WALTER RIBEIRO DE OLIVEIRA JUNIOR

O USO DE REDES NEURAIS CONVOLUCIONAIS E TOTALMENTE CONECTADAS PARA A CATEGORIZAÇÃO DE TEXTOS EM IDIOMAS PORTUGUÊS E INGLÊS

Curitiba - PR, Brasil

2018

WALTER RIBEIRO DE OLIVEIRA JUNIOR

O USO DE REDES NEURAIS CONVOLUCIONAIS E TOTALMENTE CONECTADAS PARA A CATEGORIZAÇÃO DE TEXTOS EM IDIOMAS PORTUGUÊS E INGLÊS

Tese apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de doutor em Informática.

Pontifícia Universidade Católica do Paraná - PUCPR

Programa de Pós-Graduação em Informática - PPGIa

Orientador: Prof. Dr. Edson J. R. Justino

Coorientador: Prof. Dr. Flávio Bortolozzi

Curitiba - PR, Brasil

2018

WALTER RIBEIRO DE OLIVEIRA JUNIOR
O USO DE REDES NEURAIS CONVOLUCIONAIS E TOTALMENTE CONECTADAS PARA A CATEGORIZAÇÃO DE TEXTOS EM IDIOMAS PORTUGUÊS E INGLÊS
WALTER RIBEIRO DE OLIVEIRA JUNIOR. – Curitiba - PR, Brasil, 2018-
129 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Edson J. R. Justino

Tese –
Pontifícia Universidade Católica do Paraná - PUCPR
Programa de Pós-Graduação em Informática - PPGIa, 2018.

1. Palavra-chave1. 2. Palavra-chave2. 2. Palavra-chave3. I. Orientador.

II. Universidade xxx. III. Faculdade de xxx. IV. Título

WALTER RIBEIRO DE OLIVEIRA JUNIOR

O USO DE REDES NEURAIS CONVOLUCIONAIS E TOTALMENTE CONECTADAS PARA A CATEGORIZAÇÃO DE TEXTOS EM IDIOMAS PORTUGUÊS E INGLÊS

Tese apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de doutor em Informática.

Trabalho aprovado. Curitiba - PR, Brasil, 8 de agosto de 2018:

Prof. Dr. Edson J. R. Justino
Orientador(a)

Prof. Dr. Flávio Bortolozzi
Convidado 1

Prof. Dr. Paulo Junior Varela
Convidado 2

Prof. Dr. Júlio Cesar Nievola
Convidado 3

Curitiba - PR, Brasil
2018

Este trabalho é dedicado às mulheres da minha vida: Luciane e Carolina. Sem elas, a palavra "felicidade" não teria definição.

Agradecimentos

Inicialmente gostaria de agradecer a Deus. Sem o Arquiteto que planeja tudo que temos, vemos e sentimos, nada mais existiria.

Também gostaria de agrader às minhas famílias: uma na qual nasci, outra que constituí. Estas duas famílias me sustentam, completam e são responsáveis por toda e qualquer qualidade que eu tenha.

Gostaria também de agradecer à PUC, ao PPGIa, à Direção da professora Dra. Andreia Malucelli e à Cheila (que sempre auxilia e salva os alunos).

E, especialmente, ao prof. Justino, que me orientou desde o Mestrado e continuou neste caminho cobrando, exigindo e acalmando durante o Doutorado.

"Um pessimista vê dificuldades em toda oportunidade;
um otimista vê oportunidades em cada dificuldade."
(Winston S. Churchill)

Resumo

A categorização de documentos em classes pré-definidas é um problema ainda em aberto, especialmente para documentos de língua Portuguesa. Neste trabalho é verificado se é possível a utilização de características extraídas por meio de técnicas de TF-IDF e Doc2Vec para que documentos jornalísticos em português e mensagens de grupos de notícias em inglês sejam classificados em classes pré-definidas utilizando-se redes neurais. São testadas redes neurais totalmente conectadas e redes neurais convolucionais, obtendo-se taxas de acerto de aproximadamente 86% em média para documentos jornalísticos em português e 89% em mensagens de grupos de notícias em inglês, com menos de 5.000 documentos sendo utilizados para treinamento.

Palavras-chave: Redes Neurais. Categorização de documentos. Doc2Vec. TF-IDF.

Abstract

The categorization of documents into predefined classes is still an open problem, especially for Portuguese language documents. In this work it is verified the possibility to use characteristics extracted using TF-IDF and Doc2Vec techniques to classify Portuguese journalistic documents and English Newsgroup messages in predefined classes with neural networks based classifiers. Densely connected neural networks and convolutional neural networks are used, with an average accuracy of approximately 86% in Portuguese journalistic documents and 89.44% in English newsgroup messages, with less than 5,000 documents being used to train the neural network.

Keywords: Neural network, Deep learning, Documents, Categorization.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

CNN	*Convolutional Neural Network*
Doc2Vec	*Document to Vector*
FCNN	*Fully Connected Neural Network*
IDF	*Inverse Document Frequency*
LSTM	*Long Short Term Memory*
NGFull	*Base de documentos em idioma Inglês - base completa*
NG05	*Base de documentos em idioma Inglês - base com 5 temas*
NG10	*Base de documentos em idioma Inglês - base com 10 temas*
NG20	*Base de documentos em idioma Inglês - base com 20 temas e menor quantidade de documentos*
NLP	*Natural Language Processing*
Port10	*Base de documentos em idioma Português*
ReLU	*Rectified Linear Unit*
RNN	*Recurrent Neural Network*
TF	*Term Frequency*
TF-IDF	*Term Frequency / Inverse Document Frequency*
VSM	*Vector Space Model*

Sumário

1 Introdução

A popularização dos meios de comunicação em massa virtuais, seja por meio de revistas eletrônicas, *blogs*, plataformas de interação social ou mesmo *e-mails* fez com que a quantidade de informação disponível aumentasse exponencialmente (TAKC; SOGUKPINAR, 2004). De acordo com o dicionário Oxford (Oxford Dictionary, 2017) a expressão "explosão de informações" é utilizada desde a década de 40 para destacar o rápido crescimento da quantidade de informações disponíveis, principalmente devido ao uso, disponibilidade e sofisticação das ferramentas de tecnologia da informação.

Esta alta disponibilidade das informações pode levar à sobrecarga de informações, conforme mencionado por (TOFFLER, 1970): uma quantidade tão grande de informações que se torna difícil acompanhar todo o conhecimento disponível, mesmo que seja sobre um tópico específico, e com isto há dificuldade até mesmo para a tomada de decisões.

Torna-se necessário, então, que existam métodos que auxiliem a busca destas informações para sua posterior assimilação. Uma abordagem é a categorização dos documentos, processo pelo qual os documentos são agrupados em diferentes classes ou categorias (ISLAM, 2017). Esta categorização, por exemplo com a atribuição de cada informação a uma classe, ou a associação a palavras-chave, facilita que a informação seja recuperada quando necessário. Por exemplo, a categorização de livros segundo o Número Internacional Padronizado e sua ficha de catalogação (BRASIL, 2003). Nem sempre, entretanto, as informações são geradas com estes metadados, por exemplo, quando o autor de um *blog* da *internet* não gera nem disponibiliza estas informações mas apenas um conteúdo textual, ou mesmo pode ocorrer a dissociação entre o conteúdo e seus metadados, por exemplo quando o conteúdo de um documento é copiado e enviado a outras pessoas.

A categorização de documentos, atribuindo uma classe principal (que pode ser preestabelecida arbitrariamente) à informação contida neste mesmo documento, pode auxiliar na pesquisa e recuperação de informações, servindo ao mínimo como um filtro que separe informações em categorias desejadas. Ou seja, dado um conjunto de

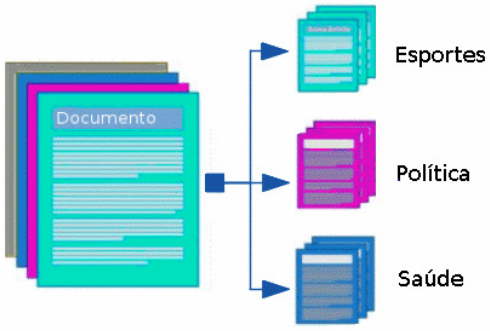
documentos N, deseja-se que a cada documento seja atribuído uma categoria, conforme ilustrado na Figura 1, e desta forma a pesquisa em grandes bases de dados pode ser otimizada restringindo-se o escopo a um subconjunto de todos os documentos. Quando informações sobre as categorias estão disponíveis, rapidamente são utilizadas por pessoas realizando pesquisas e se tornam parte de seus hábitos, auxiliando principalmente quando o *ranking* de resultados de uma pesquisa não compreende corretamente o que era buscado e assim coloca os resultados relevantes em posições desfavoráveis (KÄKI, 2005).

Este trabalho, entretanto, nem sempre pode ser realizado manualmente. Mesmo que se considere que o autor tenha indicado a categoria à qual o documento pertence, esta informação estaria limitada à própria opinião do autor. E, por óbvio, mesmo que existam categorias preestabelecidas sobre áreas de conhecimento humano (por exemplo, as categorias da Classificação Decimal de Dewey (DEWEY, 1876) e as da Classificação Decimal Universal (HARPER, 1954)), estas categorias não são obrigatórias e nem sempre atendem à necessidade pretendida. Para grandes bases de dados o uso do aprendizado de máquina é uma ferramenta possível, onde os dados são analisados por meio de algoritmos que representam o aprendizado extraído de uma série de exemplos e fornecem modelos analíticos/estatísticos (STAMATATOS, 2009)(BISHOP, 2006).

Conforme mencionam (HASHIMI; HAFEZ; MATHKOUR, 2015), necessita-se de sistemas rápidos, automáticos e inteligentes que possam lidar com grande quantidade de dados, extrair informações e assim fornecer subsídios para a tomada de decisões.

O aprendizado de máquina, para obter resultados satisfatórios, deve ser alimen-

Figura 1 – Atribuição de categorias



Fonte: autor (2018)

tado com dados que representem corretamente as informações a serem classificadas. Para isto se faz necessário um pré-processamento destas informações, extraindo-se as características desejadas e suficientes para representar o conjunto de categorias que se deseja obter ao final. Tratando-se de documentos cujo conteúdo é relevante e que serão processados com ferramentas computacionais, neste trabalho pretende-se utilizar apenas as informações disponíveis nos textos, ou seja, o conteúdo da informação, sendo ignorado qualquer outra informação, da mesma forma que considerada por (SEBASTIANI, 2002). Torna-se necessário que as características representem o seu conteúdo independentemente do meio onde foram disponibilizadas (meio físico ou digital), que sejam desconsideradas informações gráficas (por exemplo, a presença de imagens, a disposição do texto no documento e até mesmo a tipografia utilizada) e que todas as metainformações prévias (por exemplo, revista onde ocorreu a publicação) sejam descartadas.

O escopo é limitado apenas ao conteúdo produzido, e assim a análise feita utilizará apenas a linguagem humana. O processamento de linguagem natural (*Natural Language Processing* - NLP) é a área de pesquisa que explora o uso de processamento computacional para o entendimento e a manipulação de linguagem natural para a realização de atividades (CHOWDHURY, 2005). Por meio de suas técnicas, informações podem ser extraídas de documentos para, posteriormente, serem utilizadas em classificadores que analisarão cada conjunto de informações correspondentes a um documento e atribuirão o documento a uma classe pré-definida.

O processo de "atribuição a uma classe" significa que há um documento, cuja classe é ignorada, que será analisado e que por meio de tarefas de classificação terá uma classe mais provável atribuída. Diferencia-se, por exemplo, da verificação de classificação, onde já há uma classe atribuída e deseja-se verificar se o documento realmente corresponde a ela. Diferencia-se, ainda, da extração de informações, tais como palavras-chave, resumo, autoria ou período/época de elaboração.

Entre as diversas técnicas existente para a classificação de documentos, uma abordagem pouco explorada em relação aos documentos escritos em língua portuguesa está o uso de redes neurais. Conforme (FAUSETT et al., 1994), redes neurais artificiais são sistemas de processamento de informações que generalizam modelos matemáticos de como funciona o conhecimento humano. Este sistema é dotado de aprendizado,

definido como um processo de busca por um estado que otimize uma função pré-definida em um espaço multidimensional de parâmetros (HASSOUN, 1995).

Estas redes neurais são caracterizadas por sua arquitetura (padrão pelo qual os seus elementos constitutivos, denominados neurônios, são conectados), algoritmo de aprendizado (método pelo qual são estabelecidos os pesos entre as conexções dos neurônios) e função de ativação (função matemática que determina qual saída será obtida a partir dos valores disponíveis nas entradas) (FAUSETT et al., 1994).

Neste trabalho propõe-se uma abordagem computacional para a categorização de documentos de textos jornalísticos em português e de notícias de grupos de discussão em inglês em classes pré-definidas. São apresentadas diversas características de extração de conteúdo, buscando-se comparar quais são suficientes para a obtenção de um resultado superior aos obtidos em (OLIVEIRA Jr., 2011) e semelhantes aos obtidos por (WITTLINGER; SPANAKIS; WEISS, 2015) em suas bases de dados, mais bem explicadas na seção 2.5. A extração de características é feita com o uso de duas técnicas de NLP: TF-IDF e DOC2VEC. Estas características são utilizadas para alimentar classificadores de redes neurais, e os resultados são comparados com os obtidos anteriormente por (OLIVEIRA Jr., 2011) que utilizou a compressão de dados para a classificação de documentos.

1.1 Objetivo Geral

O objetivo principal deste trabalho é verificar a acurácia da classificação, em categorias pré-definidas, de documentos jornalísticos em Português e de mensagens de grupos de notícias em Inglês com o uso de características extraídas por TF-IDF e Doc2Vec e classificadores baseados em redes neurais totalmente conectadas e redes convolucionais.

1.2 Objetivos específicos

Como objetivos específicos tem-se:

- Obter as bases de dados em português e inglês para realizaão de testes, sendo a base em português composta por documentos jornalísticos e a base em inglês por

mensagens em grupos de discussão;

- Extrair características de documentos utilizando-se TF-IDF e Doc2Vec, gerando vetores adequados para a classificação com redes neurais;
- Realizar a categorização destes documentos com o uso de redes neurais do tipo totalmente conectadas (FCNN) e redes neurais convolucionais (CNN);
- Verificar a robustez do método, aplicando-se o mesmo algoritmo, com os mesmos parâmetros, para bases de dados compostas por documentos em português e em inglês;
- Verificar qual a taxa de acerto média obtida;
- Analisar os resultados com auxílio de matriz de confusão.

1.3 Justificativa/Motivação

Entre os diversos fatores que motivam o presente trabalho está o aumento constante da quantidade de informações disponíveis por meio de documentos nem sempre estruturados. Isto acarreta uma maior dificuldade na realização de pesquisas para a recuperação de informações relevantes. Estes documentos nem sempre estão disponíveis publicamente, existindo bases de dados de acesso limitado (por exemplo, nos sistemas de processos judiciais eletrônicos) com uma grande quantidade de documentos que possuem informações relevantes a um público específico, mas com difícil recuperação e aproveitamento.

Outro fator motivador é a verificação da utilidade de redes neurais para a categorização de documentos em língua portuguesa após a extração de características. Sabe-se que apesar do português ser o 4º idioma mais falado no mundo e o 5º idioma mais falado na *internet* (Instituto Camões, 2017), nem sempre é objeto de pesquisas internacionais. Por exemplo, no grupo de NLP da Universidade de Stanford, os idiomas considerados para a produção de ferramentas são o alemão, arábico, chinês, espanhol, francês e inglês (STANFORD, 2018).

Há também a motivação de que o método estudado seja robusto e genérico o suficiente para ser aplicado a documentos de outros idiomas, com o mínimo de

alterações necessárias no método, evitando-se que a solução proposta seja especializada e de alcance restrito. O desenvolvimento de soluções de categorização de documentos que atendam a mais de um idioma permite uma maior utilização e colaboração em seu desenvolvimento.

Como mencionam (LIU; QIU; HUANG, 2016), redes neurais, em geral, requerem uma grande quantidade de documentos. Isto ocorre devido ao grande número de parâmetros existente em suas camadas, sendo difícil treinar uma rede que consiga generalizar bem com poucos dados. Desta forma, outro fator motivador é verificar se com uma quantidade pequena de documentos (aproximadamente 3.000 documentos em português e 5.000 em inglês) é possível obter bons resultados de categorização de documentos.

Por fim, espera-se obter taxas de acerto médias superiores às obtidas por (OLIVEIRA Jr., 2011), apresentando um método que apresente uma acurácia melhor com testes na mesma base de dados.

1.4 Originalidade/Contribuições

A principal contribuição deste trabalho é a verificação da categorização de documentos jornalísticos em português utilizando redes neurais do tipo *deep learning* a partir de bases de dados contendo um conjunto reduzido de documentos. Há também a verificação se uma solução tecnológia proposta pode ser utilizada em bases de dados contendo documentos em outro idioma (inglês), utilizando-se uma base com mensagens trocadas em grupos de discussão, também contendo uma quantidade reduzida de documentos, com o mínimo de alterações em seus parâmetros.

1.5 Organização do Trabalho

Este trabalho está organizado em 6 capítulos. Este primeiro capítulo, Introdução, apresentou uma breve introdução ao trabalho e tratou dos Objetivo Geral, Objetivos específicos, Justificativa/Motivação, Originalidade/Contribuições e Organização do Trabalho.

O segundo capítulo tratará da Revisão da Literatura, estabelecendo as premissas

conceituais necessárias sobre a extração de características de linguagem natural e o uso de rede neurais como classificadores. O terceiro capítulo diz respeito à Metodologia, detalhando as bases de dados utilizadas, ferramentas de extração de características e o uso dos classificadores. O quarto capítulo trata da Proposta, mostrando as etapas a serem seguidas para o desenvolvimento da abordagem proposta. Por fim, o capítulo 5 trata de Resultados Experimentais e Discussão, para a seguir serem apresentadas as conclusões deste trabalho.

Nos anexos são disponibilizados o código-fonte que serviu de base para esta pesquisa e um pequeno teste realizado com a base de documentos em Português, no qual foi utilizada a mesma abordagem deste trabalho para verificar se é possível realizar a atribuição de autoria de documentos eletrônicos.

2 Revisão da Literatura

Este capítulo tem por objetivo apresentar as premissas conceituais necessárias para a compreensão da proposta bem como o estado da arte dos trabalhos já desenvolvidos em classificação de documentos. São discutidos temas como classificação automática de documentos, processamento de linguagens naturais (NLP), a extração de características dos documentos para as tarefas de treinamento/classificação (sendo escolhidas as técnicas de TF-IDF e Doc2Vec), redes neurais do tipo *Fully Connected Neural Network (FCNN)* e *Convolutional Neural Network (CNN)* e o estado da arte na classificação de documentos.

2.1 Classificação automática de documentos

A classificação automática de documentos é definida como a tarefa de atribuir os documentos a uma ou mais categorias pré-definidas, tendo-se por base o seu conteúdo (SEBASTIANI, 2002)(GOLLER et al., 2000). O uso de ferramentas computacionais tem aprimorado a quantidade e a qualidade do armazenamento, acesso e modificação de dados (BHATTACHARYYA et al., 2008). Para isto, diversas técnicas podem ser utilizadas, entre as quais podem ser citadas o aprendizado de máquina, o uso de classificadores estatísticos e o uso de redes neurais.

Entre as vantagens de se utilizar uma classificação automática de documentos, comparando-se a realização desta tarefa por seres humanos, podem ser citadas o aumento de desempenho (superando, em geral, a quantidade de documentos classificados por seres humanos no mesmo período de tempo), a reprodutibilidade de resultados e a redução da subjetividade na classificação. Como mencionado na Introdução, a quantidade de documentos produzidos sofreu um aumento significativo e isto impacta diretamente no tempo necessário para a categorização, sendo útil o uso de sistemas automatizados para esta tarefa.

A categorização de documentos atende a diversos objetivos, entre os quais podem ser citados: melhorar o desempenho da recuperação de informações com resultados mais relevantes e em menor tempo, automatizar a geração de resumos ou a

extração de outras informações dos documentos e facilitar a organização de grupos de documentos (STEIN BENNO; EISSEN, 2003).

A categorização de documentos possui duas fases principais: uma fase de aprendizado/treinamento e uma fase de classificação.

Na fase de aprendizado ou treinamento, as categorias são definidas e são fornecidos documentos de exemplo ou conjuntos de características que definem a qual categoria cada documento pertencerá (GOLLER et al., 2000). Os classificadores buscam maximizar os exemplos que tornam cada categoria única, por exemplo, atribuindo um maior peso a um conjunto de vetores que identificam cada uma das classes.

A fase de classificação, por sua vez, ocorre quando já há um modelo treinado e os classificadores escolhidos são aplicados a documentos obtendo-se assim a realização da tarefa pretendida. Os documentos para os quais as categorias de pertencimento são ignoradas são submetidos à ferramenta de classificação, obtendo-se uma (ou mais) categorias atribuídas (GOLLER et al., 2000)(DHILLON; KOGAN; NICHOLAS, 2004).

2.2 Linguagem natural

Quando uma pessoa lê ou ouve uma frase, ela utiliza todo o seu conhecimento previamente adquirido e toda sua inteligência para compreendê-la (WINOGRAD, 1972). Isto significa que houve um conhecimento prévio, adquirido ao longo do tempo, onde sons e símbolos foram criados, articulados, interligados e processados, fazendo com que a comunicação fosse possível. Esta comunicação depende de dois grandes elementos: um transmissor e um receptor que, compreendendo o mesmo conjunto de sons ou símbolos, transmitem a informação de um para outro. A linguagem natural, assim, seria qualquer linguagem que evoluiu juntamente com os seres humanos sem que houvesse uma criação consciente. Desta forma, algumas linguagens propostas formalmente, como o esperanto, não seriam linguagens naturais.

Há discussão que a linguagem natural surgiu exatamente junto com a evolução humana, seguindo os mesmos princípios da seleção natural, pois as linguagens humanas são mecanismos complexos de transmissão de informações que possuem uma estrutura definida, com convenções específicas que são adaptadas conforme a sua aceitação social e o seu compartilhamento as propagam (PINKER; BLOOM, 1990).

Figura 2 – Símbolos dos alfabetos cirílico e latino

Fonte: http://sunsite.icm.edu.pl/untpdc/incubator/rus/tpmos/rus-alph.gif, com adaptações do autor

O fato de haver mais de 6.000 línguas diferentes, com quantidade de falantes variados e diferentes abrangências geográficas não invalida a existência de linguagem natural, ao contrário, reafirma que diferentes pressões evolutivas deram origem a diferentes idiomas e dialetos (PINKER; BLOOM, 1990)(GREENBERG, 2000).

Cada idioma e dialeto, por sua vez, possuem características que os tornam peculiares e únicos. Mesmo considerando que esta pesquisa abrange apenas a linguagem escrita, podem ser feitas as seguintes observações. Primeiramente, que o conjunto de símbolos gráficos utilizados para representar uma determinada linguagem não é necessariamente igual ao de outras linguagens, por exemplo, mesmo entre as escritas alfabéticas (onde os símbolos representam letras, e não sílabas ou palavras) existem diferentes representações gráficas para os mesmos sons, como pode ser visto na figura 2.

Em segundo lugar, a frequência da distribuição dos caracteres em cada idioma costuma ser diferente, com peculiaridades sobre quais símbolos são mais utilizados a cada linguagem escrita. Por exemplo, as 5 letras mais frequentes em palavras em diversos idiomas é mostrada na tabela 1.

A frequência de distribuição de caracteres em um documento, tendo-se por base a frequência de distribuição média do idioma ou dos outros documentos que compõe a base de referência, já foi utilizada em diversos problemas relacionados a documentos, por exemplo para a identificação de autoria de documentos (DIURDEVA; SHALYMOV, 2016) e classificação de documentos quanto ao seu idioma (TAKC; SOGUKPINAR, 2004). Apesar de bons resultados para estas tarefas, principalmente pelo desempenho

Tabela 1 – Caracteres mais frequentes em diversos idiomas

	Português	Inglês	Alemão	Polonês
1ª	A	E	E	A
2ª	E	T	N	I
3ª	O	A	S	E
4ª	S	O	R	O
5ª	R	I	I	N

Fonte: o autor, com informações de (WINDISCH; CSINK, 2005).

obtido na identificação de alguns idiomas, o resultado obtido para a categorização de documentos quanto a classes arbitrárias costuma ser insuficiente.

Desta forma, é necessário que os documentos sejam processados, para que seja possível transformar o conjunto de caracteres e palavras em características apropriadas a serem utilizadas em um sistema informatizado.

2.2.1 *Natural Language Processing*

O processamento natural de linguagens - *Natural Language Processing* (NLP) - é uma área de pesquisa que busca descobrir o quanto computadores podem ser utilizados para compreensão e manipulação de documentos que estão codificados em linguagem natural (CHOWDHURY, 2005). Conforme menciona (CAMBRIA; WHITE, 2014), o uso de algoritmos para o processamento básico de documentos tornou-se trivial, havendo grande desempenho nas atividades de recuperar documentos armazenados, separá-los em pedaços delimitados por marcadores, verificar a correção ortográfica e contagem do número de caracteres, palavras ou parágrafos. Grandes desafios existem, entretanto, quando são necessários processamentos mais simbólicos, por exemplo na representação de conceitos abstratos ou compreensão do significado de frases.

Por exemplo, (QIN; XU; GUO, 2016) utilizaram redes neurais para a classificação da função semântica de palavras dentro de frases para a construção de sistemas de pergunta e resposta; (MOU et al., 2016) utilizaram redes neurais para o processamento de conhecimentos armazenados no código-fonte de programas de computador; (PORIA et al., 2016) utilizaram redes neurais para a detecção de sarcasmo e assim verificar a classificação de sentimentos na avaliação de produtos; (MAJUMDER et al., 2017) utilizaran redes neurais para detecção da personalidade dos autores de documentos;

(LEVI; HASSNER, 2015) para a detecção de idade e gênero a partir de imagens; (TRAN; KAVULURU, 2017) utilizaram redes neurais para predizer estados mentais a partir de notas psiquiátricas;(ADEVA et al., 2014) para a revisão de literatura de medicina; (HUYNH et al., 2016) pesquisaram a presença de narrativas de efeitos adversos de drogas em redes de relacionamento; (YAN; SONG; WU, 2016) para construção de sistemas de pergunta/resposta, além de diversos trabalhos que envolvem a análise de sentimentos positivos/negativos em análise de produtos (TANG; QIN; LIU, 2015) - inclusive buscando detalhar qual aspecto do produto foi o objeto do sentimento (PORIA; CAMBRIA; GELBUKH, 2016)-, filmes (SANTOS; GATTI, 2014) e mensagens do Twitter utilizando representações de palavras (SEVERYN; MOSCHITTI, 2015) (ATTARDI et al., 2015) ou caracteres (PRUSA; KHOSHGOFTAAR, 2017). Redes neurais encontram, inclusive, aplicações militares, com (RICHTER; WRONA, 2017) utilizando classificação automática de documentos para auxiliar a tarefa de atribuir graus de confidencialidade a documentos militares.

2.3 Redes Neurais

Conforme menciona (GOLDBERG, 2016), tradicionalmente as principais técnicas para NLP utilizavam aprendizado de máquina com abordagens lineares, tais como regressões logística e *support vector machines*, em geral com vetores de alta dimensionalidade mas com características bastante esparsas.

Estes classificadores tem sido substituídos por modelos de redes neurais, com relativo sucesso, ao serem utilizados vetores com dimensionalidades iguais ou menores (GOLDBERG, 2016).

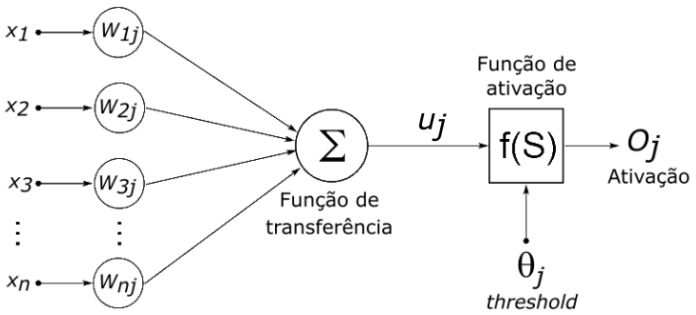
Como observado por (JOHNSON; ZHANG, 2014), a classificação de sentimentos e de tópicos de documentos com o uso de redes neurais pode obter bons resultados, em alguns casos até mesmo com palavras ou frases que apareceram na fase de treinamento mas não estavam presentes no documento testado sendo úteis para a classificação.

As redes neurais são redes de processamento paralelo formadas por diversas unidades de processamento distribuídas e conectadas, denominadas neurônios. Entre as características desejáveis de uma rede neural, podem ser citadas: possibilidade de

estrutura linear ou não-linear, mapeamento de entrada-saída, adaptatividade, resposta com grau de confiança, tratamento de informação contextual e resistência à falhas (HAYKIN, 1999). Inspiradas na forma como o cérebro humano funciona, alguns resultados indicam que o uso de determinado tipo de rede neural (*Matching Network*) apresenta resultados testáveis e previsíveis de como seres humanos aprendem novas palavras (RITTER et al., 2017).

A figura 3 mostra a representação de um neurônio de uma rede neural artificial.

Figura 3 – Neurônio artificial



Autor: Geetika Saini, disponível em https://commons.wikimedia.org/wiki/File:Artificial_neural_network.png

É possível observar que o neurônio possui n entradas, denominadas $x_1 \dots x_n$, cada uma com um peso associado w_{nj}, ou seja, para cada uma de suas sinapses, há uma entrada x que é multiplicada por um peso w, isto para cada sinapse j. Todas as sinapses são agrupadas em uma função de transferência, que é o somatório de cada entrada com seu respectivo peso. Esta função é uma combinação linear, com o resultado u_j, que pode ser expressa pela equação 2.1.

$$u_j = \sum_{j=1}^{n} w_{nj} x_n \tag{2.1}$$

Este resultado u_j é submetido a uma função de ativação $f(S)$, resultando em uma saída o_j. A função de ativação leva em conta um limiar de ativação θ_j, ou seja, o valor da saída o_j dependerá da função $f(S)$ que só será aplicada se a combinação linear u_j for superior ao limiar de ativação θ_j.

