8-9-2021

Algoritmos de Classificação e Representação Word Embedding em Dados de Patentes

Henrique C. Farias
Universidade Federal de Mato Grosso, harrycamachofarias@hotmail.com

Claudia A. Martins
Universidade Federal de Mato Grosso, claudia@ic.ufmt.br

Rafaela S. Francisco
Universidade Federal de Mato Grosso, rafaela.souzaf@hotmail.com

Algoritmos de Classificação e Representação *Word Embedding* em Dados de Patentes

Artigo Completo

Henrique Camacho Farias
Aluno de graduação
Instituto de Computação/UFMT
harrycamachofarias@hotmail.com

Claudia Aparecida Martins
Professora
Instituto de Computação/UFMT
claudia@ic.ufmt.br

Rafaela Souza Francisco
Aluna de graduação
Instituto de Computação/UFMT
rafaela.souzaf@hotmail.com

Abstract

In this work, a study of Machine Learning algorithms combined with various forms of word embedding vector representation of patent documents was carried out in order to analyze the performance of classifiers for an automatic process of searching and retrieving information in the patent domain. Data were obtained from WIPO and were selected with a view to recovering the most discriminating data, using a methodology for selecting documents based on the centroids of the classes, reducing the data set by 78%. The classifiers were built using the HyperOpt automatic learning tool to adjust the hyperparameters. A comparative analysis was performed between the eight classifiers combined with four distinct vector representations of the document. The best result, using the Light Gbm classifier and the Word2Vec vectorizer, obtained a performance of 83.36% accuracy in the test set, considered competitive when compared to other works that used the same data set and language.

Keywords

Classification. Centroids. Word Embedding. Patent.

Resumo

Neste trabalho foi realizado um estudo de algoritmos de Aprendizado de Máquina combinados com diversas formas de representação vetorial *word embedding* de documentos de patentes com o objetivo de analisar o desempenho dos classificadores para um processo automático de busca e recuperação de informações no domínio de patentes. Os dados foram obtidos da WIPO e foram selecionados visando à recuperação dos dados mais discriminantes, usando uma metodologia de seleção de documentos baseada nos centroides das classes, reduzindo em 78% o conjunto de dados. Os classificadores foram construídos utilizando a ferramenta de aprendizado automático HyperOpt para o ajuste dos hiperparâmetros. Foi realizada uma análise comparativa entre oito classificadores combinados com quatro distintas representações vetoriais do documento. Comparativamente, o melhor resultado, dentre os classificadores utilizados, foi o classificador Light Gbm com o vetorizador Word2Vec, com acurácia de 83.36% no conjunto de teste, um resultado competitivo se comparado a outros trabalhos que utilizaram o mesmo conjunto de dados e idioma.

Palavras-chave

Classificação. Centroides. Vetorização de palavras. Patente.

Introdução

O volume de informações disponíveis nos últimos anos foi devido, principalmente, aos avanços tecnológicos, cuja expansão permite captar e receber qualquer tipo de dado numa fração de tempo cada vez menor e, com isso, produzir uma quantidade maior de dados estruturados ou não. O armazenamento, a manipulação e o tratamento desse volume de informações são alguns dos novos desafios computacionais e que poderá beneficiar a diversos domínios (Risch 2019).

O processamento de dados baseados em textos e, portanto, não-estruturados, vai muito além de técnicas de armazenamento. É necessário fundamentalmente compreender os dados por meio de uma esquematização linguística capaz de interpretar as suas relações. Isto geralmente é realizado usando técnicas computacionais tais como Processamento da Linguagem Natural (PLN) e Aprendizado de Máquina (AM) que, entre suas diversas aplicações, podem ser utilizadas no processamento automático para análise, compreensão e manipulação dos dados textuais.

Uma das áreas de aplicação de processamento de textos complexa pela natureza intrínseca de seu conteúdo, é a classificação automática de documentos de patentes. Uma patente consiste em um título de propriedade temporária sobre uma invenção ou modelo de utilidade, outorgado pelo Estado aos inventores, ou autores, detentoras de direitos sobre a criação[1].

