Seleção de documentos baseado em centróides para classificação de patentes usando Word2Vec e KNN

Henrique Camacho Farias, Andreia Gentil Bonfante, Claudia Aparecida Martins

[1]Instituto de Computação (IC) – Universidade Federal de Mato Grosso (UFMT)
Fernando Correa da Costa nº 2367 – 78060-90 – Cuiabá - MT – Brazil

harrycamachofarias@hotmail.com, {andreia.bonfante,claudia}@ic.ufmt.br

Abstract. *This paper presents a patent categorization method based on deep learning word embedding vectors (Word2Vec), centroid-based document filtering and K-Nearest Neighbor (KNN) algorithm to classify patent documents down to section level of the IPC hierarchy from the WIPO dataset. The experimental results indicate that the proposed classification method reached an accuracy of 75%.*

Resumo. *Este artigo apresenta um método de categorização de dados de patentes baseado na representação vetorial utilizando word embedding vectors (Word2Vec), na seleção de documentos por meio do cálculo dos centróides das classes e no algoritmo K-Nearest Neighbor (KNN), com o objetivo de classificar documentos de patentes no nível de Seção da hierarquia IPC do conjunto de dados WIPO. Os resultados experimentais indicam que o método de classificação proposto alcançou a acurácia de 75%.*

1. Introdução

Nos últimos anos, o aumento do volume de dados disponíveis na Internet, principalmente dados não estruturados, tornou-se tanto uma valiosa fonte de informações e conhecimento quanto um grande desafio relacionado ao processo de armazenamento, manipulação e processamento desses dados. O tratamento de grandes volume de dados consiste na criação e/ou utilização de algoritmos e técnicas computacionais com o objetivo de automatizar e auxiliar tarefas para extração de conhecimento e padrões nos dados.

Considerando o domínio cujos dados são essencialmente textos, ou seja dados não estruturados, a tarefa de processar e extrair conhecimento dos dados se torna ainda mais custosa. É, portanto, natural a utilização de diversas técnicas das áreas de Ciência da Computação e da Computação Linguística, tais como Processamento de Linguagem Natural (PLN) e Aprendizado de Máquina (AM) que, entre outras coisas, fornecem mecanismos para auxiliar sistemas computacionais a automatizar a tarefa de entender, processar e manipular informações relacionadas com a linguagem humana.

Neste trabalho, é proposto o uso de técnicas de PLN e AM aplicadas em dados textuais, no domínio de documentos de patentes. Uma patente é uma concessão pública, conferida pelo Estado, que garante ao seu titular a exclusividade de explorar comercialmente a sua criação. Em outras palavras, uma patente pode ser definida como um direito exclusivo que se concede sobre uma invenção, permitindo ao seu criador a decisão da invenção poder ser usada por terceiros ou não[1].

[1]http://www.inpi.gov.br/

Para garantir o direito de uso e a invenção ser patenteada, é necessário que não haja registro de invenções semelhantes já protegidas. Escritórios especializados são responsáveis por armazenar os documentos das patentes em repositórios, tornando-os facilmente manipuláveis para que possam ser consultados. As patentes são armazenadas de forma hierarquizadas em categorias de acordo com as características de seu conteúdo, no qual cada escritório ou país decide qual sistema de classificação utilizará. Uma das classificações mais comuns utilizada é a *International Patent Classification* (IPC[2]), que abrange todas as áreas tecnológicas e utilizada por mais de noventa países [Fall et al. 2003].

De forma geral, o código definido pelo IPC é dividido em uma hierarquia de classificação com diferentes níveis, como: Seção, Classe, Subclasse, Grupo e Subgrupo, e a classificação completa pode ser encontrada em [Wipo 2019]. Assim, documentos com assuntos similares são mantidos dentro de uma mesma hierarquia ou código e cada patente possuiu várias informações textuais como título, depositante, inventor, resumo (*abstract*), produto ou processo que se está protegendo (*claims*), sua descrição (*text*), etc.

A atividade de buscar no repositório patentes similiares à nova invenção, se feita manualmente, é bastante custosa e exige que os agentes envolvidos tenham um grau de especialização considerável. Por esse motivo, muito tem sido investido na produção de ferramentas computacionais capazes de processar, armazenar, categorizar e analisar as patentes, com o mínimo de intervenção possível com a automatização da tarefa de busca e classificação.

