CLASSIFICAÇÃO DE DOCUMENTOS DE PATENTES USANDO O DOC2VEC

TAMARA A. T. MASCAREMHAS[1]; ANDREIA G. BONFANTE[2]; ALBERTO W. MASCARENHAS[3]

[1,2] *Instituto de Computação – Universidade Federal de Mato Grosso (UFMT)*
[3] *Departamento de Eletroeletrônica – Instituto Federal de Mato Grosso (IFMT)*
Campus Cuiabá – MT – Brazil
E-mails: tammi.aguiar@gmail.com, andreia.bonfante@ic.ufmt.br, awmascarenhas@gmail.com

Abstract— Patents are considered extremely useful sources for activities related to the search and analysis of information and the generation of new knowledge. In this article, we use a paragraph vector algorithm doc2vec, an extension of word2vec, which learns representations of sentences in a document, in a supervised deep learning scheme for automatic patent classification. The classification was carried out in documents with summaries of patents in english, in a hierarchical process comprising sections, classes, subclasses according to the International Patent Classification (IPC). The tests were developed in four steps, required, due to the large number of classes and subclasses, with the objective of identifying primary or secondary IPC codes, if it is associated with a set of classifications related to other aspects expressed in the patent. The tests presented very promising results in the classification of patents. The next steps will be to produce qualitative assessments and compare them with other models of machine learning present in the literature

Keywords— Patents, similarity, doc2vec, deep learning.

Resumo— As patentes são consideradas fontes extremamente úteis para atividades relacionadas à busca e análise de informações e para a geração de novos conhecimentos. Neste artigo, usamos um algoritmo de vetor de parágrafo *doc2vec*, uma extensão do *word2vec*, que aprende representações de frases em um documento, em um esquema de aprendizagem profunda supervisionada para a classificação automática de patentes. A classificação foi realizada em documentos com resumos de patentes em inglês, em um processo hierárquico que compreende seções, classes, subclasses, de acordo com a Classificação Internacional de Patentes (IPC). Os testes foram desenvolvidos em quatro etapas, necessária, devido ao grande número de classes e subclasses, com o objetivo de identificar códigos IPC primário ou secundários, caso esteja associado a um conjunto de classificações relacionadas a outros aspectos expressos na patente. Os testes apresentaram resultados bastante promissores na classificação de patentes. Os próximos passos serão produzir avaliações qualitativas e compará-las com outros modelos de aprendizagem de máquina presentes na literatura.

Palavras-chave— Patentes, similaridade, *doc2vec*, aprendizagem profunda.

1 Introdução

Atualmente, os documentos de patentes são considerados uma das melhores opções de fontes de consulta pelo fato de possuírem mais detalhe técnico, organizacional e fácil acessibilidade (Abbas and Khan, 2014). Além disso, é uma fonte de informação de expressiva importância para recuperação de informações tecnológicas e possui características e propriedades que o torna uma fonte de informação extremamente útil para atividades relacionadas à busca e análise de informações e para a geração de novos conhecimentos. Esse modelo de documento possui informações bibliográficas e técnicas altamente estruturadas e internacionalmente padronizadas, dispondo até mesmo de um sistema de classificação especialmente criado para gerenciar e agrupar as invenções, numa tentativa de torná-la mais fácil de recuperar e analisar (Lima et al., 2014). Segundo o Instituto Nacional da Propriedade Industrial - base de patentes brasileira, a documentação de patentes é considerada a mais completa entre as fontes de pesquisa, e estudos apontam que 70% das informações tecnológicas contidas nestes documentos não estão disponíveis em qualquer outro tipo de fonte de informação (INPI, 2016).