Para garantir o direito de uso e a invenção ser patenteada, é necessário que não haja registro de produtos ou processos semelhantes já protegidos. Alguns escritórios especializados são responsáveis por armazenar os documentos das patentes em repositórios, disponibilizando-os para serem facilmente consultados. As patentes são armazenadas de forma hierarquizadas em categorias de acordo com as características de seu conteúdo, como área médica ou alimentícia, e cada escritório define qual sistema de classificação utilizar. O *International Patent Classification* (IPC) (WIPO 2019) é um dos sistemas de classificação, cuja organização se baseia numa hierarquia de níveis e subníveis, utilizado em mais de noventa países e abrange as áreas tecnológicas (Fall et al. 2003).

O processo de automatização e análise de patentes tem sido foco de diversos trabalhos, 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 prospecção de oportunidades e, também, para garantir que novos depósitos de patentes não infrinjam leis de propriedade intelectual.

Classificadores automáticos podem auxiliar na busca, análise e atribuição de códigos às patentes, necessárias para que seus conteúdos sejam indexados e consultados dentro das diversas categorias nos repositórios (Benites et al. 2018). Nesse contexto, este trabalho investigou o processamento de algoritmos de AM combinados com diversas formas de representação vetorial de documentos de patentes com o objetivo de analisar o desempenho dos classificadores para um processo automático de busca e recuperação no domínio de patentes.

Este trabalho está dividido nas seguintes seções: Trabalhos Relacionados que apresenta alguns trabalhos semelhantes com o mesmo domínio; Metodologia Utilizada que apresenta as técnicas utilizadas; Experimentos e Resultados que apresenta como foi realizado o processamento dos algoritmos e os resultados obtidos e, por fim, Conclusão e os trabalhos futuros.

Trabalhos Relacionados

São muitos os esforços na busca por modelos que contribuam na melhoria do estado da arte na categorização automática de patentes, com as mais variadas combinações de representações vetoriais como tf-idf, Word2Vec etc., e métodos de classificação como Regressão Logística, K-Nearest Neighbor (KNN), Suport Vector Machine (SVM), Redes Neurais Recorrentes com aprendizado profundo - Convolutional Neural Network (CNN) (Kim 2014), Long Short Term Memory (LSTM) (Hochreiter e Schmidhuber 1997),

[1] https://www.gov.br/inpi/pt-br

etc. Em trabalhos mais recentes, comentados a seguir, são encontrados alguns resultados relacionados ao tema desenvolvido neste trabalho.

Lyu e Han (2019) utilizam várias técnicas de aprendizado profundo, comparando-as com as abordagens tradicionais, como tf-idf + Regressão Logística, na classificação de patentes chinesas. Trabalhando na hierarquia IPC no nível de seção, documentos distribuídos em oito categorias, usaram 8 mil documentos do repositório INCOPAT, contendo título, resumo e texto do primeiro pedido de proteção (*claim*). No pré-processamento realizaram a remoção de palavras comuns *stopwords* e atribuição da morfossintaxe (*part-of-speech tagging*). Usaram na vetorização Word2Vec e tf-idf. Como modelos de treinamento, Regressão Logística e uma combinação de redes neurais recorrentes *Text Convolutional Neural Network* (TextCNN) (Gong e 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. Nos modelos neurais, a melhor combinação obteve acurácia de 81,8%, com Word2Vec + GRU + TextCNN.

Gomez e 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 F1 de 72,56%. Seus experimentos reportaram ainda o desempenho do algoritmo KNN de 64,29% de acurácia e 61,99% de F1.

Molla e Seneviratne (2018) mostram 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 3972 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 F1 (*Micro-averaged*) e acurácia de 78%.

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 documentos, 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. (2017) utilizam o treinamento *Continuous 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 do repositório USPTO Bulk Data. 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. (2018) propõem a classificação de patentes usando Word2Vec e CNN nos 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%.