Dessa forma, como contribuição para o processo de classificação automática em documentos de patentes, neste trabalho foi investigada a viabilidade de seleção de documentos com base na proximidade dos seus respectivos centróides, associada às técnicas de redes neurais de aprendizado profundo (Word2Vec), para representação vetorial, e o método de aprendizado supervisionado baseado em similaridades, o algoritmo K-Nearest Neighbor (KNN). Foram utilizadas as informações textuais extraídas dos campos *título* e *resumo* dos documentos das patentes disponíveis nos conjuntos WIPO-alpha e WIPO-gamma[3], juntamente com as suas classificações representadas nas oito diferentes seções da hierarquia IPC.

Este trabalho está dividido da seguinte forma: na Seção 2 são apresentados os trabalhos relacionados à classificação de patentes, na Seção 3 é apresentada a metodologia proposta, na Seção 4 são apresentados os experimentos e resultados e, por fim, na Seção 5 é apresentada a conclusão.

2. Trabalhos relacionados

A automatização do processo de classificação e análise de patentes tem sido foco de grande atenção por parte dos pesquisadores, uma vez que os repositórios são fontes inesgotáveis e ricas de captação de ideias potenciais de negócios. Ferramentas de busca e recuperação de conteúdos podem ser utilizadas na prospeção de oportunidades e, também, para garantir que novos depósitos de patentes não infrinjam leis de propriedade intelectual. Dessa forma, classificadores automáticos podem auxiliar na atribuição de códigos às patentes, necessários para que seus conteúdos sejam hierarquizados dentro das diversas categorias nos repositórios.

[2]https://www.wipo.int/classifications/ipc/en/

[3]https://www.wipo.int/classifications/ipc/en/ITsupport/Categorization/dataset/

São muitos os esforços na busca por modelos que contribuam na melhoria do processo de categorização automática de patentes, utilizando as mais variadas combinações de representações vetoriais como tf-idf e Word2Vec, além de métodos de classificação como Regressão Logística, *K-Nearest Neighbor* (KNN), *Suport Vector Machine* (SVM), *Long Short Term Memory* (LSTM) [Hochreiter and Schmidhuber 1997], Redes Neurais Recorrentes com Aprendizado Profundo (CNN[4]) [Kim 2014], entre outros. Em trabalhos recentes, são encontrados alguns métodos relacionados ao desenvolvido neste trabalho, comentados a seguir.

Lyu e Han [Lyu and Han 2019] utilizam várias técnicas de aprendizado profundo (*deep learning*), comparando-as com as abordagens tradicionais, como tf-idf + Regressão Logística, na classificação de patentes chinesas. Trabalhando na hierarquia IPC no nível de Seção, ou seja, com documentos distribuídos em oito categorias, recolheram 8.000 documentos do repositório INCOPAT[5], contendo título, resumo e texto do primeiro pedido de proteção (*claim*). No pré-processamento realizaram a remoção de palavras comuns (*stopwords*) e atribuição da morfo-sintaxe (*part-of-speech tagging*). Para a vetorização usaram Word2Vec e tf-idf. Como modelos de treinamento, Regressão Logística e uma combinação de Redes Neurais Recorrentes (TextCNN[6]) [Gong and Ji 2018], Gated Recurrent Unit (GRU) [Cho et al. 2014] e Attention Networks [Yang et al. 2016]). Obtiveram acurácia de 72% com a combinação de tf-idf + Regressão Logística. Dentre os modelos neurais, a melhor combinação obteve acurácia de 81,8%, utilizando Word2Vec + GRU + TextCNN.

Gomez et al. [Gomez and Moens 2014] trabalharam com o conjunto WIPO-alpha, utilizando as informações do título, do resumo e as primeiras 30 (trinta) linhas da descrição. Utilizaram como pré-processamento a remoção de *stopwords* e termos com frequência inferior a cinco documentos, além do método proposto *Minimizer of Reconstructor Error*, uma extensão da propriedade de minimização de erro do método de Análise de Componente Principal (PCA) [Jolliffe 1986], para a extração de features. Obtiveram acurácia de 74,59% e medida F_1 (*F-score*) de 72,56%. Seus experimentos reportaram ainda a performance do algoritmo KNN com valores de 64,29% de acurácia e 61,99% de F_1.