Porém, a análise de patentes apresenta três grandes desafios: (I) o número de patentes no mundo cresce de forma constante, (II) tentar entender uma patente é uma tarefa demorada, que pode ser tratada apenas com mão-de-obra qualificada e (III) análise de patentes em escritórios de propriedade intelectual não é tão boa quanto deveria ser (Moehrle, 2010). Essa dificuldade em analisar as patentes é, em grande parte, devido ao uso de termos não padronizados, principalmente quando o produto ainda está no estágio inicial de desenvolvimento.

Recentemente, diversas pesquisas têm convergido para o uso do aprendizado profundo. Este novo paradigma corresponde a uma classe bastante ampla de técnicas e arquiteturas de aprendizagem de máquina. Os algoritmos dessa linha de pesquisa envolvem a característica de usar muitas camadas de processamento de informações não-lineares que são de natureza hierárquica. Percebe-se na literatura que essas técnicas vêm evoluindo e conseguem processar bases de dados com alto volume de informações. A demanda por esse tipo de técnica de aprendizagem é crescente por consequência da maior disponibilidade de dados nos últimos anos, principalmente com o au-

mento da velocidade da internet, crescimento das redes sociais e avanço das tecnologias de comunicação e marketing. Os métodos de aprendizagem profunda buscam simular o comportamento do cérebro humano em tarefas como reconhecimento visual, reconhecimento de fala e processamento de linguagem natural (Medeiros Neto et al., 2017).

Dentre essas técnicas destaca-se o *doc2vec*, recentemente proposto como um método não supervisionado para aprender representações distribuídas para textos. Os textos podem ser de tamanhos variados que vão desde sentenças e parágrafos a documentos que leva em consideração o contexto semântico da frase no qual uma palavra está inserida (Le and Mikolov, 2014).

O objetivo deste trabalho foi utilizar o *doc2vec* num esquema de aprendizagem profunda supervisionada para facilitar e acelerar o processo de classificação de uma patente, assim como, recuperar documentos de patentes com conteúdos similares mesmo pertencendo a classes distintas.

Neste trabalho foi usado uma base de dados de teste com resumos de patentes para treinar e testar esse algoritmo. Também foi feito uma análise baseada no desempenho de respostas as consultas realizadas. Essa análise foi realizada manualmente.

Diversos trabalhos propõem a utilização de algoritmos de aprendizagem para o processamento e a recuperação de textos a incorporação de palavras vetorizadas, cujas similaridades relativas se correlacionam com a similaridade semântica.

Quoc Le e Tomas Mikolov (Le and Mikolov, 2014) demonstram o bom desempenho do *doc2vec* na captura da semântica dos parágrafos. Eles fizeram uma comparação entre os algoritmos *bag-of-word* e *doc2vec*. Nos vários experimentos realizados na classificação de textos no que se refere a análise de sentimentos e recuperação de informações contidos no texto, o *doc2vec* se mostrou com potencial para superar muitas das deficiências do modelo *bag-of-words*. Long Ma e Yanqing Zhang (MA and ZHANG, 2015) desenvolveram uma aplicação útil para grandes dados usando o *word2vec* com uma técnica de agrupamento de vetores de palavras para criar uma dimensão reduzida de dados, e com isso, diminuir o tempo no processo de aprendizagem de máquina e melhorar a capacidade de aprendizagem da aplicação.

Trabalhos com documentos de patentes são encontrados na literatura em (Verma and Varma, 2011), que desenvolveram um trabalho para encontrar patentes similares usando informações e citações do IPC (Classificação Internacional de Patentes) com combinações lineares de similaridade em três níveis de classificação hierárquica e proporção de 0,1, 0,2 e 0,7, respectivamente. (Fall et al., 2003) aplicou o método SVM para classificação em nível de subclasse e conseguiu uma acurácia de 41%. (Tikk et al., 2008) conseguiu acurácia de 53,25% em um sistema de classificação hierárquica baseada em níveis, utilizando um modelo de rede neural para aprender no nível de subclasse. (Verberne et al., 2010) e (Beney, 2010) também trabalharam em níveis de subclasse, em classificação não hierárquica, e conseguiram 56% e 68% nas medidas de F1. (Aiolli et al. 2009) utilizou classificação hierárquica em subclasses com medida de F1 de 52,98%. (Young and Kuhn, 2016) desenvolveram uma medida automatizada da similaridade entre patentes utilizando VSM (*Vector Space Model*). Nenhum desses trabalhos utilizam o *doc2vec* na classificação e recuperação de patentes.