2.3.1 Funções de ativação

Diversas funções de ativação são possíveis, sendo as mais comuns as seguintes (DUCH; JANKOWSKI, 1999):

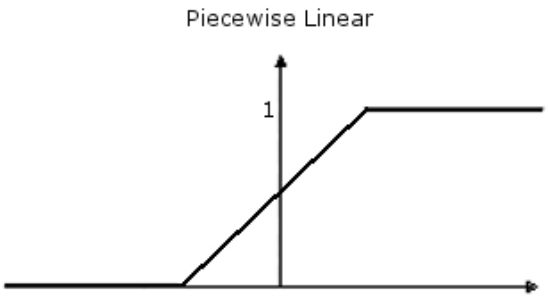
- *step function*: a ativação é feita conforme a equação 2.2. Ou seja, se o valor obtido na função de transferência das entradas for superior a um valor de ativação, a saída terá o valor 1; se o valor estiver abaixo, a saída terá o valor 0.

$$o_j = \begin{cases} 1 & u_j \geq \theta_j \\ 0 & u_j < \theta_j \end{cases} \tag{2.2}$$

- *piece-wise linear:* a ativação apresenta um trecho de linearidade proporcional à entrada, sendo definida pela equação 2.3. Ou seja, a saída dependerá do valor de entrada, com um valor fixo ou linear conforme esteja acima ou abaixo de limites estabelecidos. Neste caso, os limites estabelecidos também podem ser dados pelo liminar de ativação θ_j. A saída desta função pode ser ilustrada conforme a figura 4.

$$o_j = \begin{cases} 1 & u_j \geq \theta_j \\ u_j & -\theta_j < u_j < \theta_j \\ 0 & u_j \leq -\theta_j \end{cases} \tag{2.3}$$

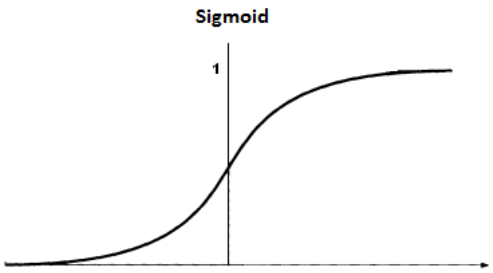
Figura 4 – Função Piece-wise



Fonte: http://www.saedsayad.com/artificial_neural_network.htm, com alterações.

- função padrão logística: é uma função sigmóide, com a saída adquire um formato de S. A função de ativação é dada pela equação 2.4. Nesta equação, β é o parâmetro

Figura 5 – Função sigmóide



Fonte: http://www.saedsayad.com/artificial_neural_network.htm, com alterações.

de inclinação da curva. A saída desta função é ilustrada na figura 5.

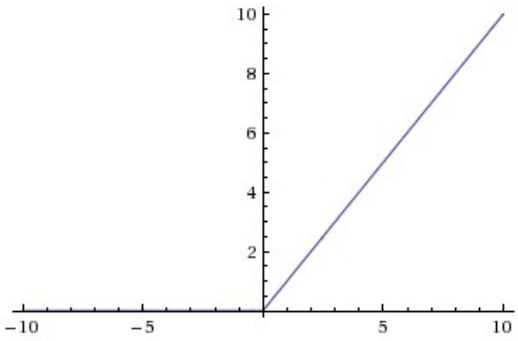
$$o_j = \frac{1}{1 + e^{-\beta u_j}} \tag{2.4}$$

- retificador linear: ReLU (*Rectified Linear Unit*) é uma função de retificação, ou seja, dados os valores de entrada, os valores de saída corresponderão a 0 ou a um valor positivo. O seu processamento, em geral, costuma ser rápido por não envolver cálculos exponenciais (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), obtendo bons resultados em processamento de imagens. Esta função é dada na equação 2.5.

$$o_j = \max(0, j) \tag{2.5}$$

A figura 6 ilustra os resultados da saída quando se aplica a função ReLU.

Figura 6 – Função ReLU



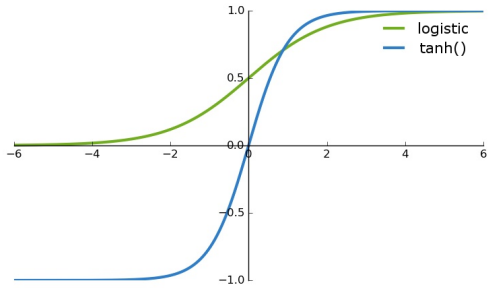
Fonte: https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/relu_layer.html.

- hiperbólica tangente: conforme (LECUN et al., 2012), a função padrão logística sofre do problema de transformar a saída em valores 0 ou positivos, perdendo-se

assim os valores negativos de entrada. Para isto, uma solução é utilizar outra função sigmóide, sendo indicada a hiperbólica tangente. Esta função é dada na equação 2.6. A figura 7 ilustra os resultados obtidos quando se aplica a função padrão logística e a hiperbólica tangente

$$o_j = \tanh(u_j) \tag{2.6}$$

Figura 7 – Funções padrão logística e a hiperbólica tangente



Fonte: http://ronny.rest/blog/post_2017_08_16_tanh/, com alterações.

- *softmax*: a função de exponencial normalizada é conhecida pelo nome de *softmax* por ser uma versão suavizada de uma função que retornaria os valores máximos (BISHOP, 2006)(DUCH; JANKOWSKI, 1999). Esta função transforma um vetor v, com K dimensões, em um vetor $\sigma(v)$, cujos elementos estão compreendidos entre $[0, 1]$ e cujo somatório de todos os seus elementos resulta em 1. Esta função é definida na equação 2.7, sendo v o vetor com K dimensões, j a posição do elemento dentro do vetor v e $\sigma(v)$ o vetor resultante .

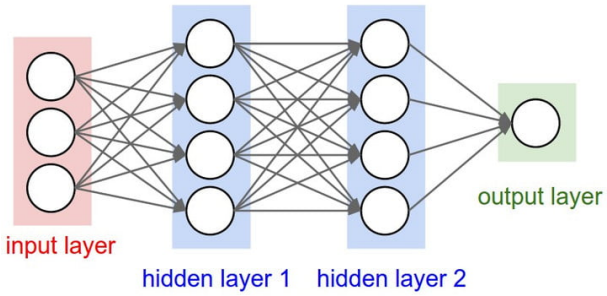
$$\sigma(v)_j = \frac{e^{v_j}}{\sum^K e^{v_k}} \tag{2.7}$$

Por exemplo, se o vetor $v = [1, 2, 3, 4, 3, 2, 1]$ for submetido à função *softmax*, o resultado será um vetor $\sigma(v) = [0.024, 0.064, 0.175, 0.475, 0.175, 0.064, 0.024]$, cuja soma resulta em 1.

2.3.2 Rede Neural

Uma rede neural é formada pela união de diversos neurônios, separados em camadas que desempenham funções específicas. Em geral, as redes neurais são formadas por uma camada de entrada, uma camada de saída, e uma ou mais camadas intermediárias, que representam as camadas escondidas (*hidden layers*).

Figura 8 – Exemplo de rede neural



Fonte: https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/

Um exemplo de rede neural é mostrada na figura 8. Nesta rede de exemplo, todos os neurônios de cada camada anterior são interligados a todos os neurônios das camadas subsequentes, em uma configuração denominada *totalmente conectada*.

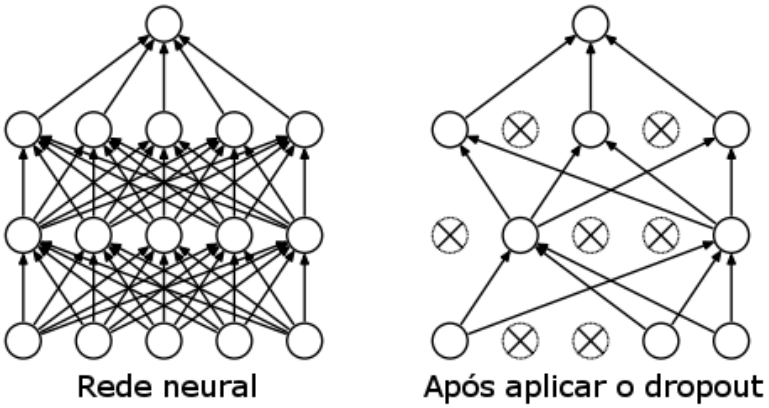
2.3.3 *Dropout*

Conforme mencionam (SRIVASTAVA et al., 2014), as redes neurais sofrem do problema de *overfitting* principalmente pela quantidade de ruído que é criado durante a fase de treinamento. Para minimizar este problema, os autores propõe uma abordagem denominada de *dropout*. Isto é feito removendo-se temporariamente um neurônio da rede neural, incluindo-se todas as suas ligações de entrada e saída. A figura 9 ilustra este processo, com os neurônios removidos sendo marcados com um X e nenhuma conexão sendo feita a eles, nem como entrada, nem como saída.

O processo de remoção é aleatório, ou seja, para cada um dos neurônios, é estabelecida uma probabilidade para sua permanência ou remoção temporária, sendo que o mais comum é esta probabilidade ser estabelecida como parâmetro da rede neural (ou de suas camadas). Desta forma, a cada nova fase de treinamento, alguns neurônios são removidos e outros retornam, fazendo com que a rede neural esteja em constante alteração. Isto equivale a treinar n redes neurais diferentes, sendo que ao final todas serão combinadas e então a fase de testes será realizada com a rede completa (SRIVASTAVA et al., 2014).

Conforme (SRIVASTAVA et al., 2014), houve ganho em todos os testes realizados comparando-se redes neurais com e sem o uso de *dropout*. A quantidade do ganho, entretanto, é variável, sendo muito mais pronunciada em tarefas relacionadas a imagens

Figura 9 – Dropout



Fonte: (SRIVASTAVA et al., 2014)

do que outros tipos de dados.

2.3.4 Retropropagação

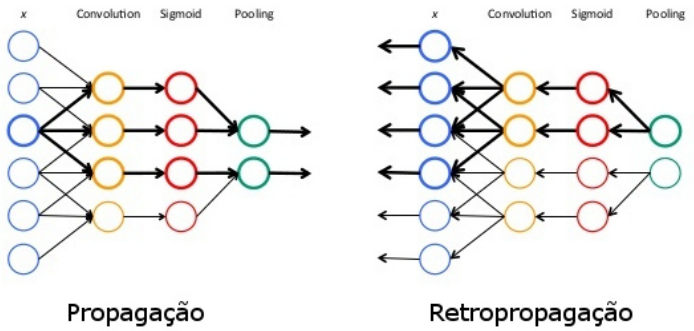
Para que a rede neural possa ser treinada e utilizada como classificador, é necessário que haja um mecanismo para atualização dos pesos dos neurônios, diminuindo-se ou aumentando-se conforme haja uma maior contribuição de cada neurônio para o resultado final. Isto é feito por meio da retropropagação, que é um dos meios de se otimizar uma rede neural (HAYKIN, 1999).

A rede neural, inicialmente, é treinada com a informação desejada (vetores de entrada) sendo propagados desde a camada de entrada até a camada de saída, passando por todas as camadas intermediárias existentes. Assim, obtem-se o valor de saída. Em seguida, é calculado o erro entre o resultado esperado e o efetivamente obtido. A seguir, este erro é retropropagado, seguindo o caminho inverso (indo da camada de saída em direção à camada de entrada), permitindo-se calcular o gradiente para a otimização do peso dos neurônios (AL, 2017) (GOODFELLOW; BENGIO; COURVILLE,).

Conforme menciona (GOODFELLOW; BENGIO; COURVILLE,), o termo retropropagação é erroneamente entendido como todo o processo de aprendizagem, quando na verdade retropropagação é apenas o mecanismo de cálculo do gradiente, enquanto outros algoritmos (por ex., descida estocástica de gradiente) são utilizados para o aprendizado.

A figura 10 ilustra as fases de propagação e retropropagação.

Figura 10 – Propagação e retropropagação



Fonte: https://www.datasciencecentral.com/profiles/blogs/self-learning-machines-deep-convolutional-neural-networks, com alterações

2.3.5 Deep Neural Networks

As redes neurais existem há vários anos (SCHMIDHUBER, 2015) . Se for considerado que redes neurais são variações de métodos de regressão linear, as primeiras referências existem desde o início dos anos 1800, com trabalho de Gauss (1809, 1821) e Legendre (1805). Redes neurais sem aprendizado foram propostas nos anos 1940 por McCulloch e Pitts ((MCCULLOCH; PITTS, 1943)), com redes de aprendizado não supervisionado sendo propostas em 1943 por (HEBB, 1949). Em anos subsequentes foram propostas redes neurais simples com aprendizado supervisionado (e.g., (NARENDRA; THATHACHAR, 1974) e (ROSENBLATT, 1958)).

Os mecanismos de retropropagação, que permitiam o aprendizado supervisionado em redes discretas de profundidade arbitrária, foram desenvolvidos nos anos 1960 e 1970 (SCHMIDHUBER, 2015). Conforme menciona (SCHMIDHUBER, 2015), sua aplicação em redes neurais data de 1981 (WERBOS, 1982), mas a aplicação em redes neurais com muitas camadas não era viável pelo esforço computacional necessário, e apenas a partir dos anos 2000 as redes neurais profundas (*deep neural networks*) ganharam uma maior atenção. Conforme (LECUN; BENGIO; HINTON, 2015), as *deep neural networks* são as redes que possuem diversos níveis de representação, obtidas pela composição de diversos módulos não-lineares que transformam a representação original, sucessivamente, para níveis mais abstratos.

Conforme (LECUN; BENGIO; HINTON, 2015), um dos aspectos principais é que estas camadas de processamento não são definidas previamente e manualmente, ou

seja, as características que são extraídas a cada camada de transformação são treinadas a partir da observação dos dados de entrada por mecanismos de aprendizado gerais. As estruturas abstratas de alta dimensionalidade dos dados são descobertas durante o treinamento, fazendo com que em geral o aumento do poder computacional e a grande disponibilidade de dados para treinamento produzam resultados melhores.

2.3.6 *Topologias de redes neurais*

A topologia de uma rede neural representa a maneira pela qual os neurônios estão conectados para formar esta rede (FIESLER; BEALE, 1996). Diversas topologias para a construção de redes neurais são possíveis. Apenas em caráter ilustrativo, diversas destas redes são mostradas na figura 11.

Para o presente trabalho buscou-se comparar o desempenho de duas destas redes: redes totalmente conectadas e redes neurais convolutivas (CNN). Conforme (GOLDBERG, 2016), redes neurais CNN apresentam bons resultados para classificação de documentos, motivo pelo qual foram escolhidas juntamente com as redes totalmente conectadas para os testes deste trabalho.

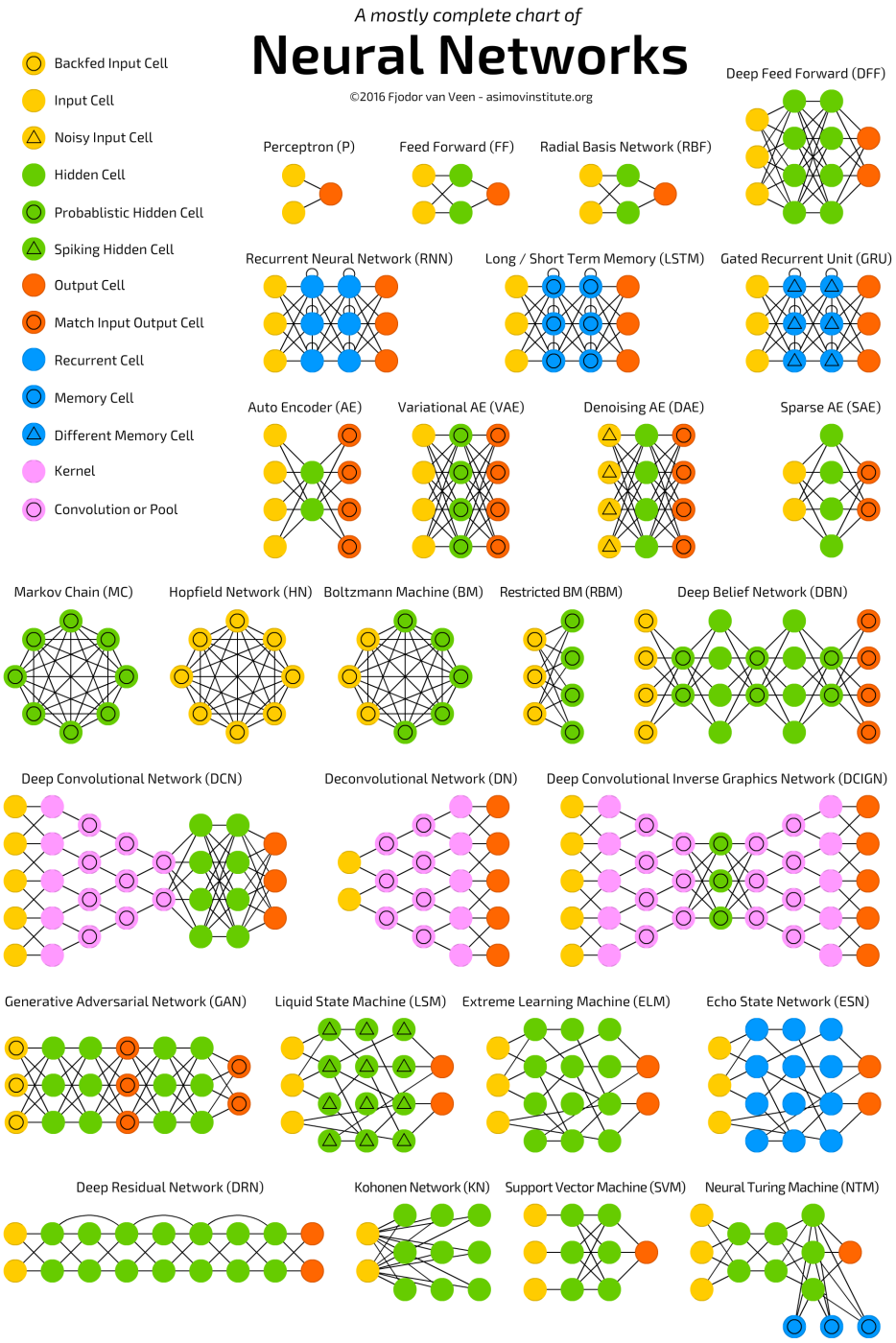
2.3.7 FCNN - Redes neurais totalmente conectadas

Conforme mencionado anteriormente, há um modelo comum de redes neurais originárias, utilizadas didaticamente para explicar as demais. Esta rede é a rede neural totalmente conectada (*Fully Connected Neural Network*), também denominada de redes *Deep Feed Forward*. Nela, todos os neurônios de uma determinada camada conectam-se a todos os neurônios das camadas anteriores e posteriores.

Ou seja, conforme pode ser visto na figura 12, há uma camada de entrada, formada por um número n de neurônios. A saída destes neurônios são ligados às entradas de todos os neurônios da camada seguinte, e assim sucessivamente, até se chegar à camada de saída.

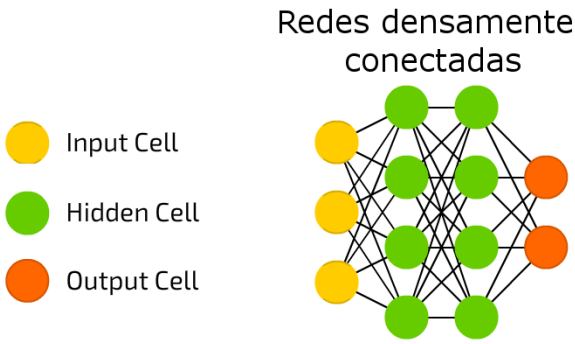
Este tipo de rede neural costuma apresentar como vantagem a facilidade de sua construção e a baixa quantidade de parâmetros necessários, sendo que a maior parte de seus ajustes depende do treinamento realizado e das correções feitas pela retropropagação. Por outro lado, estas redes costumam utilizar um processamento

Figura 11 – Topologias de Redes Neurais



Fonte: http://www.asimovinstitute.org/wp-content/uploads/2016/09/neuralnetworks.png

Figura 12 – Topologia: Redes totalmente conectadas



Fonte: http://www.asimovinstitute.org/wp-content/uploads/2016/09/neuralnetworks.png, com alterações do autor.

intensivo, pois há uma grande quantidade de conexões a serem calculadas. Estas redes podem apresentar uma quantidade variável de camadas escondidas.

2.3.8 CNN - *Convolutional Neural Networks*

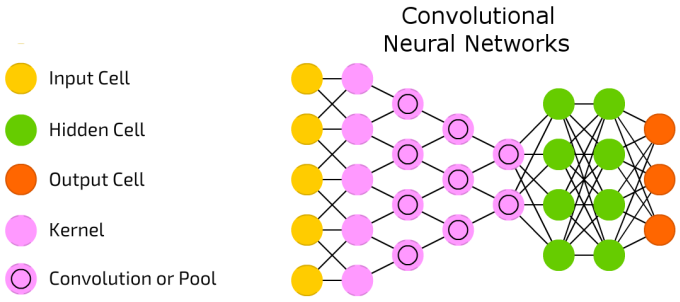
As redes neurais convolucionais (CNN - *Convolutional Neural Networks*) foram introduzidas por (FUKUSHIMA, 1979), e são redes onde um campo receptivo (em geral retangular) é deslocado, passo-a-passo, por uma matriz bidimensional de valores de entrada. A matriz 2D resultante pode então ser utilizada para prover entrada para camadas subsequentes (SCHMIDHUBER, 2015).

Enquanto as redes neurais tradicionais conectam todos os neurônios de saída à entrada de todos os neurônios da camada seguinte (camadas totalmente conectadas), as redes CNN utilizam convoluções, existindo apenas conexões locais(LOPEZ; KALITA, 2017), como pode ser visto na figura 13. As CNN alavancam as relações espaciais, reduzindo assim o número de parâmetros da rede e melhorando o desempenho com mecanismos de retropropagação (BENGIO, 2009)(AL, 2017).

Conforme menciona (COLLINS; DUFFY, 2002), podem ser utilizadas em tarefas de NLP. Em geral as redes CNN são formadas por diversas camadas de dois tipos: camadas de convolução e camadas de subamostragem (BENGIO, 2009) (AL, 2017).

A figura 14 mostra um exemplo simples de como funcionam as CNN. Resumidamente, a partir dos vetores de entrada, são aplicados filtros em diversas regiões da entrada que se sobrepõe ao longo dos vetores (convolução), e em outra camada são

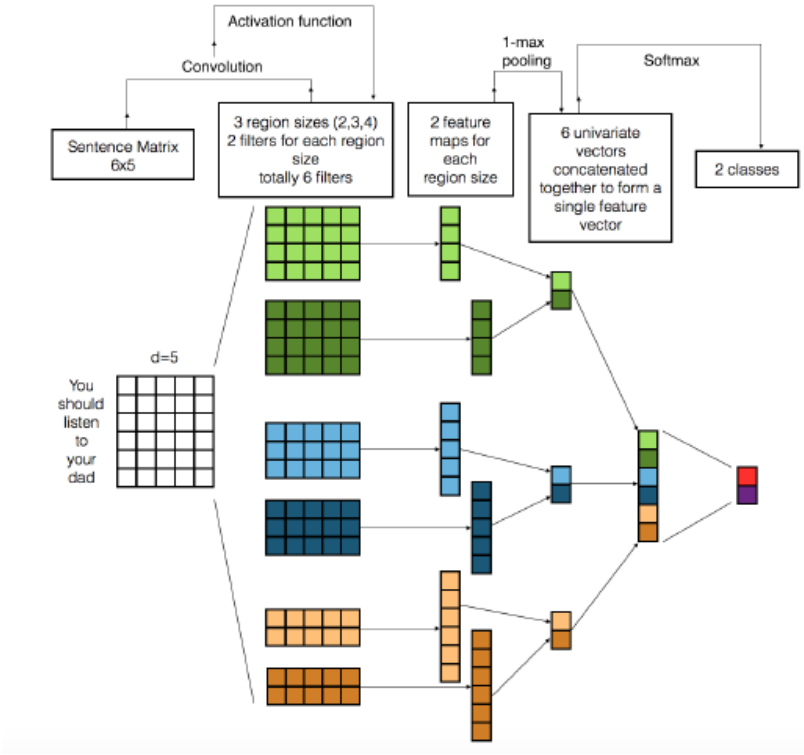
Figura 13 – Topologia: redes CNN



Fonte: http://www.asimovinstitute.org/wp-content/uploads/2016/09/neuralnetworks.png, com alterações do autor.

retiradas amostras. Por fim, uma camada aplica uma função de ativação, tendo-se na saída um vetor com as classes possíveis (no caso, 2 classes).

Figura 14 – Redes CNN: convolução e *pooling*



Fonte: (LOPEZ; KALITA, 2017)

2.3.9 Treinamento, validação e testes de redes neurais

Em geral, para o uso de redes neurais, a base de dados costuma ser divida em 3 categorias (FRIEDMAN; HASTIE; TIBSHIRANI, 2001).

A primeira divisão é feita separando-se documentos para o treinamento. Estes documentos serão utilizados para alimentar a rede neural, obtendo-se o resultado na camada de saída, e realizar a retropropagação, ajustando todos os fatores necessários para que a rede obtenha a máxima performance.

Entretanto, a aplicação somente do treinamento pode implicar no *overfitting*, ou seja, o modelo torna-se altamente especializado em classificar os documentos de teste mas perde sua capacidade de generalizar para demais casos de documentos ainda não vistos. Para evitar isto, uma parcela dos documentos é separada para a validação. Estes documentos constituem uma "reserva" de documentos não vistos pelo classificador, e desta forma o treinamento pode ser repetido com os documentos de treinamento enquanto houver melhora dos resultados do classificador em relação aos documentos de validação.

Por fim, quando o modelo está treinado sem que tenha ocorrido *overfitting*, procede-se ao seu efetivo teste utilizando documentos que ainda não foram utilizados (nem para treinamento nem para validação), que foram previamente separados como documentos de teste.

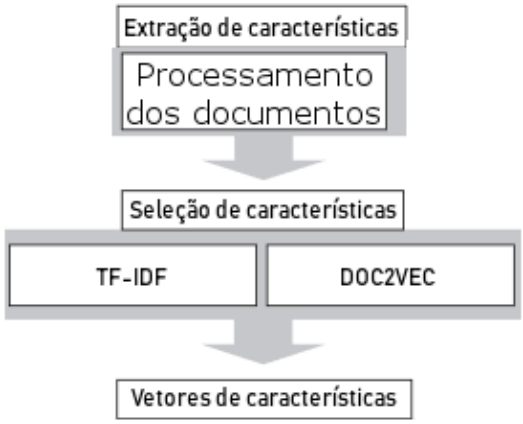
2.4 Extração de características

2.4.1 Pré-processamento

Muitas vezes os documentos não estão organizados com todas as informações necessárias estruturadas, ou mesmo não apresentam meta-informações necessárias para a sua correta classificação. Desta forma, em geral se faz necessária uma etapa de pré-processamento, buscando-se eliminar informações que apenas prejudicarão a tarefa de classificação (evitando-se o efeito *garbage in, garbage out* (DAI; KAKKONEN; SUTINEN, 2011)). Este pré-processamento é feito de duas maneiras: extração de características e seleção de características. A figura 15 ilustra este processo, com as características selecionadas para o presente trabalho.

Na extração de características os documentos são processados para que as características desejáveis, previamente estabelecidas, sejam extraídas e possam ser utilizadas para o treinamento e para a classificação de documentos. Diversas formas de

Figura 15 – Pré-processamento



Fonte: (IRFAN et al., 2015), com alterações do autor.

extração de características são possíveis, por exemplo:

- análise morfológica: onde as palavras do documento são consideradas individualmente. Neste tipo de pré-processamento as palavras são separadas umas das outras (*tokenization*), alguns símbolos podem ser removidos (por ex., sinais de pontuação), as palavras podem ser transformadas (por ex., todas as letras e palavras reduzidas à letras minúsculas), ou até mesmo a retirada de sufixos e prefixos para se obter apenas a raiz das palavras;
- análise sintática: onde as palavras são consideradas em suas relações, formando sequencias longas como frases e orações, observando-se a estrutura gramatical das frases (VARELA, 2017). Nesta técnica é feita a segmentação do texto e/ou a rotulagem das frases que o compõe;
- análise semântica: esta análise busca compreender o significado das frases, com técnicas que filtram o conteúdo útil dos documentos e a classificação de palavras de acordo com os sentimentos expressados ou com sinônimos (IRFAN et al., 2015).

A seleção de características, por sua vez, busca eliminar informações que sejam irrelevantes ou redundantes, reduzindo o documento ao seu conteúdo mínimo necessário. Em geral, isto é feito atribuindo-se um valor a cada uma das palavras do texto, sendo que este valor pode ser determinado por diversas técnicas (IRFAN et al., 2015).

Conforme (JOHNSON; ZHANG, 2015), as redes neurais do tipo CNN foram inicialmente desenvolvidas para o processamento de imagens, que são dados densos que apresentam um tamanho fixo e baixa dimensionalidade. Documentos de texto apresentam tamanho variável, alta dimensionalidade e dados bastante esparsos, então o uso de de redes CNN exige que primeiro sejam feitas modificações, e em geral isto é feito transformando-se o texto em vetores de palavras, caracteres ou outras representações de baixa dimensionalidade. Assim, pequenas regiões de um texto (por exemplo, a sequência de palavras "bom resultado") são transformadas em vetores que são utilizadas nas camadas mais profundas da rede neural, onde convoluções posteriores representam as sequências mais relevantes para a classificação.

Entre as ténicas existentes, duas foram selecionadas para o presente trabalho, e por isto são descritas mais detalhadamente a seguir.

2.4.2 TF-IDF

Esta técnica representa o documento como um modelo de vetores de espaço (VSM - *vector space model*). Cada dimensão representa uma palavra ou frase, sendo possível reduzir vários documentos a uma matriz $m \times n$, com m documentos e n palavras. Desta forma, qualquer entrada da matriz que possua valor superior a 0 indica que aquela palavra está presente no documento, e assim os vetores de características representam o documento.

Dois métodos básicos existem para a obtenção dos vetores de características: a frequencia dos termos (TF - *term frequency*) e a frequência inversa do documento (IDF - *inverse document frequency*) (SALTON; WONG; YANG, 1975) (ROBERTSON, 2004).

Conforme explica (SALTON; WONG; YANG, 1975), em situações onde a recuperação de documentos depende da comparação com outros documentos ou com padrões específicos de busca, uma boa propriedade para indexar estes documentos é a que coloca uma entidade tão longe quanto possível de outras.

E o IDF, proposto por (JONES, 1972), é uma medida discriminante que corrobora esta proposta. O IDF busca quantificar quantas vezes um termo desejado aparece em documentos, considerando que termos que sejam muito frequentes (por ex., artigos como "a", "o") devem possuir um peso menor, e termos mais raros devem ter seu

peso aumentado, sendo possíveis indicadores que a sua presença é importante para determinada classificação.

Conforme (ROBERTSON, 2004), muitos autores tentam explicar de maneira teórica porque o IDF apresenta resultados bons em testes. Dentre as várias explicações possíveis, pode-se considerar que o IDF é apenas uma função probabilística, desde que se assuma que o espaço para os eventos probabilísticos representados no IDF não é conhecido. Também tenta-se explicar conforme a teoria da informação clássica de Shannon, onde o IDF representaria a quantidade de informação, o peso que determinada palavra representa para o documento.

Já o TF, que é a frequência com que um termo aparece em um documento, costuma ser um indicador que tal termo é relevante dentro do documento, não se diferenciando muito do conceito de *bag-of-words*. Associando-se estas duas métricas, tem-se a medida de TF-IDF, onde os vetores são construídos indicando o quanto um termo é frequente em um documento e quanto é raro em outros documentos.

A fórmula de cálculo mais comumente utilizada para TF-IDF, segundo (RAMOS, 2003), é a mostrada na equação 2.8:

$$w_d = f_{w,d} \times \log\left(\frac{|D|}{f_{w,D}}\right) \tag{2.8}$$

sendo que D é o conjunto de documentos dados, w é uma palavra qualquer, d é um documento que pertence ao conjunto D; $f_{w,d}$ é o número de vezes que a palavra w aparece no documento d, $|D|$ é a quantidade de documentos do conjunto D, e $f_{w,D}$ é o número de vezes que a palavra w aparece no conjunto de documentos D.

