence*-BERT). O SBERT foi proposto por Reimers e Gurevych (2019), onde é utilizada redes triplas e siamesas, sendo estas capazes de derivar frases com significados semânticos, permitindo que novas tarefas possam ser realizadas utilizando este modelo. Essas novas tarefas incluem a comparação de semelhanças semânticas em larga escala, agregação e recuperação de informações através de pesquisa semântica. Os autores ressaltam que a criação do SBERT objetivou resolver problemas de desempenho no processamento de sentenças longas. A arquitetura siamesa é essencial nessa tarefa, pois permite que vetores de tamanho fixo de sentenças de entrada possam ser derivados utilizando uma medida de semelhança como a similaridade do cosseno ou a distância de manhattan/euclidiana, sendo possível encontrar frases semanticamente semelhantes. A complexidade para encontrar o pares de frases mais semelhantes em uma coleção de 5 dígitos cai de horas para apenas alguns segundos, justificando o uso do SBERT no método proposto neste trabalho.

3. TRABALHOS CORRELATOS

Através de buscas na literatura científica foram encontrados e selecionados trabalhos referentes à classificação de patentes, utilizando a incorporação de palavras juntamente com modelos pré-treinados, bem como outras abordagens. As pesquisas foram realizadas nas seguintes bases de dados científicas: Scopus®, Web of Science®, ScienceDirect®, IEEE Xplore®, ACM Digital Library® e Spring Link®. Considerou-se artigos científicos em língua inglesa publicados entre os anos de 2018 até 2023 levando em conta a seguinte string de busca: ("*Patent Classification*" OR "*Patent Document Classification*" OR "*Patent Text Classification*" OR "*Patent Document Categorization*") AND ("*Deep Learning*" OR "*Neural Network*" OR "*Neural Networks*") AND ("*Transformer*" OR "*Transformers*"). A Tabela 2 apresenta a quantidade de artigos encontrados nas bases de dados utilizadas e, na sequência, estão descritos os trabalhos considerados mais relacionados ao tema.

[11] Projeto disponível em https://www.sbert.net/

Tabela 2 : Resultados da revisão da literatura

Base de Artigos Acadêmicos	Quantidade de Artigos Resultantes
ACM Digital Library®	10
Science Direct®	34
Scopus®	4
Web of Science®	4
IEEE Xplore®	1
Spring Link®	12
Todos os Resultados	**65**

Fonte: Elaborado pelo autor (2023).

A Tabela 2 apresenta os resultados obtidos através da expressão de busca. Do total de 65 trabalhos encontrados, 13 apresentaram títulos adequados ao objetivo da busca. Os resumos foram lidos na íntegra e 12 desses artigos foram selecionados para análise da introdução. No fim, os 8 trabalhos que possuem maior semelhança com o tema desta pesquisa foram escolhidos e são descritos a seguir.

O artigo publicado por Kang *et al.* (2020) apresenta uma abordagem para a realização de uma pesquisa de arte prévia (*prior art search*) buscando encontrar patentes válidas utilizando DL. A base de dados utilizada foi obitda através do site WIPS'[12] em março de 2016, incluindo patentes da Europa, China, Estados Unidos e do PCT, cada uma das patentes contendo título, abstract, número de aplicação e código padrão IPC. Realizando métodos para limpar a base de dados, como remoção de caracteres especiais números, foram selecionadas cerca de 10 mil patentes. Os autores ainda descrevem o uso do modelo BERT para a classificação de patentes válidas e inválidas, atingindo uma acurácia de 94,29%.

Li *et al.* (2022) propõem uma nova arquitetura para a classificação de texto utilizando uma combinação de *transformer* e CNN. A proposta utiliza uma arquitetura de dois níveis que permite uma compreensão do texto de entrada e uma maior precisão na classificação. No primeiro nível, uma arquitetura *transformer* é usada para codificar a entrada e capturar informações de contexto. No segundo nível, uma arquitetura CNN hierárquica é aplicada para extrair características de nível mais baixo e classificar as categorias. O desempenho do modelo proposto é analisado utilizando o conjunto de dados RCV1 e AAPD (sendo o primeiro o domínio das notícias e, o segundo, um domínio de artigos científicos e tecnológicos) considerados clássicos para a tarefa de classificação de múltiplas categorias e seus resultados experimentais mostram que a abordagem proposta supera outras abordagens de referência em diversas métricas de avaliação.

Já Jung, Shin e Lee (2023) abordam o impacto do pré-processamento e do *embedding* de palavras em tarefas de classificação de patentes com um conjunto de dados de aproximadamente 1 milhão de documentos de patentes sem pré-processamento vindos da Europa, Estados Unidos e Japão. Os autores investigaram como diferentes técnicas de pré-processamento como *stemming* e remoção das *stop-words*, afetam a qualidade da classificação e como diferentes modelos de incorporação de palavras, como o GloVe e o CBOW, influenciam no desempenho do modelo de classificação. Os resultados mostram que o pré-processamento de texto e a escolha da técnica de incorporação de palavras são cruciais para a precisão da classificação de patentes com múltiplas etiquetas.

Risch, Garda e Krestel (2020) propõem um novo método de classificação hierárquica de documentos, que utiliza modelos de geração de sequências para criar uma representação

[12] Disponível em: https://www.wipson.com/service/mai/main.wips

hierárquica de um documento. O método consiste em dividir o documento em seções, onde cada seção representa um nível hierárquico diferente. Em seguida, um modelo de geração de sequências é treinado para prever a próxima seção de um documento com base nas seções anteriores por meio de uma arquitetura do tipo *transformer*. Os experimentos realizados mostram que a abordagem proposta supera outras técnicas de classificação hierárquica em termos de precisão e eficiência.

Roudsari *et al* (2022) descreve um modelo de classificação de documentos de patente utilizando DP e NLP, o artigo usa um conjunto de dados não rotulados vindos do USPTO-2M® e o M-patent® ambos com a arquitetura IPC. Através do conjunto de dados é proposto uma nova abordagem para a classificação de patentes com múltiplas etiquetas chamado PatentNet, e utilizando modelos pré-treinados como BERT, XLNet, RoBERTa e ELECTRA ajustados para ajudar nessa classificação. O autor faz uma comparação de abordagens desses modelos pré-treinados com os que têm como base redes neurais, chegando a conclusão que em diversas métricas de avaliação a classificação com o uso de modelos pré-treinados se sobressai, atingindo resultados melhores mostrando ser um modelo eficaz.

Lee e Hsiang (2020) propõem uma abordagem de ajuste fino (do inglês - *fine-tuning*) do modelo pré-treinado BERT em uma base de dados com mais de 4 milhões de patentes. O ajuste é efetuado nos parâmetros do modelo pré-treinado para que ele possa ser utilizado para uma tarefa específica, como a classificação de patentes. Os resultados experimentais mostram que a abordagem proposta supera outras técnicas de classificação de patentes em termos de acurácia e *f1-score*, obtendo uma precisão média de 92% na classificação de patentes em 35 categorias diferentes. Os autores também compararam o desempenho do modelo BERT com outros modelos de classificação de patentes existentes e mostraram que esse modelo superou todos os outros modelos em termos de precisão. Além disso, a análise da importância das palavras-chave para a classificação de patentes sugere que o modelo BERT é capaz de aprender relações semânticas complexas entre as palavras.

Choi *et al.* (2022) propõe o uso de modelos baseados em *transformers* e *embeddings* de grafos para representar informações sobre patentes em uma estrutura flexível que possa ser facilmente adaptada a diferentes conjuntos de dados. O método proposto utiliza a arquitetura *transformer* para extrair recursos das descrições das patentes e o algoritmo de incorporação de grafos para capturar as relações entre as patentes. Esses recursos são usados para criar um modelo de ML capaz de identificar tendências tecnológicas emergentes em tempo real. O modelo é treinado em um grande conjunto de dados de patentes, para aprender a identificar padrões e relações entre os dados. Uma das principais vantagens do método proposto é sua capacidade de adaptar-se a possíveis mudanças de alguma tendência ao longo do tempo. À medida que novas patentes são concedidas e o conhecimento em um determinado campo evolui, o modelo pode ser atualizado para refletir essas mudanças e fornecer *insights* atualizados.