2 Desenvolvimento

O *doc2vec* foi introduzido por (Mikolov et al., 2013b) como uma extensão do *word2vec* que aprende representações de frases em um documento em um esquema de aprendizagem profunda supervisionada.

Inicialmente, Mikolov colaborador do Google, desenvolveu junto com a sua equipe de trabalho algoritmo *word2vec* para calcular representações vetoriais de palavras.

Segundo (Wang et al., 2016), o *word2vec* projeta cada palavra em um espaço vetorial e prediz suas palavras circundantes. O modelo *word2vec* é uma rede neural de três camadas, incluindo camada de entrada, camada de projeção e camada de saída. A camada de entrada corresponde a sinais de contexto (palavras circundantes), a camada de projeção faz a ponte entre a camada ajustando os pesos da rede e a camada de saída identifica os significados semânticos e a relação entre palavras de uma forma muito eficiente de computação. O *word2vec* possui duas arquiteturas para produzir uma representação distribuída de palavras: A arquitetura *CBOW* prevê a palavra atual com base no contexto e o *Skip-gram* prevê as palavras em torno da palavra atual (Mikolov et al., 2013a). As arquiteturas do modelo são apresentadas na figura 1.

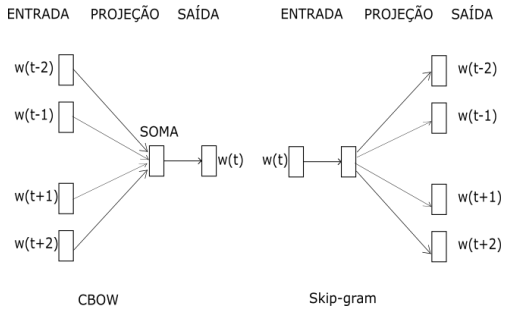

Figura 1. Arquiteturas do *word2vec*.

Depois, foi desenvolvido o *doc2vec* como uma extensão do *word2vec* com o objetivo de criar uma representação numérica de um documento, independentemente do seu comprimento, correlacionando rótulos e palavras, ao invés de palavras com outras palavras. A ideia é chegar a um vetor que representa

o significado de um documento para associar documentos com rótulos.

Esse modelo sugere duas arquiteturas de representação de vetores de parágrafos: um vetor de memória para o modelo de linguagem padrão que visa capturar os tópicos do documento, chamado de ”Memoria Distribuída”. Esse vetor de parágrafo é concatenado ou calculado como média com vetores de palavras de contexto local para prever a próxima palavra. A tarefa de previsão altera os vetores da palavra e o vetor de parágrafo. O vetor de parágrafo pode ser simplificado com o modelo “*Distributed Bag of Words*”, quando não é necessário usar um contexto local na tarefa de previsão. No momento da inferência, os parâmetros do classificador e os vetores da palavra não são necessários e o *backpropagation* é usado para sintonizar os vetores de parágrafo (Dai et al., 2015). As arquiteturas são apresentadas na figura 2 e 3.