Observa-se que a função w_d terá um valor maior, portanto indicando uma maior relevância da palavra w, quando $f_{w,d}$ tiver um valor elevado e $f_{w,D}$ tiver um valor pequeno, ou seja, quando a palavra w for frequente em um determinado documento d mas for rara no conjunto de documentos D.

Conforme (ASTRAKHANTSEV; FEDORENKO; TURDAKOV, 2015), a equação 2.8 pode ser reescrita de uma forma mais direta e simples, como pode ser visto na equação 2.9 a seguir, sendo *TF(d)* a frequência que a palavra *d* aparece em um documento

e $TF_D(d)$ a frequência que a palavra d aparece no conjunto de documentos considerados.

$$TF - IDF(d) = TF(d) \times \log\left(\frac{1}{TF_D(d)}\right) \tag{2.9}$$

Desta forma, é possível formar um conjunto de vetores que representem a função TF-IDF das palavras presentes no conjunto de documentos, sendo associado um peso a cada vetor correspondente à frequência desta palavra em um documento específico e no conjunto de documentos.

2.4.3 Doc2Vec

A abordagem Doc2Vec, proposta por (LE; MIKOLOV, 2014) em 2014, aprimora a abordagem Word2Doc proposta por (MIKOLOV et al., 2013). Nestas abordagens, são utilizadas redes neurais para gerar vetores de palavras que conseguem prever, de maneira bastante razoável, quais serão as palavras que estarão próximas no documento.

Inicialmente proposta, a técnica *Skip-gram* (MIKOLOV et al., 2013) busca resolver o problema de que em muitas abordagens de NLP as palavras são tratadas como unidades atomicas, representadas como entradas em dicionários, sem que seja consideradas características similares entre estas palavras ou o seu relacionamento. Desta forma são perdidas informações que podem ser relevantes, por exemplo, que as palavras similares tendem a ficar próximas e também que existem níveis de similaridades entre as palavras (MIKOLOV et al., 2013).

Para a geração destes vetores são utilizadas redes neurais recorrentes (RNN - *Recurrent Neural Network*). Neste caso, a camada de entrada recebe as palavras em um determinado momento t, há uma camada escondida que mantém o histórico das frases, e a camada de saída é dada por um vetor que representa a distribuição probabilística das palavras (MIKOLOV; YIH; ZWEIG, 2013).

Por exemplo, descobriu-se que que é possível realizar operações algébricas nos vetores de palavras e obter assim vetores derivados (MIKOLOV; YIH; ZWEIG, 2013). Desta forma, é possível realizar uma operação de $vetor(\text{"}Rei\text{"}) - vetor(\text{"}Homem\text{"}) + vetor(\text{"}Mulher\text{"})$ obtendo-se um vetor resultante $vetor(1)$ que é muito próximo ao vetor que seria obtido para o *vetor "Rainha"*. Isto se dá porque é possível assumir que, no

espaço de representação de vetores, todas as palavras que guardam uma determinada relação possuem o mesmo *offset* constante, conforme representado na figura 16.

Figura 16 – Vetores representados no espaço

Fonte: (MIKOLOV; YIH; ZWEIG, 2013)

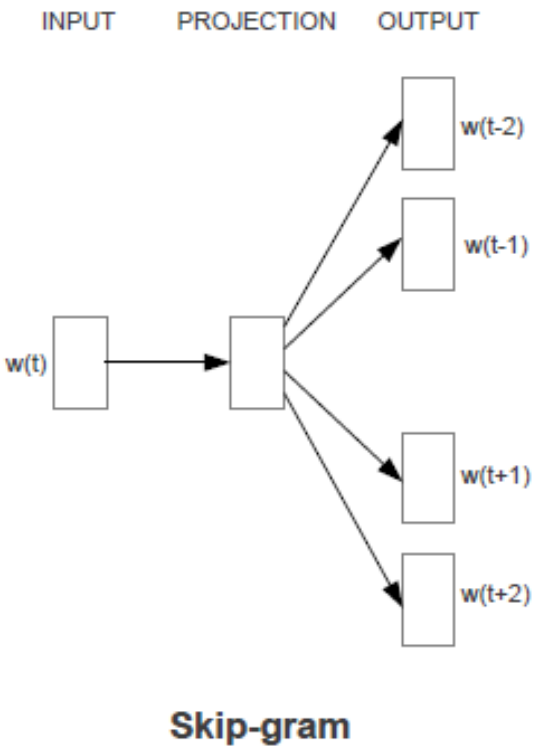
Ou seja, os vetores que representam gênero possuem todos o mesmo deslocamento, como pode ser visto no lado esquerdo da figura. E mesmo relações mais complexas podem ser observadas, no lado direito da figura 16, onde se verifica que a relação entre singular e plural pode ser obtida quando se considera o gênero da palavra. Partindo-se da palavra "King" e de seu plural "Kings", é possível se deslocar para a palavra "Queen" e se chegar ao seu plural "Queens" (MIKOLOV; YIH; ZWEIG, 2013).

Conforme observam os autores, não é fornecida nenhuma informação, neste modelo, sobre semântica, sintaxe ou morfologia das palavras mas mesmo assim os vetores obtidos capturam propriedades de sintática e semântica que permitem, inclusive, executar as operações aritméticas mencionadas anteriormente e obter resultados satisfatórios (MIKOLOV; YIH; ZWEIG, 2013).

Neste modelo, tenta-se maximizar a classificação de uma palavra tendo-se por base outras palavras existentes na mesma frase. A cada palavra considerada, o modelo *Skip-gram* implementado por meio de um classificador de rede neural recorrente (RNN) busca predizer palavras que estejam a uma determinada distância definida, antes e depois da palavra considerada. Este modelo é mostrado na figura 17, onde se observa uma camada de entrada, uma camada escondida de projeção e uma camada de saída, onde são gerados os vetores.

Os autores observam que há uma relação direta entre a distância das palavras, a qualidade dos vetores obtidos e a complexidade computacional para o cálculo (MIKOLOV; YIH; ZWEIG, 2013). E como as palavras mais distantes entre si costumam apresentar uma menor relação do que as palavras que estão mais próximas, há uma

Figura 17 – Modelo Skip-gram



Fonte: (MIKOLOV et al., 2013)

otimização ao se atribuir o peso conforme a distância entre as palavras.

A partir destes conceitos, o método Word2Vec é proposto por (MIKOLOV et al., 2013). Observa-se que o termo "Word2Vec" não foi cunhado pelos autores, mas proposto por (REHUREK; SOJKA, 2010), em sua biblioteca Gensim[1]. Tendo-se por base o modelo *Skip-gram*, algumas otimizações foram feitas, como será mencionado a seguir.

Primeiramente, o modelo *Skip-gram* busca maximizar as probabilidades de se verificar a presença de palavras antes e depois de cada palavra considerada. Desta forma, em geral, quanto maior a distância-limite entre as palavras, maior a qualidade dos vetores obtidos. Como ponto negativo, entretanto, tem-se que o tempo de treinamento é proporcional à quantidade de palavras e esta distância-limite, consumindo-se um grande tempo computacional para pequenos ganhos. Para otimizar este fator, a abordagem Word2Vec deixa de utilizar uma função *softmax* na saída da rede neural e passa a adotar uma abordagem *softmax* hierárquica, reduzindo-se o tempo computacional em aproximadamente $log_2(W)$, sendo W a quantidade de palavras processadas no

[1] https://radimrehurek.com/gensim/

total no treinamento (MIKOLOV et al., 2013).

Outra otimização importante foi em relação à frequência de determinadas palavras. Algumas palavras funcionais, por exemplo artigos e preposições, ocorrem com uma frequência maior nos documentos do que outras palavras, e em geral estas palavras fornecem menos significado útil que as palavras mais infrequentes. Por exemplo, é mais significativo que as palavras "França" e "Paris" apareçam próximas do que as palavras "a" e "França". Para reduzir a utilização de palavras muito frequentes, os autores utilizaram uma fórmula para descartar palavras cuja frequência seja muito elevada em relação às demais. Desta forma, mantendo-se a mesma acurácia, o tempo necessário para processamento pode ser reduzido em aproximadamente 50% (MIKOLOV et al., 2013).

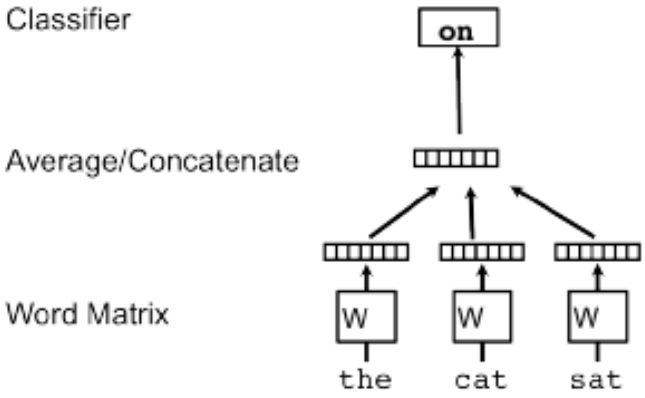
Os autores destacam que com o uso de Word2Vec não apenas as transformações algébricas dos vetores resultam em conhecimentos semelhantes, como no caso dos deslocamentos para verificar gênero e plural, mas que até a adição de vetores pode resultar em vetores com informações relacionadas. Por exemplo, se durante o treinamento as palavras "rio Volga" aparecem na mesma frase que "rio" e "russo", a adição dos vetores $vetor("rio") + vetor("russo")$ resulta em um vetor que é muito próximo ao vetor obtido por $vetor("rio\ Volga")$ (MIKOLOV et al., 2013).

Conforme explicam (LE; MIKOLOV, 2014), neste abordagem, cada palavra é relacionada a um vetor, e o conjunto de vetores é armazenado em uma matriz W, juntamente com a posição da palavra dentro do dicionário. A concatenação dos vetores é usada como característica para predição da próxima palavra dentro de uma frase. Ou seja, dado um conjunto de palavras $w_1, w_2 \ldots w_N$, sendo w cada uma das palavras e N a quantidade total de palavras, o objetivo do modelo de vetores de palavras é maximizar a probabilidade média logaritmica mostrada na equação 2.10:

$$\frac{1}{N} \sum_{n=k}^{N-k} \log p\left(w_n \mid w_{n-k}, \ldots, w_{n+k}\right) \tag{2.10}$$

sendo que w é cada uma das palavras, N a quantidade total de palavras, n é a posição da palavra dentro do dicionário, k é a quantidade de palavras ao redor da

Figura 18 – Modelo de aprendizado de vetores de palavras



fonte: (LE; MIKOLOV, 2014)

palavra w a serem consideradas, log é a função logaritmica a ser maximizada:

$$p\left(w_n \mid w_{n-k}, \ldots, w_{n+k}\right) = \frac{e^{yw_n}}{\sum_i e^{yi}} \tag{2.11}$$

sendo i cada uma das palavras de saída e y_i é uma probabilidade logaritmica não normalizada, definida por:

$$y = b + Uh(w_{n-k}, \ldots, w_{n+k}, W) \tag{2.12}$$

Nesta equação 2.12, U e b são parâmetros da função *softmax* e h é construído pela concatenação dos vetores de palavras extraídos da matriz W.

Esta abordagem pode ser melhor visualizada figura 18, onde é possível se observar a matriz de palavras, os vetores correspondentes, a função que faz a sua concatenação e o resultado obtido.

Posteriormente, (LE; MIKOLOV, 2014) propuseram uma nova abordagem denominada de *Paragraph Vector*, que graças ao trabalho de (REHUREK; SOJKA, 2010) em sua biblioteca Gensim[2], tornou-se conhecido como Doc2Vec. Doc2Vec é um *framework* não supervisionado que apresenta vetores distribuídos de aprendizado contínuo a partir de textos que podem ter tamanhos variados, desde frases até documentos completos (LE; MIKOLOV, 2014).

[2] https://radimrehurek.com/gensim/

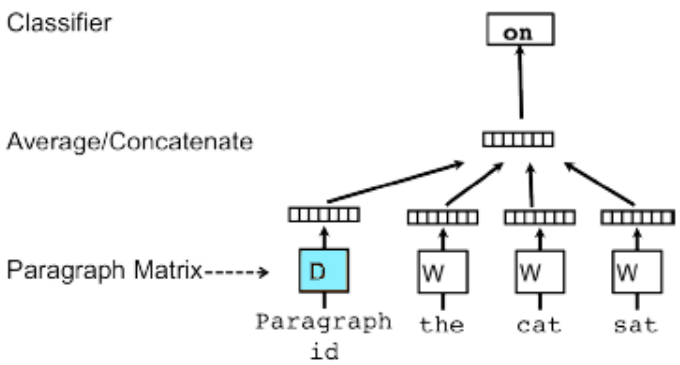
Em seu funcionamento, os vetores de representação são treinados para que consigam predizer palavras que virão a seguir de trechos de texto arbitrários. Isto é feito treinando-se vetores de palavras e de parágrafos com métodos estocásticos de gradiente descendente e retropropagação (LE; MIKOLOV, 2014). Desta forma os vetores dos parágrafos são únicos mas os vetores de palavras são compartilhados entre os parágrafos. Obtem-se, assim, uma representação do documento em forma de parágrafos e as relações existentes entre as palavras (nos parágrafos e no documento como um todo).

Entre suas vantagens, pode ser destacado que são aplicáveis a textos de qualquer tamanho (sejam frases, parágrafos ou documentos completos), não dependem de nenhuma otimização em relação ao peso que deve ser atribuído a palavras e também não dependem de árvores para segmentação ou rotulagem das palavras (LE; MIKOLOV, 2014).

Este modelo Doc2Vec é derivado de Word2Vec e o estende com os vetores de parágrafo sendo utilizados conjuntamente para a predição da próxima palavra. Desta forma, cada parágrafo do documento é mapeada em um vetor de parágrafo, e estes vetores são armazenados em uma matriz D, e posteriormente são concatenados com os vetores de palavra da matriz W, mencionada anteriormente, para a previsão da próxima palavra.

A figura 19 exibe este modelo, destacando-se a nova matriz D que é acrescida juntamente com as informações das palavras.

Figura 19 – Modelo de aprendizado de vetores de parágrafos



Fonte: (LE; MIKOLOV, 2014), com modificações do autor.

Formalmente, a única alteração existente no Doc2Vec em relação ao Word2Vec

é na equação 2.12, que passa a ser representada da seguinte maneira:

$$y = b + Uh(w_{n-k}, \ldots, w_{n+k}, W, D) \tag{2.13}$$

sendo acrescentada a matriz D que armazena os vetores de parágrafo e com os demais termos permanecendo inalterados.

Conforme mencionam (LE; MIKOLOV, 2014), o vetor de um parágrafo pode ser considerado como se fosse apenas uma outra palavra, ou um tópico do parágrafo, que é aplicado localmente apenas ao parágrafo que está sendo trabalhado. Assim, enquanto a matriz W é utilizada no documento inteiro, em todos os parágrafos, a matriz D é utilizada apenas para o parágrafo sendo processado.

Entre as vantagens dos vetores de parágrafo, (LE; MIKOLOV, 2014) mencionam que eles mantém a semântica das palavras, assim como fazem os vetores de palavras, e desta forma o nível de proximidade entre palavras é mantida, enquanto em modelos do tipo *bag-of-words* ou TF-IDF perdem esta informação. Outra vantagem mencionada é que a ordem das palavras é capturada sem a necessidade de uma alta dimensionalidade, isto é, a mesma ordem das palavras que poderia ser mantida por outros métodos (por exemplo, *n-grams* de palavras) é obtida sem que a dimensionalidade se torne inviável.

2.5 Estado da arte

Estudos já utilizaram abordagens semelhantes com o uso de vetores de TF-IDF, Doc2Vec e outros para a classificação de documentos em categorias pré-definidas. Para isto foram utilizados diversos classificadores, tais como compressão de dados, SVM e redes neurais. Alguns destes estudos são relacionados a seguir.

Em 2014, (KIM, 2014) utilizou um conjunto de vetores pré-treinados por (MIKOLOV et al., 2013)[3] para treinar uma rede CNN com apenas uma camada de convolução. Foram testadas 7 base de dados diferentes, com 4 abordagens diferentes para testes, que foram comparados com resultados obtidos por outros autores. As bases de dados utilizadas eram:

[3] Este conjunto de vetores está disponível em https://code.google.com/archive/p/word2vec/ e consiste em uma base de 1.5 GB, com vetores de 300 dimensões para 3 milhões de palavras e frases em inglês, obtidas a partir da base de dados do Google News, de cerca de 100 bilhões de palavras.

- MR: opiniões sobre filmes, a serem classificadas em positivas/negativas;
- SST-1: opiniões sobre filmes, a serem classificadas em 5 categorias de sentimentos, de muito positiva a muito negativa;
- SST-2: a mesma base anterior, mas com apenas 4 categorias de sentimentos, sem haver uma classificação para "neutro";
- Subj: base de dados contendo diversas frases a serem classificadas em subjetivas ou objetivas;
- TREC: base de dados de 500 questões a serem classificadas em 6 categorias, conforme o tipo da questão: se a questão é sobre pessoas, informações numéricas, localizações etc. (LI; ROTH, 2002) (NIST, 2000) ;
- CR: base de dados sobre opiniões de consumidores sobre alguns produtos, devendo ser classificadas em positivas ou negativas;
- MPQA: base de dados contendo opiniões a serem classificadas em positivas ou negativas.

As abordagens utilizadas foram:

- CNN-*rand*: as palavras só foram treinadas a partir das bases de dados utilizadas, sem a utilização de vetores pré-treinados;
- CNN-*static*: vetores pré-treinados, mencionados anteriormente, foram utilizados e permanceram estáticos, sem serem alterados no treinamento;
- CNN-*non-static*: os vetores pré-treinados foram utilizados mas eram alterados durante o treinamento, sendo assim refinados para cada base de dados;
- CNN-*multichannel*: neste modelo são utilizados dois conjuntos de vetores de treinamento, para que um seja otimizado e o outro permaneça estático.

Os resultados obtidos pelos autores foram superiores à literatura de referência em 4 das 7 bases de dados, e nos outros 3 conjuntos os resultados ficaram bastante próximos. Para a classificação de questões, foram usados vetores pré-treinados e o resultado obtido foi de 93,60%.

Em 2015, (JOHNSON; ZHANG, 2015) utilizaram a mesma base de dados de (JOHNSON; ZHANG, 2014), com 734.402 documentos de notícias em inglês e 55 categorizações possíveis, para realizar seus experimentos. Foi testada a geração de vetores de palavras do tipo *two-view-embedding*, que utiliza uma rede CNN não-supervisionada de apenas uma camada escondida para verificar o relacionamento de duas palavras que estejam em um determinado trecho do texto. A seguir, estes vetores são alimentados em uma rede CNN supervisionada, formada de 1.000 neurônios e com apenas uma camada de convolução, com documentos de textos com classes definidas sendo utilizados para atualizar os pesos dos neurônios e assim aprender como utilizar os vetores na classificação. Foi obtida uma taxa de acerto de 92,29%.

Em 2015, (LEI; BARZILAY; JAAKKOLA, 2015) utilizaram redes CNN para a classificação de sentimentos e para a categorização de notícias jornalísticas em idioma chinês. Esta base de dados possui um total de 99.400 documentos, sendo que 80% deles (79.520 documentos) foram utilizados para o treinamento, 9.940 para o desenvolvimento do método proposto e 9.940 (10% dos documentos) utilizados para testes, em um modelo que utilizou vetores de *n-gramas* e o uso de redes neurais com tensores, obtendo-se taxas de acerto de 80% para a categorização de notícias jornalísticas em idioma chinês.

Em 2015, (WITTLINGER; SPANAKIS; WEISS, 2015) estudaram a possibilidade de usar redes neurais flexíveis para a tarefa de processamento de linguagem natural. Os testes foram feitos em uma base de dados de 18.845 documentos de grupos de notícias a serem categorizados em 20 classes com o uso de vetores de palavras. Os melhores resultados ocorreram com um método proposto para que o *dropout* ocorresse de maneira determinística, obtendo-se uma taxa de acerto de 82,9%.

Em 2015, (ZHANG; LECUN, 2015) realizaram testes de categorização de documentos, entre outros. Para isto criaram duas bases de dados:

- Base de dados em Inglês: feita com 4 categorias (mundo, esportes, negócios e ciência/tecnologia) em um total de 241.000 documentos, contendo apenas o título e o campo de descrição das notícias. Por exemplo, um dos documentos desta base da categoria *negócios* é formado pelo título *"ANZ sells project finance unit"* e o conteúdo é a descrição *"Australia amp; New Zealand Banking Group said today it would transfer most of its London-based project finance business to Standard Chartered."*.

O tamanho médio de cada doumento é de 232 caracteres.

- Bases de dados de notícias chinesas: formada por 5 categorias (esportes, finanças, entretenimento, automóveis e tecnologia) em um total de 1.477.649 documentos que foram latinizados, transcrevendo-se ideogramas para a sua pronúncia fonética e limitados a 1014 caracteres.

Na base de notícias em inglês os autores tiveram uma taxa de acerto de 76,73% utilizando-se vetores de palavras extraídos com Word2Vec, 86,69% utilizando-se de vetores *bag-of-words* e 87,18% em uma abordagem CNN com vetores de caracteres (letras minúsculas, dígitos e alguns símbolos especiais) e com ampliação da base utilizando sinônimos.

Na base de notícias em chinês os autores tiveram uma taxa de acerto de 92,78% utilizando a abordagem *bag-of-words* e 95,12% utilizando uma rede CNN com vetores de caracteres.

Posteriormente, em 2015, (ZHANG; LECUN, 2015) publicaram outro trabalho onde detalhavam melhor o seu trabalho de categorização de documentos, especialmente de como a rede CNN era utilizada para a geração de vetores de caracteres.

Em 2016, (LEE; DERNONCOURT, 2016) compararam o desempenho de redes CNN e de redes recorrentes do tipo *Long Short Term Memory* (LSTM) para a classificação de diálogos de frases curtas. Foram testadas 3 bases de dados com 89 categorias, 5 categorias e 43 categorias, respectivamente. Em quase todos os testes as redes CNN tiveram um desempenho melhor que a rede LSTM, com o melhor resultado sendo 84,60%.

Em 2016, (WANG et al., 2016) realizaram a categorização de textos curtos. Conforme observam os autores, textos curtos não fornecem informação suficiente de contexto e assim há o problema de dados muito esparços não fornecerem informação discriminatória suficiente para produzirem bons resultados. Em sua proposta utilizaram vetores de palavras e redes CNN. Para os vetores de palavras foram utilizados vetores já pré-treinados em bases maiores, disponíveis publicamente[4], que foram otimizados para a tarefa por meio de um agrupamento de palavras ao redor de picos de densidade. As palavras do documento eram associadas a estes vetores, submetidas a uma camada

[4] https://www.code.google.com/p/word2vec/

de convolução e a seguir tinham dimensionalidade reduzida em uma camada *K-max pooling*, com classificação sendo realizada na camada de saída por uma função *softmax*. A base de dados era constituída de 12.340 documentos, com tamanho médio de 18 palavras, distribuídos em proporções diferentes em 8 temas (negócios, computadores, cultura-arte-entretenimento, educação-ciência, engenharia, saúda, política-sociedade e esportes). A classificação obteve uma taxa de acerto de 85,5% em vetores obtidos por meio de *Word2Vec*.

Em 2016, (YANG et al., 2016) realizaram testes para a classificação de perguntas e respostas em 10 categorias diferentes: sociedade e cultura, ciência e matemática, saúde, educação e referências, computadores e internet, esportes, negócios e finanças, entretenimento e música, família e relacionamentos e política e governo. A base de dados é composta por 1.450.000 documentos diferentes, com um tamanho médio de 108 palavras por documento. Foi obtida uma taxa de acerto de 75,8% para uma abordagem proposta que utiliza "Redes de atenção hierárquica", onde o contexto das palavras é considerado para verificar quais palavras são mais significativas.

Em 2017, (CONNEAU et al., 2017) utilizaram uma rede CNN com até 49 níveis de convolução para tarefas de classificação de texto. Foram testados 8 bases de dados, sendo 4 para classificação de sentimentos, 1 para classificação em tópicos, 1 para classificação de ontologias e 2 para categorização de notícias jornalísticas (em inglês e em chinês, na base de dados criada por (ZHANG; LECUN, 2015)). Para a categorização de notícias em inglês, foram utilizados 120.000 documentos para treinamento e 7.600 documentos para teste, havendo 4 categorias possíveis para categorização, sendo obtidos resultados inferiores a outros autores, com uma taxa de acerto de 91,33%. Para documentos em chinês, o treinamento foi feito com 450.000 documentos e 60.000 documentos para testes, havendo 5 categorias possíveis de categorização. A taxa de acerto obtida foi de 96,82%. Os autores observam que a taxa de acerto com redes mais profundas (com o maior número de camadas escondidas) é significativamente melhor quando a base de dados é maior.

Em 2017, (JOHNSON; ZHANG, 2017) realizaram pesquisa onde utilizaram as mesmas bases de dados de (ZHANG; LECUN, 2015) com uma nova abordagem denominada de *Deep Pyramid* CNN. Nesta abordagem cada camada subsequente tem seus parâmetros reduzidos para que o tempo computacional seja reduzido à metade,

obtendo-se assim um formato de pirâmide na representação gráfica do custo de cada camada. Foram realizados testes para diversas quantidades de camadas, sendo que com 15 camadas intermediárias os melhores resultados foram obtidos, respectivamente 93,13% de taxa de acerto para a base de notícias em inglês e 98,16% para a base de notícias em chinês.

A tabela 2 apresenta o resumo do estado da arte sobre a classificação de notícias (ou documentos jornalísticos) que são importantes para o presente trabalho.

2.6 Considerações do Capítulo

Neste capítulo foi feita uma breve revisão da literatura aplicável ao trabalho proposto. Foi analisada a relevância da classificação automática de documentos, incluindo as suas fases de aprendizado/treinamento e de classificação. Foi verificado no que consiste a linguagem natural e as peculiaridades existentes para que sejam aplicadas diretamente em sistemas computacionais, existindo uma área de pesquisa específica denominada NLP para que sistemas computacionais possam tratar e realizar atividades com a linguagem natural.

Revisou-se, brevemente, alguns mecanismos de extração de características relevantes para o presente trabalho: TF-IDF e Doc2Vec, apresentando-se conceitos mínimos necessários para a compreensão de sua utilização. Foram introduzidas as redes neurais e conceitos básicos de seu funcionamento, incluindo-se as funções de ativamento dos neurônios e a utilização de *dropout*. Foi dado destaque às redes neurais que serão utilizadas neste trabalho: redes totalmente conectadas (FCNN) e redes convolucionais (CNN).

Por fim, foi feita uma revisão do estado da arte do uso de redes neurais com *deep learning* para a classificação de documentos jornalísticos ou de notícias que possuem relevância para o presente trabalho. Foi possível verificar que há influência do idioma nos resultados obtidos, com o mesmo método obtendo variações de acerto de até 8 pontos percentuais entre idiomas inglês e chinês. Verificou-se, também, que a menor base de dados utilizada para a tarefa de classificação era composta de mais de 12 mil documentos, com a maioria dos testes sendo realizados em bases maiores (acima de 120 mil documentos). Verificou-se, ainda, a escassez de trabalhos em documentos de língua

Tabela 2 – Resumo do Estado da Arte

Autor	Ano	Base de dados	Idioma	Método	Resultados (taxa de acerto)
(KIM, 2014)	2014	7 bases de dados: 6 para classificação de sentimentos, 1 para tópicos de perguntas	Inglês	CNN	93.60%
(JOHNSON; ZHANG, 2015)	2015	734.402 documentos de notícias	Inglês	CNN não supervisionada + CNN supervisionada	92.29 %
(LEI; BARZILAY; JAAKKOLA, 2015)	2015	99.400 documentos jornalísticos	Chinês	tensores de n-gramas	80.00 %
(WITTLINGER; SPANAKIS; WEISS, 2015)	2015	18.845 documentos de grupos de notícias	Inglês	CNN	82.90 %
(ZHANG; LECUN, 2015)	2015	127.600 documentos de notícias	Inglês	CNN	87.18 %
(ZHANG; LECUN, 2015)	2015	510.000 documentos de notícias	Chinês	CNN	95.12 %
(LEE; DERNONCOURT, 2016)	2016	32.000, 109.000 e 221.000 frases	Inglês	CNN e LSTM	84.60 %
(WANG et al., 2016)	2016	12.340 documentos de notícias	Inglês	Word2Vec + CNN	85.50 %
(YANG et al., 2016)	2016	1.450.000 documentos de perguntas e respostas	Inglês	Rede de atenção hierárquica	75.80 %
(CONNEAU et al., 2017)	2017	127.600 documentos de notícias	Inglês	CNN e MaxPooling	91.33 %
(CONNEAU et al., 2017)	2017	510.000 documentos de notícias	Chinês	CNN e MaxPooling	96.82 %
(JOHNSON; ZHANG, 2017)	2017	127.600 documentos de notícias	Inglês	Deep Pyramid CNN	93.13 %
(JOHNSON; ZHANG, 2017)	2017	510.000 documentos de notícias	Chinês	Deep Pyramid CNN	98.16 %

portuguesa, causando uma ausência de parâmetros de comparação para resultados obtidos.

3 Metodologia

Neste capítulo é apresentada a metodologia utilizada para o desenvolvimento do trabalho. Inicialmente são apresentadas as bases de documentos utilizadas. São informadas as fontes de coleta, de disponibilização e a organização e características do conjunto de textos que compõe a base. A seguir são dadas informações sobre a extração de características dos documentos, explicando-se de que forma as representações textuais são transformadas em vetores com representações numéricas de características dos documentos, permitindo assim o uso de rede CNN para a categorização dos documentos.

3.1 Bases de documentos

A classificação de documentos em categorias pré-definidas depende muito do trabalho pretendido, dada a variedade de possibilidades de fontes de dados e as necessidades específicas de categorização pretendida. Como visto na seção 2.5 - Estado da arte, as bases de dados de documentos são compostas de documentos curtos (aproximadamente 100 palavras ou menos), ou de documentos muito curtos (por exemplo, *twitters* de no máximo 140 caracteres, ou título e linha de resumo de notícias, com aproximadamente 50 palavras). Em alguns casos de textos longos, em geral obras literárias, são utilizados capítulos inteiros de livros, resultando em textos de mais de 1.000 palavras. Há uma ausência, em geral, de textos intermediários, representativos de notícias jornalísticas.

Portanto, optou-se pelo uso de duas bases de dados já existente para o presente trabalho. Estas bases serão detalhadas a seguir.

3.1.1 Base de dados Port10

A base de dados denominada Port10 é formada por textos em língua portuguesa extraídos de colunas de jornais, disponibilizados durante os anos de 2008 e 2009, tendo sido coletada originalmente por (VARELA, 2010). Para a composição desta base foram utilizados os seguintes jornais, observando-se que, entre estes, alguns não existem mais,

tendo encerrado suas atividades: A Gazeta do Acre, A Gazeta do Povo, A Notícia, Colunistas IG, Diário do Grande ABC, Folha UOL Online, Jornal de Beltrão, Jornal de Brasília, O Estado do Paraná, O Extra, O Gerente, O Povo, O Tempo, Paraná On-Line e Zero Hora.