Por fim, Henriques, Ferreira e Castelli (2022) descrevem uma abordagem para classificar patentes usando DL com transferência de aprendizado. O estudo realizou experimentos em um conjunto de dados de patentes dos Estados Unidos, com resultados promissores em termos de precisão de classificação. O método proposto envolve o treinamento de um modelo de Aprendizado Profundo em um conjunto com grande volume de dados de patentes previamente classificadas. Em seguida, esse modelo é ajustado usando uma abordagem de transferência de aprendizado para se adaptar a um conjunto menor de patentes de interesse. Os resultados experimentais mostraram que o método proposto alcançou uma precisão de classificação superior a 90% para as classes existentes e uma precisão de cerca de 80% para uma nova classe adicionada ao modelo.

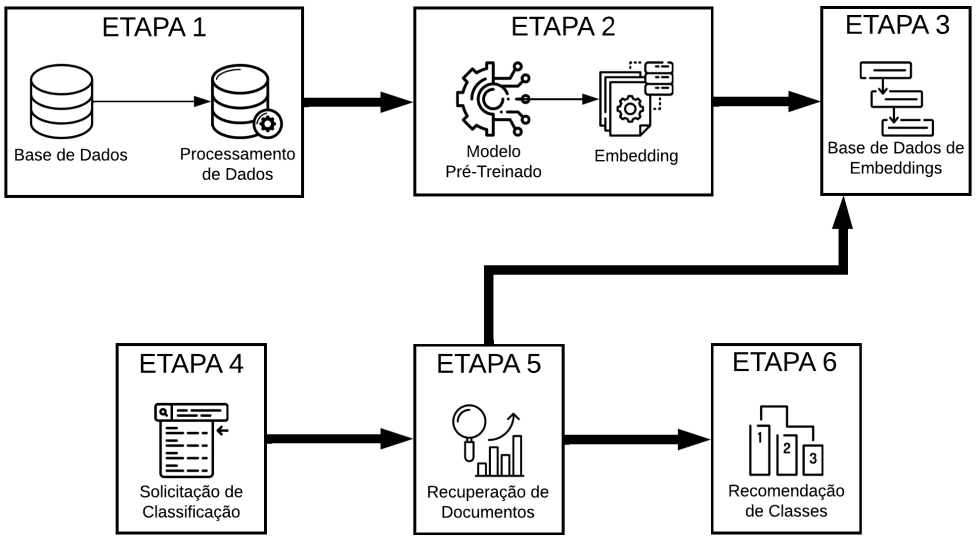
As pesquisas aqui resumidas apresentam um panorama da área de classificação de documentos de patentes e alguns de seus diferentes métodos. Apesar de muitas abordagens

serem semelhantes e existirem diversos métodos para a sua realização, não foram encontrados nestes artigos propostas baseadas no armazenamento e recuperação de *embeddings* utilizado-se de modelos pré-treinados de NLP para recomendar listas ordenadas de classes/subclasses de patentes levando-se em conta determinado texto de entrada (uma patente) de interesse, servindo assim de método de classificação de texto multi-saída. Outra característica presente no método proposto é que, ao utilizar o LLM para a geração de embeddings tem-se um custo computacional fixo (linear) à medida que mais documentos são adicionados à base de patentes já classificadas. Ademais, muitos estudos não especificam a quantidade de classes dos conjuntos de teste, visto que impacta no resultado final de acurácia de qualquer modelo. Maiores detalhes sobre o método proposto são promovidos na seção seguinte.

4. MÉTODO PROPOSTO

Esta seção detalha o método proposto para a recomendação ordenada de subclasses de patentes. Para tal, considerando um conjunto de documentos (*corpus*), cada instância é transformada em uma representação vetorial densa (*embedding*) e armazenada em um repositório (base de dados de *embeddings*). Adicionalmente, as subclasses de cada patente e outros dados também são armazenados. A partir disso, considerando determinada patente de interesse esta também é transformada em um *embedding* e comparada com os vetores mais similares do repositório. Os vetores retornados permitem então, através da utilização de alguma estratégia, elaborar uma lista ordenada de subclasses de patentes, possibilitando que estas sejam recomendadas como as mais aderentes para uma patente utilizada como entrada. A Figura 4 apresenta uma visão geral do método, sendo: i) Coleta e processamento de dados, ii) Transformação dos dados com o uso de um modelo pré-treinado para NLP; iii) Formação da base de dados de *embeddings*; iv) Solicitação de classificação a partir de uma nova patente; v) Recuperação de documentos (representados na forma de *embeddings*); e vi) Recomendação de subclasses de maneira ordenada.

Figura 4 - Fluxo (etapas) do método proposto



Fonte: Elaborado pelo autor (2023)[13].

[13] As imagens utilizadas na representação do método proposto foram obtidas a partir do site Flaticon.com.

4.1. ETAPA 1: Coleta e processamento de dados

A primeira etapa do método proposto é representada pela constituição da base de dados por meio da coleta de documentos de patentes para posterior indexação. A base de patentes utilizada neste trabalho foi disponibilizada publicamente e será descrita em detalhes na Seção 5.1. De modo geral, cada patente é composta por seu identificador, título, resumo e a relação de subclasses (Figura 5). Uma vez obtido o conjunto de dados, alguns passos são requeridos. Essencialmente, são retiradas pontuações de seus campos mais importantes e, na sequência, os campos de título e resumo são concatenados de modo que se tenha um conteúdo único de entrada para a devida transformação na etapa seguinte.

Figura 5 - Exemplo de um documento de patente


```
{
    "Subclass_labels": [
        "A43B",
        "A43C"
    ],
    "Abstract": "an upper for an article of footwear may have material layers and a plurality of strand segments the
    "Title": "methods of manufacturing articles of footwear with tensile strand elements",
    "No": "US08925129"
},
```


Fonte: Elaborado pelo autor (2023).

4.2. ETAPA 2: Transformação dos Dados através de um Modelo Pré-treinado para NLP

Esta etapa é caracterizada pela utilização de um modelo pré-treinado de NLP, também chamado de LLM. De maneira simplificada, estes modelos produzem, a partir de um *token* (por exemplo, uma palavra), uma sentença ou texto, um vetor denso ou *embedding*. Como já explicado na Seção 2, *embeddings* permitem uma representação semântica em um espaço *n*-dimensional habilitando a localização, a partir de um vetor de entrada, de vetores similares. A dimensionalidade de um *embedding* depende do LLM utilizado. Para este projeto utilizou-se o modelo "all-MiniLM-L6-v2"[14], devido principalmente ao seu tamanho em *megabytes* reduzido, baixo número de dimensões geradas em comparação com outros modelos (384[15] dimensões) e resultados expressivos em operações envolvendo similaridades vetoriais, principalmente, a similaridade pelo cosseno.. O Quadro 2 apresenta um exemplo de *embedding*.

Quadro 2 - Exemplo de um *embedding*

0,0203627	-0,0361268	0,0252454	0,0604007	0,0051878
0,0528063	0,0779504	0,0483506	-0,0281695	-0,0278436
		…		
-0,0619734	0,0201325	-0,0459939	-0,0341154	0,0015293
-0,0076992	0,0150959	-0,0461266	0,0590581	0,0650232

Fonte: Elaborado pelo autor (2023).

Salienta-se que o modelo "all-MiniLM-L6-v2" é utilizado na geração de *embeddings* a partir do texto das patentes (título e resumo) tanto para a formação da base de *embeddings*

[14] Informações básicas disponível em https://www.sbert.net/docs/pretrained_models.html
[15] A dimensionalidade de 384 refere-se à escolha realizada durante o desenvolvimento do modelo no projeto SBERT, considerando questões relativas à eficiência computacional e acurácia em diferentes cenários.

(etapa 3), quanto para a solicitação de classificação (etapa 4). Por fim, vale destacar que este modelo é instalado pela aplicação em sua primeira execução, ou seja, é realizado o *download* para o servidor ou computador local. Em execuções subsequentes a aplicação utiliza o modelo previamente instalado o que otimiza o tempo de geração dos *embeddings*.

4.3. ETAPA 3: Formação da base de dados de *embeddings*

Após a geração dos vetores densos (*embeddings*) considerando o conjunto de documentos patentes, estes devem ser armazenados para posterior consulta, permitindo recuperar os demais dados de uma patente. Neste trabalho, utilizou-se o banco de dados NoSQL (do inglês *Not only Structured Query Language*) com capacidade de armazenamento de vetores densos chamado ElasticSearch®. Esta classe de bancos de dados também é denominada por Bancos de Dados Vetoriais.