Jafery et al. (2019) propõem a classificação de patentes baseada nos pilares da indústria 4.0, utilizando o repositório de dados da *Intellectual Property Corporation of Malasya* (MyIPO), para identificar se uma dada instituição está adequada para enfrentar a quarta revolução industrial. No entanto, o conjunto de dados contou apenas com o título e ano de publicação da patente. A metodologia consistiu na coleta de dados, pré-processamento dos documentos, comparação entre cinco diferentes algoritmos de classificação de acordo com os nove pilares da indústria 4.0 e, finalmente, o melhor classificador foi usado para categorizar as patentes baseado nos pilares dessa indústria. Para medir a performance dos classificadores, os dados foram divididos em conjuntos de treino e conjunto de teste com poucas amostras que,

respectivamente, contaram com 202 amostras (70%) e 87 amostras (30%). O melhor classificador foi o *Support Vector Machines* (SVM) que atingiu 97% de precisão.

Lu et.al (2019) apresentam um novo modelo de rede neural, C3-BIGRU-AT, baseado em uma fusão de redes neurais multivariadas. O modelo integra *Convolutional Neural Networks* (CNN) e *Bidirectional GRU* (BIGRU) com um mecanismo de atenção. O modelo apresenta camadas de *word embeddings*, convolucionais, BIGRU, de atenção e softmax. Para medir o desempenho do modelo, as patentes foram coletadas de 5320 (cinco mil trezentos e vinte) artigos de cinco assuntos diversos, distribuídos em 3724 amostras para o conjunto treino (70%) e 1596 amostras para teste (30%). Os documentos foram vetorizados utilizando Word2Vec e o modelo obteve 86.80% de precisão.

Metodologia Utilizada

Como mencionado, os escritórios de patentes possuem formas específicas e utilizam diferentes sistemas de classificação para o armazenamento de patentes. O sistema de classificação IPC é um sistema hierarquizado em diferentes níveis (ou camadas) como Seção, Classe, Subclasse, Grupo e Subgrupo. Para que uma patente possa ser classificada quanto ao nível de Subclasse, a patente precisa obrigatoriamente ter sido classificada quanto ao nível anterior, no caso Seção.

A complexidade no processo de classificação de dados de patentes consiste, além da dimensionalidade inerente dos dados e da sobreposição das classes, também a especialização do conteúdo nessa hierarquia de níveis, pois à medida que vai percorrendo na hierarquia, o assunto ou área que a patente abrange, vai se especializando e assumindo, consequentemente, diversos níveis de classificações, conhecido como um problema multirótulo.

Neste trabalho, foi utilizada a metodologia proposta em Farias et al. (2020), que realizou a seleção de documentos por meio de centroides e um processo automático para realizar o ajuste de parâmetros do classificador escolhido. Considerando os dados de patentes, foi escolhida a hierarquia IPC e o atributo *abstract*. O atributo *abstract* é uma descrição textual resumida da invenção e foi utilizado em conjunto com o atributo Seção da hierarquia IPC, para o processamento dos algoritmos de classificação.

Nesse primeiro momento, apenas o primeiro nível - Seção - é considerado. No nível Seção existem 8 (oito) possíveis classificações, representadas por meio das letras de A até H, as quais descrevem diferentes áreas relacionadas aos documentos das patentes, como química, física e necessidades humanas. A partir de agora, as Seções de A-H serão denominadas de classes de A-H.

O conjunto de dados de patentes utilizado nesta pesquisa foi o WIPO-alpha[2] extraído do repositório da *World Intellectual Property Organization*. O conjunto WIPO-alpha tem aproximadamente 75K documentos escritos na língua inglesa, distribuídos em arquivos de treinamento e de teste, disponibilizados no formato csv, sendo que cada documento diz respeito a uma única patente com os seus atributos de identificação. A distribuição do conjunto de dados pode ser visualizada na Figura 1.