Foram estabelecidos 10 temas para a categorização de notícias:

- Assuntos Variados
- Direito
- Economia
- Esportes
- Gastronomia
- Literatura
- Política
- Saúde
- Tecnologia
- Turismo

Observa-se que estes 10 temas possuem alguma semalhança com os temas utilizados em trabalhos mencionados na seção 2.5 - Estado da arte, mas com algumas adições (por exemplo, literatura e direito são temas que não costumam aparecer em outras bases de dados).

Para cada um destes 10 temas foram selecionados 10 autores. Ou seja, há um total de 100 autores, com cada tema possuindo 10 autores diferentes. Os autores selecionados possuem relevância nacional, portanto não é incomum que seus textos sejam reproduzidos em mais de um jornal ou mesmo estejam presentes em livros de coletâneas. Por exemplo, para o tema Saúde, foram escolhidos os seguintes autores: Claudio Lima, Drauzio Varela, Fabio Cesar dos Santos, Fernanda Aranda, Flavio Settanni, John Cook Lane, Leandro Perché, Leo Kahn, Liliane Ferrari e Loir Carlos Costa.

Para cada um dos autores foram selecionados e coletados 30 documentos, cuidando-se para que não houvesse repetição, ou seja, evitando-se que um mesmo texto publicado em mais de um jornal fosse selecionado repetidamente. Desta forma cada um dos temas possui 300 documentos diferentes, havendo um total de 3.000 documentos na base de dados. Estes documentos possuem, aproximadamente em média, 550 palavras e 3.300 caracteres, com a base de dados totalizando um tamanho de 8,35 MB.

Figura 20 – Exemplo de documento em Português

Mais quatro olhos em campo
Um tempo de meditação de Michel Platini, hoje um dos homens de confiança de Joseph Blatter na Fifa, pode introduzir, em breve, uma saudável inovação no sistema de arbitragem mundial. Platini falou para os maiores cartolas do futebol do planeta em Nassau, nas Bahamas, quando do sorteio das cidades que serão sedes da Copa de 2014, que o experimento de mais dois assistentes atrás das metas utilizado pela Uefa, recentemente, terá continuidade em todos os campeonatos da entidade a partir de agosto do ano em curso.
O maior craque francês de todos os tempos entende que, se hoje o quarteto de árbitros não consegue detectar todos os incidentes que ocorrem no campo de jogo em função de que o futebol está muito rápido e os jogadores estão cada vez mais bem preparados, é necessário a presença de mais dois homens de preto, atrás das metas, objetivando observar lances que fujam da percepção visual do trio de árbitros na área de pênalti.
Entendo que é uma estratégia que deve ser cultivada, já que o autor da idéia, como craque que foi, sabe que o problema dos árbitros está na frente das traves, mas que para ser percebido melhor, tem que ser observado por trás das mesmas, o que virá solucionar os momentos "dramáticos" que vivem os árbitros na atualidade. É hora de pensar neste novo instrumento humano que visa colocar num jogo de futebol exatamente cinco agentes do apito.
Promessa auspiciosa
Ricardo Marques Ribeiro, 29 anos, (Fifa-MG), o mais jovem árbitro do Brasil na Fifa, dirigiu na quarta-feira passada, pelas semifinais da Copa do Brasil, Coritiba 1 x 0 Inter-PR. Em que pese o seu noviciado, realizou a melhor arbitragem do ano em Curitiba. Seu estilo de arbitragem encaixou-se dentro dos parâmetros da Fifa, que determina que as partidas devem ser disputadas como espetáculos de entretenimento ao público, com um mínimo de interrupções, com o árbitro tendo o poder discricionário de distinguir um choque casual de uma falta. Isso foi possível, porque Ricardo Marques demonstrou extremo equilíbrio, critérios equânimes na marcação de faltas, nos aspectos disciplinares, persuasão, ótimo posicionamento em todo o transcorrer do jogo. E principalmente, bom senso em todos os incidentes protagonizados pelos atletas das equipes nominadas. Passou num teste dificílimo.Tomara que não altere a sua característica de apitar, pois quem ganha com isso é o futebol brasileiro.
PS: Sugiro à Comissão de Árbitros da FPF, exibir o DVD da arbitragem aqui mencionada ao quadro de arbitragem do Paraná, que há muito tempo está carente de ensinamentos com qualidade.

Fonte: Valdir Bicudo

Considerando-se que os documentos foram retirados de *sites* de *internet* dos jornais, foram feitos processamentos em relação aos elementos existentes nas páginas e em seus elementos gráficos para que apenas o conteúdo fosse extraído. Por exemplo, retiradas todas informações de *links*, características gráficas de fontes ou de itálicos ou negritos, entre outras.

Por exemplo, o conteúdo de um documento da categoria Esportes é mostrado na figura 20.

Como pode ser observado, foram mantidas as acentuações gráficas, sinais de pontuação e divisão em parágrafos. Tendo-se o conteúdo do texto, foi realizada uma limpeza dos dados, quando necessário, para a retirada do nome/assinatura do autor, indicativos da seção do jornal ou do assunto do documento. Por exemplo, se a primeira linha do documento fosse "Drauzio Varela/Saúde", esta informação era suprimida, para que não restasse no documento referências à sua categoria. Por outro lado, não foram feitas substituições no corpo do documento, portanto se em um documento da categoria "Esporte" houvesse uma frase "O esporte no Brasil", a palavra "esporte" não

sofreria qualquer alteração.

Estes documentos foram armazenados em arquivos simples de dados, do tipo `txt`, com codificação ISO/IEC 8859-1:1998 - Latin 1 (International Organization for Standardization, 1998), com cada documento constituindo um arquivo.

É sabido que esta base de dados apresenta um elevado grau de dificuldade em muitos de seus documentos para os testes de classificação automática. Isto se dá por diversos motivos, propositalmente estabelecidos quando da coleta dos documentos, que são melhor descritos a seguir.

Primeiramente, tratam-se de documentos jornalísticos que, apesar de possuírem um tema principal conforme a seção do jornal de onde foram extraídos, não foram selecionados e excluídos da base caso tratassem de temas diversos. Por exemplo, se um documento do tema Esportes (localização na seção de esportes de um determinado jornal) estivesse tratando dos bastidores políticos da eleição de algum comitê, ainda assim seria mantido dentro do tema Esportes.

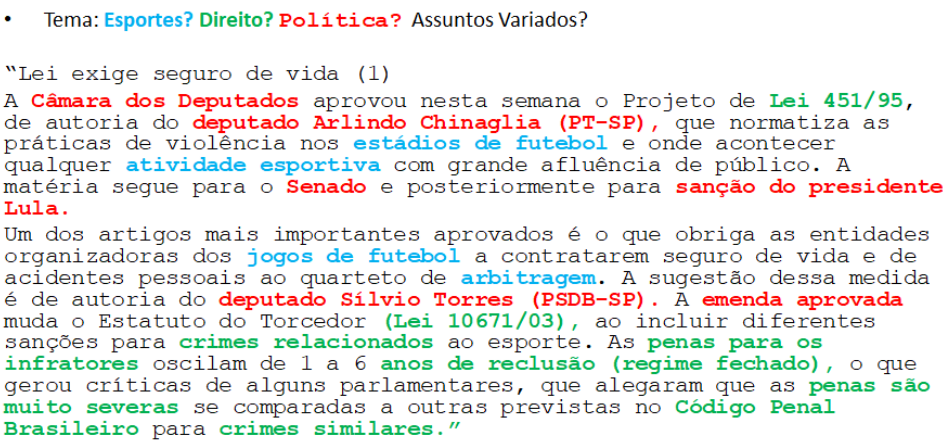
Em segundo lugar, cada tema possuir 10 autores diferentes, o que gera grande variação de vocabulário e estrutura sintática utilizada na escrita, por cada autor. Desta forma, a classificação pode apresentar resultados inadequados se considerar vocábulos que não sejam suficientes para delimitar o tema.

Há também a dificuldade de cada documento pertencer apenas a uma classe. Como pode ser visto no exemplo da figura 21, um documento do tema Esportes, mas destacando os vocábulos que poderiam fazer este mesmo documento ser classificado como Direito, Política ou Assuntos Variados.

Também há dificuldade na classificação por haver um tema denominado "Assuntos Gerais", composto por documentos de autores que escrevem em temas variados, sem estarem restritos a uma seção específica de um jornal. Estes documentos trazem tanto a variação e confusão com os outros temas, como o fato de que um mesmo autor pode estar tratando de temas diferentes a cada documento.

Por fim, existe também a dificuldade de alguns temas serem bastante próximos entre si, sendo que em trabalhos anteriores (por exemplo, em (OLIVEIRA Jr., 2011)) era comum a confusão entre temas Literatura e Direito, Gastronomia e Saúde, Esportes e

Figura 21 – Exemplo de documento em Português com possíveis multiplos temas



Fonte: o autor

Saúde, Assuntos Variados e Política.

3.1.2 Base de dados NG

A segunda base de dados, em idioma inglês, é conhecida por *20 Newsgroups*[1] e foi criada por (LANG, 1995). Os documentos em inglês estão armazenados em arquivos simples de texto, com codificação UFT-8, com cada documento constituindo um arquivo. Esta base é bastante utilizada em tarefas de NLP e é composta por 19.997 documentos, distribuídos em 20 classes, cada classe com aproximadamente 1.000 documentos.

As classes que a compõe são:

- comp.graphics
- comp.os.ms-windows.misc
- comp.windows.x
- comp.sys.mac.hardware
- comp.sys.ibm.pc.hardware
- misc.forsale
- rec.autos
- rec.motorcycles
- rec.sport.baseball
- rec.sport.hockey
- talk.politics.misc
- talk.politics.guns

[1] Disponível em https://archive.ics.uci.edu/ml/machine-learning-databases/20newsgroups-mld/

Figura 22 – Exemplo de documento em Inglês

```
Benedikt Rosenau writes, with great authority:>
IF IT IS CONTRADICTORY IT CANNOT
EXIST."Contradictory" is a property of language.  If
I correct this to       THINGS DEFINED BY
CONTRADICTORY LANGUAGE DO NOT EXISTI will object to
definitions as reality.  If you then amend it to
THINGS DESCRIBED BY CONTRADICTORY LANGUAGE DO NOT
EXISTthen we've come to something which is plainly
false.  Failures indescription are merely failures
in description.(I'm not an objectivist, remember.)--
C. Wingate        + "The peace of God, it is no
peace,                  +    but strife closed in
the sod.mangoe@cs.umd.edu +  Yet, brothers, pray for
but one thing:tove!mangoe       +    the marv'lous
peace of God."
```

Fonte: o autor

- talk.politics.mideast
- sci.crypt
- sci.electronics
- sci.med
- sci.space
- talk.religion.misc
- alt.atheism
- soc.religion.christian

Como pode ser observado, algumas classes são bem semelhantes, por exemplo *comp.sys.mac.hardware* e *comp.sys.ibm.pc.hardware*. Os documentos possuem o tamanho médio de 2,3kB, com a quantidade média de 307 palavras e 2360 caracteres por documento.

A base foi utilizada como originalmente fornecida, não sendo realizado nenhum tratamento sobre o conteúdo de seus arquivos, seguindo-se o mesmo uso feito por outros autores em relação a esta base. Esta base foi denominada $NGfull$ no restante deste trabalho. Um exemplo de trecho de um documento desta base pode ser visto na figura 22.

A partir de seus documentos foram criadas outras 3 bases para os testes a serem realizados, tendo então um total de 4 bases em inglês.

A segunda base foi denominada de $NG20$ e é composta por 7.378 documentos divididos nas mesmas 20 classes. Desta forma puderam ser separados 5.000 documentos para treinamento, com 250 documentos de treinamento em cada classe, e aproximadamente 119 documentos para teste em cada classe. Desta forma a quantidade de

documentos para treinamento ficou semelhante à quantidade de documentos utilizados na base de dados *Port*10, mas com uma quantidade maior de documentos para testes.

A terceira base foi denominada de *NG*10 e é composta por 3.690 documentos: 2.500 documentos para treinamento e 1.190 documentos para testes. Foi reduzido, entrentanto, a quantidade de classes existentes, passando-se a apenas 10 classes: *alt.atheism, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, misc.forsale, rec.motorcycles, rec.sport.hockey, sci.electronics, sci.space, talk.politics.guns* e *talk.politics.misc*. Desta forma tem-se a mesma quantidade de classes que a base *Port*10, podendo ser verificado nos testes se a quantidade de classes possui influência sobre a categorização de documentos. Estas classes foram escolhidas selecionando-se toda segunda classe da lista, ou seja, da lista de classes existentes da base de dados, ordenada aleatoriamente, foram selecionadas as classes que ocupavam a 2ª, 4ª, 6ª, 8ª, 10ª, 12ª, 14ª, 16ª, 18ª e 20ª posições.

Por fim, foi criada uma quarta base, denominada de *NG*05, composta por 1.843 documentos: 1.250 documentos para treinamento e 593 documentos para testes, reduzindo-se a quantidade de classes disponíveis para apenas 5 classes: *soc.religion.christian, talk.religion.misc, talk.politics.mideast, talk.politics.misc* e *talk.politics.guns*. Estas classes ocupavam, respectivamente, a 4ª, 8ª, 12ª, 16ª e 20ª posições na lista de classes.

Observa-se que a atribuição de documentos à classes era feita pelos próprios usuários que mandavam mensagens ao *newsgroup*, ou seja, não há qualquer interferência ou verificação se o conteúdo corresponde à classe atribuída.

3.1.3 Separação de documentos

A separação de documentos consiste em definir quantos documentos serão utilizados para o treinamento, validação e testes.

Conforme mencionado anteriormente , os documentos de treinamento são utilizados para treinar o modelo com a otimização dos parâmetros da rede neural (pesos entre as conexões neuroniais), os documentos de validação são utilizados para estimar a taxa de erro do modelo de seleção que está sendo treinado e os documentos de testes servem para efetivamente testar o modelo com documentos ainda não vistos (FRIEDMAN; HASTIE; TIBSHIRANI, 2001).

Para a base de dados *Port*10 os documentos foram separados de forma a permi-

tir a realização de 30 testes, sem repetição dos documentos testados. Para isto, foram separados 2.900 para treinamento e 100 documentos para testes, sendo 10 documentos para cada classe testada. Esta configuração foi repetida 30 vezes, separando-se novos 100 documentos a cada rodada de testes, e o resultado obtido foi calculado a partir do valor médio das 30 repetições.

Dado o custo computacional de se repetir a quantidade de testes e o valor médio obtido não apresentar grandes variações, para as demais bases foi adotada uma nova divisão.

Para a base $NG05$, que possui a menor quantidade de documentos, foram separados aproximadamente 66% de documentos de treinamento 33% de documentos de testes. Desta forma, 1.250 documentos foram utilizados para o treinamento (125 documentos para cada classe) e 593 documentos para testes. Os testes foram repetidos 10 vezes, alterando-se quais documentos eram usados para treinamento e quais eram usados para testes. Alguns documentos participaram do grupo de teste mais de uma rodada.

Para as bases $NG10$ e $NG20$, a mesma proporção de documentos de treinamento e de testes foi mantida. Como para a base $NG10$, a quantidade de classes foi reduzida para 10, sendo retirados os documentos que pertenciam às demais classes. Desta forma a quantidade de documentos total foi reduzida aproximadamente à metade, com 2.500 documentos para treinamento (250 documentos para cada classe) e 1.190 documentos para testes (com 119 documentos para cada classe).

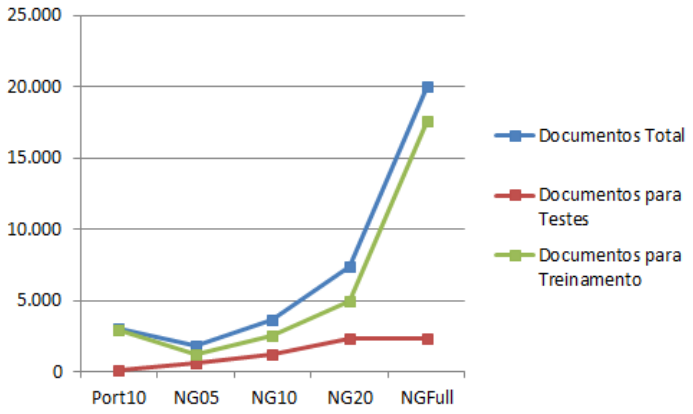
Para a base $NG20$ a quantidade de documentos de treinamento foi reduzida a 5.000, com 250 documentos de treinamento para cada classe, mas com todas as 20 classes originais. O número de documentos de teste foi mantido em 2.378. Para a base $NGfull$ foram separados os mesmos 2.378 documentos para testes, aproximadamente 119 documentos por classe. Para ser possível verificar se a quantidade de documentos disponíveis para treinamento influenciava o resultado, todos os demais documentos (17.619) foram utilizados para treinamento. E, da mesma maneira, foram repetidos por 10 vezes, com todos os documentos participando mais de uma vez do grupo de teste.

Tabela 3 – Bases de dados utilizadas - Resumo

Base de dados	Classes	Documentos totais	Documentos de treinamento e validação	Documentos de testes
*Port*10	10	3.000	2.900	100
NGfull	20	19.997	17.619	2.378
*NG*20	20	7.378	5.000	2.378
*NG*10	10	3.690	2.500	1.190
*NG*05	5	1.843	1.250	593

Fonte: o autor.

Figura 23 – Comparativo das bases de dados



Fonte: o autor

3.1.4 Resumo das bases de dados

As bases de dados utilizadas neste trabalho estão em idiomas português e inglês. Para a realização dos testes a base de dados em inglês foi subdividida em 4 bases, com características próprias, conforme mencionado anteriormente.

A tabela 3 resume as principais informações destas bases de dados.

A figura 23 mostra, graficamente, a comparação entre as bases de dados, ilustrando a quantidade de documentos totais, documentos de teste e de documentos de treinamento.

3.2 Processamento de Linguagem Natural

Conforme mencionado na seção 2.4 - Extração de características, é necessário extrair as características desejáveis dos documentos para que estes possam ser submetidos

à categorização, no caso deste trabalho, com o uso de redes neurais.

Além da possibilidade de se desenvolver ferramentas para o processamento de linguagem natural, existem diversas soluções prontas, de código aberto ou fechado, que se propõe a realizar as mais diversas atividades. Entre as diversas alternativas, destaca-se a linguagem de programação Python[2] e suas inúmeras bibliotecas. Python é uma linguagem interpretada, de alto nível, que suporta diversos paradigmas de programação (procedural, imperativa, funcional ou orientação a objeto), bastante utilizada para o processamento de linguagem natural. Entre outras bibliotecas, duas são extremamente úteis para NLP:

- NLTK[3] (*Natural Language Toolkit*) se destaca por ser gratuíta, de código aberto e fornecer um conjunto de ferramentas para se trabalhar com linguagem natural: classificação, *tokenization*, rotulagem, análise etc (BIRD; KLEIN; LOPER, 2009);
- Scikit-learn[4], por sua vez, fornece diversas ferramentas para *data mining* e análise de dados, também sendo de código aberto (PEDREGOSA, 2011).

Para a extração de características foi considerada apenas uma análise morfológica do conteúdo dos documentos. Entre as técnicas existentes, optou-se por duas: TF-IDF e Doc2Vec, conforme mencionado anteriormente. Com esta extração de características foi possível transformar as informações textuais em informações numéricas, na forma de vetores, que puderam alimentar as redes neurais para a tarefa de classificação.

3.2.1 TF-IDF

Como explicado na seção 2.4.2 - TF-IDF, a TF-IDF busca representar o quanto um termo está presente em um documento e o quanto é raro nos demais documentos, indicando assim a sua relevância apenas para o documento atual.

Para a geração de vetores de TF-IDF foi utilizada uma função da biblioteca Scikit-Learn[5]. Para retirar, quando necessário, as palavras mais comuns que possuem funções sintáticas mas pouco conteúdo agregado, foram utilizadas as palavras constantes na

[2] Disponível em https://www.python.org
[3] Disponível em http://www.nltk.org
[4] Disponível em http://scikit-learn.org/stable/
[5] Disponível em http://scikit-learn.org/stable/

lista de *stopwords* da biblioteca NLTK[6], em português e em inglês. Por exemplo, entre estas palavras, podemos citar: *a, ao, aos, aquela, aquilo* para o idioma português e *a, the, on* para o idioma inglês.

Conforme menciona (YU, 2008), sabe-se que *stopwords* são palavras extremamente comuns e que em geral não carregam significado próprio, sendo muitas vezes consideradas palavras funcionais, tais como pronomes e artigos. Entretanto, em algumas tarefas, o seu poder discriminante não pode ser desconsiderado, pois podem ser marcadores estilísticos em tarefas onde o estilo de escrita é significativo. Por exemplo, em tarefas de análise de gênero, de estilística ou mesmo na atribuição de autoria. Por outro lado, há um custo de dimensionalidade quando as palavras mais comuns não são removidas, desperdiçando-se capacidade computacional e, principalmente, gerando ruído entre as características a serem classificadas. Desta forma, a retirada ou a manutenção de *stopwords* dos testes realizados foi feita em cada caso, conforme o objeto do teste.

As palavras foram todas transformadas em minúsculas, supondo-se que não há relevância para a categorização de documentos se uma determinada palavra aparece no ínicio de uma frase (com a primeira letra em maiúscula), no meio da frase (com todas as letras minúsculas), se foi grafada como nome próprio (tendo então a primeira letra maiúscula mesmo no meio de uma frase) ou se foi destacada (sendo grafada toda em maiúscula).

3.2.2 Doc2Vec

Para a geração de vetores Doc2Vec foi utilizada a biblioteca Gensim[7] (REHUREK; SOJKA, 2010). Assim como feito com TF-IDF (seção 3.2.1), em alguns casos foram removidas *stopwords* e o conteúdo dos documentos foi transformado para letras minúsculas.

Na geração de vetores Doc2Vec existem alguns parâmetros que podem ser otimizados:

- *min_count*: desconsidera as palavras que estejam presentes em um número menor

[6] Disponível em http://www.nltk.org
[7] Disponível em https://radimrehurek.com/gensim/index.html

de vezes que a estabelecida neste parâmetro. Nos testes foi estabelecido para considerar todas as palavras, mesmo que presentes apenas uma vez;

- *window*: a distância máxima entre a palavra considerada e a palavra a ser prevista;
- *size:* dimensionalidade dos vetores,
- *iter:* o número de iterações que deverão ser utilizadas para a geração dos vetores.

Estes parâmetros foram objetos de alguns testes, conforme será detalhado.

3.3 Treinamento

Após a extração de características, os documentos são submetidos a treinamento. Isto é feito por meio de redes neurais, conforme detalhamento a seguir.

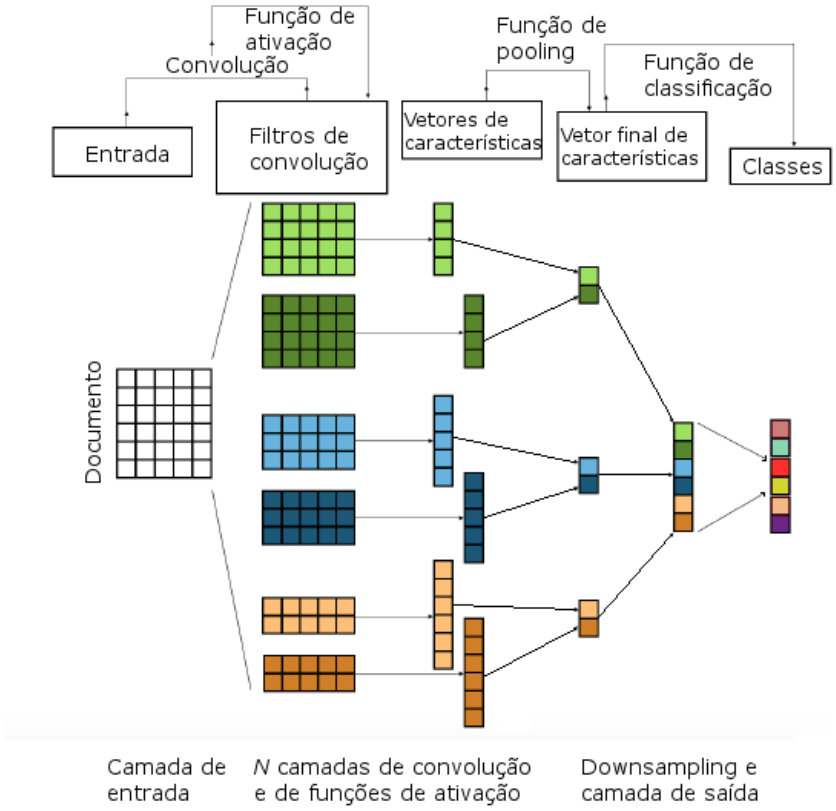
Para o uso de redes FCNN, são criadas no mínimo 3 camadas (entrada, camada escondida e saída). Estas camadas possuem alguns parâmetros, sendo o mais relevante a quantidade de neurônios e as funções dea ativação. Conforme indicação de literatura, foram utilizadas principalmente as funções de retificação linear (ReLU) e hiperbólica tangente, por apresentarem resultados satisfatórios com menor custo computacional.

O uso de rede CNN, por sua vez, exige que diversos parâmetros sejam configurados: dimensão da camada de entrada, quantidade de camadas escondidas, quantidade de camadas de convolução e seus parâmetros, camadas de ativação e seus parâmetros, camadas de *downsamplig* e camadas de conexão para a obtenção de classificação (ou seja, uma camada de saída). Não há uma regra sobre a quantidade de camadas, sua intercalação ou seus parâmetros, devendo ser configuradas conforme a aplicação desejada e os resultados obtidos.

Conforme mencionam (ZHANG; WALLACE, 2017), um ponto negativo das redes CNN é justamente a necessidade de se especificar o modelo exato da arquitetura e todos seus hiperparâmetros. Explorar todas as combinações possíveis é próximo ao impossível, seja pelos recursos computacionais exigidos, pelo tempo necessário e pela explosão combinacional de configurações possíveis.

A camada de entrada é a que receberá os vetores de características, portanto sua dimensão deve ser igual à dimensionalidade dos vetores. Se esta quantidade for

Figura 24 – Arquitetura de uma rede CNN para categorização de documentos



Fonte: (LOPEZ; KALITA, 2017) com alterações.

inferior, informações dos vetores serão perdidas. Se for superior, existirão neurônios que não serão alimentados, resultando então em desperdício de processamento (se os seus valores forem atribuídos como 0) e/ou ruído (se os seus valores forem determinados aleatoriamente no início, e por falta de informação nos vetores seus valores permanecerem aleatórios até serem eventualmente corrigidos com a retropropagação).

As camadas seguintes são as camadas escondidas. Em geral são utilizados grupos formados com camadas de convolução/funções de ativação para a redução da dimensionalidade, extraindo-se informações mais densas em forma de vetores. A seguir são aplicadas as camadas de *downsampling* e camada de saída, sintetizando os vetores e fazendo-se a categorização. A figura 24 ilustra um exemplo de arquitetura possível.

A cada execução, a rede CNN verifica o resultado obtido e realiza a retroprogração, ajustando os pesos de cada neurônio visando sua otimização. Os documentos separados para validação são então aplicados, verificando-se se houve *overfitting* ou se a rede consegue generalizar o suficiente. Ao final, obtém-se uma rede CNN treinada.

3.4 Considerações do Capítulo

As bases de dados são fundamentais para a tarefa de categorização de documentos. Tanto a fase de treinamento como a fase de testes são realizadas considerando-se as categorias pré-definidas para os documentos. Conforme menciona (SELIVANOVA; RYABKO; GUSKOV, 2017), a verificação da acurácia depende da qualidade dos dados originais, pois erros de categorização poderão levar a treinamentos imperfeitos ou a resultados imprecisos dos testes.

Nas bases de teste utilizadas neste trabalho garantiu-se que todas as categorias propostas possuam a mesma quantidade de documentos representativos, evitando-se que os resultados apresentem distorções. Por exemplo, se uma base de testes possuir 20% de seus documentos em uma classe X e 80% de seus documentos em uma classe Y, o simples fato de se atribuir todos os documentos de testes à classe Y fará com que exista uma taxa de acerto de 80%, puramente pelo desequilíbrio da representatividade dos documentos, e não por méritos do classificador.

E, por suas características, a base de teste $NGfull$ pode ser subdividia em mais três bases, reduzindo-se a quantidade de documentos para treinamento ($NG20$) e também reduzindo-se a quantidade de classes possíveis ($NG10$ e $NG05$), podendo ser verificado o desempenho da proposta em bases com características diferentes.

A proximidade da época da coleta dos documentos também é interessante, evitando-se que diferenças sejam estabelecidas não pela categoria dos documentos, e sim pela época em que foram produzidos. Sabe-se que qualquer idioma é vivo, ou seja, representa o estágio linguístico falado por pessoas em um determinado momento e sofre variações constantes com a introdução e supressão de vocábulos, bem como o aumento e decréscimo de popularidade de algumas palavras. Ou, em alguns casos, pela variação de significação, o que faz com que o mesmo vocábulo passe a representar idéias ou conceitos diferentes.

Por fim, foi comentado sobre a extração de características utilizando-se duas abordagens distintas, para que o conteúdo dos documentos seja transformado de palavras em vetores de características que serão utilizados para a tarefa computacional de classificação.

4 Proposta

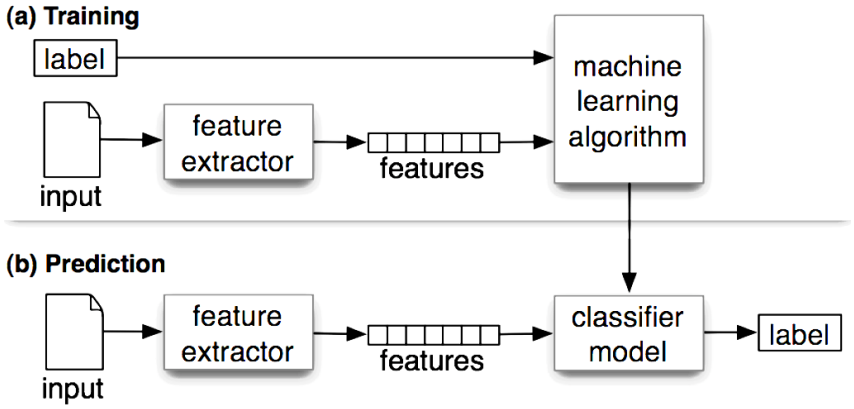
Este capitulo trata da proposta de trabalho utilizada para os problemas de categorização de documentos jornalísticos em Português e de notícias de grupos de discussão em Inglês. São detalhadas as etapas realizadas desde o tratamento inicial das amostras de documentos, a geração de vetores de informações, a construção das redes de treinamento e o processo de categorização.

4.1 Visão Geral

A categorização de documentos, em geral, segue os mesmos passos de um aprendizado por máquina supervisionado, mostrado na figura 25. Resumidamente, um conjunto de entradas e suas respectivas categorias são utilizadas para o treinamento. É feita a extração de características dos dados da entradas e estas características são aplicadas a uma ferramenta computacional de aprendizado que gera um modelo de classificação. Na fase de testes o objeto a ser testado tem suas características extraídas e é submetido à classificação, tendo-se por base o modelo de classificação obtido na fase de treinamento. Com isto, uma categoria é atribuída, e a verificação da correção desta atribuição determina a taxa de acerto do modelo.