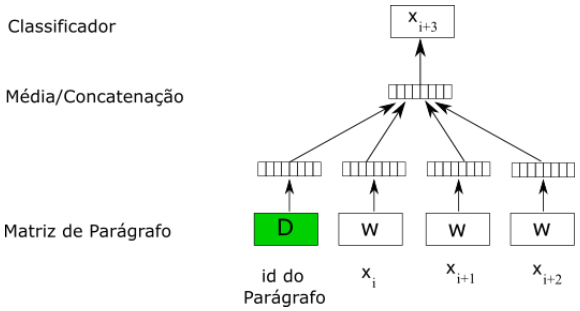

Figura 2. Arquitetura Memória distribuída

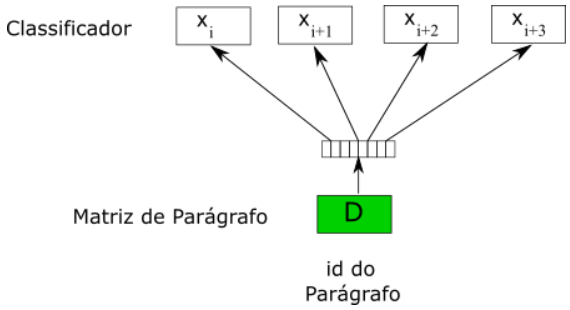

Figura 3. Arquitetura Saco de Palavras Distribuídas.

O desenvolvimento deste trabalho foi dividido em quatro etapas. A primeira foi a organização da base de dados com resumos de patentes em pastas seguindo a mesma classificação internacional de patentes, a segunda foi a organização de grupos de dados para a realização do treinamento, a terceira foi a criação de um algoritmo para busca e classificação de patentes. Esse algoritmo foi desenvolvido a partir de um exemplo fornecido pelo Dl4j (*Deeplearning4j*, 2017). Nesse trabalho foi realizado modificações no algoritmo para que o mesmo classifique textos em três níveis hierárquicos: Seção, Classe e Subclasse. A quarta etapa foi realizar os testes para analisar a precisão do algoritmo na classificação e busca por patentes similares.

2.1. Organização da base de dados

A base de dados foi baixada da internet e organizada de acordo com o sistema de Classificação Internacional de Patentes (IPC), representado por um código alfanumérico para indicar o campo técnico ou os campos técnicos de uma patente. Esse código é organizado de forma hierárquica, numa estrutura em quatro níveis. No nível hierárquico mais alto estão as Seções, representadas pelas letras de A à H. As Seções são subdivididas em Classes representadas por números de 0 a 99, que são subdivididas em Subclasses representadas por letras de A até Z, e, por fim, o ultimo nível, divididos em Grupos Principais e Subgrupos. Um exemplo do código IPC com todos os níveis é apresentado na Figura 4.

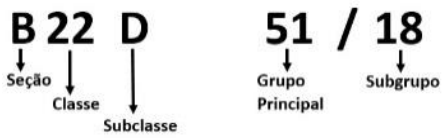

Figura 4. Código do IPC (D22D 51/18).

As publicações de patentes têm uma estrutura padronizada, composta por: Titular, Inventor, Classificação, Prioridade, Anterioridades, Resumo, Relatório descritivo, Reivindicações e Desenhos (eventualmente). Embora uma patente seja constituída de várias partes, neste trabalho utilizou-se apenas os resumos, distribuídos em três conjuntos:

Conjunto 1 – Separadas apenas por Seções, cada uma com 1.000 amostras, totalizando 8.000 amostras;

Conjunto 2 – Separadas em hierarquia de classificação abrangendo Seções e Classes, cada classe com 600 amostras, totalizando 77.400 amostras;

Conjunto 3 – Separadas em hierarquia de classificação abrangendo Seções, Classes e Subclasses, cada subclasse com 300 amostras, totalizando 142.500 amostras.

2.2. Organização da base de treinamento

Os treinamentos foram realizados com 4.000 amostras do conjunto 1 para classificar as seções, com 38.700 amostras do conjunto 2 para a classificar as classes e com 114.000 amostras do conjunto 3 para a classificar as subclasses. Esses dados foram selecionados de forma alternada da base original para criar um vocabulário de palavras mais diversificado.

Na figura 5 é apresentado a hierarquia das pastas: Seção, Classe, Subclasse e os Resumos das patentes

Figura 5. Hierarquia das pastas: Seção, Classe, Subclasse e os Resumos das patentes.