Mollá e Seneviratne [Mollá and Seneviratne 2018] mostraram os resultados da tarefa proposta pela *Australasian Language Technology Association* (ALTA) 2018, cujo objetivo foi produzir classificadores para textos de patentes australianas, segundo o nível de seção da classificação IPC. Foram utilizados 3.972 documentos para treinamento e 1000 para testes, com alto grau de desbalanceamento entre as classes, como nos conjuntos WIPO-alpha. Foi utilizado todo o conteúdo textual das patentes, com extração de *features* baseado em tf-idf com unigramas e bigramas e classificador SVM, com inclusão de treinamento adicional, treinados com termos e com caracteres. O melhor resultado obteve 77,8% de medida F_1 e acurácia de 78% [Benites et al. 2018].

Xiao et al. [Xiao et al. 2018] utilizam Word2Vec e LSTM para classificar documentos de patentes no campo de segurança. Como pré-processamento, foram filtrados termos comuns na área e remoção das *stopwords*. Foi utilizado um modelo pré-treinado de Word2Vec com a Wikipedia Chinesa, e a rede LSTM treinada com 50.000 documen-

[4]*Do inglês Convolutional Neural Network*

[5]https://www.incopat.com/

[6]Do inglês *Text Convolutional Neural Network.*

tos, com tamanho máximo de 200 termos cada. Na comparação com a utilização sem o Word2Vec, o modelo LSTM sozinho teve acurácia de 85,76% enquanto o combinado com Word2Vec pré-treinado teve acurácia de 93,48%. Foram comparados com modelos KNN (33,51%), CNN (80,59%) e CNN+Word2Vec (81,18%).

Grawe et al. [Grawe et al. 2017] utilizam o treinamento *Continous Bag of Words* (CBOW) do Word2Vec com aprendizado baseado LSTM para classificar patentes seguindo a hierarquia IPC no nível de subclasse, com 50 categorias diferentes. Na construção do modelo Word2Vec foram utilizados 167.876 documentos extraídos de USPTO Bulk Data[7] repositório. Já no LSTM foram utilizados 15.050 documentos para treinamento e 550 para testes. Os resultados mostram acurácia de 63%, melhorando os 41% obtidos no trabalho de [Fall et al. 2003], também em nível de subclasse, usando SVM.

Li et al. [Li et al. 2018] propõem a classificação de patentes usando Word2Vec e CNN no textos de títulos e resumos (100 termos de entrada) das 637 subclasses da classificação IPC disponíveis nos repositórios USPTO-2M, com precisão de 73,88%. Utilizaram mais de 1,95 milhões de patentes para treinamento e 49,9 mil para testes. Em outro teste com 580.586 patentes da EPO e 161.555 da WIPO obtiveram precisão de 83,98%.

3. Metodologia Proposta

Neste trabalho, é proposta uma metodologia que utiliza algoritmos de aprendizado profundo para auxiliar o processo de identificação das classes, no contexto de documentos de patentes. É apresentado um processo de seleção de documentos, visando encontrar os documentos que possam ser mais relevantes na discriminação da classe. Isto pode útil em contextos para classes desbalanceadas e/ou em situações cujos documentos estão próximos, ou sobrepostos, dos limites das classes.

Os passos utilizados nessa metodologia são apresentados no Algoritmo 1. Inicialmente, foi obtido o conjunto de dados utilizado junto à *World Intellectual Property Organization* (WIPO), mais especificamente, foram selecionados documentos das coleções WIPO-alpha(2002) e WIPO-gamma(2015). Este conjunto de dados está classificado de acordo com *International Patent Classification* (IPC) em cinco níveis: Seção, Classe, Subclasse, Grupo e Sub-grupo. O nível de Seção, mais alto nível de classificação, representa o campo técnico e é composto por oito diferentes seções denominadas de A até H, descrevendo as possíveis categorias às quais a patente está relacionada, por exemplo, física, eletricidade, transporte, entre outros. Neste trabalho, foi utilizada a classificação no nível de Seção e, a partir de agora, para efeito dos algoritmos de classificação será denominada de ”classe”os símbolos A, B, C, D, E, F, G e H.

O pré-processamento dos documentos consistiu na padronização de todas as palavras para caixa baixa, remoção de *stopwords* e transformação das palavras em *tokens*. A remoção de *stopwords* foi realizada utilizando o pacote NLTK[8], ferramenta de código aberto para Processamento de Linguagem Natural bastante utilizada na literatura.