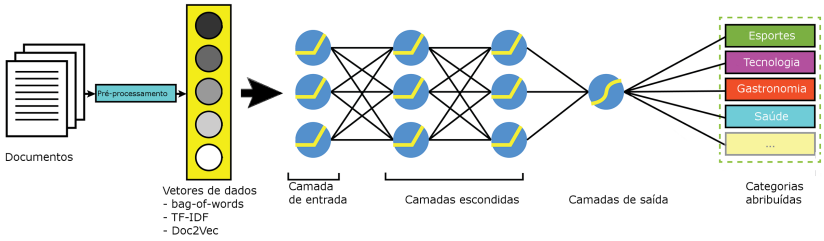
No presente trabalho este mesmo modelo pode ser aplicado, apenas devendo

Figura 25 – Classificação supervisionada



Fonte: http://www.nltk.org/images/supervised-classification.png, com alterações do autor

Figura 26 – Classificação de documentos com uso de rede FCNN conectadas

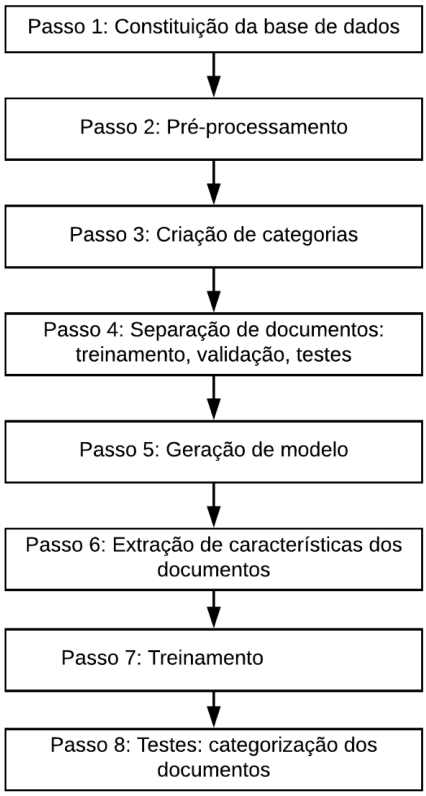


Fonte: http://blog.aylien.com com modificações do autor.

ser considerado que o aprendizado e a classificação são feitas por redes neurais. A figura 26 ilustra este processo de classificação, onde o treinamento e a categorização dos documentos é feita utilizando-se uma rede neural totalmente conectada.

O trabalho desenvolvido pode ser demonstrado em um fluxograma (figura 27), que é detalhado nas seções deste capítulo.

Figura 27 – Fluxograma de etapas de execução desta pesquisa



Fonte: o autor

4.2 Constituição da base de dados

Conforme detalhado nas seções 3.1.1 e 3.1.2, foram selecionadas bases de dados em dois idiomas para a realização dos testes propostos neste trabalho.

4.3 Pré-processamento

O pré-processamento dos dados, para a realização do trabalho, consistiu nas seguintes atividades. Primeiramente, a troca de todos os caracteres em suas versões minúsculas, preservando-se as acentuações existentes. A seguir, procedeu-se à separação das palavras para serem consideradas individualmente, em um processo de *tokenização*. Isto é feito escolhendo-se quais caracteres são delimitadores de palavras, permitindo que cada palavra seja considerada individualmante. Para este trabalho, esta segmentação foi feita considerando como separadores os espaços em branco e os sinais de pontuação (por exemplo, . , ; - ( ) " ") . Este processo é ilustrado na figura 28. Nota-se que o resultado é um conjunto de palavras, consideradas individualmente, e que os sinais utilizados como separadores deixam de fazer parte do texto.

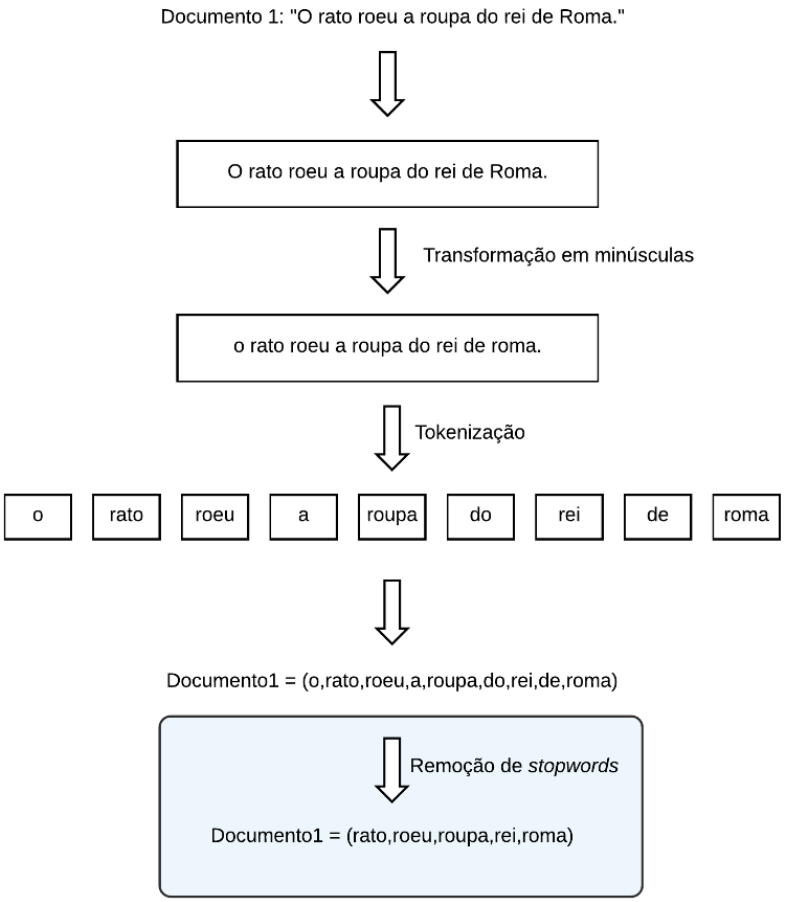
Por fim, ainda foi feita mais uma etapa de pré-processamento para alguns testes, consistindo na retirada de palavras funcionais mais comuns, conforme mencionado na seção 3.2.1, o que é destacado na figura 28 por meio de um sombreamento.

Como as bases de dados estão em dois idiomas distintos, para cada base de dados foi utilizada uma lista de *stopwords* diferentes, apropriada a cada idioma, obtidas a partir do pacote NLTK[1].

Na base de dados com documentos em Português foram retiradas, entre outras, as seguintes palavras: *de, a, o, que, e, do, da, em, um, para*. Nas bases de dados com documentos em Inglês foram retiradas, entre outras palavras, as seguintes: *i, me, my, myself, we, our, ours, ourselves, you*.

[1] Disponível em http://www.nltk.org

Figura 28 – Pré-processamento de documentos



fonte: o autor

4.4 Geração de modelo e extração de características

No presente trabalho foram utilizadas duas abordagens para a extração de características, conforme mencionado anteriormente: TF-IDF e Doc2Vec.

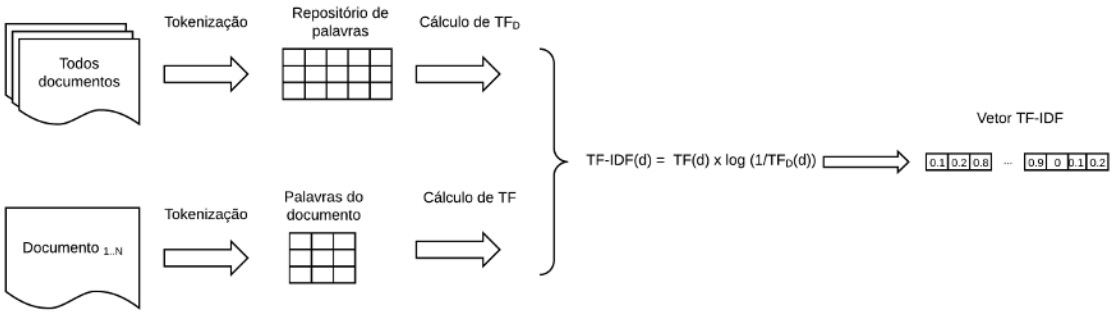
Primeiramente são gerados os modelos que correspondem a cada uma destas abordagens, e em seguida estes modelos são aplicados aos documentos a serem treinados/testados. Este procedimento é detalhado a seguir para cada uma das abordagens propostas.

4.4.1 TF-IDF

Conforme mencionado nas seções 2.4.2 e 3.2.1, a extração de características utilizando-se de TF-IDF é bastante semelhante à criação de vetores de *bag-of-words*.

Primeiramente, as palavras de todos os documentos são *tokenizadas*, criando-

Figura 29 – Geração de vetor TF-IDF



Fonte: o autor

se um repositório de palavras. Para a maioria dos testes foram então eliminadas as *stopwords*, restando apenas as palavras mais significativas. A seguir calcula-se a frequência com que cada palavra aparece em cada documento e a frequência com que aparece em todos os documentos, gerando-se assim um valor de TF-IDF para cada uma das palavras.

Foi escolhido que os vetores de características gerados teriam uma dimensionalidade 300, conforme indicado em resultados obtidos por (NEELAKANTAN et al., 2014). Para isto, os 300 melhores valores de TF-IDF foram escolhidos para a geração de vetores. Em alguns testes verificou-se a influência da dimensionalidade dos vetores, e neste caso outros valores foram utilizados.

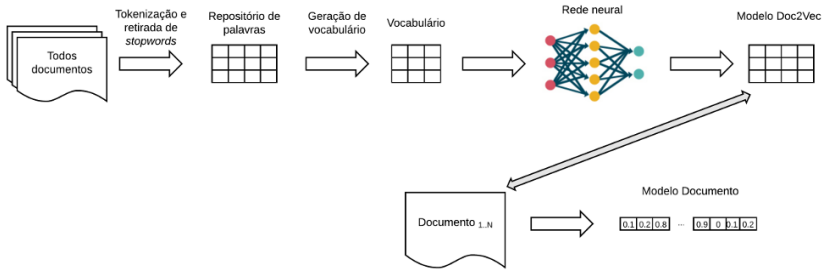
A figura 29 representa estas etapas. A equação apresentada é a mesma da seção 2.4.2, sendo *TF(d)* a frequência que a palavra *d* aparece em um documento e $TF_D(d)$ a frequência que a palavra *d* aparece no conjunto de documentos considerados.

4.4.2 Doc2Vec

Conforme mencionado nas seções 2.4.3 e 3.2.2, a extração de características utilizando-se de Doc2Vec faz uso de uma rede neural do tipo RNN. Para isto, são seguidos as seguintes etapas.

Primeiramente, todos os documentos são processados e então são eliminadas as *stopwords*, restando apenas as palavras mais significativas. A seguir é utilizada a biblioteca Gensim (REHUREK; SOJKA, 2010) para a construção de um vocabulário (a partir de uma sequência de frases), e este vocabulário é treinado em uma rede

Figura 30 – Geração de vetor Doc2Vec



Fonte: o autor

Figura 31 – Exemplo de vetor Doc2Vec

```
[ 0.71362358 -1.12171006 -1.26174808  0.03368234  0.86513954 -0.23822314
 -0.59973741  0.43381354  0.46723211 -0.03759978 -0.03090817 -0.65990639
  0.44740394  0.38632751 -0.64299488 -0.76582974  0.70809776  0.5341633
 -0.24220999  1.96894312 -1.0738765   0.53481847  1.21064591 -0.19786125
 -0.23021105 -0.98338068 -0.4087083  -1.53231645 -0.59389532 -0.82156634
 -0.90052021 -0.20107487  0.70761055  0.04062408 -1.92294586  0.14372818
  0.07599016  0.10100252  0.23549418 -0.37859419 -0.96258956  0.11801355
  0.57515281 -0.81004196  1.43082905  1.10528874 -1.81704438 -1.07322013
  2.33850288  1.47429836  0.03617498  0.2772817  -0.90406901 -0.47681484
 -1.20219553  1.56235576 -1.16914308  0.07812656 -0.50174516 -1.10441196
 -0.16680019 -2.44420886 -0.55436939 -0.51729727  1.31880331 -0.3882584
  0.49777809 -1.2955296   0.79085428  1.71913671  0.44602805  0.7781868
  0.6923762   0.51886469  0.18433619 -0.68494338 -1.30090511  1.8505969
 -0.97047472 -0.1628852   0.08385519 -1.37567282  0.11794516  1.42104471
 -1.06521463  0.04866452  0.2210784   0.15269053  0.75906527  0.03948998
 -1.71322286 -2.31707668 -0.38312119 -1.94798696 -0.46897316  1.16689515
  1.01112998 -0.30907783 -0.16540338 -1.07148397]
```

Fonte: o autor

neural RNN, gerando um modelo de vetores do tipo Doc2Vec que, como explicado anteriormente, guarda a relação das palavras entre si e também o contexto (no caso, documento) onde as palavras estiveram presentes. A dimensionalidade destes vetores é arbitrada, sendo que neste trabalho utilizou-se uma dimensionalidade 300, conforme recomendações de parâmetros feita por (LAU; BALDWIN, 2016), para a maioria dos testes.

Em seguida o modelo é aplicado para cada documento, gerando assim os vetores Doc2Vec específicos de cada documento.

A figura 30 representa estas etapas.

Um exemplo de vetor Doc2Vec gerado é mostrado na figura 31.

4.5 Treinamento

O treinamento é realizado separando-se a quantidade de documentos desejada para treinamento e para testes. A seguir, os documentos de treinamento são pré-processados, como mencionado anteriormente, retirando-se as palavras mais comuns.

A seguir é feita a extração de características destes documentos, gerando-se vetores que representam as características TF-IDF ou Doc2Vec.

Estes vetores são utilizados para o treinamento da rede neural com os parâmetros de treinamento da rede:

- *topologia*: a rede é configurada com a topologia desejada, especificando-se a quantidade de camadas, quantidade de neurônios, funções de ativação, existência ou não de *dropout* e seu valor;
- *função de perda e otimizador*: configura-se qual o parâmetro de perda que será otimizado pelo treinamento da rede (por exemplo, se o objetivo é minimizar as perdas na acurácia) e qual método de otimização será adotado;
- *repetições*: configura-se, por fim, como será realizado o treinamento: quantos documentos serão utilizados por vez para o treinamento, quantas vezes cada um destes lotes de treinamento serão utilizados, quantos documentos serão utilizados para treinamento e para validação, se os documentos de treinamento e validação serão alterados a cada execução ou se permanecerão os mesmos.

Com o treinamento a rede neural verifica quais as classificações são obtidas após a aplicação dos documentos de treinamento, e é feita a correção dos parâmetros da rede (por ex., peso das conexões) pela retropropagação. Em seguida são aplicados os documentos de validação, verificando quais são os resultados obtidos pela rede neural. São também configurados os valores de *early stop*, ou seja, se os resultados obtidos com os documentos de validação são satisfatórios, encerra-se o treinamento da rede neural sem executar todas as repetições previstas anteriormente.

4.5.1 Funções de ativação

Dentre as funções de ativação mencionadas na 2.3.1, foram realizados testes iniciais com as funções Piece-wise, ReLU e SoftMax. Foi verificado que os resultados eram ligeiramente inferiores com a função Piece-wise e bastante semelhantes com as funções ReLU e SoftMax, com uma diferença negligenciável no tempo de processamento. Desta forma, optou-se pela utilização da função SoftMax na primeira camada das redes neurais, e utilizando-se a função ReLU em todas as demais camadas.

4.6 Testes

Após a conclusão de treinamento, passa-se à fase de testes. Nesta fase o modelo de rede treinada é utilizada para categorizar os documentos a serem testados, obtendo-se então a taxa de acerto da categorização dos documentos.

Como mencionado na seção 3.1.3, para cada base de dados foi feita uma separação de documentos em treinamento e testes.

Na base $Port10$ foram separados 100 documentos para testes, repetindo-se os testes por até 30 vezes sem que os documentos de teste fossem repetidos.

Nas bases $NGfull$ e $NG20$ foram testados 2.378 documentos a cada rodada de testes. Na base $NG10$ foram testados 1.190 documentos e na base $NG05$ foram testados 593 documentos a cada rodada de testes. Para cada uma destas bases, os testes foram repetidos 10 vezes, obtendo-se a acurária pela média destas execuções.

4.7 Ambiente

Os treinamentos e testes foram executados em um equipamento com as seguintes características:

Item	Descrição
Processador	Intel Core i7-7700K, 4.2GHz
Memória	16 GB DDR4, posteriormente expandida para 32 GB
Discos	SDD OCZ-Vertex Plus
GPU	Intel HD Graphics
Processamento em GPU	não

Foi utilizada a linguagem de programação Python 3.6.4[2] com bibliotecas Pandas[3], Numpy[4], Keras[5], Scikit-Learn[6], NLTK[7] e TensorFlow[8]. Foram recompiladas todas as bibliotecas possíveis para otimização em relação ao equipamento disponível.

4.8 Considerações do Capítulo

Este capítulo complementa as observações feitas no capítulo 3, apresentando a abordagem proposta para a categorização de documentos jornalísticos em idioma português (base *Port*10) ou documentos de mensagens de grupos de discussão (bases $NGfull$, $NG20$, $NG10$ e $NG05$) em classes pré-definidas.

São detalhadas algumas características das redes neurais, destacando-se principalmente a quantidade de parâmetros passíveis de ajustes e a necessidade de testes de alguns destes parâmetros para verificação de seus impactos no desempenho da categorização.

O capítulo 5, a seguir, apresenta os resultados obtidos com esta proposta.

[2] Disponível em https://www.python.org
[3] Disponível em https://pandas.pydata.org
[4] Disponível em http://www.numpy.org
[5] Disponível em https://keras.io
[6] Disponível em http://scikit-learn.org/stable/
[7] Disponível em http://www.nltk.org
[8] Disponível em https://www.tensorflow.org

5 Resultados Experimentais e Discussão

Neste capítulo são apresentados os resultados obtidos na condução dos experimentos propostos em diversos cenários. São também apresentados resultados comparativos obtidos com a mesma base de dados e outras abordagens, servindo como referência do desempenho da proposta deste trabalho. Juntamente a cada cenário são apresentadas discussões e considerações, havendo uma seção final para a conclusão dos resultados obtidos.

5.1 Resultados anteriores

Inicialmente, apresenta-se resultados anteriores obtidos para as bases de dados, permitindo-se desta forma a comparação de desempenho. Sabe-se que não apenas o desempenho de atribuições corretas é relevante mas também considerações que permitam evitar pontos fracos de metodologias anteriores que tenham trabalhado com os mesmos dados. Desta forma, é estabelecido uma referência mínima para verificar se a metodologia proposta possui desempenho semelhante para a tarefa.

Em 2011, (OLIVEIRA Jr., 2011) utilizou a mesma base de dados *Port*10 (VARELA, 2010) para o teste de autoria de documentos, tendo sido colocado também o resultado de atribuição de temas aos documentos. Naquele trabalho, a classificação era feita utilizando-se compressores de dados como classificadores em uma abordagem denominada *Normalized Compression Distance* (CILIBRASI; VITÁNYI, 2005).

Em relação aos temas de documentos, a mesma abordagem demonstrou que era possível obter resultados para a base *Port*10, conforme mostrado na tabela 4, na coluna "Taxa de acerto". Estes resultados são os melhores obtidos para cada uma das classes e a taxa média de acerto seria de 79,61% para a categorização correta de temas aos documentos questionados.

Em relação à base $NG20full$, foram considerados os resultados anteriores obtidos por (WITTLINGER; SPANAKIS; WEISS, 2015). Naquele trabalho, os autores testaram diferentes técnicas de *dropout* e obtiveram, como melhor resultado, a taxa de

Tabela 4 – Categorização de documentos com NCD

Tema	Taxa de acerto
Assuntos Variados	86,96 %
Direito	83,48 %
Economia	74,35 %
Esportes	96,96 %
Gastronomia	65,65 %
Literatura	57,39 %
Política	83,91 %
Saúde	81,74 %
Tecnologia	91,74 %
Turismo	73,91 %
Média	79,61 %

Fonte: o autor

acerto de 82,9%. Quando foram realizados testes em uma parte da base $NGfull$, composta de apenas 5.000 documentos, os autores relatam que os resultados não variaram significativamente, permanecendo entre 81% e 83%.

5.2 Extração de características com Doc2Vec

Inicialmente, foram realizados testes utilizando-se a extração de características com Doc2Vec (seções 2.4.3 e 3.2.2). Seguindo-se a Proposta (capítulo 4), inicialmente as bases de dados foram separadas em documentos de treinamento, validação e testes.

Para cada uma das 5 bases de dados foram extraídas as características Doc2Vec, a partir de seus próprios dados. Ou seja, os vetores Doc2Vec a serem utilizados para cada base consideraram apenas os seus documentos, sem que o modelo gerado com uma base fosse reutilizado para outra. Então, no total, para os primeiros testes foram gerados 5 modelos Doc2Vec, com tamanhos variáveis conforme a quantidade de documentos e vetores trabalhados.

O primeiro teste realizado utilizou os parâmetros de Doc2Vec expressos na tabela 5. Estes parâmetros são os normalmente utilizados na literatura.

Apenas para ilustração, o tamanho do arquivo contendo o modelo Doc2Vec gerado para cada uma das bases de teste é expresso na tabela 6.

Tabela 5 – Parâmetros Doc2Vec

Parâmetro	Valor	Observações
Algoritmo de treino	*Bag-of-words*	
Min count	5	Frequência minima de palavras
Windows	5	Distância máxima entre as palavras
Size	300	Dimensionalidade do vetor
Iter	100	Quantidade de iterações

Fonte: o autor

Tabela 6 – Tamanho de arquivos dos modelos Doc2Vec

Base de dados	Tamanho do Arquivo (MB)
Port10	74
NG05	30
NG10	41
NG20	71
NGFull	127

Fonte: o autor

Tabela 7 – Taxas de acerto dos modelos Doc2Vec e FCNN

Base de dados	Taxa de acerto média
Port10	82%
NG05	76,56%
NG10	84,79%
NG20	77,17%
NGFull	83,10%

Fonte: o autor

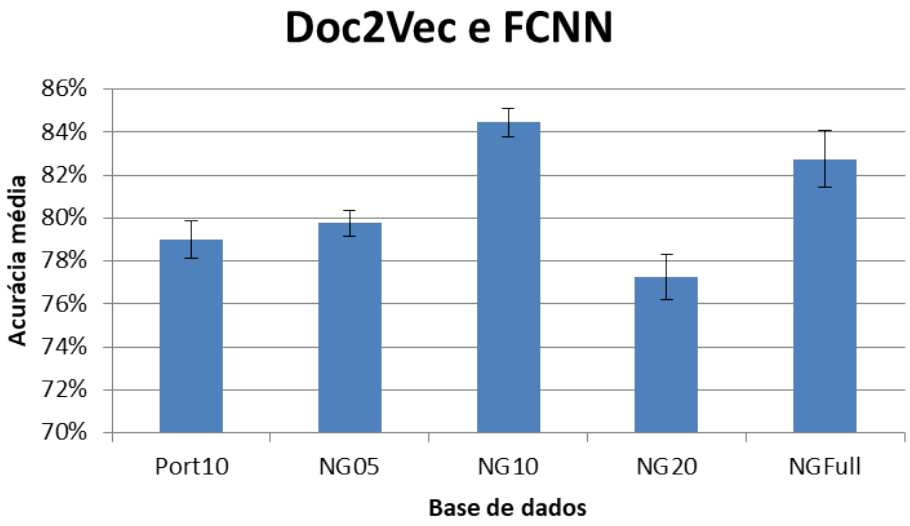
5.2.1 FCNN - Rede Neural Totalmente Conectada

Testes foram realizados considerando o modelo Doc2Vec e a classificação com redes FCNN. A taxa de acerto é mostrada na tabela 7.

A figura 32 mostra, graficamente, os resultados médios obtidos e o desvio padrão.

Para verificar o desempenho obtido, foi gerada a matriz de confusão de um dos testes da classe $NG05$, que é mostrada na figura 33. Nesta matriz é possível observar que muitos documentos foram atribuídos erroneamente à classe *soc.religion.christian*, com exatamente 148 documentos sendo atribuídos erroneamente a ela neste teste. Verificando-se o conteúdo dos arquivos desta base, entretanto, não foi possível visualizar

Figura 32 – Doc2Vec e FCNN: resultados



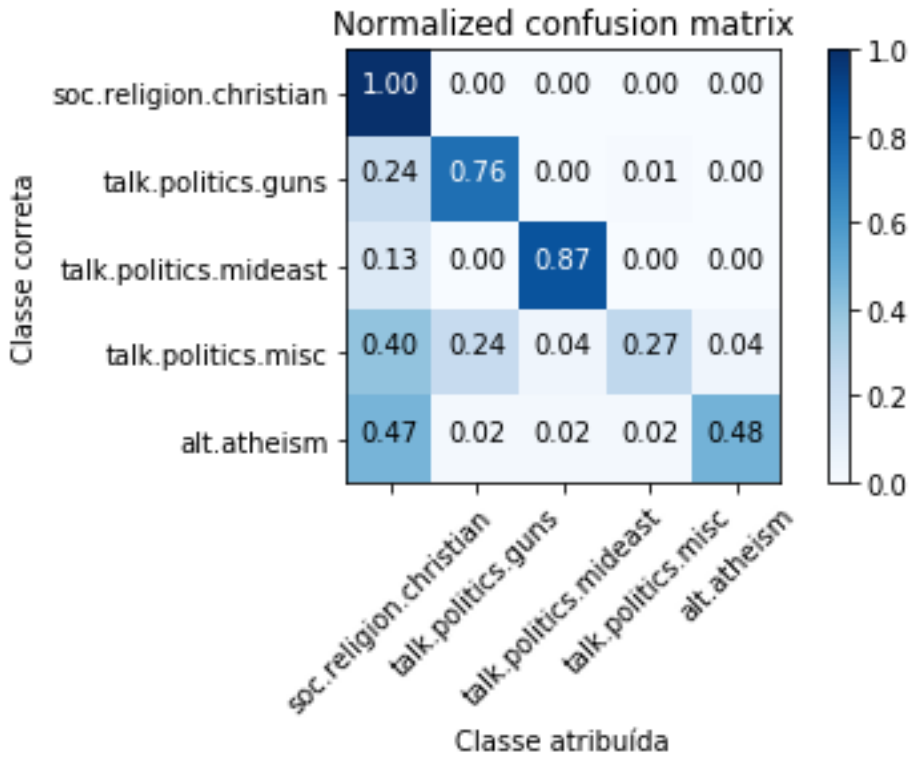
Fonte: o autor.

nenhuma situação que pudesse ter treinado erroneamente o Doc2Vec para que ele associasse termos de outros arquivos a palavras que pudessem ser encontrados neste documentos. A classe *talk.politics.misc* também teve muitos documentos confundidos com a classe *talk.politics.guns*, mas também não foi possível verificar o uso de palavras que pudessem levar a este equívoco.

A matriz de confusão da base *Port*10 é mostrada na figura 34. Como pode ser observado, a classe *Assuntos Variados* é a que teve o maior número de atribuições incorretas, com 40% de seus documentos indo para as classes *Economia*, *Literatura* e *Política*. A classe *Política* foi a que mais recebeu atribuições incorretas, superando a classe *Assuntos Variados*, que por ser a mais abrangente costuma ser a destinatária de atribuições errôneas em trabalhos anteriores. Chama a atenção a pouca confusão envolvendo as classes *Direito* e *Literatura*, e também as classes *Sade* e *Gastronomia*. No trabalho de (OLIVEIRA Jr., 2011) estas classes eram objeto de confusão por utilizarem vocabulário assemelhado entre si, por exemplo, ao envolver receitas de alimentação saudável.

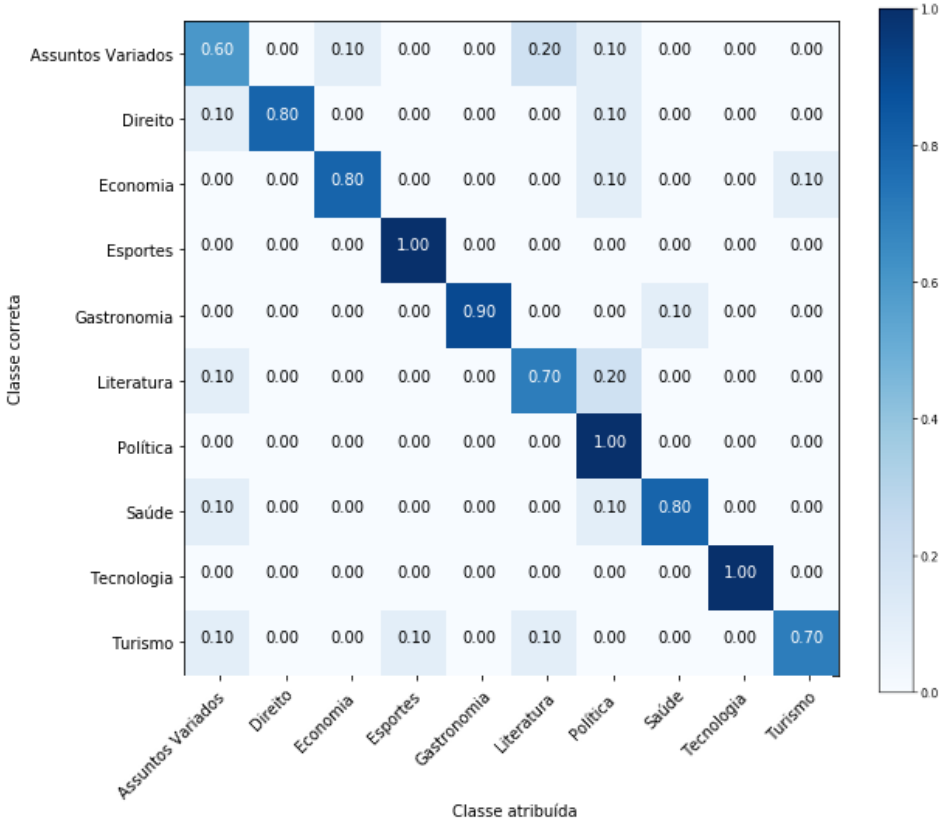
Para ter-se uma referência de desempenho, foi medido o tempo de execução dos testes em cada base, e o resultado é expresso na tabela 8. Como pode ser verificado, a execução do teste na base *Port*10 levou, em média, 493 segundos. A mesma abordagem utilizada por (OLIVEIRA Jr., 2011) foi novamente executada, no mesmo equipamento,

Figura 33 – Doc2Vec e FCNN: matriz de confusão da base NG05



Fonte: o autor.

Figura 34 – Doc2Vec e FCNN: matriz de confusão da base Port10



Fonte: o autor.

Tabela 8 – Tempo de execução: Doc2Vec e rede FCNN

Base de dados	Tempo de execução (segundos)
Port10	493
NG05	261
NG10	511
NG20	1.115
NGFull	5.059

Fonte: o autor

medindo-se o tempo gasto, e verificou-se que utilizar a Distância Normalizada de Compressão, mesmo em um equipamento mais moderno, levou aproximadamente 535 segundos. Isto é relevante porque na abordagem de (OLIVEIRA Jr., 2011) todos os arquivos são compactados em pares e há crescimento exponencial do tempo com o aumento da base de dados.