[2] https://www.wipo.int/classifications/ipc/en/ITsupport/Categorization/dataset/index.html

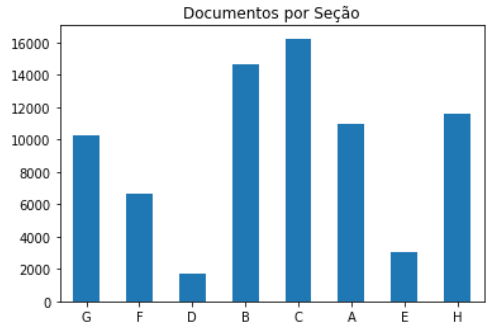

Figura 1. Distribuição dos documentos por Seção

O domínio e a qualidade dos dados, os algoritmos de aprendizado e a representação dos dados, são fatores determinantes no êxito do processo de classificação. Assim, o objetivo desse trabalho foi analisar o desempenho dos algoritmos de AM combinados com diversas formas de representação vetorial de documentos de patentes. Para isso, foram selecionados 4 (quatro) vetorizadores de palavras e 8 (oito) algoritmos de AM, além de um método de seleção de documentos visando à seleção dos documentos mais representativos da classe.

De forma geral, a metodologia desse trabalho segue os seguintes passos. Inicialmente, os dados foram obtidos do repositório WIPO e foi realizado o pré-processamento dos dados, como a limpeza dos documentos e a redução do conjunto de dados diminuindo dessa forma, o tempo de processamento computacional requerido na execução dos algoritmos de AM.

Basicamente, o pré-processamento consistiu nos seguintes passos: (i) eliminação de números e caracteres especiais, como @, _, \#, além da padronização de todas as palavras para caixa baixa; (ii) tokenização dos documentos, na qual o conteúdo do *abstract* foi separado em palavras únicas, chamadas de uni-grama; (iii) remoção das *stop words*, ou seja, palavras comuns que não discriminam significado ao texto, como os artigos e as preposições; (iv) lematização das palavras, com diferentes formas verbais de uma palavra sendo reduzidas para sua forma canônica, por exemplo *studying*, *studies* e *study*.

Após o pré-processamento, foram utilizadas técnicas de *word embedding* para a representação das palavras codificando o seu significado na forma de vetores numéricos, tal que palavras mais próximas no espaço n-vetorial são consideradas mais similares, sintaxe e semanticamente. Para esta tarefa, foram selecionados quatro vetorizadores baseados em *Skip-Gram*, que são o Word2Vec (Mikolov et al. 2013a), o GloVe (Pennington et al. 2014), o Bert (Devlin et al. 2019) e o Doc2Vec (Mikolov et al. 2013b).

Todos esses vetorizadores utilizados trabalham com palavras, com exceção do Doc2Vec que trabalha com documentos. Para estender essa ideia a documentos inteiros, para cada documento *i* são selecionadas as suas *n* palavras. Em seguida, as *n* palavras são vetorizadas obtendo, portanto, *n* vetores numéricos. Após, a partir dos *n* vetores, será calculada a média aritmética simples encontrando a representação vetorial do documento *i*.

Como mostrado na Figura 1, a distribuição dos documentos nas classes está totalmente desbalanceada, sendo que a classe D contém a menor quantidade de documentos (2.3%) e a classe C a maior quantidade (21.6%). Considerando o fato de as classes estarem desbalanceadas e muitas delas estarem com classes sobrepostas no limite do espaço vetorial, decidiu-se por selecionar os documentos mais representativos para discriminar a classe e, assim, possibilitar a redução do tempo de processamento e melhorar o desempenho dos algoritmos de classificação.

A seleção dos documentos mais relevantes foi baseada no cálculo dos centroides de cada classe, consistindo dos seguintes passos: (i) para cada classe *C (i=1...8)*, soma-se os vetores de cada documento *d* e calcula-se a média aritmética, obtendo o centro da classe C_i; (ii) os *n* documentos mais próximos de cada centro C_i são selecionados como sendo os mais representativos daquela classe.

Considerando que para diferentes classificadores existem diferentes hiperparâmetros e a pesquisa manual destes se torna muito custosa quanto ao tempo necessário, foi utilizada a biblioteca HyperOpt[3] que é uma biblioteca de código aberto em Python em AM, para encontrar os hiperparâmetros ótimos para cada um dos oito classificadores selecionados.