A Figura 6 apresenta o mapeamento de determinado documento (patente) no banco de dados, composto pelo número da patente (*no*), as subclasses de determinada patente (*subclass_labels*), o título (*title*), o resumo (*abstract*) e o vetor denso gerado a partir da concatenação de título e resumo (*embedding*). Além disso, pode-se perceber que cada campo na estrutura possui algumas propriedades, entre elas, o tipo (*type*) e a informação se o campo deve ou não ser indexado (*index*) para permitir buscas em texto completo (característica não utilizada neste trabalho) ou no vetor denso. Em relação ao tipo tem-se: a) *text* indicando que um texto longo será armazenado; b) *keyword* indicando que uma lista de *strings* será armazenada, geralmente utilizada para textos curtos, e; c) *dense_vector* indicando que um vetor denso será armazenado respeitando determinada dimensionalidade, neste caso, identificada pelo atributo *dims* com 384 posições. Existe ainda o atributo *similarity* que indica a métrica pela qual será estabelecida a comparação vetorial. No exemplo, utiliza-se a similaridade do cosseno (*cosine*). Por fim, o atributo *_source* indica quais campos devem ser excluídos do armazenamento. Isto reduz o tamanho do índice e aumenta a eficiência no momento da recuperação de documentos similares. Visto que um *embedding* não possui um significado explícito, tão pouco possui utilidade para a projeção/apresentação em alguma interface, este é indexado, mas não armazenado.

Figura 6 - Exemplo do mapeamento de documento na base de dados vetorial

```
mapping = {
    "_source": {"excludes": ["embedding"]},
    "properties": {
      "no":    { "type": "text", "index": False },
      "subclass_labels":    { "type": "keyword", "index": False },
      "title":  { "type": "text", "index": False  },
      "abstract":   { "type": "text", "index": False  },
      "embedding":   { "type": "dense_vector", "dims": 384, "index": True, "similarity": "cosine"  }
    }
  }
```

Fonte: Elaborado pelo autor (2023).

A partir do mapeamento do índice determinado documento pode ser composto e armazenado na base de dados. A Figura 7 apresenta um exemplo em formato JSON (*JavaScript Object Notation*) contendo o resumo (*abstract*), o número da patente (*no*), as subclasses da patente (*subclass_labels*), o título (*title*), o identificador (*_id*) gerado pelo banco de dados, o nome do índice (_index) em que o documento será salvo/armazenado e o *score* com valor padrão para igual a 1. Todavia, este valor irá se alterar em função de determinada consulta indicando a similaridade do documento em resposta a determinada consulta, ou seja, a determinada patente de interesse no contexto do trabalho. Ademais,

percebe-se também o campo *embedding*. Como mencionado anteriormente, este campo não é armazenado na versão final do índice, constando inicialmente somente para questões de conferência. De qualquer modo, é possível verificar que o campo é composto por um conjunto de valores que representam determinado documento (patente) em um espaço *n*-dimensional, onde *n* representa a quantidade de dimensões definidas no mapeamento do índice (Figura 6).

Figura 7 - Exemplo de uma patente indexada com *embedding*


```
{
  "abstract": ["a method is presented to address quantitative assessment of facial ..."],
  "embedding": [
    0.00925423577427864,
    -0.004118036478757858,
    ...
    -0.06906228512525558,
    -0.002283700043335557
  ],
  "no": [
    "US08777630"
  ],
  "subclass_labels": [
    "A61B",
    "G09B"
  ],
  "title": [
    "method and system for quantitative assessment of facial emotion sensitivity"
  ],
  "_id": "NHKKBYgBhUfUKtHdVBsf",
  "_index": "patents_tcc_rodrigo",
  "_score": 1
}
```


Fonte: Elaborado pelo autor (2023).

4.4. ETAPA 4: Solicitação de classificação a partir de uma nova Patente

Esta etapa é caracterizada pela utilização de determinada patente para a localização das patentes mais similares e, após isso, na recomendação (conforme explicação nas etapas seguintes) das suas possíveis subclasses. Uma vez recebida a patente como entrada, esta é enviada para a próxima etapa sendo transformada em um *embedding* levando-se em conta a concatenação do título e do resumo. O resultado será, assim como já apresentado no Quadro 2 e Figura 7, um vetor denso *n*-dimensional que objetiva permitir buscas aproximadas e sugerir, ao final do método, recomendações de subclasses.

4.5. ETAPA 5: Recuperação de documentos

Uma vez que o embedding tenha sido gerado, este é utilizado como elemento de entrada para uma consulta no banco de dados vetorial que irá retornar os *n* documentos mais similares. A consulta envolve basicamente quatro parâmetros que podem ser vistos na Figura 8, sendo o primeiro o atributo onde a consulta será executada (*field*), o segundo o próprio *embedding* (*query_vector*), o terceiro o número de documentos similares que serão retornados e, por fim, um número pré-definido de documentos candidatos a serem comparados com o vetor denso que representa a patente de entrada. Por fim, é possível definir quais atributos de cada documento serão retornados (neste caso o número da patente e as subclasses), servindo de base para a geração da recomendação na etapa seguinte.

Figura 8 - Exemplo de uma consulta enviada ao banco de dados vetorial

```
POST patents_tcc_rodrigo/_search
{
 "knn": {
  "field": "embedding",
  "query_vector": [0.00925423577427864, -0.004118036478757858, ...,
                -0.0272693932056427, 0.014938917011022568],
  "k": 100,
  "num_candidates": 200
 },
 "fields": [ "no", "subclass_labels"],
 "_source": false
}
```

Fonte: Elaborado pelo autor (2023).

4.6. ETAPA 6: Recomendação de classes (lista ordenada de classes)

A partir da lista de documentos retornados deve-se implementar uma estratégia de ordenação e recomendação de subclasses. Cada documento (representando uma patente) retornado possui uma ou mais subclasses, conforme pode ser observado na Figura 5.

A estratégia base implementada neste método considera a frequência com que as subclasses aparecem no conjunto de documentos recuperados, produzindo um histograma sendo ordenado da subclasse com maior frequência para a de menor frequência. A partir disso, deve-se determinar um número *k* de classes que serão recomendadas. Neste trabalho, aplicou-se uma estratégia simples em que o *k* é fixado. O Quadro 3 traz um exemplo mostrando o *ranking* resultante do método, sendo a primeira coluna *k* representando a posição da subclasse, a segunda coluna representando a subclasse recomendada, a terceira coluna mostrando a frequência com que as subclasses apareceram nos documentos e a quarta coluna mostrando a relevância que a aquela subclasse tem.

Quadro 3 - *Ranking* de subclasses resultante do método de recomendação

k	**Subclasse**	**Frequência**	**Relevância**
1	B01J	16	0.177
2	H01L	12	0.133
3	B05B	12	0.133
4	B32B	8	0.088
5	B08B	8	0.088
6	A61M	8	0.088
7	A61K	8	0.088
8	B05C	6	0.066
9	B01L	6	0.066
10	B01F	6	0.066

Fonte: Elaborado pelo autor (2023).

Considerações acerca dos resultados utilizando diferentes valores de *k* (número de subclasses recomendadas) e diferentes valores de *n* (número de documentos retornados na consulta), serão discutidos na próxima seção. Ademais, serão demonstrados exemplos de recomendação de subclasses levando-se em conta determinada patente de interesse.

5. RESULTADOS EXPERIMENTAIS

5.1. CENÁRIO DE ESTUDO E AVALIAÇÃO

O cenário de estudo deste trabalho envolve a recomendação ordenada de patentes, levando em conta sua classe/subclasse, servindo de auxílio aos examinadores de patentes. Ou seja, através de uma entrada de determinada patente são recomendadas as respectivas subclasses mais similares de maneira ordenada, da mais relevante para a menos relevante. Vale mencionar que a tarefa deste trabalho refere-se à classificação de patentes, enquanto que o objetivo principal diz respeito a recomendação de classes, mais especificamente de subclasses, visto que no conjunto de dados é esta a informação disponível e representa o 3º nível da taxonomia IPC.

Para a realização do trabalho foi utilizado dois conjuntos de dados disponibilizados por (LI *et al.*, 2018), sendo compostas por patentes obtidas a partir do USPTO-2M® referentes aos anos de 2014 e 2015. Esse conjunto de dados é amplamente utilizado no cenário de classificação de patentes e, apesar de ser constituído por patentes até o ano de 2015, possui um número expressivo de patentes e baseia-se na taxonomia IPC. O conjunto de 2014 possui ao todo 303.334 documentos de patentes (conforme Figura 5), sendo que, após o pré-processamento restaram 303.332 documentos, visto que 2 documentos não possuíam a indicação da(s) subclasse(s). Já o conjunto de 2015, possui um total de 49.900 documentos de patentes também em formato similar ao que consta na Figura 5, divididos em mais de 600 subclasses. Vale ressaltar que o conjunto de dados completo possui em torno de 2 milhões de patentes.