Após a remoção de *stopwords*, foi necessário reduzir a um mesmo termo as várias formas verbais de uma palavra. Por exemplo, as palavras LOVE, LOVING e LOVED pos-

[7]https://bulkdata.uspto.gov/

[8]https://www.nltk.org/api/nltk.html

suem o mesmo radical, ou elemento básico e significativo da palavra, e caso não seja feito o tratamento, o algoritmo pode identificar como sendo palavras distintas em um texto. Neste trabalho, foi realizada a lematização das palavras, representando as palavras em sua forma canônica: infinitivo, masculino singular dos substantivos e adjetivos. No Algoritmo 1 estão sintetizados os passos até agora descritos.

Algoritmo 1: Passos para Classificação de Patentes

Entrada: Conjunto de Documentos de Patentes
Saída: Modelo de Classificação de Patentes

```
1  início
2      seleciona o conjunto de dados – WIPO;
3      pré-processa os dados: remove stopwords e transforma palavras em
         tokens;
4      vetoriza as palavras usando modelo skip-gram – Word2Vec;
5      enquanto documento i faça
6          seleciona os vetores das palavras;
7          calcula a média aritmética dos vetores x⃗;
8      fim
9      seleciona m documentos de acordo com os centróides das classes;
10     seleciona os parâmetros e algoritmos de classficação – TPOT;
11     classifica os m documentos – KNN, SVM, XGBoost;
12     valida o resultados de acordo com a métrica escolhida.
13 fim
```

Anterior ao processamento dos dados em um algoritmo de aprendizado, geralmente, é necessário que os documentos estejam no formato de vetores numéricos, considerando que muitos algoritmos irão realizar operações matemáticas com o objetivo de otimizar alguma função de custo. Para a vetorização das palavras, neste trabalho, foi utilizado o modelo *skip-gram* do Word2Vec, que é uma técnica de predição do contexto ao qual se inserem as palavras (termos). Nesse espaço vetorial, palavras que compartilham um mesmo contexto (semântico, sintático, etc.) tendem a serem colocadas próximas, conforme ilustrado na Figura 1 no contexto de alimentação.

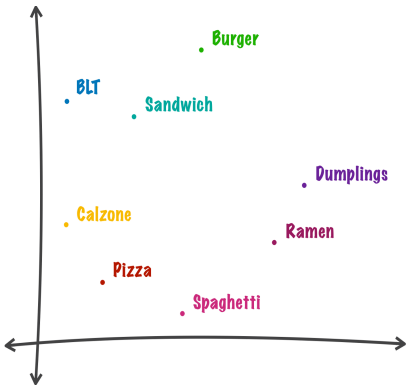

Figura 1. Representação espacial dos termos Word2Vec

A fim de processar documentos relavantes na discriminação das classes, e consi-

derando que as classes estão totalmente desbalanceadas, foi realizada uma nova seleção e redução do conjunto de dados. Essa redução foi realizada da seguinte forma: para cada classe n ($n = 1..8$), soma-se os vetores de cada documento i, calcula-se a média aritmética e, assim, o centro da classe n é encontrado. Após o centro da classe ter sido encontrado, os m documentos mais próximos desse centro são selecionados como sendo os mais representativos daquela classe.

Neste trabalho, foi definido o valor de $m = 2000$ para que as classes fiquem com a mesma quantidade de documentos. Considerando a classe D, com a menor quantidade de documentos (pouco mais de mil e setecentos) no conjunto WIPO-alpha, optou-se por complementá-la com trezentos documentos escolhidos da mesma Seção no conjunto WIPO-gamma, garantindo-se assim, a mesma quantidade de documentos em todas classes.

A definição dos parâmetros e do algoritmo a ser utilizado pode ser realizado por meio de um processo automático que auxilia a escolha dos algoritmos mais adequados a uma determinada tarefa de aprendizado. Neste trabalho, com a finalidade de auxiliar nesta tarefa, foi utilizada a ferramenta TPOT[9] (*Tree-Based Pipeline Optimization Tool*), disponível na linguagem de programação Python. TPOT é uma ferramenta de aprendizado de máquina automática (autoML[10]) que utiliza algoritmos genéticos para auxiliar e automatizar na busca por modelos mais adequados rapidamente, como por exemplo, seleção do algoritmo de aprendizado e otimização dos hiperparâmetros do algoritmo selecionado.