Para realização dos experimentos foi utilizada a plataforma Google Colab[4], que disponibiliza, ainda que de forma limitada, recursos computacionais por meio de um *notebook* utilizando a linguagem de programação Python. A configuração disponibilizada pelo Colab foi um processador Intel Xeon, placa de vídeo tesla T4, 12GB de memória RAM e 128GB de espaço em disco.

Experimentos e Resultados

O resultado do pré-processamento dos dados consistiu na redução da dimensionalidade, devido a limpeza e a tokenização das palavras, além da redução de todo conjunto de dados. Como mencionado, o conjunto WIPO-alpha contém aproximadamente 75K documentos e, neste trabalho, foi realizada a seleção dos documentos mais relevantes de acordo com o centroide da classe.

O valor n = 2000 foi definido para que as classes ficassem com a mesma quantidade de documentos, com exceção da classe D que contém apenas 1710 documentos. Assim, o conjunto final de treinamento foi reduzido de 75k para aproximadamente 16k (15710 documentos), uma redução de mais de 78% na quantidade de documentos utilizados se comparada ao conjunto original.

Na Figura 2, é possível visualizar os dados distribuídos em três dimensões, sendo que o gráfico esquerdo representa o conjunto de dados original com 75K documentos e o gráfico direito representa os 16K documentos, após a seleção dos documentos mais descritivos. É possível visualizar uma melhor separabilidade dos dados no gráfico direito, após a redução e seleção dos documentos mais discriminantes. O método utilizado para caracterizar as classes foi o *Linear Discriminant Analysis* (LDA) (McLachlan 1992) no qual o algoritmo tenta diminuir o número de dimensões dos dados, ao mesmo tempo que tenta preservar o máximo de informação. As informações foram codificadas em três dimensões.

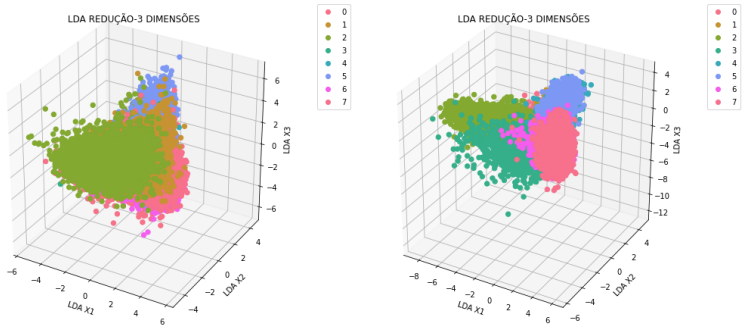

Figura 2. Distribuição das classes em 3 dimensões

Após a seleção dos documentos, os dados foram divididos em conjunto de treino e teste contando com 70% e 30%, respectivamente. O otimizador de parâmetros HyperOpt foi executado com um número de épocas igual a 100 e *cross validation* igual a 5. Os classificadores, escolhidos previamente, foram o K-Nearest Neighbor (KNN), Naive Bayes, Decision Tree, Random Forest, Catboost, XGBoost, Adaboost e LightGBM. Para que os hiperparâmetros sejam encontrados pelo HyperOpt é necessário que seja passado alguma métrica como função a ser minimizada. A métrica escolhida foi a acurácia e, como o HyperOpt tentará minimizar a perda (*loss*), a acurácia foi passada com um sinal negativo.

[3] https://hyperopt.github.io/hyperopt/
[4] https://colab.research.google.com/

Devido ao processamento demorado do algoritmo KNN utilizando a biblioteca `sklearn`, foi utilizada a biblioteca `cuml`, que disponibiliza uma série de algoritmos de classificação implementados para a utilização com placa de vídeo, possibilitando o aumento na velocidade de processamento. Apesar disso, a biblioteca permite apenas a modificação de um único hiperparâmetro, no caso o número de vizinhos, limitando dessa forma a melhora na acurácia que poderia ser buscada modificando outros hiperparâmetros.