Do tempo total de execução dos testes, verificou-se que a geração do modelo Doc2Vec era responsável por, aproximadamente, 10% do tempo total do processamento, com um pequeno valor (inferior a 5%) sendo gasto para a inicialização do programa e leitura dos arquivos e a grande maioria (aproximadamente 85% do tempo) sendo consumido pelo processamento da rede FCNN.

Verifica-se que a base $Port10$ possui uma taxa de acerto de 82%. Este valor é superior ao obtido anteriormente por (OLIVEIRA Jr., 2011), com o teste t de Student apresenta um resultado $p > 0,05$ (aproximadamente 0,5456), mostrando que a hipótese nula de que a diferença entre $Port10$ e o trabalho de (OLIVEIRA Jr., 2011) se dever exclusivamente a fatores aleatórios pode ser rejeitada.

A base de dados $NG10$ é composta por aproximadamente a mesma quantidade de arquivos e a mesma quantidade de classes da base $Port10$, mas com documentos em Ingles. Verificou-se que a sua taxa de acerto foi de 84,79%, bastante próxima ao resultado obtido pela base $Port10$ em português. Realizando-se o teste t de Student para o resultado destas duas bases de dados, obteve-se um valor $p < 0,05$.

A base de dados $NG05$, por sua vez, obteve uma taxa de acerto de 76,56%. Este valor é inferior ao obtido na base $NG10$. A quantidade de classes, em geral, influencia no resultado, sendo esperado resultados melhores quando os documentos pertencem a um número menor de classes. Entretanto, conforme mencionado anteriormente, as

classes utilizadas para a base $NG05\%$ são bastante semelhantes entre si, o que gera bastante confusão nos resultados.

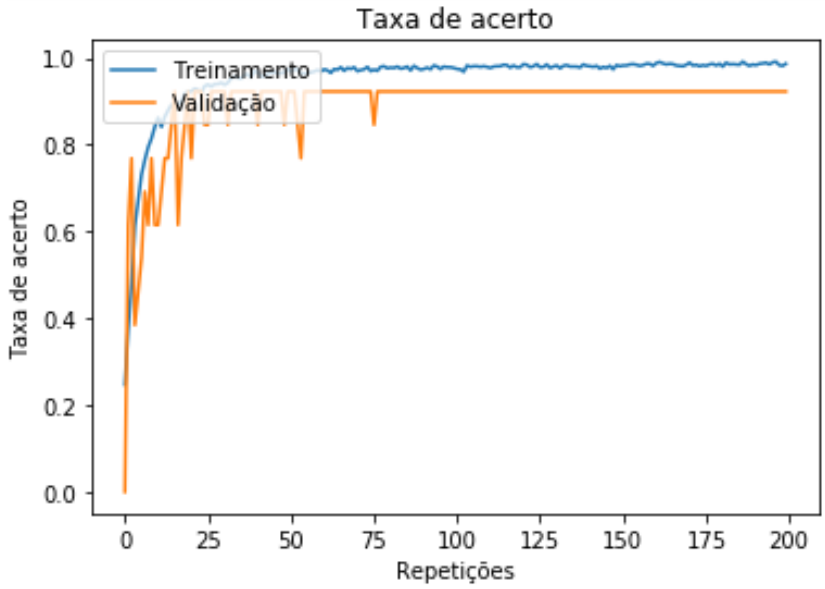
A taxa de acerto na base $NG20$, por sua vez, foi de 77,17%. Neste caso, a maior quantidade de classes influenciou o resultado, com a quantidade de documentos utilizados para treinamento tendo sido um fator importante para a taxa obtida. Isto é observado quando se compara este resultado com o obtido na base $NGFull$. Esta última base apresenta a mesma quantidade de classes mas uma maior quantidade de documentos para treinamento (19.997 contra 7.378), e sua taxa de acerto foi de 83,10%. Como todos os demais parâmetros de teste foram mantidos iguais, a única diferença existente é a quantidade de documentos. Realizando-se o teste t de Student para o resultado destas duas bases de dados, obteve-se um valor $p < 0,05$.

A taxa de acerto da base de dados $NG05$ é exibida na figura 35. Como pode ser observado, há uma relativa estabilidade na curva da taxa de acerto (tanto de treinamento como de validação). Isto permite que seja programado o encerramento do treinamento quando a taxa de acerto da validação deixa de apresentar ganhos (superiores a um valor configurável). Configurando-se os testes para a base de dados $NG05$ para este encerramento seja antecipado quando o ganho for inferior a 1% em um período de 20 ciclos, observou-se que a taxa de acerto do modelo teve variação positiva (de 76,56% em média para 79,76%), mas com significativa redução do tempo de processamento (261 segundos para 54 segundos). Esta técnica é denominada de *early stop*.

Para a base $NG10$,a aplicação de *early stop* com os mesmos parâmetros mencionados anterioremente levou a uma redução do tempo de processamento de aproximadamente 80% (de 511 para 95 segundos) com pequena alteração na taxa de acerto (de 84,79% para 84,45%). Na base $NG20$, o tempo de processamento foi reduzido em aproximadamente 60% (de 1.115 para 422 segundos) com também uma pequena variação na taxa de acerto (77,17% para 77,24%). Na base $NGFull$ a redução foi de aproximadamente 75% (5.059 para 1.308 segundos) com variação da taxa de acerto de 83,10% para 82,76%. Por fim, na base $Port10$ a redução foi de aproximadamente 70% (493 para 154 segundos) com variação da taxa de acerto de 82% para 79%.

Como houve redução no tempo total de processamento computacional com a adoção de *early stop*, com pequenas alterações nos valores obtidos na taxa de acerto,

Figura 35 – Doc2Vec e FCNN: curva de treinamento e validação da base NG05



Fonte: o autor.

Tabela 9 – Dimensionalidade dos vetores: base *NG*05

Dimensionalidade	Taxa de acerto
100	77,40%
200	77,74%
300	79,76%
400	80,10%
500	78,41%

Fonte: o autor

decidiu-se que todos os demais testes utilizariam a condição de *early stop*.

A seguir verificou-se a influência dos parâmetros Doc2Vec nos resultados obtidos. Primeiramente, foi verificado o impacto da alteração da dimensionalidade do vetor gerado pelo modelo na base *NG*05. Os resultados obtidos são mostrados na tabela 9.

O tempo de execução dos testes permaneceu aproximadamente o mesmo, com pequenas variações lineares proporcionais à quantidade de vetores. Isto era esperado, pois como verificado anteriormente, o fator dominante no tempo de execução é o processamento da classificação com a rede neural.

Observa-se que houve pouca variação nos resultados em função da dimensionalidade dos vetores. Do valor original utilizado (300), houve pequeno ganho ao se utilizar uma dimensionalidade 400, e a perda de pouco mais de 2 pontos percentuais ao

Tabela 10 – Dimensionalidade dos vetores: base $NG20$

Dimensionalidade	Taxa de acerto
100	74,64%
200	74,89%
300	77,24%
400	77,25%
500	78,47%

Fonte: o autor

se reduzir a dimensionalidade para 100. Realizou-se o teste de *Anova de fator único* em relação à alteração da dimensionalidade dos vetores, obtendo-se que o valor $F = 15,51$, superior ao valor $F_{critico} = 2,86$. Ou seja, verifica-se que a média das amostras é efetivamente diferente, sendo rejeitada a hipótese que as médias seriam equivalentes com diferenças possíveis apenas por fatores aleatórios.

Para verificar se a taxa de acerto obtida com dimensionalidades diversas é influenciada pela composição da base de dados, o mesmo teste foi realizado com a base $NG20$. Os valores obtidos são expressos na tabela 10. Como pode ser observado, também houve pequena variação nos resultados. Para comprovar se a dimensionalidade era significativa, realizou-se o teste de *Anova de fator único* obtendo-se o valor $F = 26,37$, superior ao valor $F_{critico} = 2,86$. Ou seja, verifica-se que a média das amostras é efetivamente diferente, sendo rejeitada a hipótese que as médias seriam equivalentes com diferenças possíveis apenas por fatores aleatórios. Foram realizados então testes t de Student, obtendo-se valores $p < 0,05$ entre as dimensionalidades 300 e 500. Desta forma, optou-se por manter a dimensionalidade 300 do vetor Doc2Vec para os demais testes realizados.

Outro parâmetro que foi testado foi a frequência mínima de palavras (parâmetro *min_count*). Desta forma, as palavras mais infrequentes nos documentos são desconsideradas. Testou-se a base $NG05$ e o resultado obtido é mostrado na tabela 11.

Como pode ser observado, quando mais palavras são consideradas (com a linha de corte sendo a frequência mínima de aparição das palavras em todos os documentos igual a 10 vezes ou menos), os resultados ficam bastante próximos. Quando a frequência torna-se mais elevada (sendo consideradas apenas as palavras que aparecem no mínimo 25 a 200 vezes nos documentos), a taxa de acerto começa a sofrer perdas. Considerando

Tabela 11 – Frequência mínima de palavras: base *NG*05

Frequência mínima	Taxa de acerto
1	77,74%
3	80,22%
5	79,76%
7	78,08%
10	78,41%
25	74,70%
50	75,38%
100	71,16%
200	63,91%
368	58,52%

Fonte: o autor

que esta base é formada por 1.843 documentos e 5 classes (aproximadamente 368 documentos por classe), verifica-se que as palavras que apareçam no mínimo 200 vezes em 1.843 documentos (ou seja, palavras mais comuns, que aparecem no máximo 1 vez a cada 9 documentos de todas as classes, podendo entretando aparecerem mais de uma vez em um documento) são insuficientes para uma boa classificação. Quando se restringe a casos de palavras ainda mais comuns, que aparecem no mínimo 368 vezes nos documentos (ou seja, no máximo 1 vez a cada 5 documentos, que poderiam ser palavras comuns e utilizadas em todos os documentos de uma classe e não aparecerem em documentos de outra classe), a taxa de acerto diminui para 58,52%. Este resultado ainda é superior a uma classificação aleatória (pois ao se classificar aleatoriamente 1 documento em 5 classes possíveis, a taxa de acerto esperada seria de 20%), mas é bastante inferior ao obtido quando são permitidas palavras mais raras (palavras com no mínimo 10 aparições em todos os documentos).

Para verificar se a dimensionalidade do vetor impactava neste resultado, mais dois testes foram realizados. Com uma dimensionalidade 1.000, a taxa de acerto foi de 56,49%. Com uma dimensionalidade 2.000, a taxa de acerto foi de 55,82%. Ou seja, mesmo buscando-se capturar uma quantidade maior de palavras comuns nos vetores, a taxa de acerto permanece aproximadamente a mesma. Desta forma, verifica-se que o uso de palavras comuns é insuficiente para uma boa classificação, sendo necessário a captura de palavras mais raras para uma melhor classificação. Desta forma, foi mantido o valor de 5 para a frequência mínima das palavras nos documentos para os demais testes.

Tabela 12 – Distância entre palavras: base *NG*05

Distância entre palavras	Taxa de acerto
1	78,75%
3	77,07%
5	79,76%
7	79,26%
10	80,78%
20	79,93%

Fonte: o autor

Foi também testada a influência da distância entre as palavras para a elaboração dos vetores Doc2Vec. Como o modelo Doc2Vec busca estabelecer associação entre palavras, ou seja, gera vetores cujos valores representam a proximidade com que palavras aparecem no corpo dos documentos, ao se alterar a distância máxima permitida entre as palavras os vetores podem capturar sentidos diferentes.

Por exemplo, considerando-se a frase "O pato sabe nadar, voar e andar, mas tudo mal". Se for considerada uma distância = 2, a palavra "pato"será associada às palavras "o", "sabe"e "nadar". Com uma distância maior, também poderá ser capturada a relação entre a palavra "pato"e "voar"e "andar". Ao se utilizar uma distância muito grande, entretanto, associações poderão ser feitas incorretamente ao ultrapassar as palavras de uma frase e capturar a relação com palavras de outras frases que não fariam sentido. O método Doc2Vec minimiza isto ao atribuir um peso maior às palavras mais próximas e ao reforço dado pela frequência que as associações ocorrem (ou seja, se na base de documentos é comumente visto as palavras "pato"e "nadar", o vetor entre estas duas palavras é mais forte que entre as palavras "pato"e "gaita", que provavelmente aparecerá um número reduzido de vezes).

Para a base *NG*05, o teste da distância permitida entre as palavras produziu o resultado que é exibido na tabela 12.

Verifica-se que entre o valor utilizado originalmente (distância máxima = 5) e os demais testes realizados há pouca diferença, variando-se no máximo 3 pontos percentuais no pior caso e com um ganho de 1 ponto percentual no melhor caso. Desta forma, verifica-se que a classificação de documentos nesta base utiliza apenas as palavras mais próximas para a geração de vetores, sendo pouco significativas as palavras distantes. Houve pequena alteração no tempo de execução, com um aumento

Tabela 13 – Distância entre palavras: base $NG20$

Distância entre palavras	Taxa de acerto
3	78,30%
5	77,24%
10	76,96%

Fonte: o autor

máximo de 2 segundos. Considerando que a única alteração em relação à linha de base estabelecida com o *early stop* foi a distância entre as palavras, verifica-se que distância maiores entre as palavras leva a uma maior complexidade do processamento, como esperado.

Para verificar se a distância entre palavras é influenciada pelo tamanho da base de documentos utilizada, foram feitos testes na base $NG20$, com os valores que produziram alterações mais extremas na base $NG05$, produzindo-se os resultados mostrados na tabela 13.

O impacto verificado foi de no máximo 1 ponto percentual para mais ou para menos, com um aumento máximo no tempo de processamento de 5 segundos. O tempo de processamento é compatível com a quantidade de documentos existente nesta base (trata-se de base 4 vezes maior que a $NG05$), mas ainda pouco significativo com o tempo necessário para o processamento da rede neural.

Um outro teste de parâmetros foi realizado na base $NG05$. Foram alterados os parâmetros mencionados anteriormente para os valores que produziram melhor resultado nesta base. A quantidade de vetores foi alterada para 400, a frequência minima das palavras foi alterada para 3 e a distância máxima para as palavras foi alterada para 10. A taxa de acerto obtida foi de 79,26%, muito próxima ao resultado base de 79,76% obtido com o uso de *early stop*. Retirando-se o *early stop*, o resultado foi uma taxa de acerto de 78,75%.

Verifica-se, então, que para os testes realizados na base $NG05$, a alteração de parâmetros do método Doc2Vec para os valores "ótimos"obtidos em testes separados não resulta em alteração significativa na taxa de acerto obtida (havendo, inclusive, uma pequena perda). Alguns parâmetros também foram alterados e testados na base $NG20$, havendo pouca alteração nos resultados. Como alguns testes nas demais bases ($Port10$,

Tabela 14 – Quantidade de camadas: base *NG*05

Quantidade de camadas	Taxa de acerto
1	77,94%
2	79,76%
3	79,93%
4	76,67%
5	77,24%

Fonte: o autor

*NG*10 e *NGFull*) também não apresentavam alterações significativas, nem todos os testes de alteração de parâmetros foram realizados.

Por fim, o último teste realizado na base *NG*05 verificando-se a influência da quantidade de camadas escondidas na taxa de acerto. Em todos os testes eram utilizadas duas camadas escondidas, e o resultado de alterar a quantidade de camadas é mostrado na tabela 14.

Ou seja, houve pouca variação em função da quantidade de camadas escondidas existentes. O impacto no tempo de processamento, entretanto, foi mais significativo. Isto é coerente com o processamento necessário: se todos os neurônios das camadas encontram-se conectados, a adição de uma nova camada aumenta exponencialmente a quantidade de cálculos necessários tanto durante a aplicação dos valores de entrada como para a retropropagação.

5.2.2 CNN - Rede Neural Convolucional

Os vetores Doc2Vec também foram utilizados para a tarefa de classificação de documentos utilizando-se uma rede neural convolucional (CNN).

Inicialmente foi criada uma rede convolucional com os parâmetros estabelecidos na tabela 15.

Ao final da rede convolucional foi utilizada uma totalmente conectada, reduzindo as saídas das camadas anteriores para o número de classes possíveis de categorização. Os resultados obtidos são mostrados na tabela 16, juntamente com os resultados anteriores de Doc2Vec com rede totalmente conectada e *early stop*.

A figura 36 mostra, graficamente, os resultados médios obtidos e o desvio padrão. São também mostrados os resultados anteriores (Doc2Vec e rede FCNN), para

Tabela 15 – Rede convolucional

Camada	Filtros	Kernel
1	64	2
2	32	2
3	16	2
4	16	2

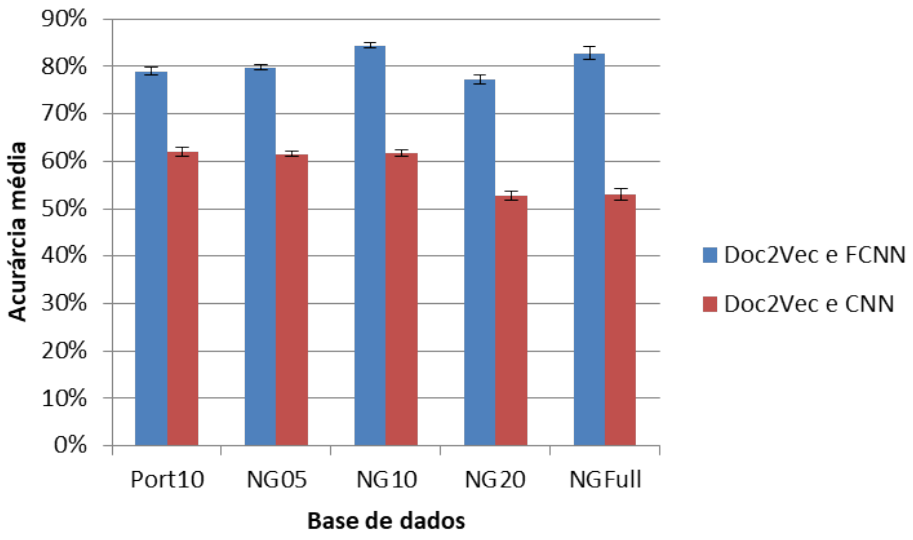
Fonte: o autor

Tabela 16 – Resultado : Doc2Vec e CNN

Base de dados	**Doc2Vec e CNN**	Doc2Vec e FCNN
Port10	62%	79%
NG05	61,55%	79,76%
NG10	61,76%	84,45%
NG20	52,73%	77,24%
NGFull	53,07%	82,76%

Fonte: o autor

Figura 36 – Doc2Vec e CNN: taxa de acerto

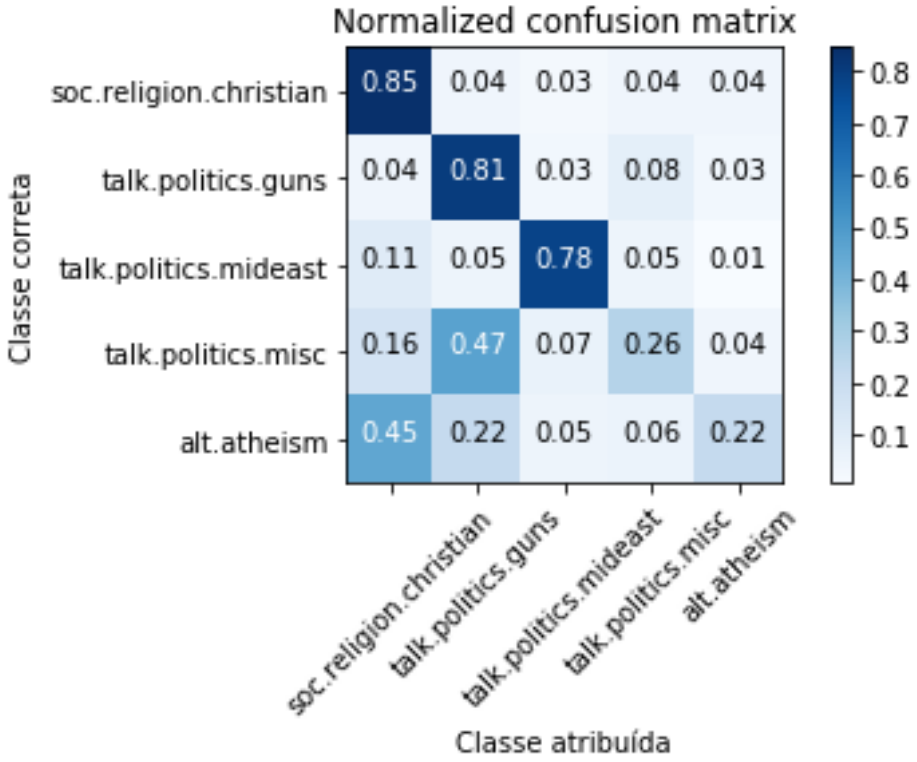


Fonte: o autor.

comparação.

Como pode ser observado, os resultados obtidos foram bastante inferiores aos obtidos quando a classificação era feita com uma rede FCNN, chegando a aproximadamente 30 pontos percentuais a menos no caso da base $NGFull$. Há pouca diferença entre os resultados obtidos na base $Port10$ em relação à base $NG10$, que possuem tamanho e

Figura 37 – Doc2Vec e CNN: matriz de confusão da base NG05



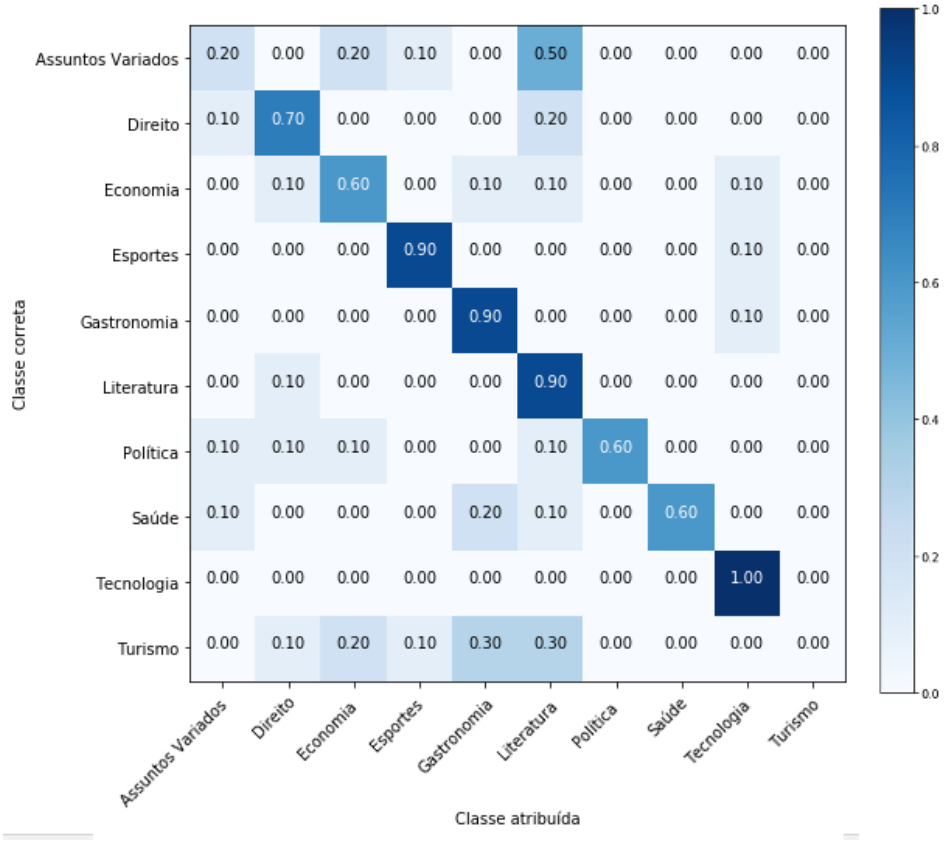
Fonte: o autor.

quantidade de classes semelhantes. Realizando-se o teste t de Student para o resultado destas duas bases de dados, obteve-se um valor $p < 0,05$. E a base $NGFull$ possui resultado pouco superior ao obtido pela base $NG20$, apesar de possuir uma quantidade maior de documentos.

A matriz de confusão da base $NG05$ é mostrada na figura 37. Como pode ser observado, há uma grande confusão entre documentos das classes *talk.politics.misc* com as demais classes, com quase metade de seus documentos sendo classificados como *talk.politics.guns*. A classe *alt.atheism* também teve deiversos documentos atribuídos erroneamente, com quase metade de seus documentos sendo classificados como *soc.religion.christian* e aproximadamente 25% de seus documentos sendo classificados como *talk.politics.guns*. São confusões semelhantes às verificadas na abordagem de rede neural totalmente conectadas, mas com uma quantidade maior de documentos.

A matriz de confusão da base $Port10$ é mostrada na figura 38. Nesta matriz observa-se que a classe *Assuntos Diversos* apresentou a maior taxa de confusão, com 80% de seus documentos sendo atribuídos a outras classes, metade deles indo para a classe

Figura 38 – Doc2Vec e CNN: matriz de confusão da base Port10



Fonte: o autor.

Literatura. A classe *Turismo*, por sua vez, deixou de ser considerada totalmente pelo classificador, não tendo nenhum documento atribuído a ela, nem mesmo os próprios documentos. Isto indica que ou os vetores Doc2Vec não conseguiram captar nenhuma associação de palavras que fosse mais forte nesta classe, ou o uso de rede convolucional fez com que os vetores de palavras desta classe tivessem valores pequenos e fossem desconsiderados quando abrangidos pela janela de convolução. A classe *Literatura*, por sua vez, recebeu 70% a mais documentos do que deveria. Comparando-se esta matriz com a matriz de confusão da abordagem de rede neural totalmente conectada, verifica-se que aqui a confusão teve classes que atraíram (*Literatura*) ou repeliram (*Turismo*) documentos muito mais fortemente, enquanto naquela abordagem houve uma homogeneidade de confusão.

O tempo de execução também foi superior, por exemplo, na base *NG*05 o tempo de execução foi de 155 segundos, bastante superior aos 54 segundos em média da rede FCNN em teste com a mesma base.

Tabela 17 – Base NG05: Rede CNN com topologia piramidal

Camada	Filtros	Kernel
1	200	10
2	100	5
3	50	3
4	30	2

Fonte: o autor

A seguir foram feitos outros testes na base *NG*05, variando-se a topologia da rede CNN. Em um primeiro teste foram alterados os parâmetros de filtro e *kernel*, utilizando-se com a configuração mostrada na tabela 17. Estes parâmetros equiparam-se às redes CNN de topologia piramidal.

Com esta alteração, a base de teste *NG*05 apresentou uma taxa de acerto de 61,55%, superior ao valor obtido no primeiro teste. Este ganho é obtido, apesar de ainda inferior à rede totalmente conectada, por ser feito um refinamento progressivo dos valores dos vetores de entrada. Os valores de entrada são processados pela camada convolucional e os melhores valores são selecionados para a camada seguinte, por meio de uma função de *maxpooling*, restando ao final apenas os valores mais significativos.

Por fim, testou-se a redução de camadas convolucionais, deixando-se apenas uma camada de convolução (sendo testados diversos valores de filtros e *kernel*. Todos os resultados foram insatisfatórios, com os melhores casos possuindo um desempenho inferior em 5 pontos percentuais.

5.2.3 Conclusão do método Doc2Vec

Conforme os resultados obtidos é possível verificar que a abordagem Doc2Vec pode produzir resultados satisfatórios quando a classificação é feita por uma rede neural totalmente conectada (FCNN). Nesta abordagem, são gerados vetores de semelhança entre as palavras dos documentos por meio do Doc2Vec e, a seguir, são gerados vetores que representam cada documento em relação ao modelo Doc2Vec obtido anteriormente. Este modelo é utilizado para treinar uma rede neural e promover a classificação. Como neste momento os modelos não guardam mais relação com a ordem original das palavras, a rede neural totalmente conectada consegue capturar as palavras mais significativas para distinguir os documentos e assim promover a classificação.

Tabela 18 – Rede de topologia mista CNN e FCNN

Camada	Tipo de camada	Observações
1	convolucional	filtros=200, kernel=20
2	totalmente conectada	tamanho=100
3	totalmente conectada	tamanho=50
4	totalmente conectada	tamanho=25

Fonte: o autor

A rede CNN, por sua vez, enfrenta dificuldades ao verificar quais palavras são mais relevantes por serem aplicadas sucessivas reduções em sua dimensionalidade. Isto é semelhante ao processo de se utilizar apenas palavras muito comuns na geração do modelo Doc2Vec, pois apenas as palavras mais comuns possuem vetores que são sucessivamente transmitidos às camadas seguintes. Para verificar esta hipótese, foi criada uma nova topologia com a configuração mostrada na tabela 18.

Para a base de testes $NG05$, esta topologia obteve uma taxa de acerto de 48,06%. Este valor é inferior aos demais testes realizados com redes convolucionais e Doc2Vec, exigindo um tempo maior de processamento computacional mesmo com o uso de *earlystop*. Neste caso, os ganhos obtidos pelo uso de camadas convolucionais que reduzem menos a dimensionalidade e o uso de camadas totalmente conectadas para fazer o restante da classificação não apresenta benefícios comparando-se ao uso direto de camadas totalmente conectadas.

5.3 Extração de características com TF-IDF

Outra abordagem proposta é a extração de características estatísticas do tipo TF-IDF (seções 2.4.2 e 3.2.1).

Os parâmetros que foram utilizados para a extração de características TF-IDF dos documentos estão demonstrados na tabela 19. A remoção de palavras mais comuns, conforme mencionado na seção 3.2.1, reduz o processamento necessário ao remover palavras que, em regra, não apresentam conteúdo semântico, mesmo que elas fossem posteriormente descartadas pelos valores do TF-IDF obtido.

Como mencionado anteriormente, os documentos das 5 bases foram divididos em treinamento, validação e testes. Para a extração da TF-IDF todos os documentos

Tabela 19 – Parâmetros TF-IDF

Parâmetro	Valor	Observações
Cut-off	5	Frequência mínima de palavras
Size	300	Dimensionalidade do vetor
n-grams	1-3	Tamanho do n-grams de palavras a serem considerados
Converter maiúsculas	Sim	Transforma todas as letras em minúsculas
Stopwords	Sim	Remove as palavras mais comuns

Fonte: o autor

Tabela 20 – Topologia das redes FCNN

Camada	Tamanho
Entrada	512
1a escondida	2048
2a escondida	2048
3a escondida	2048
Saída	variável

Fonte: o autor

Tabela 21 – Taxa de acerto do modelo TF-IDF com redes FCNN

Base de dados	**TF-IDF e FCNN**	Doc2Vec e redes FCNN
Port10	86%	79%
NG05	90,22%	79,76%
NG10	89,83%	84,45%
NG20	88,14%	77,24%
NGFull	89,44%	82,76%

Fonte: o autor

eram considerados e, posteriormente, cada documento era representado por um vetor considerando a TF-IDF calculada previamente.