O primeiro conjunto de dados (2014) foi utilizado para o treinamento, isto é, no contexto do método proposto para a formação do banco de dados vetorial em que, cada documento, após a transformação em um *embedding* é armazenado para posterior consulta. Por outro lado, o segundo conjunto de dados (2015) serviu para a fase de teste do método proposto, em que cada documento, após a transformação, é consultado na base de dados retornando os documentos mais similares. A partir dos documentos retornados é formulada uma lista ordenada de subclasses permitindo avaliar se a(s) subclasse(s) da patente de entrada constam na lista de recomendação para determinado *k*. Em caso afirmativo é computado um acerto sendo incrementada em 1 (um) a variável *TA=total de acertos*, caso contrário, um erro sendo incrementada em 1 (um) a variável *TE*=total de *erros*. Analisando todos os documentos do conjunto de teste, ao final é estabelecida a acurácia do método conforme Equação 1.

$$Acurácia = \frac{TA}{TA + TE} \quad (1)$$

A fase de tese, conforme seção 5.3, ocorreu variando a recomendação de diferentes subclasses (variável *k*) em diferentes conjuntos de documentos de retorno (variável *n*).

5.2. IMPLEMENTAÇÃO DO MÉTODO PROPOSTO

O método proposto neste trabalho foi desenvolvido através da linguagem de programação Python® visto que esta possui um conjunto extenso de bibliotecas que facilitam diversas atividades de implementação em cenário de Análise de Dados, Aprendizado de Máquina e Inteligência Artificial como a própria biblioteca *SentenceTransformer*. Com foco nesta biblioteca, utilizou-se o LLM *all-MiniLM-L6-v2*, modelo pré-treinado do SBERT. Este LLM possibilita a transformação de determinado conteúdo textual em uma representação vetorial densa, ou seja, um *embedding*.

Os *embeddings* são então armazenados em um banco de dados NoSQL, mais especificamente um banco de dados com suporte para o processamento de vetores densos, chamado de ElasticSearch®. Este banco de dados permite a definição de um mapeamento de índice com suporte para dados estruturados e não estruturados na forma de texto, assim como representações vetoriais densas para qualquer tipo de dado não estruturado, isto é, textos, imagens, vídeos ou áudios. Ou seja, desde que determinado tipo possa ser transformado em um *embedding*, este pode ser armazenado para futuras consultas através de algum algoritmo de busca aproximada.

Desta forma, o ElasticSearch® utiliza o algoritmo kNN (*k-nearest neighbor*) para encontrar os *k* vetores mais próximos de um vetor de consulta em específico, realizando a comparação por meio de alguma métrica. No contexto do trabalho utilizou-se a métrica do cosseno (Equação 2) para determinar a similaridade entre o vetor de entrada (determinada patente) e os vetores armazenados na base de dados, sendo:

$$cos(\theta) = \frac{A.B}{||A||.||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (2)$$

onde o denominador trata da relação entre o produto de dois vetores (A) e (B), dividido pelo produto da norma desses dois vetores (A) e (B). Desta forma, a equação do cosseno determina a similaridade entre -1 e 1, sendo -1 totalmente dissimilar e 1 completamente similar.

5.3. APRESENTAÇÃO DOS RESULTADOS

Para a avaliação do método proposto e, adicionalmente ao cenário principal descrito na seção 5.1, foram estabelecidos alguns cenários específicos de modo a verificar e explicitar, a partir de algumas patentes selecionadas no conjunto de testes, o comportamento da estratégia de recomendação de subclasses. Ou seja, verificar se as subclasses da patente aparecem na relação recomendada, em que posição isto ocorre e as possíveis considerações sobre as recomendações efetuadas.

Para o primeiro cenário específico foi utilizada como entrada a patente representada no Quadro 4 com as subclasses A45F e A41D. Foi também definido um k=10, permitindo analisar se a(s) subclasse(s) da patente constam até a décima posição recomendada (Quadro 5) considerando os 50 documentos mais relevantes selecionados para a avaliação (n=50). Analisando o Quadro 5 é possível verificar que a somatória da frequência das subclasses da patente em questão, dentro dos 50 documentos analisados, é 59, onde as duas subclasses ao qual a patente pertence constam dentre as dez subclasses recomendadas (marcadas em verde no Quadro 5). A primeira subclasse A45F referente a patente, é recomendada na quinta posição, ou seja k=5, com uma relevância de 0,118 (11,8%) e presente em 7 documentos. Já a segunda subclasses A41D, é recomendada na oitava posição, ou seja k=8, com uma relevância de 0,050 (5%) e presente em 3 documentos.

Quadro 4 - Exemplo de patente com duas subclasses

Número da Patente	US08925115
Título	*Low profile medical kit*
Resumo	*A first aid systems for an ultra compact first aid pouch configured to fit behind the ballistic plates of a protective vest are disclosed in this configuration the first aid kit is protected from shrapnel and tearing is easily locatable and removable and does not affect the users freedom of movement when the first aid pouch is removed from its protected location it presents the first aid equipment in a logical and easily viewable manner low profile medical kit*
Subclasses	A45F e A41D

Fonte: Elaborado pelo autor (2023).

Quadro 5 - *Ranking* de subclasses obtido utilizando o método para a patente do Quadro 4

k	**Subclasse**	**Frequência**	**Relevância**
1	A61F	12	0.203
2	A61B	10	0.169
3	A61M	8	0.135
4	B65D	7	0.118
5	A41D	7	0.118
6	A42B	4	0.067
7	A62B	3	0.050
8	A45F	3	0.050
9	A45C	3	0.050
10	F41H	2	0.033

Fonte: Elaborado pelo autor (2023).

O resultado se mostra satisfatório, visto que ambas as subclasses são recomendadas, ainda que com uma relevância e frequência reduzidas. É possível notar que as subclasses mais recomendadas A61F, A61B e A61M com relevância de 0,203 (20,3%), 0,169 (16,9%) e 0,135 (13,5%), respectivamente, todas são referentes a seção da IPC-WIPO® de Necessidades Humanas (do inglês - *HUMAN NECESSITIES*), mais especificamente a classe A61 se refere à Ciência Veterinária ou Médica; Higiene (do inglês - *MEDICAL OR VETERINARY SCIENCE; HYGIENE*). Analisando o conteúdo presente no título e resumo verificam-se a presença de palavras voltadas à essa seção médica, visto que *medical* e *first aid* se repetem diversas vezes. Isto justifica as três primeiras posições do *ranking* relacionadas às subclasses A61F, A61B e A61M, ao invés das subclasses A41D e A45F que se referem à Vestuário de Proteção (do inglês - *WEARING APPAREL*) e Artigos de Mão ou Viagem (do inglês - *HAND OR TRAVELLING ARTICLES*), respectivamente. Entre as possibilidades deste fato, encontra-se a possibilidade de modificação da classificação da patente ao longo do tempo ou evolução da taxonomia de classes da IPC-WIPO®. Desta forma, uma patente previamente classificada em um conjunto de subclasses pode sofrer alterações. Como foi utilizado um conjunto de dados produzido há algum tempo, algumas divergências na classificação da patente podem ocorrer. Todavia, em função dos resultados gerais obtidos a partir do método, percebe-se que isto não promove impactos relevantes.

Seguindo o mesmo processo do primeiro cenário específico e considerando os mesmos valores de *k* e *n*, este teste utiliza a patente descrita no Quadro 6. Analisando o *ranking* de subclasses gerado no Quadro 7 é possível verificar que a frequência total das subclasses da patente em questão é 78, e que a subclasse G06K se encontra em primeiro lugar do *ranking* (marcada em verde no Quadro 7), ou seja, a subclasse teria sido recomendada corretamente considerando *k*=1, com relevância de 0,410 (41%) e presente em 32 documentos.

Quadro 6 - Exemplo de patente com uma subclasse

Número da Patente	US08942511
Título	*Apparatus and method for detecting object from image and program*
Resumo	*An image processing apparatus includes an input unit configured to input an image a determining unit configured to determine a foreground area and a background area in the image input by the input unit an expansion unit configured to expand the foreground area determined by the determining unit a calculating unit configured to calculate a feature amount of the foreground area expanded by the expansion unit and a detecting unit configured to detect an object from the image using the feature amount*
Subclasses	G06K

Fonte: Elaborado pelo autor (2023).