A execução do TPOT para a seleção do modelo e para a otimização dos hiperparâmetros apresentou que a métrica escolhida para ser otimizada foi a acurácia, a medida de distância foi a euclidiana, na qual são verificadas a distância D do documento p em relação ao documento q:

$$D(p, q) = \sqrt[2]{(q_1 - p_1)^2 + (q_2 - p_2)^2}$$

e o modelo de classificação no qual obteve-se a maior acurácia foi o algoritmo KNN, técnica de aprendizado supervisionado não paramétrica, no qual os pontos de dados para uma determinada categoria são classificados a partir do conjunto de treinamento com base em uma similaridade, como descrito de forma genérica no Algoritmo 2.

Na validação, é possível verificar se o resultado dos classificadores é estatisticamente significativo ou se o resultado foi simplesmente devido ao acaso ou ruído nos dados. Neste trabalho, a métrica utilizada foi a acurácia juntamente com o teste de hipóteses *t*. Este teste é um procedimento estatístico que permite tomar uma decisão entre duas ou mais hipóteses, utilizando os dados observados de um determinado experimento. O teste *t* verifica a menor ocorrência de erros do tipo 1, ou seja, quando a hipótese nula é rejeitada sendo ela verdadeira.

Usando a estatística do teste *t*, o valor de *p*[11] pode ser calculado e comparado com um nível de significância escolhido anteriormente α = 0.05. Se o valor de *p* for menor que α, a hipótese nula é rejeitada e aceita-se que há uma diferença significativa nos dois modelos.

[9] http://automl.info/tpot/

[10] Do inglês *Automated Machine Learning*.

[11] O *p* ou *p–valor* é a probabilidade do resultado ser pelo menos tão extremo quanto teste estatístico, assumindo que a hipótese nula é verdadeira.

Algoritmo 2: Algoritmo KNN

Entrada: Conjunto de m Documentos de Patentes
Saída: Classificação de Patentes

1 **início**
2 determina o parâmetro k – número de vizinhos;
3 computa a distância de todos os documentos com o documento que se deseja saber a predição;
4 ordena as distâncias, selecionando os k documentos mais próximos;
5 seleciona a categoria dos k documentos mais próximos;
6 seleciona a categoria mais numerosa por meio de uma votação simples;
7 **fim**

4. Experimentos e Resultados

A partir da metodologia proposta, foram realizados diversos experimentos para verificar o seu desempenho nos documentos de patentes. Inicialmente, foram selecionados 75.239 documentos disponíveis pela WIPO-alpha, divididos entre treinamento e teste. Como pode ser observado na Figura 2, a quantidade de documentos nas classes (Seções) de A-H está totalmente desbalanceada, visto que a classe C tem 21,6% de todos os documentos (16.244), enquanto a classe D possui apenas 2,3% (1.710).

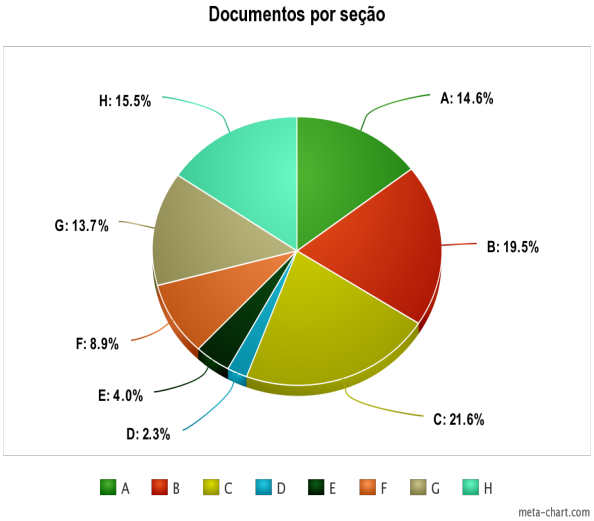

Figura 2. Distribuição de documentos nas classes A-H

Buscando melhorar o desempenho dos algoritmos de classificação, optou-se aqui pelo balanceamento do número de documentos e pela seleção daqueles mais relevantes, conforme já mencionado na Seção Metodologia (3). A seleção dos documentos foi realizada baseando-se em sua proximidade espacial com o centróide calculado de sua respectiva classe. Esta opção foi considerada como uma forma de tentar garantir da melhor forma possível as fronteiras de separação linear entre as classes, uma vez que nesse conjunto, os centróides de algumas classes estão bastante próximos, conforme mostrado na Figura 3.