2.3. Algoritmo

O algoritmo utilizado neste trabalho segue um processo de classificação hierárquica iniciando a classificação de um documento pela categoria mais genérica (seção), desdobrando-se para as categorias mais específicas (classes e subclasses), tanto na classificação como na busca por patentes similares. Os parâmetros utilizados no algoritmo foram: *LearningRate* (Taxa de Aprendizagem) - define a taxa de aprendizado inicial, com valor igual a 0,001; *MinLearningRate* (Taxa Mínima de Aprendizagem) - define a taxa mínima de aprendizado após a decadência aplicada, com valor igual a 0.0001; *BatchSize* (Tamanho do Lote) - define o número de amostras que serão propagadas através da rede, com valor igual a 1.000; e *Epochs* (Épocas) - define quantas iterações devem ser feitas em todo o corpo de treinamento durante a modelagem, com valores iguais a 40 (para classificar a seção), 10 (para classificar a classe) e 20 (para classificar a subclasse).

No processo de classificação, o algoritmo faz a separação dos conjuntos para o treinamento e realiza a classificação em quatro etapas. Na primeira etapa o algoritmo calcula a pontuação da similaridade de cada seção e classifica as duas seções (P1 e P2) com maior pontuação de similaridade. Na segunda etapa as classes dessas seções são copiadas em uma pasta e a classificação é novamente realizada para selecionar as duas classes com maior pontuação de similaridade. Na terceira etapa o algoritmo divide as subclasses em conjuntos menores e faz a classificação. Na quarta etapa, o processo é repetido e selecionado a melhor subclasse. Essa classificação é ilustrada na Figura 6.

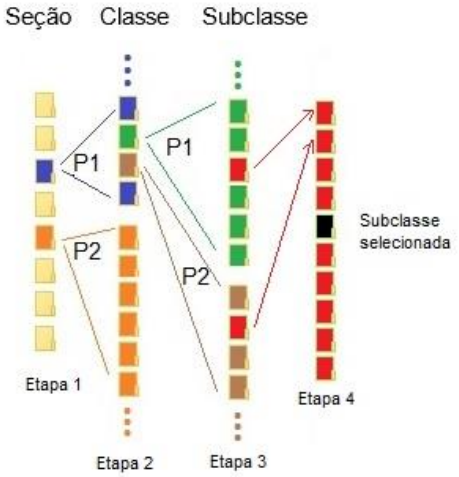

Figura 6. Etapas da classificação de uma patente

3 Resultados

O algoritmo faz a classificação seguindo a hierarquia, isto é, seção, classes e subclasses. Nos testes, os parâmetros utilizados pelo algoritmo foram ajustados em: *LearningRate* = 0.001, *MinLearningRate* = 0.0001, *BatchSize* = 0.0001, *Epochs* = 40.

O algoritmo acertou a classificação das seções em todos os testes realizados, obtendo uma precisão de 100%. Embora esse resultado seja muito bom, o número de seções é pequeno quando comparado ao número de classes e subclasses. Só existem oito se-

ções, cada uma composta por textos com assuntos diferentes. Essa distância entre os textos facilita a classificação realizada pelo algoritmo.

Na classificação da classe o algoritmo seleciona as duas seções com maior pontuação de similaridade (resultado do passo anterior). Nos testes, após ter sido detectado que arquivos com patentes próximas tem mais palavras similares e isso deixa o algoritmo tendencioso, as classes para o treinamento foram escolhidas de forma alternada para abranger todo o conjunto de dados. Essa característica aparece, talvez, por seguir um contexto histórico de desenvolvimento tecnológico. Também contribuiu para essa escolha a visível melhora nos resultados comparados com a escolha de arquivos na forma sequencial. Nos testes realizados o algoritmo utilizou os mesmos parâmetros da etapa anterior, exceto o número *Epochs*, que nesse estágio da classificação é igual a 10. Nessa etapa o algoritmo obteve uma precisão de 80%. Nos 20% restantes, a classe que deveria ser escolhida ficou classificada do segundo ao quinto lugar. Depois dos testes foi realizado uma análise e ficou constatado que esses erros do algoritmo ocorreram por se tratar de assunto que realmente poderia se encaixar em outras classes.