Quadro 7 - *Ranking* de subclasses obtido utilizando o método para a patente do Quadro 6

k	**Subclasse**	**Frequência**	**Relevância**
1	G06K	32	0.410
2	H04N	17	0.217
3	G06T	10	0.128
4	G09G	6	0.076
5	G06F	6	0.076
6	G03B	3	0.038
7	H04B	1	0.012
8	G08B	1	0.012
9	G02B	1	0.012
10	A61B	1	0.012

Fonte: Elaborado pelo autor (2023).

O resultado se mostra consistente e promissor, visto que a primeira subclasse sugerida apresenta uma relevância elevada dentro do *ranking*. Esta consistência ocorre devido a patente ser classificada em somente uma subclasse e, analisando a secção referente ao IPC-WIPO® de Física (do inglês - *PHYSICS*) e, mais especificamente, a subclasse G06 que se refere à Informática, Cálculo ou Contagem (do inglês - *COMPUTING; CALCULATING OR COUNTING*), percebe-se que o conteúdo presente no título e o resumo da patente está completamente aderente a subclasse G06. A ocorrência das palavras *calculating e calculate* reforçam esta constatação.

A análise do terceiro cenário específico foi baseada nas informações da patente presentes do Quadro 8, utilizando os mesmos valores de *k* e *n* apresentados anteriormente. Analisando o *ranking* de subclasses gerado no Quadro 9 é possível observar que a frequência total ficou em 44, ou seja, indicando que várias outras subclasses foram identificadas para além de *k*=10. Na composição do *ranking* verifica-se que quatro das seis subclasses presentes na patente foram recomendadas (marcadas em verde no Quadro 9). A primeira subclasse F04D é recomendada na segunda posição, tendo seu *k*=2, com uma relevância de 0,136 (13,6%) presente em 6 documentos. A segunda subclasse F01L aparece na quarta posição, tendo seu *k*=4, com uma relevância de 0,136 (13,6%), estando presente em 6 documentos. A terceira subclasse F04C é apresentada na quinta posição, tendo seu *k*=5, com relevância de 0,090 (9%) e presente em 4 documentos. Por fim, a quarta subclasse identificada dentre as 6 esperadas é a F01C, na décima posição, tendo seu *k*=10 com importância de 0,068 (6,8%) e estando presente em 3 documentos. Vale mencionar que a(s) subclasse(s) que constam em determinada patente de análise não indicam qualquer ordem esperada no processo de recomendação.

Quadro 8 - Exemplo de patente com seis subclasses

Número da Patente	US08967113
Título	*Vacuum pump mounting structure*
Resumo	*There is provided a vacuum pump mounting structure a vacuum pump is connected to an axial end portion of a first cam shaft of an engine and is configured to be driven by the first camshaft the first camshaft is disposed in a cylinder head a cam housing rotatably supports the first cam-shaft on the cylinder head a cam housing side boss portion and a cylinder head side boss portion are formed in the cam housing an engine upper side of a body portion of the vacuum pump is fixed to the cam housing side boss portion an engine lower side of the body portion of the vacuum pump is fixed to the cylinder head side boss portion*
Subclasses	F01C, F01L, B60T, F04C, F04D e F02B

Fonte: Elaborado pelo autor (2023).

Quadro 9 - *Ranking* de subclasses obtido utilizando o método para a patente do Quadro 8

k	**Subclasse**	**Frequência**	**Relevância**
1	A47L	7	0.159
2	F04D	6	0.136
3	F04B	6	0.136
4	F01L	6	0.136
5	F04C	4	0.090
6	F16L	3	0.068
7	F16F	3	0.068
8	F03C	3	0.068
9	F02M	3	0.068
10	F01C	3	0.068

Fonte: Elaborado pelo autor (2023).

O resultado se mostra consistente, apesar da recomendação não retornar todas as seis subclasses referentes a patente, sendo as subclasses B60T e F02B as duas não listadas no *ranking*. Antes de se analisar esta ausência é importante ressaltar também o fato da patente ser classificada em seis subclasses diferentes, o que dificulta a geração do *ranking*. Verifica-se pelo *ranking* que na primeira posição consta uma subclasse da seção referente à IPC-WIPO® de Necessidades Humanas, o que difere totalmente das outras subclasses presentes na patente relacionadas a Engenharia Mecânica; Iluminação; Aquecimento; Armas; Explosão (do inglês - *MECHANICAL ENGINEERING; LIGHTING; HEATING; WEAPONS; BLASTING*). Isso pode ser explicado analisando o conteúdo do título e do resumo verifica-se uma certa frequência na palavra *vacuum* o que levou o modelo a interpretar isso como utensílio doméstico, como um aspirador de pó (do inglês - *vacuum cleaner*) e mesmo a utilização da palavra *housing,* justificando a recomendação da primeira subclasse A47. Por outro lado, a ausência da subclasse B60T referente à Execução de Operações e Transporte (do inglês - *PERFORMING OPERATIONS; TRANSPORTING*), mais especificamente Veículos no Geral (do inglês - VEHICLES IN GENERAL) no *ranking* se deve a ausência de palavras relacionadas à veículos no conteúdo do título e no resumo. Já a subclasse F02B se refere a seção referente à Motores à Pistão de Combustão Interna; Motores de Combustão em Geral (do inglês *INTERNAL-COMBUSTION PISTON ENGINES; COMBUSTION ENGINES IN GENERAL*), sendo que no texto da patente ocorrem palavras como *engine*, *cylinder*. Neste sentido, o método deveria ter recomendado a subclasse. Isto ocorreu, mas em um *k* mais elevado .

A divergência dos resultados obtidos nas recomendações nestes três cenários específicos permite questionar se a patente foi classificada corretamente dentro de suas subclasses levando em consideração o título e o resumo. Conforme mencionado anteriormente, uma patente pode ter tido sua classificação alterada, o que não estaria refletido no conjunto de dados utilizado que foi composto em 2018. Os resultados do primeiro e terceiro cenários, poderiam ser explicados em parte pela própria evolução da taxonomia de secções, classes, subclasses, grupos e subgrupos da WIPO® ao longo do tempo. A cada nova versão anual da taxonomia IPC reclassificações podem ocorrer, com a possível criação de novas entradas/elementos, ou mesmo, modificações ou exclusões de entradas/elementos (BRASIL, 2022).

Antes de apresentar a análise do cenário final é importante ressaltar o funcionamento do algoritmo kNN (explicado em linhas gerais na seção 5.2). Inicialmente, determinado vetor é enviado ao banco de dados que calcula a distância entre a instância de consulta (determinada patente) e as prováveis instâncias (patentes) que constam na base de conhecimento. Este parâmetro é indicado pelo número de candidatos para se obter uma solução para a consulta. Em seguida, o algoritmo seleciona os *k* vizinhos mais próximos com base em alguma medida de distância, por exemplo, a medida do cosseno. Depois de obter os vizinhos (instâncias de patentes) mais próximos, é executada a estratégia de *ranking* que identifica quais são as subclasses mais relevantes a serem recomendadas para determinada patente.

Foram realizados alguns testes gerais para determinar valores adequados para os parâmetros *k* e número de candidatos considerando o algoritmo kNN. Por exemplo, com um *k*=100 e 200 candidatos e, analisando a quinta posição de recomendação (*k*=5) com número de documentos *n*=50, chegou-se ao resultado de 0.7745 (77.45%) de acurácia, sendo que até a décima posição com o mesmo *n* chegou-se ao resultado de 0.8617 (86.17%). Por outro lado, usando uma configuração com menos vizinhos (número de vizinho igual a 100), visto que isto impacta no tempo de resposta do banco de dados uma vez que menos dados são analisados, *k*=100 e também levando-se em conta a quinta posição de recomendação (*k=5*),

chegou-se ao resultado de 0.7738 (77.38%) de acurácia, sendo possível verificar uma diferença muito pequena.

Outro dado que é interessante trazer para a análise é o tempo de processamento do algoritmo kNN, usando *k*=100 e 200 candidatos foram necessários 150 minutos para avaliar as 49900 patentes do conjunto de teste. Já com *k*=100 e 100 candidatos, a mesma quantidade de patentes foi avaliada em 114 minutos, como uma acurácia muito próxima. Neste sentido, para efeitos do trabalho, a utilização de *k*=100 e 100 candidatos ofereceu resultados adequados, sendo apresentados no Quadro 10. De modo geral, a redução do espaço de busca foi benéfico, uma vez que, além de reduzir o tempo computacional, manteve a acurácia num nível adequado, muito próximo de configurações com espaços de busca maiores.