É importante ressaltar que o objetivo foi buscar os documentos mais relevantes

na discriminação das classes. Documentos localizados na fronteira linear das classes, apesar da diversidade que possam representar, não auxiliam no processo de classificação e, portanto, outras técnicas seriam necessárias para lidar com documentos nesse limiar com documentos de classes diferentes sobrepostos.

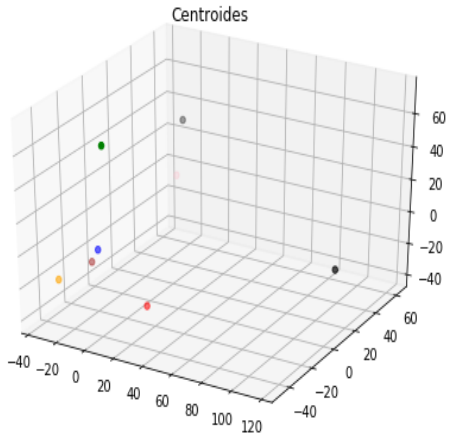

Figura 3. Centróides das classes A-H

Com a seleção dos dois mil documentos, a quantidade de *tokens* acabou também sendo reduzida, gerando uma considerável redução de dimensionalidade, conforme pode ser observado na Tabela 1, na qual *# Doc* apresenta a quantidade de documentos, *# Tokens* a quantidade de *tokens*, `wipoO` o conjunto de dados inicial e `wipoS`, o novo conjunto de dados após a seleção dos documentos.

Conjunto de Dados	*# Doc*	*# Tokens*
`wipoO`	75529	58477
`wipoS`	16000	31279

Tabela 1. Conjunto de dados

Como ilustração, a partir do conjunto de dados `wipoS` foi gerada uma nuvem de palavras, mostrando os termos mais frequentes de cada classe (Figura 4). Uma funcionalidade de correlação e identificação entre os principais termos e suas frequências em cada classe está sendo investigado.

O conjunto de dados `wipoS`, após o pré-processamento, foi submetido ao algoritmo de classificação. No entanto, considerando que existem muitos algoritmos de aprendizado de máquina, a escolha adequada do algoritmo para o contexto e os dados em questão pode ser uma tarefa não trivial, não sendo raro, por exemplo, a situação em que a escolha tende a ser os algoritmos mais complexos e/ou recentes como o XGBoost[12] e o *Support Vector Machine* (SVM). Mas, apesar dos algoritmos mais atuais de fato se saírem melhor na maioria das vezes, por razões óbvias como atualização de códigos já testados, em determinadas ocasiões um algoritmo mais simples pode ter um desempenho melhor.

A partir disso, como já mencionado, o TPOT foi utilizado para auxiliar nesta tarefa. O TPOT foi executado 7 (sete) vezes com o intuito de confirmar o resultado obtido

[12]https://github.com/dmlc/xgboost

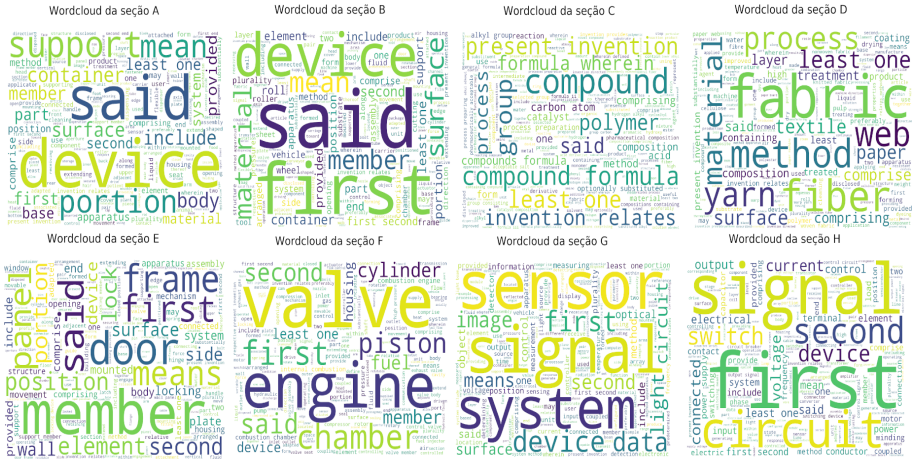

Figura 4. Termos mais frequentes de cada classe A-H

e verificar a tendência de convergência na escolha do algoritmo KNN. A execução de sete vezes do TPOT já foi suficiente para verificar a finalidade de testar a convergência na escolha dos algoritmos. Com relação aos hiperparamêtros do algoritmo KNN, foi necessário a modificação de três hiperparâmetros com a finalidade de resultar em uma melhor acurácia, sendo: i) o número de vizinhos, ii) o peso que cada vizinho teria na classificação e, por fim, iii) a métrica de distância a ser utilizada.