Após a definição dos hiperparâmetros e o processamento dos algoritmos, foram obtidos os resultados mostrados na Tabela 1, na qual os valores no formato "xx.x / yy.y" representam a acurácia obtida no conjunto de treino (xx.x) e teste (yy.y) respectivamente.

	Word2Vec	*Glove*	*Doc2Vec*	*Bert*
KNN	80.6 / 80.3	34.6 / 37.7	38.9 / 40.3	75.8 / 77.3
Naïve Bayes	73.7 / 75.3	45.3 / 46.0	69.4 / 67.8	74.0 / 74.6
Decision Tree	61.7 / 61.0	31.0 / 32.1	43.2 / 42.6	51.4 / 51.7
Random Forest	75.8 / 76.8	41.5 / 43.0	58.1 / 59.5	72.2 / 72.6
Catboost	81.8 / 62.5	47.0 / 37.1	71.4 / 52.9	79.0 / 59.6
Light GBM	**82.7 / 83.3**	48.4 / 49.4	72.4 / 72.1	80.9 / 82.0
XGBoost	82.6 / 63.6	47.5 / 43.8	71.4 / 68.3	81.5 / 74.8
Adaboost	65.5 / 38.5	42.2 / 26.6	61.0 / 36.3	64.2 / 33.7

Tabela 1. Resultado dos Classificadores

Como pode ser observado na tabela, o melhor resultado foi obtido pelo algoritmo de classificação Light GBM com o vetorizador Word2Vec, cuja acurácia foi de 82.74% no conjunto de treino e 83.36% no conjunto de teste.

Na Figura 3 é apresentada a matriz de confusão do classificador com melhor desempenho. A partir da matriz em questão, é possível observar as principais dificuldades que o classificador encontrou para discernir entre as diferentes classes, como, as classes G (Física) e H (Eletricidade); as classes A (Necessidades humanas) e B (Transporte e operações); as classes B e E (Construção Fixa). Também é perceptível que dentre as oito classes, o classificador teve o melhor desempenho com a classe C (Química e metalurgia) e o pior na classe D (Têxtil e papel).

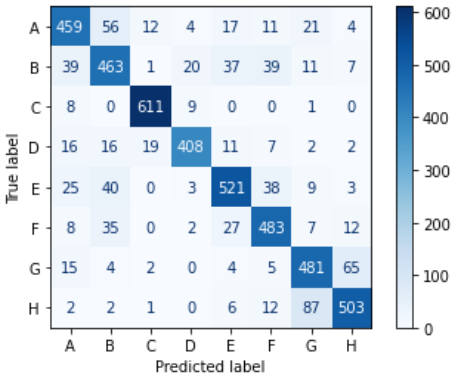

Figura 3. Matriz de confusão LightGBM+Word2Vec

A partir dessas evidências, estão sendo investigados novos métodos para trabalhar com a sobreposição das classes, aplicadas tanto para o conjunto reduzido para 16K quanto para o conjunto total de 75K. Uma outra forma de visualizar e analisar os resultados mostrados na Tabela 1, é apresentada no gráfico da Figura 4 uma comparação da acurácia obtida no treinamento. É possível verificar que a representação de palavras usando Word2Vec teve o melhor desempenho para todos os algoritmos, seguido muito próximo pelo Bert,

enquanto o GloVe obteve o pior desempenho. Também, o classificador Light GBM teve o melhor desempenho em todas as representações de palavras, enquanto o Decision Tree teve o pior desempenho.