Quadro 10 - Resultados obtidos com *k*=100 e 100 candidatos

k	***n*=10**	***n*=25**	***n*=50**	***n*=75**	***n*=100**
1	0.4004	0.4016	0.3982	0.3934	0.3903
2	0.5720	0.5823	0.5802	0.5755	0.5712
3	0.6567	0.6747	0.6760	0.6733	0.6693
4	0.7042	0.7306	0.7345	0.7333	0.7308
5	0.7343	0.7667	0.7738	0.7743	0.7720
6	0.7558	0.7930	0.8021	0.8028	0.8017
7	0.7714	0.8109	0.8229	0.8251	0.8245
8	0.7828	0.8240	0.8395	0.8416	0.8415
9	0.7905	0.8343	0.8513	0.8555	0.8557
10	0.7963	0.8425	0.8612	0.8659	0.8675

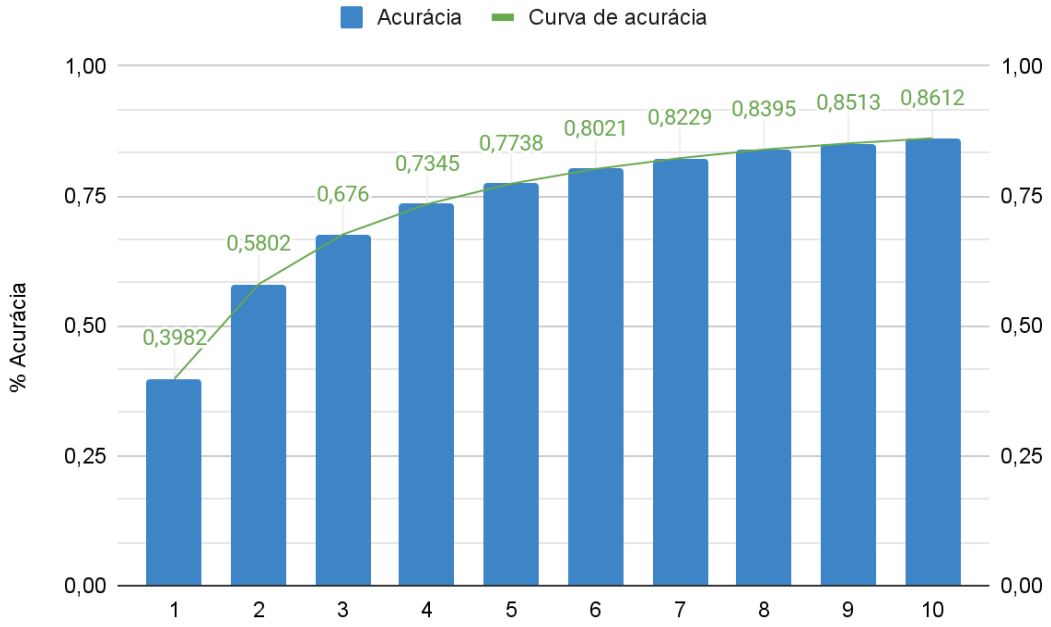
Fonte: Elaborado pelo autor (2023).

O Quadro 10 apresenta as acurácias calculadas para cada posição *k* (neste caso *k* significa o número de subclasses sugeridas), e para a quantidade de documentos retornados durante a consulta, representado pelo *n*, que varia entre 10, 25, 50, 75, 100. Avaliando o quadro pode-se determinar que a quantidade de subclasses a ser recomendada com níveis satisfatórios de acurácia fica entre *k*=5 e *k*=6 e com *n*=50.

A justificativa para a determinação do *n*=50 ocorre através da comparação da acurácia à medida que o próprio *n* aumenta, visto que é possível notar que até o valor 50 ocorre um acréscimo na acurácia para a maioria dos *k.* Já para os demais valores de *n* (75 e 100) o acréscimo é pequeno ou ocorre uma inflexão. Isso ocorre devido ao aumento na quantidade de documentos serem inseridos na análise, gerando uma maior disparidade de subclasses o que potencialmente afeta o resultado. Vale ressaltar que a estratégia de *ranking* utilizada neste trabalho não analisa a ordem dos documentos retornados para determinar a importância das subclasses, mas somente a frequência com que estas aparecem no conjunto de documentos retornados.

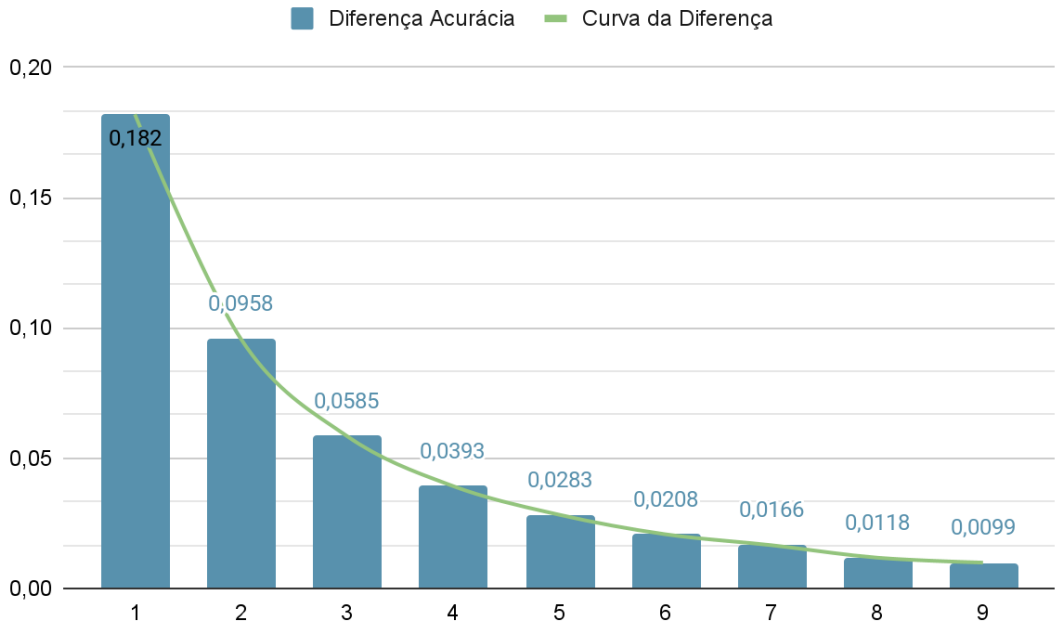
Visando justificar a escolha da quantidade de subclasses recomendadas (*k*) foram produzidas duas figuras. A primeira (Figura 9) destaca a acurácia conforme o aumento de *k.* Já a Figura 10 apresenta a diferença entre as acurácias a medida que *k* é incrementado.

Figura ´9 - Gráfico de acurácia variando *k* para *n*=50 (conforme Quadro 10)



Fonte: Elaborado pelo autor (2023).

Figura 10 - Gráfico da diferença entre as acurácias para *n*=50 (considerando o Quadro 10)



Fonte: Elaborado pelo autor (2023).

Analisando a Figura 9 é possível notar que perto da coluna 5 e 6 a curva de acurácia tende a uma menor intensidade de crescimento. Isso fica mais evidente na Figura 10 quando é analisada a curva da diferença entre as acurácias, em que se nota uma diferença de 0.182 (18.20%) comparando-se *k*=1 e *k*=2 (posição 1). Da mesma forma, quando se analisa a diferença entre *k*=5 e *k*=6 (posição 5) tem-se uma diferença de 0.0283 (2.83%). O valor da

diferença tenderia a zero à medida que mais subclasses fossem recomendadas, mas, de modo geral, percebe-se que valores de k=5 ou k=6 parecem adequados.

A determinação de um k ideal neste cenário não é uma tarefa fácil, sendo mais plausível determinar este número pela indicação explícita de algum critério. Por exemplo, quando a diferença da acurácia em um dado ponto (k) for inferior à 5% em relação à medição anterior, o k corrente poderia ser considerado. Por outro lado, impor um limite pode não ser adequado neste cenário de estudo, uma vez que ofertar as 10 primeiras subclasses para um avaliador de patentes pode ser relevante. Considerando que determinado examinador é quem irá tomar a decisão final, ou seja, irá concordar ou não com as classes recomendadas atribuindo-as à patente, seria pertinente viabilizar a escolha deste parâmetro pelo próprio examinador.

6. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Este trabalho teve como o objetivo o desenvolvimento de um método de classificação de patentes baseado em Aprendizado Profundo e modelos pré-treinados de NLP, com foco na recomendação ordenada (*ranking*) de subclasses de patentes. Para a construção deste documento, pesquisas bibliográficas foram realizadas voltadas à análise de patentes, técnicas de NLP e aprendizado profundo, assim como identificou-se um conjunto de dados adequado para a etapa avaliativa do método. Das pesquisas elencadas na seção de trabalhos correlatos não foram encontrados trabalhos com proposta similar à apresentada neste trabalho. Vale ressaltar ainda, que o método proposto possui custo computacional linear à medida que mais documentos são adicionados à base de patentes, com impacto em futuras recomendações sem a necessidade de novos treinamentos. O método proposto consiste de 6 etapas, começando pela coleta e pré-processamento de dados, passando pela transformação de patentes em *embeddings* e posterior formação da base de dados que promove suporte à consultas baseadas em vetores densos, finalizando com a recomendação de subclasses de maneira ordenada (*ranking*).

Nesta pesquisa, um conjunto de dados com patentes coletadas a partir de dados públicos disponibilizados pela USPTO® (Seção 5.1) foi utilizado. Deste conjunto de dados, as patentes referentes ao ano de 2014 foram consideradas na criação da base de dados (conjunto de treinamento), sendo que cada patente é transformada em um vetor denso (*embedding*) para posterior armazenamento. Para a criação dos vetores densos utilizou-se o LLM do projeto SBERT identificado pelo nome “all-MiniLM-L6-v2”. Já as patentes referentes ao ano de 2015 foram consideradas para a etapa de teste do método proposto, em que os passos para a transformação em *embeddings* são os mesmos aplicados ao conjunto de treinamento.

A partir do conjunto de teste foram apresentados três cenários específicos e um geral, ambos analisados na seção 5.3. Em cada um dos cenários específicos foi verificado o método proposto considerando as variáveis k e n com valores de 5 e 50, respectivamente, ou seja, a recomendação das 5 subclasses mais relevantes levando-se em conta os primeiros 50 documentos recuperados a partir de uma patente específica do conjunto de testes. Já o cenário geral teve o objetivo de avaliar de maneira mais ampla o comportamento do método utilizando a variável k com valores entre 1 e 10 e a variável n com os valores de 10, 25, 50, 75, 100.

De maneira geral, os resultados presentes nos cenários específicos se mostram consistentes, se alinhando com o esperado do método de recomendação de subclasses no contexto de classificação de patentes. Isso pode ser verificado no segundo cenário, onde existe apenas uma subclasse atrelada à patente. Mesmo os outros dois cenários se mostram interessantes pois, mesmo que não tenham retornado todas as subclasses pertencentes a patente de entrada, como ocorre no terceiro caso, indicam a possibilidade de equívocos na

classificação da própria patente. Adicionalmente, promovem também uma indagação sobre como o conteúdo presente no texto se relaciona com as subclasses em questão. Neste sentido, os resultados globais obtidos pelo método poderiam ter sido ainda melhores.

Da mesma forma, considerando a análise do cenário geral onde delimitou-se o ponto de corte entre k=5 ou 6, as acurácias próximas a 80% mostram que o método tem potencial na recomendação de subclasses. Ressalta-se que esse ponto de corte é uma sugestão deste trabalho em relação ao método proposto, não havendo qualquer impedimento na recomendação de valores maiores para k. Em uma situação real de recomendação de subclasses para uma patente, pode ser interessante para o examinador a obtenção de mais subclasses, visto que isto pode, inclusive, facilitar a análise do mesmo auxiliando em uma tomada de decisão mais precisa. Menciona-se também que os possíveis usuários do método proposto são tanto examinadores dos escritórios de patentes, quanto gestores de departamentos de pesquisa, desenvolvimento e inovação (PD&I).

Apesar das contribuições deste trabalho, percebem-se limitações que podem ser tratadas em trabalhos futuros. Primeiro, a fim de obter acurácias mais elevadas é possível utilizar um modelo superior ao "all-MiniLM-L6-v2", visto que esta é a versão mais simples do SBERT com dimensionalidade de 384 posições. A expectativa é que modelos com uma maior dimensionalidade produzam uma representação semântica mais apurada, resultando consequentemente em um aumento da acurácia. Todavia, o aumento da dimensionalidade pode impactar na escolha do banco de dados vetorial. Por exemplo, o ElasticSearch® possui atualmente um tamanho máximo de 1024 posições para representar um vetor denso.

Adicionalmente, a melhoria ou disponibilização de outras estratégias de *ranking* promoveriam opções diferenciadas com possíveis impactos na acurácia do método proposto. A estratégia atual considera apenas a frequência com que determinada subclasse aparece no conjunto de documentos retornados. Uma possibilidade seria considerar o *score* do documento, isto é, a relevância do documento retornado levando-se em conta determinada patente de interesse. Quanto melhor a posição do documento, maior é o seu *score* e, consequentemente, este poderia ser utilizado para normalizar a frequência com que determinada subclasse aparece nos conjunto de documentos retornados.

Além disso, considerando um contexto atual de Inteligência Artificial Generativa, o LLM utilizado, ou outro qualquer considerado mais adequado, poderia sofrer ajustes (*fine-tuning*) a partir do conjunto de patentes utilizado para o treinamento. Segundo pesquisas recentes, esta estratégia tem potencial para incrementar resultados considerando domínios específicos, o que é aderente ao cenário discutido e analisado neste trabalho. Ainda pensando neste contexto, mas com foco mais na aplicação final, o desenvolvimento de um *chatbot* parece ser algo relevante para demonstrar o potencial da pesquisa e do método proposto facilitando a interação com usuários.

Por fim, mostra-se relevante a criação de uma estratégia de recomendação que consiga diferenciar o *ranking* de subclasses que atingiram o mesmo nível de relevância. Entre as possibilidades, vislumbra-se a utilização das informações da taxonomia da IPC-WIPO® para criar uma métrica que avalie o contexto da subclasse (por meio do texto que descreve a subclasse) em relação ao texto da patente.

7. REFERÊNCIAS

ABDELGAWAD, L. *et al*. Optimizing Neural Networks for Patent Classification. **Machine Learning and Knowledge Discovery in Databases**. v. 11908, 2020. Disponível em: <https://doi.org/10.1007/978-3-030-46133-1_41>

AGGARWAL, C. C. An Introduction to Neural Networks. **Springer International Publishing**, p. 1-52, 2018. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-319-94463-0_1>

ALOM, Md Z. *et al*. A State-of-the-Art Survey on Deep Learning Theory and Architectures. Electronics, **Computer Science & Engineering**, v. 8, p. 292, 2019. ISSN 2079-9292. Disponível em : <https://www.mdpi.com/2079-9292/8/3/292>

ABBAS, A.; ZHANG, L.; KHAN, S. U.. A literature review on the state-of-the-art in patent analysis, **World Patent Information**, v. 37, 2014, p. 3-13, ISSN 0172-2190. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0172219013001634>.

BRASIL. Classificação de Patentes (IPC/CPC): Relatório executivo, 2022. Disponível em: <https://www.gov.br/inpi/pt-br/servicos/patentes/classificacao/RelatorioExecutivoClassificacaoPatentes2021_DIRPA_14032022.pdf>

LUAN, C. *et al*. An Approach to Construct Technological Convergence Networks Across Different IPC Hierarchies and Identify Key Technology Fields. **IEEE Transactions on Engineering Management**. p. 1-13, ISSN 1558-0040, 2021 Disponível em: <doi: 10.1109/TEM.2021.3120709>

CHEN, L. *et al*.A deep learning based method benefiting from characteristics of patents for semantic relation classification. **Journal of Informetrics**, v. 16, p. 101312, Issue 3, 2022, ISSN 1751-1577. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1751157722000645>

CHOI, S. *et al*. Deep learning for patent landscaping using transformer and graph embedding. **Technological Forecasting and Social Change**. v. 175, 121413, ISSN 0040-1625, 2022. Disponível em: <https://doi.org/10.1016/j.techfore.2021.121413>

CLARKE, N. S., The basics of patent searching, **World Patent Information**, v. 54, Supplement, p. S4-S10, ISSN 0172-2190, 2018. Disponível em: <https://doi.org/10.1016/j.wpi.2017.02.006>.