Em todas as execuções do TPOT, a métrica de distância escolhida foi a euclidiana e o peso de cada vizinho foi a distância, isto é, vizinhos mais próximos do ponto a ser feito a predição irão ter uma maior importância para a classificação do mesmo. Com relação ao número de vizinhos, este por sua vez acabou variando durante as execuções do TPOT ficando entre 9-15 vizinhos nos modelos finais. Com a seleção do algoritmo e dos parâmetros, após a execução do algoritmo KNN, o resultado foi de 75% de acurácia.

A matriz de confusão gerada pelo KNN é mostrada na Figura 5. É possível verificar que os documentos da classe D, na figura representada pelo número 3, teve um desempenho pior na classificação comparado às outras classes. É um resultado que não surpreende considerando que o centróide dessa classe estava numa fronteira de separação linear muito próximo de outras classes. Além disso, é possível verificar também a dificuldade do algoritmo em classificar as classes G e H, representada pelos números 6 e 7 respectivamente.

Além do algoritmo KNN, outros experimentos foram realizados utilizando os algoritmos supervisionados SVM e XGBoost para efeitos de comparação e comprovação da seleção do KNN pelo TPOT. Para os hiperparâmetros do SVM apenas o *kernel* foi modificado devido a não linearidade dos dados. Neste caso, foi utilizado o *kernel* RBF. Já no caso do XGBoost, foram utilizados os hiperparâmetros padrões. Os resultados obtidos pelos algoritmos são apresentados na Tabela 2.

Para confirmar que o modelo KNN obteve um desempenho diferente do SVM e o XGBOOST, foi utilizado o *5x2cv paired t test*, proposto por Dietterich [Dietterich 1998] para corrigir a deficiência de outros métodos como o *resampled paired t test* e o *k-fold cross-validated paired t test*.

Analisando os resultados, para o KNN e o SVM o valor *p* foi 0.002 e para o KNN

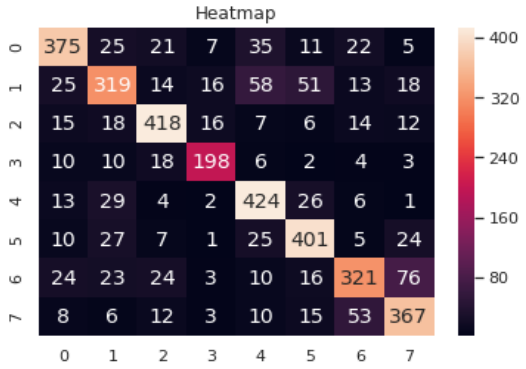

Figura 5. Matriz de confusão

Classificador	*% Acurácia*	*% F_1*	*% Recall*
KNN	75	74	75
SVM	69	69	69
XGBoost	65	65	65

Tabela 2. Tabela de resultados

e o XGBOOST o valor de p foi de 0.001. Em ambos os casos o valor p foi menor que 0.05 e, portanto, rejeita-se a hipótese nula de que o modelo KNN teve um desempenho igual aos outros dois modelos.

Um resumo das informações observadas com trabalhos relacionados é apresentado na Tabela 3. É possível verificar que a acurácia dos resultados apresentaram-se bastante promissores em relação aos trabalhos relacionados já apresentados, considerando-se o mesmo conjunto de dados e a mesma língua.

Autor	*Método*	*Dados - Língua*	*Acurácia*
Este trabalho	**w2v+SelCentróides+KNN**	**WIPO - inglês**	**75%**
[Gomez and Moens 2014]	tf-idf normalizado+mRE	WIPO - inglês	74%
[Gomez and Moens 2014]	tf-idf normalizado+KNN	WIPO - inglês	64%
[Benites et al. 2018]	tf-idf (1grama,2gramas)+SVM	ALTA2018 - inglês	78%
[Gomez and Moens 2014]	tf-idf normalizado+mRE	WIPO - alemão	69%
[Lyu and Han 2019]	w2v+GRU+textCNN	chinês	81%
[Xiao et al. 2018]	w2v(wikipedia-chinesa)+LSTM	Segurança - chinês	94%

Tabela 3. Resultados de classificadores de patentes

É importante ressaltar que, os valores mostrados na Tabela 3 indicam apenas um direcionamento favorável na investigação da metodologia apresentada neste trabalho e, posteriormente, em conjunto com novos métodos. Para efeito de análise comparativa entre os diversos resultados, é necessário a definição de parâmetros e métricas de avaliação.