Por exemplo, o arquivo escolhido da classe D03 é tecelagem e ficou classificado na classe D02 que é fios, acabamento mecânico de fios e cordas, urdidora ou tecedura; todos relacionados a indústria têxtil.

Na terceira etapa, o algoritmo realiza a classificação das subclasses. Nos testes realizados o algoritmo continuou com os mesmos parâmetros, alterando apenas o número de *Epochs* para 20. Nesses testes obteve-se uma precisão de 50 %. Nos 50 % restantes a subclasse que deveria ser escolhida ficou classificada no segundo ou terceiro lugar.

Os erros ocorridos pelo algoritmo nessa etapa apresentaram as mesmas características da etapa anterior, onde assuntos de subclasses ou de classe são muitos parecidos. Isso pode ser observado no exemplo do arquivo selecionado A61N – "Dispositivo que auxilia na deglutinação", referente a Produtos de Electroterapia; Magnetoterapia; Terapia de radiação; Terapia de ultrassons; o algoritmo classificou na primeira posição a subclasse A61H, que se refere a Aparelho de terapia física, Dispositivos para localizar ou Estimular pontos de reflexos no corpo; Respiração artificial; Massagem; Dispositivos de banho para fins terapêuticos ou higiênicos. Outro erro de classificação foi com o arquivo "Sistemas, dispositivos e métodos para um objeto móvel suscetível a roubo em um local selecionado", da subclasse G08B, referente a Sinalização ou sistemas de chamada; telegrafas de ordem; sistemas de alarme, e o algoritmo classificou a subclasse G08C que corresponde ao assunto Sistemas de transmissão para valores medidos, controle ou sinais similares.

Outro erro de classificação ocorreu com a subclasse A21C, onde o algoritmo classificou em A22C. Estas são subclasses de classes diferentes, mas ambas se referem a processamento. Enquanto a primeira foca em processamento alimentos em geral, a segunda foca em processamento de carnes, aves e peixes, dificultando a aprendizagem do algoritmo. Outro exemplo é com o arquivo da subclasse E01D, "Ponte temporária para suportar cargas pesadas" (Construção de pontes, estruturas ou viadutos elevados; montagem de pontes), que o algoritmo classificou em E02C (Dispositivos ou Mecanismos de elevação de navios).

Nesse trabalho a base de dados para os testes foi composta de resumos com um texto relativamente pequeno para a rede discriminar o seu contexto, é possível melhorar os resultados se o algoritmo for aplicado a textos completos. Além disso, esses resumos são escritos pelos autores das patentes e traz consigo um vocabulário próprio que pode não ser similar aos dos outros autores de patentes do mesmo grupo.

4 Conclusão

Neste trabalho, é apresentado um cenário de categorização de patentes com o uso do algoritmo de aprendizagem de máquina, *doc2vec*, em um processo hierárquico de classificação. Nos testes realizados, esse algoritmo teve uma eficiência de 80 % na classificação, e conseguiu determinar a seção e classe de patentes similares, mas teve problemas na determinação da subclasse. Talvez, porque nesse trabalho a base de dados é composta de resumos com textos relativamente pequenos para a rede discriminar o seu contexto. Além disso, patentes não tem uma norma de escrita muito rígida e por isso esses resumos contêm vocabulários dos seus autores que pode não ser similar aos dos outros autores de patentes do mesmo grupo. Com o uso desse algoritmo pode-se diminuir o trabalho humano e economizar tempo através de um processo automatizado de classificação de patentes, além de melhorar a qualidade da busca de patentes para avaliar a presença, principalmente, da novidade e atividade inventiva. Os próximos passos serão produzir avaliações quantitativas e compará-las com outros modelos de aprendizagem de máquina presentes na literatura.