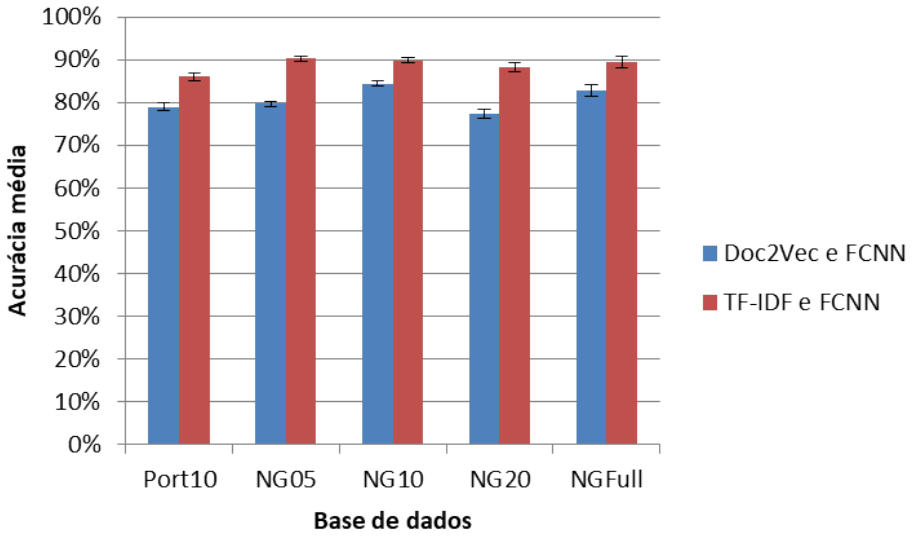
5.3.1 FCNN - Rede Neural Totalmente Conectada

Os primeiros testes foram realizados com TF-IDF em redes neurais totalmente conectadas. Estas redes possuiam a topologia mostrada na tabela 20.

Os resultados obtidos são mostrados na tabela 21, sendo mostrados também os resultados obtidos anteriormente pelo método Doc2Vec com redes totalmente conectadas.

A figura 39 mostra, graficamente, os resultados médios obtidos e o desvio

Figura 39 – TF-IDF e FCNN: taxa de acerto



Fonte: o autor.

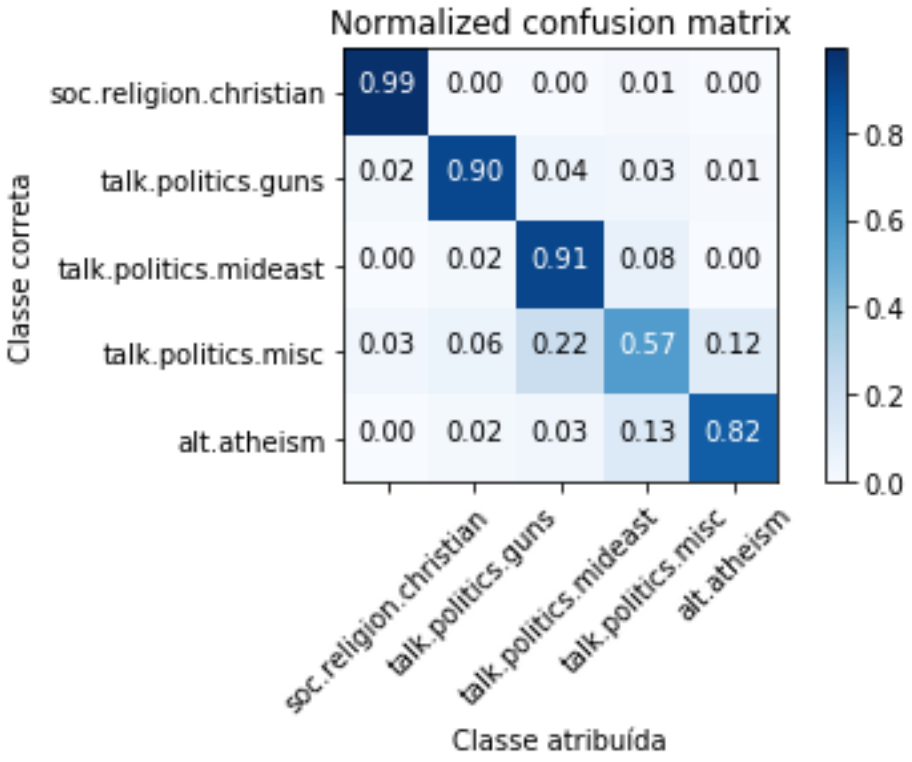
padrão. Também são mostrados os resultados obtidos por Doc2Vec e FCNN, para comparação.

Como pode ser observado, os valores obtidos são superiores aos obtidos pela abordagem Doc2Vec com redes totalmente conectadas (seção 5.2.1). A variação foi de aproximadamente 5 pontos percentuais (base $NG10$) até 12 pontos percentuais (base $NG05$). Os resultados entre as bases de dados também são próximos. Por exemplo, a diferença entre a base $Port10$ e a base $NG10$ é de aproximadamente 4 pontos percentuais, sendo que as *stopwords* removidas a cada idioma são bastante diferentes (isto é, não são meras traduções de um idioma para outro). Realizando-se o teste t de Student para o resultado destas duas bases de dados, obteve-se um valor $p < 0,05$. A quantidade de documentos nas bases também teve pouca influência, como pode ser visto comparando-se os resultados das bases $NG20$ e $NGFull$, cujos resultados também apresentam um valor $p < 0,05$ para o teste t de Student.

A matriz de confusão da base $NG05$ é mostrada na figura 40. Como pode ser observado, as maiores confusões ocorreram entre as classes *talk.politics.misc*, *talk.politics.mideast* e *alt.atheism*, com aproximadamente 30% dos documentos da classe *talk.politics.misc* sendo atribuídos erroneamente.

Também é apresentada a matriz de confusão da base $Port10$ na figura 41. Os

Figura 40 – TF-IDF e FCNN: matriz de confusão da base NG05



Fonte: o autor.

valores são representativos de um ciclo e são exibidos normalizados. É possível observar que as maiores confusões ocorreram com 30% dos documentos da classe *Assuntos Variados* sendo atribuídos a *Literatura*, 20% dos documentos de *Sade* sendo atribuídos a *Assuntos Variados* e 30% dos documentos de *Turismo* também sendo atribuídos a *Assuntos Variados*. *Literatura*, *Poltica* e *Tecnologia* também tiveram documentos atribuídos a *Assuntos Variados*. Considerando que esta classe realmente possui documentos com os mais diversos conteúdos, é de se esperar que seja a principal classe envolvida em confusão de atribuição.

Dada a diferença de bases de dados utilizadas em outros trabalhos não é possível uma comparação exata de resultados, podendo-se realizar apenas um comparativo aproximado.

O trabalho de (WITTLINGER; SPANAKIS; WEISS, 2015) utilizou aproximadamente a mesma base de dados *NGFull*, e obteve desempenho inferior em aproximadamente 7 pontos percentuais: 82,9% contra 89,44%.

O trabalho de (JOHNSON; ZHANG, 2015) utilizou uma abordagem CNN para

Figura 41 – TF-IDF e FCNN: matriz de confusão da base Port10

Fonte: o autor.

gerar a lista de vetores e obteve uma taxa de acerto de 92,29%, utilizando um total de 734.402 documentos, bastante superior aos documentos disponíveis na base $NGFull$.

Em seu trabalho, (ZHANG; LECUN, 2015) realizaram testes em bases jornalísticas compostas apenas de título e uma frase de descrição da notícia, com um total de 241.000 documentos distribuídos em 4 classes distintas: mundo, esportes, negócios e ciência/tecnologia. Obtiveram taxas de acerto de 76,73% utilizando Word2Vec, 86,68% utilizando vetores *bag-of-words* e 87,18% utilizando vetores de caracteres gerados em uma rede CNN. Este resultado é interessante porque a base $NG05$, a princípio mais complexa que a utilizada pelos autores (já que possui 5 classes possíveis e é composta por apenas 1.843 documentos) obteve uma taxa de acerto de 90.22% em nossos testes. As bases de dados e as abordagens são bastante distintas para permitirem uma comparação direta, mas a obtenção de um resultado superior a 90% é interessante. A mesma base

Tabela 22 – Comparativo de tempo das abordagens

Base de dados	**TF-IDF e FCNN** - em segundos	Doc2Vec e FCNN
Port10	24	154
NG05	28	54
NG10	58	95
NG20	143	422
NGFull	382	1.308

Fonte: o autor

foi também utilizada por (JOHNSON; ZHANG, 2017), onde o uso de uma abordagem denominada *Deep Pyramid CNN* obteve a taxa de classificação de 93,13% para as mesmas 4 classes.

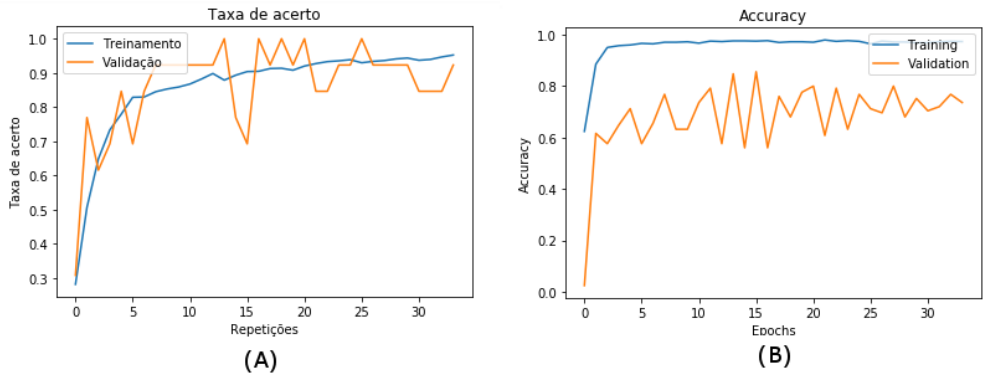
Também (CONNEAU et al., 2017) utilizaram uma base derivada do trabalho de (ZHANG; LECUN, 2015) para classificar 127.600 documentos nas mesmas 4 classes mencionadas anteriormente, obtendo uma taxa de acerto de 91,33%, superior portanto ao nosso resultado na base $NG05$.

Comparando-se com o trabalho anterior de (OLIVEIRA Jr., 2011), é possível observar que a taxa de acerto na base $Port10$ aumentou de 79,61% para 86%, um ganho de 6 pontos percentuais. Mas, como mencionado anteriormente, a realização dos mesmos testes em equipamento mais moderno resultou em um tempo de aproximadamente 535s para execução, enquanto a abordagem TF-IDF com redes neurais totalmente conectadas com uso de *early stop* consumiu um tempo substancialmente inferior de aproximadamente 24 segundos.

Em comparação com abordagem Doc2Vec com redes FCNN, o tempo de execução dos testes é bastante inferior, conforme pode ser observado na tabela 22

Ou seja, mesmo utilizando-se a mesma quantidade de camadas, com o mesmo tamanho, o tempo de execução da abordagem TF-IDF com redes FCNN é bastante inferior à abordagem Doc2Vec com redes FCNN. Medindo-se o tempo de geração do modelo (Doc2Vec ou TF-IDF), verificou-se que são bastante semelhantes, havendo no máximo uma diferença de 5 segundos em seu processamento. O grande ganho ocorre no treinamento da rede neural, onde a condição de estabilidade necessária para ativação do *early stop* ocorre com menos ciclos. Isto pode ser observado, por exemplo, na figura 42, que apresenta as curvas de treinamento da base $NG05$ nas abordagens Doc2Vec

Figura 42 – TF-IDF e FCNN: curvas de treinamento e validação da base NG05



Fonte: o autor.

Tabela 23 – Testes de dimensionalidade dos vetores TF-IDF

Dimensionalidade	*NG*05	*Port*10
100	91,41%	86%
200	90,89%	84,5%
300	90,22%	86%
400	90,73%	83,5%
500	91,06%	88,33%
1.000	91,91%	82,5%

Fonte: o autor

(figura A) e TF-IDF (figura B).

O tamanho dos vetores gerados para TF-IDF são inferiores aos gerados anteriormente para a abordagem Doc2Vec. Por exemplo, para a base *NGFull*, a abordagem TF-IDF utiliza aproximadamente 45 MB, enquanto a abordagem Doc2Vec necessitava de 127 MB para a mesma base.

Nas bases *NG*05 e *Port*10 foram realizados alguns outros testes para verificar se a alteração de parâmetros da geração de vetores TF-IDF impactava no resultado. Por exemplo, a dimensionalidade dos vetores foi alterada conforme mostrado na tabela 23. Os resultados obtidos mostram que há pouca influência na base *NG*05, com variações de no máximo 2 pontos percentuais.

5.3.2 CNN - Rede Neural Convolucional

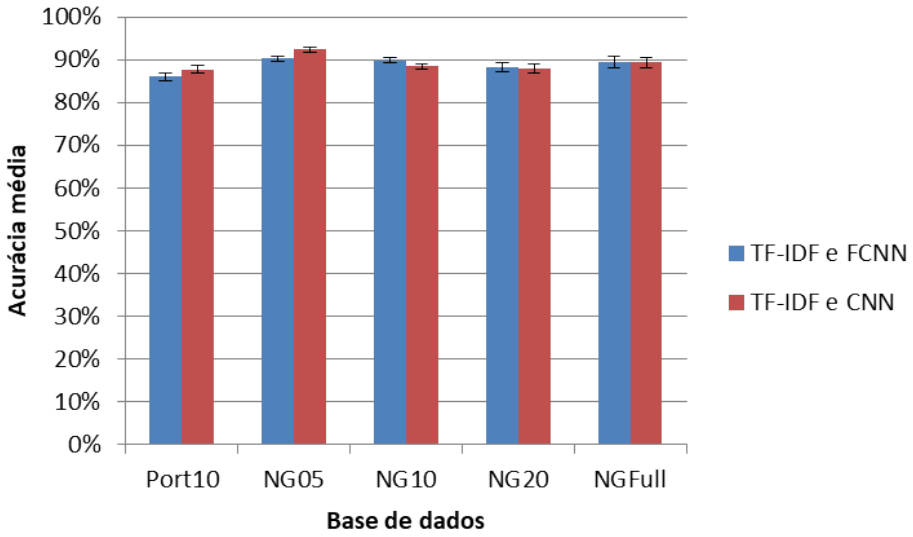
Por fim, foram realizados testes utilizando a extração de características com a abordagem TF-IDF e redes neurais convolucionais (CNN).

Tabela 24 – Taxa de acerto: TF-IDF com CNN

Base de dados	**TF-IDF e CNN**	TF-IDF e FCNN
Port10	79%	86%
NG05	92,41%	90,22%
NG10	88,40%	89,83%
NG20	87,97%	88,14%
NGFull	89,32%	89,44%

Fonte: o autor

Figura 43 – TF-IDF e CNN: taxas de acerto



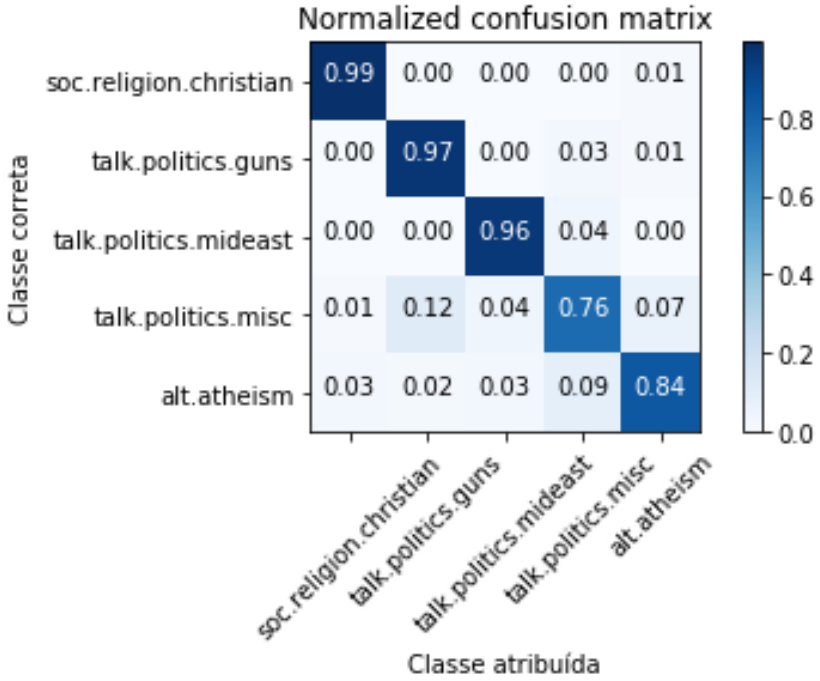
Fonte: o autor.

Os parâmetros utilizados para a extração das características de TF-IDF foram as mesmas da seção 5.3.1, sendo alterada apenas a rede neural utilizada na tarefa de classificação. Nos primeiros testes, a rede CNN utilizada possuía uma camada de convolução. Os resultados obtidos são mostrados na tabela 24.

A figura 43 mostra, graficamente, os resultados médios obtidos e o desvio padrão. São mostrados, também, os resultados obtidos por TF-IDF e FCNN, para comparação.

Como pode ser observado, alguns resultados forem ligeiramente superiores aos obtidos com o uso de redes FCNN. Por exemplo, os testes na base $NG05$ tiveram um ganho de 2 pontos percentuais. Em outras bases, por exemplo a $NG20$, os valores foram bastante próximos. Os resultados entre as bases $NG20$ e $NGFull$ foram próximos

Figura 44 – TF-IDF e CNN: matriz de confusão da base NG05



Fonte: o autor.

(com uma diferença inferior a 1,5 ponto percentual), indicando que a quantidade de arquivos disponíveis para treinamento teve pouca influência no resultado. Os testes de t de Student foram realizados comparando-se os resultados obtidos pelas bases entre as duas abordagens (redes CNN e FCNN), e em todos os casos obteve-se $p < 0,05$, sendo que em alguns casos (na base *NGFull*) o valor foi inferior a $0,002$.

A matriz de confusão da base *NG*05 é mostrada na figura 44. Observa-se que as maiores confusões foram a atribuição de 14 documentos da classe *talk.politics.misc* para *talk.politics.guns* e 19 documentos da classe *alt.atheism* para todas as demais classes. Nota-se que o perfil da matriz de confusão é diferente da abordagem que utilizou uma rede FCNN. Naquele caso, *talk.politics.misc* teve 23 documentos a mais atribuídos a classes erradas e a classe *soc.religion.christian* foi a menos afetada. Na abordagem com redes CNN, *soc.religion.christian*, *talk.politics.guns* e *talk.politics.mideast* foram menos afetadas.

A matriz de confusão da base *Port*10 é mostrada na figura 45. Nesta matriz de confusão é possível observar que 30% dos documentos de *Assuntos Diversos* foram classificados como *Literatura*. Esta mesma confusão aconteceu nas demais abordagens, sendo bastante forte na abordagem Doc2Vec com CNN. A classe *Assuntos Diversos*

Figura 45 – TF-IDF e CNN: matriz de confusão da base Port10

Fonte: o autor.

manteve-se como a classe que mais atrai atribuições incorretas, como verificado em quase todas as outras abordagens. Destaca-se que ao contrário da abordagem Doc2Vec com CNN, a classe *Turismo* teve 70% de suas atribuições corretas, perdendo 30% de seus documentos para *Assuntos Diversos* mas recebendo documentos de *Economia* e *Literatura*. Neste caso, os vetores TF-IDF foram suficientes para manter esta classe ativa. Desta forma, pode-se cogitar que os vetores Doc2Vec que não conseguiram representar os documentos desta classe quando a classificação foi feita com rede CNN. Destaca-se, ainda, que as classes de *Direito* e *Esportes* tiveram sua classificação perfeita, não recebendo nenhuma atribuição incorreta e tendo todos seus arquivos classificados corretamente.

O tempo de execução dos testes nesta abordagem também foram ligeiramente superiores aos obtidos com o uso de redes FCNN. Isto é melhor mostrado na tabela 25. Como mencionado anteriomente, a partir do uso de uma condição de *early stop*, as maiores variações de tempo são decorrentes de quanto ciclos o modelo necessita até que

Tabela 25 – Comparativo de tempo das abordagens

Base de dados	**TD-IDF e CNN** (s)	TF-IDF e FCNN (s)
Port10	65	24
NG05	60	28
NG10	56	58
NG20	184	143
NGFull	443	382

Fonte: o autor

Tabela 26 – Taxa de acerto: TF-IDF com duas camadas convolucionais

Base de dados	Duas camadas	Uma camada
Port10	87,73%	79%
NG05	92,41%	91,23%
NG10	88,40%	88,24%
NG20	87,97%	88,18%
NGFull	89,32%	89,15%

Fonte: o autor

a curva de treinamento atinja uma estabilidade. E as bases maiores ($NG20$ e $NGFull$) necessitaram de mais ciclos.

Outros testes foram executados alterando-se os parâmetros da rede convolucional. Com o uso de duas camadas convolucionais, a base $Port10$ teve uma melhora em seu resultado, atingindo a taxa média de acerto de 87,33%. Neste caso, as camadas convolucionais possuiam filtros de tamanho 50 e 25, respectivamente. As demais bases de dados tiveram um desempenho bastante próximo. Isto é melhor visualizado na tabela 26.

Observa-se que com a melhora na taxa de acerto ao serem utilizadas duas camadas convolucionais, o desempenho entre as bases $Port10$ e $NG10$ foram próximos, o que é interessante porque apesar do idioma ser diferente, apenas as *stopwords* removidas são relacionadas ao idioma, e todo o restante da abordagem é constante.

Foram realizados testes utilizando uma maior quantidade de camadas convolucionais, mas foram obtidos resultados insatisfatórios, com taxas de acerto em média de 15 a 30 pontos percentuais inferiores aos mostrados anteriormente.

Nesta abordagem, ainda, foi realizado um teste com a remoção das *stopwords*, ou seja, nenhuma palavra foi removida previamente dos documentos antes do modelo

TF-IDF ser elaborado. As diferenças obtidas na taxa de acerto foram muito pequenas. Como a abordagem TF-IDF dá um peso menor às palavras muito frequentes, a remoção de *stopwords* apenas diminui o processamento do TF-IDF, pois são palavras que teriam um peso muito pequeno e seriam eliminadas quando o vetor é gerado com uma dimensionalidade baixa (de 50 a 500).

Considerando que os valores obtidos nesta abordagem foram bastante semelhantes aos obtidos com o uso de redes FCNNs, as mesmas comparações com alguns resultados obtidos na literatura podem ser feitas. Por exemplo, (WANG et al., 2016) utilizou uma rede composta por apenas uma camada convolucional para categorizar 12.340 documentos curtos em 8 temas distintos, obtendo uma taxa de acerto de 85,5%. Apesar da quantidade de documentos ser diversa (bem como o tamanho dos documentos), a base $NG10$ obteve taxas de acerto ligeiramente superiores ao classificar documentos em 10 categorias distintas utilizando TF-IDF.

5.3.3 Conclusão do método TF-IDF

Conforme pode ser visto, a extração de características dos documentos utilizando-se o método TF-IDF produziu resultados bastante satisfatórios, com pouca diferença entre os resultados ao se utilizar uma rede neural do tipo totalmente conectada (FCNN) ou convolucional (CNN).

Os resultados obtidos foram superiores, em geral, aos obtidos com a geração de modelo com Doc2Vec em redes totalmente conectadas. Outro ponto importante é a velocidade da abordagem quando se utiliza *early stop*, com as redes neurais atingindo uma condição de estabilidade no treinamento em poucos ciclos.

5.4 Considerações do Capítulo

Neste capítulo foram apresentados os resultados obtidos pelas abordagens propostas: extração de características com Doc2Vec e TF-IDF e o uso de redes neurais totalmente conectadas e convolucionais para a classificação de documentos em categorias pré-estabelecidas.

Foi possível observar que os resultados obtidos são comparáveis ou superam os

Tabela 27 – Resumo: taxas de acerto

Base de dados	Doc2Vec e FCNN	Doc2Vec e CNN	TF-IDF e FCNN	TF-IDF e CNN
Port10	79%	62%	86%	79%
NG05	79,76%	55,65%	90,22%	92,41%
NG10	84,45%	61,76%	89,83%	88,40%
NG20	77,24%	52,73%	88,14%	87,97%
NGFull	82,76%	53,07%	89,44%	89,32%

Fonte: o autor

resultados disponíveis na literatura para bases de dados em lingua inglesa, não tendo sido localizado resultados com bases de dados em lingua portuguesa para se verificar o estado da arte e comparar o desempenho obtido.

Destaca-se que nas abordagens propostas as alterações necessárias para se utilizar o método implementado nos dois idiomas restringem-se à alteração da lista de stopwords respectiva a cada idioma. Considerando-se que na abordagem TF-IDF as palavras muito frequentes possuem um valor diminuído, o uso de *stopwords* pode ser eliminado e então a mesma solução atenderia aos dois idiomas sem qualquer outra alteração.

Observou-se que a extração de características com a abordagem TF-IDF apresentou resultados superiores. Isto é mais facilmente visualizado na tabela 27. Nesta tabela observa-se que a abordagem TF-IDF produziu resultados com taxas de acerto melhores que a abordagem Doc2Vec, e que o uso de redes neurais totalmente conectadas ou convolucionais apresentam resultados bastante próximos.

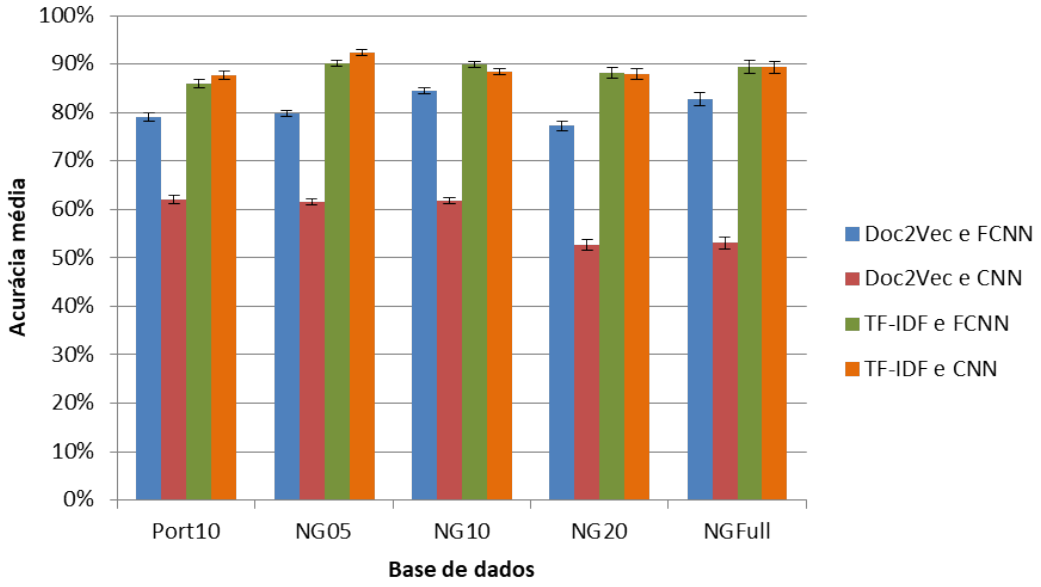
A figura 46 mostra, graficamente, os resultados médios obtidos e o desvio padrão de todos os testes executados.

Verificou-se que os resultados médios obtidos foram semelhantes (nos testes Doc2VEc com rede FCNN e TF-IDF com rede CNN) ou superiores (TF-IDF com rede FCNN) aos obtidos por (OLIVEIRA Jr., 2011) para a mesma base em Português.

A tabela 28 mostra a Topologia das redes neurais utilizadas, detalhando as camadas e alguns de seus parâmetros.

Foi possível verificar, também, que o uso de *early stop* para encerrar antecipadamente o treinamento quando uma condição de equilíbrio é encontrada reduz significativamente o tempo necessário para o treinamento da rede neural. Neste caso, a

Figura 46 – Comparativo de resultados



Fonte: o autor.

Tabela 28 – Resumo: síntese das redes neurais utilizadas

Extração de Características	Topologia	Camada de entrada	Camadas Intermediárias (neurônios)	Camada de saída (ativação)	Total de camadas
Doc2Vec	FCNN	300	5 camadas: 512, 1024, 1024, 2048, 2048	ReLU	7 camadas
Doc2Vec	CNN	300	4 convolucionais + 4 pooling	ReLU	10 camadas
TF-IDF	FCNN	300	5 camadas: 512, 1024, 1024, 2048, 2048	ReLU	7 camadas
TF-IDF	CNN	300	2 convolucionais + 2 pooling	ReLU	6 camadas

Fonte: o autor

Tabela 29 – Exemplos de palavras relevantes para a classificação de documentos

Tema	Palavras
Assuntos Variados	tempo, guerra, ano
Direito	direito, consumidor, justiça, jurídica
Economia	banco, indústria, valor
Esportes	jogo, vitória, torcedor
Gastronomia	restaurante, sabor, prato
Literatura	autor, romance, carta, livro
Política	governo, público, Brasil
Saúde	saúde, doença, mental, exames
Tecnologia	computador, site, web
Turismo	passagem, foto, dias

Fonte: o autor

abordagem TF-IDF com redes neurais totalmente conectadas apresentou, em geral, os melhores resultados. Considerando-se que os testes foram executados em equipamento sem aceleração fornecida por GPU, a adoção de *early stop* permitiu que diversos testes pudessem ser executados em tempo hábil.

Realizou-se um teste para verificar as palavras que mais contribuíram para a classificação correta na base *Port*10. Para isto, uma pequena amostra dos documentos que eram atribuídos corretamente a cada classe foram selecionados e editados, verificando-se quais palavras eram necessárias para a atribuição continuasse sendo feita corretamente. Para a base *Port*10, as palavras que mais auxiliaram na classificação corretas são exibidas na tabela 29

Disto, observa-se que apesar do tema Assuntos Variados tratar realmente de assuntos diversos, alguns documentos observados tratavam de temas associados ao final de ano, com palavras como *Natal*, *Ano-Novo*, e *paz* estando presentes. Imagina-se que muitos documentos foram coletados nos meses finais do ano, quando estes temas são abordados.

6 Conclusão e Trabalhos Futuros

A partir do trabalho realizado é possível extrair-se algumas conclusões importantes, sendo a principal delas que o uso de redes neurais pode apresentar um bom resultado mesmo com base de dados compostas por poucos documentos. Mesmo sabendo-se que a quantidade de documentos gerados aumenta constantemente, obter bons resultados com poucos documentos é significativo, principalmente em relação ao menor esforço computacional necessário para a geração de modelos de vetores e para o treinamento de redes neurais.

Foi possível realizar todos os testes pretendidos, utilizando-se duas técnicas de geração de vetores (TF-IDF e Doc2Vec) e aplicá-los em duas redes neurais diferentes (redes neurais totalmente conectadas e redes neurais convolucionais). Constatou-se que há uma grande quantidade de parâmetros possíveis de configuração em todas as abordagens vistas de redes neurais. Somando-se a isto às diferentes topologias possíveis, é possível concluir que a escolha de parâmetros adequados ainda é incipiente. Raras literaturas tratam de comparação entre os diversos parâmetros, restando a necessidade da realização de diversos testes para obter-se a configuração satisfatória para cada problema abordado. Entre os parâmetros, por vezes, há o *trade-off* entre se aumentar a taxa de acerto e minimizar o tempo necessário para o treinamento da rede neural.

Por fim, verificou-se que foi possível atingir o objetivo proposto, utilizando-se redes neurais para a classificação de documentos em língua portuguesa em classes pré-definidas, mesmo em condições onde as classes apresentam bastante semelhança e a quantidade de documentos disponíveis para o treinamento é reduzida.