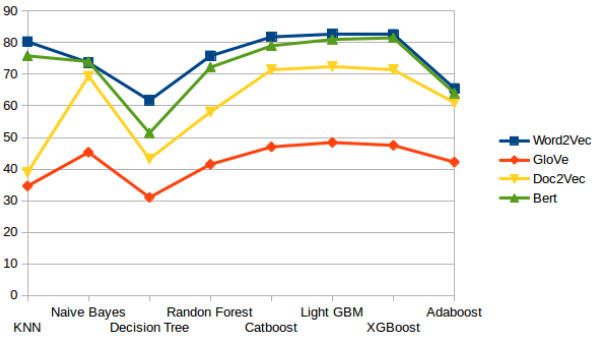

Figura 4. Acurácia do conjunto de treinamento

É interessante observar que o vetorizador Word2Vec, que gera simples vetor de palavras, obteve acurácia ligeiramente maior que o vetorizador Bert, um modelo bidirecional de representação mais recente, que gera múltiplos vetores possibilitando capturar diferentes significados semânticos. Neste trabalho, esse comportamento está sendo estudado e algumas questões abertas estão sendo investigadas. Por exemplo, para este conjunto de dados, o contexto das palavras podem não estar sendo significativamente relevante na discriminação das classes, uma vez que o vetorizador Bert é dependente do contexto enquanto o Word2Vec é independente. A sobreposição do conteúdo entre os inúmeros documentos de patentes pode ser um fator que influencia na discriminação do contexto. O vetorizador Bert gera *embeddings* contextuais de sub-palavras e representação de sentenças e, similar ao Doc2Vec que gera sentença, o desempenho foi inferior.

Ainda, para verificar se o classificador Light GBM de fato obteve melhor desempenho que os outros modelos, ou seja, que o resultado não foi aleatório, ou devido a algum ruído nos dados, foi realizado o teste de hipóteses com o classificador que obteve a segunda melhor acurácia. O teste de hipótese escolhido foi o método *5x2cvt test* (Dietterich 1998), onde, usando a estatística t, o valor de p pode ser calculado e comparado com um nível de significância escolhido anteriormente $s = 0.05$. Se o valor de p for menor que s a hipótese nula é rejeitada e aceita-se que há uma diferença significativa nos dois modelos. Para a comparação entre o LightGBM+Word2Vec e LightGBM+Bert, o valor calculado de p foi de 0.005. Neste caso, como o valor p foi menor que 0.05, a hipótese nula de que os dois algoritmos tiveram um desempenho de maneira igual é rejeitada.

Em uma análise final, é importante ressaltar que os resultados, geralmente, encontrados na literatura para esse domínio variam entre 64 a 94% de acurácia. A comparação entre diversos trabalhos é complexa devido à variação de diversos fatores, como o idioma relacionado com as patentes e o repositório de dados utilizados.

Muitos trabalhos relacionados atingiram uma acurácia em torno de 75% para os dados da WIPO. Aqueles com desempenho superior a 80% foram com idiomas diferentes, como o chinês e/ou um volume menor de dados. Nesse trabalho, o melhor desempenho entre os classificadores foi de 83.36% de acurácia utilizando o vetorizador Word2Vec e o algoritmo Light GBM. É um resultado similar em relação a trabalhos que utilizaram o mesmo conjunto de dados e o mesmo idioma.

Conclusão

Dados de patentes são naturalmente complexos e apresentam diversas regiões de sobreposições, visto que, uma determinada invenção pode conter características de mais de uma classe (Seção) ou assunto ao mesmo tempo. Além disso, a linguagem extremamente técnica e rebuscada de textos de patentes torna o processo de classificação difícil e demorado devido ao volume de documentos e à dimensionalidade inerente dos mesmos.

Neste trabalho foram investigados diversos algoritmos e representações de documentos, para análise de desempenho, a partir da seleção dos documentos mais relevantes na discriminação da classe. A combinação de quatro diferentes técnicas de vetorização dos dados com oito diferentes algoritmos de classificação, com a automatização dos hiperparâmetros, foi possível analisar a acurácia dos algoritmos, obtendo 82.7% de acurácia no treinamento e 83.3% no teste, na melhor combinação. Como próximos passos, técnicas de redes neurais profundas estão sendo investigadas, assim como o tratamento de áreas sobrepostas utilizando o conjunto de dados original com 75K documentos. Também, está sendo criada uma lista de dicionários com as palavras mais discriminantes de cada classe e seus sinônimos, além de possibilitar a identificação, e possível exclusão, de palavras que fazem a intersecção com classes distintas, visando uma melhor linearidade de separação no espaço entre as classes.