CONNEAU, A. *et al*. Very Deep Convolutional Networks for Text Classification. 2017. Disponível em: <https://doi.org/10.48550/arXiv.1606.01781>

DEGROOTE, B.; HELD, P. Analysis of the patent documentation coverage of the CPC in comparison with the IPC with a focus on Asian documentation. **World Patent Information**, v. 54, Supplement, 2018, p. S78-S84, ISSN 0172-2190, Disponível em: <https://doi.org/10.1016/j.wpi.2017.10.001>.

DEVLIN, J. *et al*. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. 2019. Disponível em: <https://arxiv.org/pdf/1810.04805.pdf>

DORAN, P., WEBSTER E., Who influences USPTO patent examiners? **World Patent Information**, v. 56, p. 39-42, ISSN 0172-2190, 2019, Disponível em: <https://doi.org/10.1016/j.wpi.2019.01.005>.

GÉRON, Aurélien. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. **O'Reilly Media, Inc.**, 2019.

GULERIA, K. *et al*. Early prediction of hypothyroidism and multiclass classification using predictive machine learning and deep learning, **Measurement: Sensors**, v. 24, p. 100482, 2022, ISSN 2665-9174. Disponível em: <https://doi.org/10.1016/j.measen.2022.100482>

HE, S.; SCHOMAKER, L. GR-RNN: Global-context residual recurrent neural networks for writer identification. **Pattern Recognition**, v. 117, 2021, 107975, ISSN 0031-3203. Disponível em: <https://doi.org/10.1016/j.patcog.2021.107975>.

HENRIQUES, R.; FERREIRA, A.; CASTELLI, M. A Use Case of Patent Classification Using Deep Learning with Transfer Learning. **Journal of Data and Information Science**. p. 49-70, 2022. Disponível em: <https://doi.org/10.2478/jdis-2022-0015>

INCITTI, F.; URL, F.; SNIDARO, L. Beyond word embeddings: A survey, **Information Fusion**, v. 89, p. 418-436, 2023, ISSN 1566-2535. Disponível em: <https://doi.org/10.1016/j.inffus.2022.08.024>

JUNG, G., SHIN, J. e LEE, S. Impact of preprocessing and word embedding on extreme multi-label patent classification tasks. **Applied Intelligence.** v. 53, p. 4047–4062, 2023. Disponível em: <https://doi-org.ez46.periodicos.capes.gov.br/10.1007/s10489-022-03655-5>

KAHRAMAN, S. Y.; DERELI, T.; DURMUŞOĞLU, A. Forty Years of Automated Patent Classification. **International Journal Of Information Technology & Decision Making**. ISSN 0219-6220, 2023, Disponível em: <10.1142/S0219622023500165>.

KANG, D. M. *et al*. Patent prior art search using deep learning language model. **In Proceedings of the 24th Symposium on International Database Engineering & Applications (IDEAS '20)**. Association for Computing Machinery, Article 1, 1–5, 2020. Disponível em: <https://dl-acm-org.ez46.periodicos.capes.gov.br/doi/10.1145/3410566.3410597>.

KRESTEL, R. *et al*. A survey on deep learning for patent analysis. **World Patent Information**, v. 65, p. 102035, 2021. ISSN 0172-2190. Disponível em: <https://www.sciencedirect.com/science/article/pii/S017221902100017X>.

LEE J.; HSIANG J. Patent classification by fine-tuning BERT language model. **World Patent Information**, v. 61, p. 101965, 2020, ISSN 0172-2190. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0172219019300742>

LI, J. *et al*. Multi-label text classification via hierarchical Transformer-CNN. **14th International Conference on Machine Learning and Computing (ICMLC)**. Association for Computing Machinery, p.120-125, 2022, ISBN 978-1-4503-9570-0. Disponível em: <https://doi-org.ez46.periodicos.capes.gov.br/10.1145/3529836.3529912>

LI, S. *et al*. DeepPatent: patent classification with convolutional neural networks and word embedding. **Scientometrics 117**, p. 721–744, 2018. Disponível em: <https://link.springer.com/article/10.1007/s11192-018-2905-5>

LIN, H. *et al*. Patent Quality Valuation with Deep Learning Models. ***Springer International Publishing***, v. 10828 p. 474-490, 2018. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-319-91458-9_29>.

LIU, X. *et al.* E2BoWs: An end-to-end Bag-of-Words model via deep convolutional neural network for image retrieval. **Neurocomputing**, v. 395, 2020, p. 188-198, ISSN 0925-2312. Disponível em: <https://doi.org/10.1016/j.neucom.2017.12.069>.

LO, H.-C.; CHU, J.-M. Pre-trained Transformer-based Classification for Automated Patentability Examination. **2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)**, Brisbane, Australia, 2021, p. 1-5, Disponível em: <10.1109/CSDE53843.2021.9718474>.

MEGURO K.; OSABE Y.; Lost in Patent Classification. **World Patent Information**. v. 57, p. 70-76, ISSN 0172-2190, 2019. Disponível em: <https://doi.org/10.1016/j.wpi.2019.03.008>.

KALIP, N. G.; ERZURUMLU, Y. Ö.; GÜN, N. A. Qualitative and quantitative patent valuation methods: A systematic literature review, **World Patent Information**, v. 69, 2022, p. 102111, ISSN 0172-2190. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0172219022000187>.

REIMERS N., GUREVYCH I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 2019. Disponível em: <https://doi.org/10.48550/arXiv.1908.10084>

RISCH, J.; GARDA, S.; KRESTEL, R. Hierarchical Document Classification as a Sequence Generation Task. **JCDL '20: Proceedings of the ACM/IEEE Joint Conference on Digital Libraries**. p. 147-155, 2020. Disponível em: <https://doi.org/10.1145/3383583.3398538>

RISCH, J.; KRESTEL, R.. Learning Patent Speak: Investigating Domain-Specific Word Embeddings, **Thirteenth International Conference on Digital Information Management (ICDIM)**, pp. 63-68, 2018, INSPEC 19013485. Disponível em: <10.1109/ICDIM.2018.8846972>

ROUDSARI, A. H. *et al*. PatentNet: multi-label classification of patent documents using deep learning based language understanding. **Scientometrics 127**, p. 207–231, 2022. Disponível em: <https://doi-org.ez46.periodicos.capes.gov.br/10.1007/s11192-021-04179-4>

RUAN L.; JIN Q. Survey: Transformer based video-language pre-training, **AI Open**, v. 3, p. 1-13, 2022, ISSN 2666-6510, Disponível em: <https://doi.org/10.1016/j.aiopen.2022.01.001>

RUIJIE, Z. *et al.* Patent text modeling strategy and its classification based on structural features, **World Patent Information**, v. 67, 102084, ISSN 0172-2190, 2021, Disponível em: <https://doi.org/10.1016/j.wpi.2021.102084>.

SREBROVIC R.; YONAMINE J. Leveraging the BERT Algorithm for Patents with TensorFlow and Big Query, **Technical Report**, Google 2020. Disponível em: <https://services.google.com/fh/files/blogs/bert_for_patents_white_paper.pdf>

SOFEAN, M. Deep learning based pipeline with multichannel inputs for patent classification. **World Patent Information**, v. 66, 2021, 102060, ISSN 0172-2190, Disponível em: <https://doi.org/10.1016/j.wpi.2021.102060>.

VASWANI, A. *et al*. Attention Is All You Need. **Advances in Neural Information Processing Systems**, 2017. Disponível em: <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>

WIPO - World Intellectual Property Organization. **World Intellectual Property Indicators 2022.** 2022. ISSN 2709-5207. Disponível em: <https://doi.org/10.34667/tind.47082>.

WIPO - World Intellectual Property Organization. Guide to the International Patent Classification, 2023. Disponível em: <https://doi.org/10.34667/tind.48084>

WIPO - World Intellectual Property Organization. **World Intellectual Property Indicators 2021**, 2021. ISSN 2709-5207. Disponível em: <https://doi.org/10.34667/tind.44461> .

YUN, J.; GEUM, Y. Automated classification of patents: A topic modeling approach. **Computers & Industrial Engineering**, v. 147, p. 106636, 2020, ISSN 0360-8352. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0360835220303703>

XINZHENG, X. *et al*. Multi-label learning method based on ML-RBF and laplacian ELM. **Neurocomputing**, v. 331, 2019, p. 213-219, ISSN 0925-2312, Disponível em: <https://doi.org/10.1016/j.neucom.2018.11.018>.

ZAINI, W. M. F.; LAI, D. T. C.; LIM, R. C. Identifying patent classification codes associated with specific search keywords using machine learning. **World Patent Information**, v. 71, 2022, 102153, ISSN 0172-2190, Disponível em: <https://doi.org/10.1016/j.wpi.2022.102153>.