Como trabalhos futuros, sugere-se que outras características sejam extraídas dos documentos, verificando-se se o uso de redes neurais apresenta um desempenho superior quando vetores representando características sintáticas ou semânticas são utilizados. Sugere-se, também, que sejam realizados testes com outros classificadores, verificando-se a matriz de confusão resultante e eventuais melhorias nos resultados pela combinação de resultados de cada classificador.

Outra sugestão para trabalhos futuros é que as técnicas aqui demonstradas

sejam utilizadas nos problemas relacionados à atribuição e verificação de autoria de documentos eletrônicos. Sabe-se que em diversas situações as informações sobre a produção de um documento eletrônico não estão presentes, restando apenas o conteúdo do documento a ser analisado. O uso de técnicas de extração de informações e a classificação com redes neurais pode apresentar resultados promissores. Por exemplo, no Anexo B do presente trabalho é apresentado o resultado de um teste básico realizado utilizando-se as técnicas descritas para a verificação de autoria de documentos.

Referências

ADEVA, JJ García; ATXA, JM Pikatza; CARRILLO, M Ubeda; ZENGOTITABENGOA, E Ansuategi. Automatic text classification to support systematic reviews in medicine. Elsevier, v. 41, n. 4, p. 1498–1508, 2014. Citado na página 27.

AL, Weibo Liu et. A survey of deep neural network architectures and their applications. Elsevier B.V., v. 234, n. October 2016, p. 11–26, 2017. Citado 2 vezes nas páginas 33 e 37.

ASTRAKHANTSEV, N A; FEDORENKO, Denis G; TURDAKOV, D Yu. Methods for automatic term recognition in domain-specific text collections: A survey. Springer, v. 41, n. 6, p. 336–349, 2015. ISSN 0361-7688. Citado na página 42.

ATTARDI, Giuseppe; GORRIERI, Laura; MIASCHI, Alessio; PETROLITO, Ruggero. Deep learning for social sensing from tweets. p. 20, 2015. Citado na página 27.

BENGIO, Y. Learning Deep Architectures for AI. v. 2, n. 1, p. 1–127, 2009. ISSN 1935-8237. Disponível em: <http://www.nowpublishers.com/article/Details/MAL-006>. Citado na página 37.

BHATTACHARYYA, Debnath; DAS, Poulami; GANGULY, Debashis; MITRA, Kheyali; DAS, Purnendu; BANDYOPADHYAY, Samir Kumar; KIM, Tai-hoon. Unstructured document categorization: A study. Citeseer, 2008. Citado na página 23.

BIRD, Steven; KLEIN, Ewan; LOPER, Edward. *Natural language processing with Python: analyzing text with the natural language toolkit*. [S.l.]: "O'Reilly Media, Inc.", 2009. ISBN 0596555717. Citado na página 66.

BISHOP, Christopher M. *Pattern Recognition and Machine Learning*. [S.l.]: Springer Science+Business Media, LLC, 2006. 758 p. Citado 2 vezes nas páginas 17 e 31.

BRASIL. *Lei Federal n. 10753/03*. 2003. Disponível em: <http://www.planalto.gov.br/CCivil_03/leis/2003/L10.753compilada.htm>. Citado na página 16.

CAMBRIA, Erik; WHITE, Bebo. Jumping NLP Curves: A Review of Natural Language Processing Research [Review Article]. v. 9, n. 2, p. 48–57, 2014. Citado na página 26.

CHOWDHURY, Gobinda G. Natural language processing. v. 37, n. 1, p. 51–89, 2005. ISSN 0066-4200. Citado 2 vezes nas páginas 18 e 26.

CILIBRASI, Rudi; VITÁNYI, Paul M B. Clustering by compression. IEEE, v. 51, n. 4, p. 1523–1545, 2005. ISSN 0018-9448. Citado na página 80.

COLLINS, Michael; DUFFY, Nigel. Convolution kernels for natural language. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2002. p. 625–632. Citado na página 37.

CONNEAU, Alexis; SCHWENK, Holger; BARRAULT, Loïc; LECUN, Yann. Very deep convolutional networks for text classification. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. [S.l.: s.n.], 2017. v. 1, p. 1107–1116. Citado 3 vezes nas páginas 53, 55 e 102.

DAI, Yue; KAKKONEN, Tuomo; SUTINEN, Erkki. MinEDec: a decision-support model that combines text-mining technologies with two competitive intelligence analysis methods. v. 3, p. 165–173, 2011. Citado na página 39.

DEWEY, M. *A classification and subject index for cataloguing and arranging the books and pamphlets of a library.* [S.l.]: Kingsport Press Inc., 1876. Citado na página 17.

DHILLON, Inderjit; KOGAN, Jacob; NICHOLAS, Charles. Feature Selection and Document Clustering BT - Survey of Text Mining: Clustering, Classification, and Retrieval. In: BERRY, Michael W (Ed.). [S.l.]: Springer New York, 2004. p. 73–100. ISBN 978-1-4757-4305-0. Citado na página 24.

DIURDEVA, Elena Mikhailova Polina; SHALYMOV, Dmitry. Writer identification based on letter frequency distribution. In: *2016 19th Conference of Open Innovations Association (FRUCT)*. IEEE, 2016. p. 24–30. ISBN 978-952-68397-4-5. Disponível em: <http://ieeexplore.ieee.org/document/7892179/>. Citado na página 25.

DUCH, Włodzisław; JANKOWSKI, Norbert. Survey of neural transfer functions. v. 2, n. 1, p. 163–212, 1999. Citado 2 vezes nas páginas 29 e 31.

FAUSETT, Laurene V et al. *Fundamentals of neural networks: architectures, algorithms, and applications*. [S.l.]: Prentice-Hall Englewood Cliffs, 1994. v. 3. Citado 2 vezes nas páginas 18 e 19.

FIESLER, Emile; BEALE, Russell. *Handbook of neural computation*. [S.l.]: CRC Press, 1996. Citado na página 35.

FRIEDMAN, Jerome; HASTIE, Trevor; TIBSHIRANI, Robert. *The elements of statistical learning*. [S.l.]: Springer series in statistics New York, 2001. v. 1. Citado 2 vezes nas páginas 38 e 63.

FUKUSHIMA, Kunihiko. Neural network model for a mechanism of pattern recognition unaffected by shift in position-neocognitron. *IEICE Technical Report, A*, v. 62, n. 10, p. 658–665, 1979. Citado na página 37.

GOLDBERG, Yoav. A Primer on Neural Network Models for Natural Language Processing. v. 57, p. 345–420, 2016. Citado 2 vezes nas páginas 27 e 35.

GOLLER, C; LÖNING, J; WILL, T; WOLFF, W. Automatic document classification. p. 145, 2000. Citado 2 vezes nas páginas 23 e 24.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. Deep Learning. v. 13, n. 1, p. 35–35. ISSN 1548-7091. Citado na página 33.

GREENBERG, Joseph H. (Joseph Harold). *Indo-European and its closest relatives : the Eurasiatic language family*. Stanford University Press, 2000. ISBN 9780804738125. Disponível em: <http://www.sup.org/books/title/?id=360>. Citado na página 25.

HARPER, Shirley F. The universal decimal classification. Wiley Subscription Services, Inc., A Wiley Company, v. 5, n. 4, p. 195–213, 1954. Citado na página 17.

HASHIMI, Hussein; HAFEZ, Alaaeldin; MATHKOUR, Hassan. Selection criteria for text mining approaches. Elsevier, v. 51, p. 729–733, 2015. ISSN 0747-5632. Citado na página 17.

HASSOUN, Mohamad H. *Fundamentals of artificial neural networks*. [S.l.]: MIT press, 1995. Citado na página 19.

HAYKIN, Simon S. *Neural networks : a comprehensive foundation*. [S.l.]: Prentice Hall, 1999. 842 p. ISBN 0132733501. Citado 2 vezes nas páginas 28 e 33.

HEBB, Donald Olding. The organization of behavior; a neuropsycholocigal theory. *A Wiley Book in Clinical Psychology.*, John Wiley and Sons Inc, p. 62–78, 1949. Citado na página 34.

HUYNH, Trung; HE, Yulan; WILLIS, Alistair; RÜGER, Stefan. Adverse drug reaction classification with deep neural networks. In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. [S.l.: s.n.], 2016. p. 877–887. Citado na página 27.

Instituto Camões. A language for the world. 2017. Citado na página 20.

International Organization for Standardization. *ISO/IEC 8859-1:1998 - Information technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1*. 1998. 10 p. Disponível em: <https://www.iso.org/standard/28245.html>. Citado na página 60.

IRFAN, Rizwana; KING, Christine K; GRAGES, Daniel; EWEN, Sam; KHAN, Samee U; MADANI, Sajjad A; KOLODZIEJ, Joanna; WANG, Lizhe; CHEN, Dan; RAYES, Ammar. A survey on text mining in social networks. Cambridge University Press, v. 30, n. 2, p. 157–170, 2015. ISSN 0269-8889. Citado na página 40.

ISLAM, Md Saiful et al. A Comparative Study on Different Types of Approaches to Bengali Document Categorization. In: *Proceedings of the International Conference on Engineering Research, Innovation and Education 2017*. [S.l.]: Association for Computational Linguistics, 2017. Citado na página 16.

JOHNSON, Rie; ZHANG, Tong. Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. p. 103–112, 2014. Citado 2 vezes nas páginas 27 e 51.

JOHNSON, Rie; ZHANG, Tong. Semi-supervised convolutional neural networks for text categorization via region embedding. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2015. p. 919–927. Citado 4 vezes nas páginas 41, 51, 55 e 100.

JOHNSON, Rie; ZHANG, Tong. Deep pyramid convolutional neural networks for text categorization. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. [S.l.: s.n.], 2017. v. 1, p. 562–570. Citado 3 vezes nas páginas 53, 55 e 102.

JONES, Karen Sparck. A statistical interpretation of term specificity and its application in retrieval. MCB UP Ltd, v. 28, n. 1, p. 11–21, 1972. Citado na página 41.

KÄKI, Mika. Findex: search result categories help users when document ranking fails. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. [S.l.]: ACM, 2005. p. 131–140. ISBN 1581139985. Citado na página 17.

KIM, Yoon. Convolutional neural networks for sentence classification. In: *2014 Conference on Empirical Methods in Natural Language Processing*. [S.l.]: Association for Computational Linguistics, 2014. p. 1746–1751. Citado 2 vezes nas páginas 49 e 55.

KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2012. p. 1097–1105. Citado na página 30.

LANG, Ken. Newsweeder: Learning to filter netnews. In: *Proceedings of the Twelfth International Conference on Machine Learning*. [S.l.: s.n.], 1995. p. 331–339. Citado na página 61.

LAU, Jey Han; BALDWIN, Timothy. An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation. 2016. Disponível em: <http://arxiv.org/abs/1607.05368>. Citado na página 76.

LE, Quoc; MIKOLOV, Tomas. Distributed representations of sentences and documents. In: *International Conference on Machine Learning*. [S.l.: s.n.], 2014. p. 1188–1196. Citado 5 vezes nas páginas 43, 46, 47, 48 e 49.

LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. *nature*, Nature Publishing Group, v. 521, n. 7553, p. 436, 2015. Citado na página 34.

LECUN, Yann A.; BOTTOU, Léon; ORR, Genevieve B.; MÜLLER, Klaus Robert. Efficient backprop. v. 7700 LECTU, p. 9–48, 2012. Citado na página 30.

LEE, Ji Young; DERNONCOURT, Franck. Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks. In: *Proceedings of NAACL-HLT*. [S.l.: s.n.], 2016. p. 515–520. Citado 2 vezes nas páginas 52 e 55.

LEI, Tao; BARZILAY, Regina; JAAKKOLA, Tommi. Molding cnns for text: non-linear, non-consecutive convolutions. 2015. Citado 2 vezes nas páginas 51 e 55.

LEVI, Gil; HASSNER, Tal. Age and gender classification using convolutional neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. [S.l.: s.n.], 2015. p. 34–42. Citado na página 27.

LI, Xin; ROTH, Dan. Learning question classifiers. In: *Proceedings of the 19th international conference on Computational linguistics-Volume 1*. [S.l.]: Association for Computational Linguistics, 2002. p. 1–7. Citado na página 50.

LIU, Pengfei; QIU, Xipeng; HUANG, Xuanjing. Recurrent neural network for text classification with multi-task learning. In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. [S.l.]: AAAI Press, 2016. p. 2873–2879. ISBN 1577357701. Citado na página 21.

LOPEZ, Marc Moreno; KALITA, Jugal. Deep Learning applied to NLP. 2017. Citado 3 vezes nas páginas 37, 38 e 69.

MAJUMDER, Navonil; PORIA, Soujanya; GELBUKH, Alexander; CAMBRIA, Erik. Deep learning-based document modeling for personality detection from text. IEEE, v. 32, n. 2, p. 74–79, 2017. ISSN 1541-1672. Citado na página 26.

MCCULLOCH, Warren S.; PITTS, Walter. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, v. 5, n. 4, p. 115–133, Dec 1943. ISSN 1522-9602. Citado na página 34.

MIKOLOV, Tomas; SUTSKEVER, Ilya; CHEN, Kai; CORRADO, Greg S; DEAN, Jeff. Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2013. p. 3111–3119. Citado 4 vezes nas páginas 43, 45, 46 e 49.

MIKOLOV, Tomas; YIH, Wen-tau; ZWEIG, Geoffrey. Linguistic regularities in continuous space word representations. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. [S.l.: s.n.], 2013. p. 746–751. Citado 2 vezes nas páginas 43 e 44.

MOU, Lili; LI, Ge; ZHANG, Lu; WANG, Tao; JIN, Zhi. Convolutional Neural Networks over Tree Structures for Programming Language Processing. In: *AAAI*. [S.l.: s.n.], 2016. v. 2, n. 3, p. 4. Citado na página 26.

NARENDRA, K. S.; THATHACHAR, M. A. L. Learning automata - a survey. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-4, n. 4, p. 323–334, July 1974. Citado na página 34.

NEELAKANTAN, Arvind; SHANKAR, Jeevan; PASSOS, Alexandre; MCCALLUM, Andrew. Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. [S.l.: s.n.], 2014. p. 1059–1069. Citado na página 75.

NIST. *Text REtrieval Conference (TREC) Home Page*. 2000. Citado na página 50.

OLIVEIRA Jr., Walter Ribeiro de. *Atribuição de Autoria de Documentos em Língua Portuguesa Utilizando a Distância Normalizada de Compressão*. 158 p. Tese (Dissertação de Mestrado) — PUC-PR, 2011. Citado 9 vezes nas páginas 19, 21, 60, 80, 83, 85, 102, 109 e 128.

Oxford Dictionary. *Oxford dictionary*. 2017. Disponível em: <https://en.oxforddictionaries.com/definition/information_explosion>. Citado na página 16.

PEDREGOSA, F et al. Scikit-learn: Machine Learning in {P}ython. v. 12, p. 2825–2830, 2011. Citado na página 66.

PINKER, Steven; BLOOM, Paul. Natural language and natural selection. Cambridge University Press, v. 13, n. 04, p. 707–727, 1990. ISSN 0140-525X. Citado 2 vezes nas páginas 24 e 25.

PORIA, Soujanya; CAMBRIA, Erik; GELBUKH, Alexander. Aspect extraction for opinion mining with a deep convolutional neural network. Elsevier, v. 108, p. 42–49, 2016. ISSN 0950-7051. Citado na página 27.

PORIA, Soujanya; CAMBRIA, Erik; HAZARIKA, Devamanyu; VIJ, Prateek. A Deeper Look into Sarcastic Tweets Using Deep Convolutional Neural Networks. In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. [S.l.: s.n.], 2016. p. 1601–1612. Citado na página 26.

PRUSA, Joseph D; KHOSHGOFTAAR, Taghi M. Improving deep neural network design with new text data representations. Springer, v. 4, n. 1, p. 7, 2017. ISSN 2196-1115. Citado na página 27.

QIN, Pengda; XU, Weiran; GUO, Jun. An empirical convolutional neural network approach for semantic relation classification. Elsevier, v. 190, p. 1–9, 2016. ISSN 0925-2312. Citado na página 26.

RAMOS, Juan. Using tf-idf to determine word relevance in document queries. In: *Proceedings of the first instructional conference on machine learning*. [S.l.: s.n.], 2003. v. 242, p. 133–142. Citado na página 42.

REHUREK, Radim; SOJKA, Petr. Software Framework for Topic Modelling with Large Corpora. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. [S.l.]: ELRA, 2010. p. 45–50. Citado 4 vezes nas páginas 45, 47, 67 e 75.

RICHTER, Marc; WRONA, Konrad. Devil in the Details: Assessing Automated Confidentiality Classifiers in the Context of NATO Documents. In: *ITASEC*. [S.l.: s.n.], 2017. p. 136–145. Citado na página 27.

RITTER, Samuel; BARRETT, David G T; SANTORO, Adam; BOTVINICK, Matt M. Cognitive Psychology for Deep Neural Networks: A Shape Bias Case Study. In: *International Conference on Machine Learning*. [S.l.: s.n.], 2017. p. 2940–2949. Citado na página 28.

ROBERTSON, Stephen. Understanding inverse document frequency: on theoretical arguments for IDF. Emerald Group Publishing Limited, v. 60, n. 5, p. 503–520, 2004. ISSN 0022-0418. Citado 2 vezes nas páginas 41 e 42.

ROSENBLATT, Frank. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958. Citado na página 34.

SALTON, Gerard; WONG, Anita; YANG, Chung-Shu. A vector space model for automatic indexing. ACM, v. 18, n. 11, p. 613–620, 1975. ISSN 0001-0782. Citado na página 41.

SANTOS, Cicero; GATTI, Maira. Deep convolutional neural networks for sentiment analysis of short texts. In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. [S.l.: s.n.], 2014. p. 69–78. Citado na página 27.

SCHMIDHUBER, Jürgen. Deep learning in neural networks: An overview. *Neural networks*, Elsevier, v. 61, p. 85–117, 2015. Citado 2 vezes nas páginas 34 e 37.

SEBASTIANI, Fabrizio. Machine learning in automated text categorization. ACM, v. 34, n. 1, p. 1–47, 2002. ISSN 0360-0300. Citado 2 vezes nas páginas 18 e 23.

SELIVANOVA, I V; RYABKO, B Ya; GUSKOV, A E. Classification by compression: Application of information-theory methods for the identification of themes of scientific texts. Springer, v. 51, n. 3, p. 120–126, 2017. ISSN 0005-1055. Citado na página 70.

SEVERYN, Aliaksei; MOSCHITTI, Alessandro. Twitter sentiment analysis with deep convolutional neural networks. In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. [S.l.]: ACM, 2015. p. 959–962. ISBN 1450336213. Citado na página 27.

SRIVASTAVA, Nitish; HINTON, Geoffrey; KRIZHEVSKY, Alex; SUTSKEVER, Ilya; SALAKHUTDINOV, Ruslan. Dropout: A simple way to prevent neural networks from overfitting. JMLR. org, v. 15, n. 1, p. 1929–1958, 2014. ISSN 1532-4435. Citado 2 vezes nas páginas 32 e 33.

STAMATATOS, Efstathios. A Survey of Modern Authorship Attribution Methods. 2009. Citado na página 17.

STANFORD, University of. *The Stanford Natural Language Processing Group*. 2018. Citado na página 20.

STEIN BENNO; EISSEN, Zu; Sven Meyer. Automatic document categorization. In: *Annual Conference on Artificial Intelligence*. [S.l.]: Springer, 2003. p. 254–266. Citado na página 24.

TAKC, Hidayet; SOGUKPINAR, Ibrahim. *Centroid-based language identification using letter feature set*. [S.l.]: Springer, Berlin, Heidelberg, 2004. 640–648 p. Citado 2 vezes nas páginas 16 e 25.

TANG, Duyu; QIN, Bing; LIU, Ting. Document modeling with gated recurrent neural network for sentiment classification. In: *Proceedings of the 2015 conference on empirical methods in natural language processing*. [S.l.: s.n.], 2015. p. 1422–1432. Citado na página 27.

TOFFLER, Alvin. *Future shock*. [S.l.]: Random House, 1970. 505 p. Citado na página 16.

TRAN, Tung; KAVULURU, Ramakanth. Predicting mental conditions based on "history of present illness"in psychiatric notes with deep neural networks. Elsevier, v. 75, p. S138–S148, 2017. ISSN 1532-0464. Citado na página 27.

VARELA, Paulo Junior. *O Uso De Atributos Estilométricos Na Identificação Da Autoria De Textos*. Tese (Dissertação de Mestrado) — PUC-PR, 2010. Citado 2 vezes nas páginas 57 e 80.

VARELA, P. J. *Uma abordagem computacional baseada em análise sintática multilingue na atribuição da autoria de documentos digitais*. 119 p. Tese (Doutorado) — PUC-PR, 2017. Citado na página 40.

WANG, Peng; XU, Bo; XU, Jiaming; TIAN, Guanhua; LIU, Cheng Lin; HAO, Hongwei. Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. v. 174, p. 806–814, 2016. ISSN 1872-8286. Citado 3 vezes nas páginas 52, 55 e 108.

WERBOS, Paul J. Applications of advances in nonlinear sensitivity analysis. Springer, p. 762–770, 1982. Citado na página 34.

WINDISCH, G.; CSINK, L. Language identification using global statistics of natural languages. In: *Proceedings of the 2nd Romanian-Hungarian Joint Symposium on Applied Computational Intelligence (SACI)*. [S.l.: s.n.], 2005. p. 243–255. Citado na página 26.

WINOGRAD, Terry. Understanding natural language. Academic Press, v. 3, n. 1, p. 1–191, 1972. Disponível em: <https://www.sciencedirect.com/science/article/pii/0010028572900023>. Citado na página 24.

WITTLINGER, Christopher; SPANAKIS, Gerasimos; WEISS, Gerhard. Flexible Deep Neural Network structure with application to Natural Language Processing. In: *BNAIC 2015 THE 27TH BENELUX CONFERENCE ON ARTIFICIAL INTELLIGENCE*. [S.l.: s.n.], 2015. Citado 5 vezes nas páginas 19, 51, 55, 80 e 100.

YAN, Rui; SONG, Yiping; WU, Hua. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In: *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. [S.l.]: ACM, 2016. p. 55–64. ISBN 1450340695. Citado na página 27.

YANG, Zichao; YANG, Diyi; DYER, Chris; HE, Xiaodong; SMOLA, Alex; HOVY, Eduard. Hierarchical attention networks for document classification. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. [S.l.: s.n.], 2016. p. 1480–1489. Citado 2 vezes nas páginas 53 e 55.

YU, Bei. An evaluation of text classification methods for literary study. Oxford University Press, v. 23, n. 3, p. 327–343, 2008. ISSN 0268-1145. Citado na página 67.

ZHANG, Xiang; LECUN, Yann. Text understanding from scratch. 2015. Citado 6 vezes nas páginas 51, 52, 53, 55, 101 e 102.

ZHANG, Ye; WALLACE, Byron. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. [S.l.: s.n.], 2017. v. 1, p. 253–263. Citado na página 68.

Anexos

ANEXO A –

Código-fonte utilizado

```
# coding: utf-8
# # Classificacao de documentos

import time
start_time = time.clock()

import os
import re
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import keras

from keras.layers import Dense, Dropout, LSTM, Embedding
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import TruncatedSVD
from sklearn.preprocessing import LabelEncoder
from nltk.corpus import stopwords

# ## 1. Carrega dados

#Configura
base_dir='d:\\diretorio\\de\\bases\\'
test_size=100
dataset_dir=base_dir + 'treinamento'
```

```
files_data=[]
for file in os.listdir(dataset_dir):
item=[file[0]]
with open(os.path.join(dataset_dir, file), "r") as infile:
item.append(infile.read())
files_data.append(item)

dataset_dir=base_dir + 'teste'
print("Dataset: " + dataset_dir)

for file in os.listdir(dataset_dir):
item=[file[0]]
with open(os.path.join(dataset_dir, file), "r") as infile:
item.append(infile.read())
files_data.append(item)

dataset=pd.DataFrame(files_data, columns=['class', 'text'])
dataset.head()

from sklearn.model_selection import train_test_split
# train, test = train_test_split(dataset, test_size=0.2,random_state=200)
train, test = train_test_split(dataset, test_size=test_size,random_state=200,shuffl

# ## 2. Pre-processamento

stop_words = set(stopwords.words('portuguese'))
tfidf = TfidfVectorizer(min_df=5, max_features=3000,
strip_accents='unicode',
lowercase =True, analyzer='word', token_pattern='\w+',
use_idf=True, smooth_idf=True, sublinear_tf=True,
stop_words = stop_words)
svd = TruncatedSVD(300)
x_train = tfidf.fit_transform(train['text'].values)
```

```
x_train = svd.fit_transform(x_train)


x_test = tfidf.transform(test['text'].values)
x_test = svd.transform(x_test)


encoder = LabelEncoder()
encoder.fit(train['class'].values)
encoded_y = encoder.transform(train['class'].values)
y_train = keras.utils.to_categorical(encoded_y)
test_encoded_y = encoder.transform(test['class'].values)
y_test = keras.utils.to_categorical(test_encoded_y)

# ## 3. Treinamento

def create_model(input_length,output_length):
model = Sequential()
model.add(Dense(512, input_dim=input_length, activation='relu'))
model.add(Dropout(0.3))

# 1a
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.3))
# 2a
model.add(Dense(2048, activation='relu'))
model.add(Dropout(0.3))
# 3a
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.3))
#Saida
model.add(Dense(output_length, activation="relu"))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy
return model
```

```
model = create_model(len(x_train[0]),len(y_train[0]))
early_stop = keras.callbacks.EarlyStopping(monitor='val_acc', min_delta=0.01, patie

estimator = model.fit(x_train, y_train, batch_size=512, epochs=150, validation_spli
print("Acuracia: %.2f%% / Validacao: %.2f%%" % (100*estimator.history['acc'][-1], 1

# ## 4. Curvas de treinamento e validacao
# Acuracia
plt.plot(estimator.history['acc'])
plt.plot(estimator.history['val_acc'])
plt.title('Acuracia')
plt.ylabel('Acuracia')
plt.xlabel('Ciclos')
plt.legend(['training', 'validation'], loc='upper right')
plt.show()

# Perdas
plt.plot(estimator.history['loss'])
plt.plot(estimator.history['val_loss'])
plt.title('Perdas do modelo')
plt.ylabel('Perda')
plt.xlabel('Repeticoes')
plt.legend(['treinamento', 'validacao'], loc='upper right')
plt.show()

# ## 5. Testa

score, acc = model.evaluate(x_test, y_test, batch_size=32)
print('Acuracia: {0:.2f}%'.format(acc*100))
print("--- %s segundos ---" % (time.clock() - start_time))
```

ANEXO B – Atribuição de autoria

Sabendo-se que a base de dados *Port10* já foi utilizada em trabalhos anteriores (por exemplo, (OLIVEIRA Jr., 2011)) para a verificação de autoria de documentos eletrônicos, foi realizado um pequeno teste para verificação da taxa de acerto na verificação de autoria.

Foi utilizada a mesma configuração mencionada anteriormente em uma das abordagens: extração de características com TF-IDF e classificação feita com uso de rede neural totalmente conectada, com os mesmos parâmetros de camadas e quantidad de neurônios explicada neste trabalho.

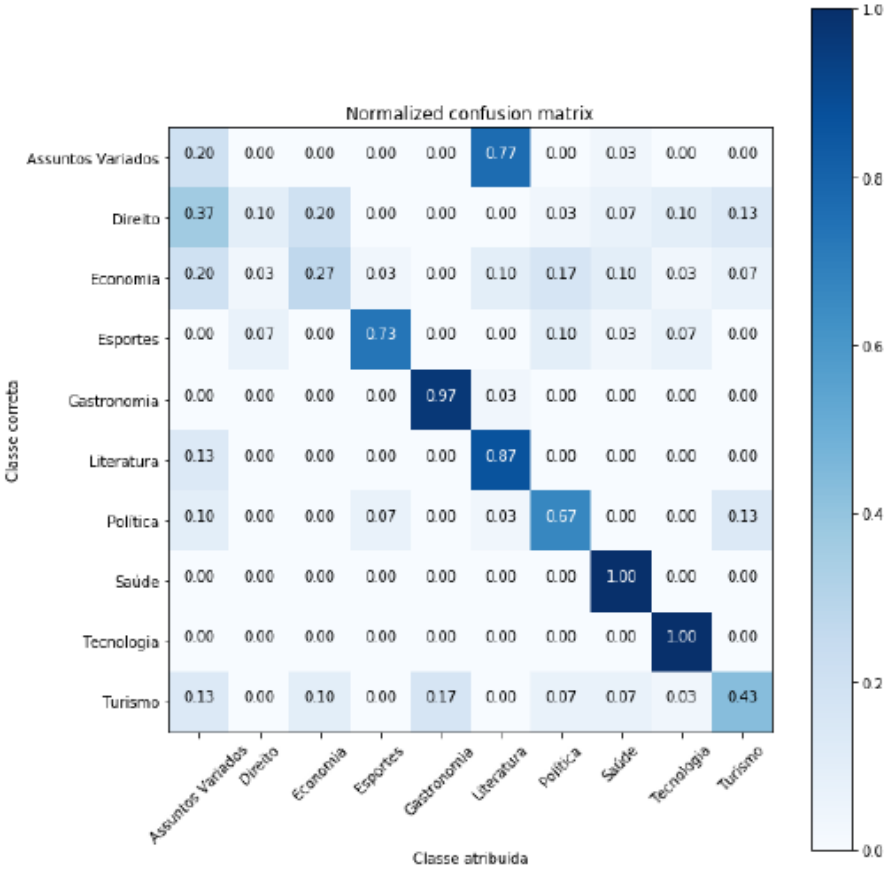
Foi obtida uma taxa de atribução correta de aproximadamente 63% para testes realizados com 100 autores possíveis. A matriz de confusão é exibida na figura 47.

Neste teste, documentos conhecidos de cada autor são utilizados para treinamento, e os documentos questionados são depois testados verificando-se para qual autor, entre os existentes, é atribuída a autoria.

Como mencionado neste trabalho, a base de dados *Port10* é desafiadora, incluindo sempre 10 autores que escrevem sobre o mesmo tema, sendo esperada a atribuição errônea de autoria mas o acerto da classe (ou seja, o documento é atribuído a outro autor que escreve sobre o mesmo tema). Neste teste simples foi possível verificar que em alguns temas houve pequena confusão entre as classes. Por exemplo, para a classe Tecnologia, todos os documentos que foram atribuídos à classe tecnologia foram atribuídos ao autor corretamente. Neste caso, nenhum documento de um autor de Esportes, por exemplo, foi atribuído a um dos autores do tema Tecnologia. Por outro lado, aproximadamente 23% dos textos escritos por autores de Tecnologia foram atribuídos a autores de outros temas.

Por outro lado, aproximadamente 37% dos documentos de autores do tema Assuntos Varuiados foram atribuídos erroneamente a algum dos autores do tema Direito.

Figura 47 – Atribuição de autoria



Fonte: o autor.

Como foi observado no trabalho que houve pouca confusão entre os temas Direito e Assuntos Variados, há possibilidade de trabalhos futuros que abordem tanto o tema como a autoria do documento, diminuindo a confusão verificada neste teste.