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”. URL http://arxiv.org/abs/1811.04695.

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, Association for Computational Linguistics (ACL).

Devlin, J., Chang,M.-W., Lee, K., and Toutanova, K. 2019. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, URL http://arXiv:1810.04805.

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

Fall, C. J., Törcsvári, A., Benzineb, K., and Karetka, G. 2003. “Automated categorization in the international patent classification”, ACM SIGIRForum(37:1), pp. 10–25. URL http://portal.acm.org/citation.cfm?doid=945546.945547

Farias, H. C., Bonfante, A. G., and Martins, C. A. 2020. "Seleção de documentos baseado em centróides para classificacão de patentes usando Word2Vec e KNN". Anais do XLVII Seminário Integrado de Software e Hardware, pp. 269-280. doi: https://doi.org/10.5753/semish.2020

Gomez, J. C., and Moens, M.-F. 2014. “A Survey of Automated Hierarchical Classification of Patents”, Springer, Cham, pp. 215–249. URLhttp://link.springer.com/10.1007/978-3-319-12511-41

Gong, L., and Ji, R. 2018. “What Does a TextCNN Learn?” ArXvi(abs/1801.0). URL http://arxiv.org/abs/1801.06287

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).

Hochreiter, S., and Schmidhuber, J.1997. “Long Short-Term Memory”, Neural Computation (9:8), pp.1735–1780. URL http://www.mitpressjournals.org/doi/10.1162/neco.1997.9

Jafery, W. A. Z. W. C., Omar, M. S. S., Ahmad, N. A., and Ithnin, H. 2019. “Classification of Patents according to Industry 4.0 Pillars using Machine Learning Algorithms”, in 2019 6th International Conference on Research and Innovation in Information Systems (ICRIIS), IEEE.

Jolliffe, I. T. 1986. “Principal Components in Regression Analysis” Springer, New York, NY, pp. 129–155.

Kim, Y. 2014. "Convolutional Neural Networks for Sentence Classification", EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, pp. 1746–1751. URL http://arxiv.org/abs/1408.5882.

Li, S., Hu, J., Cui, Y., and Hu, J. 2018. "DeepPatent: patent classification with convolutional neural networks and word embedding", Scientometrics (117:2), pp. 721–744. URL http://link.springer.com/10.1007/s11192-018-2905-5.

Lu, H., Liu, X., Yin, Y., and Chen, Z. 2019. "A Patent Text Classification Model Based on Multivariate Neural Network Fusion", in 2019 6th International Conference on Soft Computing & Machine Intelligence (ISCMI), IEEE.

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, vol. 2019-June, Institute of Electrical and Electronics Engineers Inc., vol. 2019-June.

McLachlan, G. 1992. Discriminant analysis and statistical pattern recognition, New York: Wiley.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. 2013a. "Efficient Estimation of Word Representations in Vector Space".

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. 2013b. "Distributed Representations of Words and Phrases and their Compositionality".

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.

Pennington, J., Socher, R., and Manning, C. D. 2014. "GloVe: Global Vectors for Word Representation," in Empirical Methods in Natural Language Processing (EMNLP). URL: http://www.aclweb.org/anthology/D14-1162.

Risch, J. and Krestel, R. 2018. "Learning Patent Speak: Investigating Domain-Specific Word Embeddings" in Thirteenth International Conference on Digital Information Management (ICDIM), 2018, pp. 63-68, doi: 10.1109/ICDIM.2018.8846972.

Wipo 2019. "Guide to the International Patent Classification", Tech. rep. URL http://www.wipo.int/classifications/ipc/

Xiao, L., Wang, G., and Zuo, Y.2018. "Research on Patent Text Classification Based on Word2Vecand LSTM", in Proceedings - 2018 11th International Symposium on Computational Intelligence and Design, ISCID 2018, vol. 1, Institute of Electrical and Electronics Engineers Inc., vol. 1.

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, SanDiego, California: Association for Computational Linguistics. URLhttps://www.aclweb.org/anthology/N16-1174