5. Conclusão e Trabalhos Futuros

Neste trabalho foi proposta uma metodologia que utiliza Word2Vec para a vetorização dos documentos, o uso do algoritmo *K-Nearest Neighbor* como classificador, e ainda centróides para a seleção dos documentos mais relevantes a serem utilizados no processo de treinamento do algoritmo.

Para auxiliar o processo de escolha de parâmetros e algoritmos mais adequados ao processamento dos dados, foi utilizada a ferramenta TPOT. A aplicação de ferramentas com funcionalidades semelhantes ao TPOT, auxilia no processo de seleção dos melhores métodos e parâmetros em um processo de classificação em domínios complexos, proporcionado otimização de tempo e esforço na geração dos modelos.

Num contexto genérico, os classificadores de patentes encontrados na literatura geralmente apresentam eficácia entre 64-94%, variando conforme o conjunto de dados e a língua na qual as patentes estão escritas. O melhor resultado aqui obtido chegou na acurácia de 75%, que pode não ser tão expressivo se comparado com trabalhos diversos relacionados à área de AM em outros domínios. Porém, é um resultado dentro do padrão encontrado, ou até mesmo superior, para dados relacionados com patentes na língua inglesa, indicando uma direção promissora, já que foi utilizado um dos algoritmos mais simples da literatura como o KNN.

Como trabalhos futuros, outras técnicas supervisionadas estão sendo investigados como diferentes arquiteturas de redes neurais, por exemplo. Além disso, diversas técnicas de seleção de atributos específicas para processamento de linguagem natural, como palavras ou *tokens*, estão sendo investigadas para auxiliar nesse processo. Por último, outros métodos de vetorização de documentos também estão sendo pesquisados.

6. Agradecimentos

Os autores agradecem o apoio financeiro da Fundação de Amparo à Pesquisa do Estado de Mato Grosso (FAPEMAT), Projeto n.0213429/2017, e à Universidade Federal de Mato Grosso (UFMT).

Referências

Benites, F., Malmasi, S., and Zampieri, M. (2018). Classifying Patent Applications with Ensemble Methods.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 1724–1734. Association for Computational Linguistics (ACL).

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923.

Fall, C. J., Törcsvári, A., Benzineb, K., and Karetka, G. (2003). Automated categorization in the international patent classification. *ACM SIGIR Forum*, 37(1):10–25.

Gomez, J. C. and Moens, M.-F. (2014). A Survey of Automated Hierarchical Classification of Patents. pages 215–249. Springer, Cham.

Gong, L. and Ji, R. (2018). What Does a TextCNN Learn? *ArXvi*, abs/1801.0.

Grawe, M. F., Martins, C. A., and Bonfante, A. G. (2017). Automated Patent Classification Using Word Embedding. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 408–411.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Jolliffe, I. T. (1986). Principal Components in Regression Analysis. pages 129–155. Springer, New York, NY.

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 1746–1751.

Li, S., Hu, J., Cui, Y., and Hu, J. (2018). DeepPatent: patent classification with convolutional neural networks and word embedding. *Scientometrics*, 117(2):721–744.

Lyu, L. and Han, T. (2019). A comparative study of Chinese patent literature automatic classification based on deep learning. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, volume 2019-June, pages 345–346. Institute of Electrical and Electronics Engineers Inc.

Mollá, D. and Seneviratne, D. (2018). Overview of the 2018 ALTA Shared Task: Classifying Patent Applications. In *Proceedings of the Australasian Language Technology Association Workshop 2018*, pages 84–88.

Wipo (2019). Guide to the International Patent Classification. Technical report.

Xiao, L., Wang, G., and Zuo, Y. (2018). Research on Patent Text Classification Based on Word2Vec and LSTM. In *Proceedings - 2018 11th International Symposium on Computational Intelligence and Design, ISCID 2018*, volume 1, pages 71–74. Institute of Electrical and Electronics Engineers Inc.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. (2016). Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.