Referências Bibliográficas

ABBAS, A.; ZHANG, L. e KHAN, S. U. (2014). A literature review on the state-of-theart in patent analysis. word patent information.

AIOLLI, F., CARDIN, R., SEBASTIANI, F., e SPERDUTI, A. (2009). Preferential text classification: Learning algorithms and evaluation measures. Inf. Retr., 12(5):559–580.

BENEY, J. (2010). LCI-INSA linguistic experiment for clef-ip classification track. In CLEF.

DAI, A. M., OLAH, C., e LE, Q. V. (2015). Document embedding with paragraph vectors. CoRR, abs/1507.07998.

DEEPLEARNING4J 2017, A. E. L. Deep learning for java. https://deeplearning4j.org/. Acesso: 16-05-2017.

FALL, C. J., TO¨ RCSVA´ RI, A., BENZINEB, K., e KARETKA, G. (2003). Automated categorization in the international patent classification. ACM SIGIR Forum, 37(1):10–25.

INPI 2016, I. N. D. P. I. Busca de patentes. http:/www.inpi.gov.br/menu-servicos /informacao/busca-de-patentes. Acesso: 24-06-2017.

LE, Q. V. e MIKOLOV, T. (2014). Distributed representations of sentences and documents. In Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014, pages 1188–1196.

LIMA, L. G., NASSIF, V. M. J., e QUONIAM, L. (2014). Patentes como referência e fonte de pesquisa ao campo do empreendedorismo. In III Simpósio Internacional de Gestão de Projetos (III SINGEP) e II Simpósio Internacional de Inovação e Sustentabilidade (II S2IS), page 4.

MA, L. e ZHANG, Y. (2015). Using word2vec to process big text data. In 2015 IEEE International Conference on Big Data, Big Data 2015, Santa Clara, CA, USA, October 29 - November 1, 2015, pages 2895–2897.

MEDEIROS NETO, F. G., PINTO JUNIOR, R. F., ROCHA, M. G. O., SA JUNIOR, J.J. M., e PAULA JUNIOR, I. C. (2017). Aprendizado profundo: conceitos, técnicas e estudo de caso de análise de imagens com Java, page 465. III Escola regional de informática do Piauí, Picos – Piauí.

MIKOLOV, T., SUTSKEVER, I., CHEN, K., CORRADO, G., e DEAN, J. (2013). Distributed representations of words and phrases and their compositionality. CoRR, abs/1310.4546.

MOEHRLE, M. G. (2010). Measures for textual patent similarities: A guided way to select appropriate approaches. Scientometrics, 85(1):95–109.

TIKK, D., BIRÓ, G., e TORCSVÁRI, A. (2008). A hierarchical online classifier for patente categorization A hierarchical online classifier for patent categorization. Emerging Technologies of Text Mining: Techniques and Applications, 1(1):244–267.

VERBERNE, S., VOGEL, M., e D'HONDT, E. (2010). Patent classification experiments with the linguistic classification system lcs. In CLEF.

VERMA, M. e VARMA, V. (2011). Exploring keyphrase extraction and IPC classification vectors for prior art search. In CLEF 2011 Labs and Workshop, Notebook Papers, 19-22 September 2011, Amsterdam, The Netherlands.

WANG, Z., MA, L., e ZHANG, Y. (2016). A novel method for document summarization using word2vec. In 15th IEEE InternationalConference on Cognitive Informatics & Cognitive Computing ,ICCI*CC 2016, Palo Alto, CA, USA, August 22-23, 2016, pages523–529.

YOUNG, K. A. e KUHN, J. M. (2016). Patent-to-patent similarity: A vector space model.