UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de Bauru

Luiz Carlos Felix Ribeiro

Análise de Sentimento Contextual em Diálogos Utilizando Aprendizado de Máquina

Bauru
2019

Luiz Carlos Felix Ribeiro

Análise de Sentimento Contextual em Diálogos Utilizando Aprendizado de Máquina

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, da Faculdade de Ciências da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de Bauru.

Financiadora: CAPES

Orientador: Prof. Dr. João Paulo Papa

Bauru
2019

Ribeiro, Luiz Carlos Felix.
Análise de Sentimento Contextual em Diálogos Utilizando Aprendizado de Máquina / Luiz Carlos Felix Ribeiro. – Bauru, 2019
113 f. : il., tabs.

Orientador: João Paulo Papa
Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Faculdade de Ciências, Bauru

1. Aprendizado de Máquina. 2. Processamento de Linguagem Natural. 3. Análise de Sentimento em Diálogos. 4. Análise de Sentimento Baseado em Contexto. I. Título.

R484a

Sistema de geração automática de fichas catalográficas da Unesp.
Biblioteca da Faculdade de Ciências, Bauru. Dados fornecidos pelo autor.

Essa ficha não pode ser modificada.

Luiz Carlos Felix Ribeiro

Análise de Sentimento Contextual em Diálogos Utilizando Aprendizado de Máquina

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, da Faculdade de Ciências da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de Bauru.

Financiadora: CAPES

Comissão Examinadora

Prof. Dr. João Paulo Papa
UNESP - Campus de Bauru
Orientador

Profa. Dra. Helena de Medeiros Caseli
UFSCar – Universidade Federal de São Carlos

Prof. Dr. Aparecido Nilceu Marana
UNESP - Campus de Bauru

Bauru
24 de abril de 2019

Ao meu avô, Milton Pereira (in memoriam)

Agradecimentos

Primeiramente, agradeço a Deus pelas inúmeras "coincidências" que acontecem nesta jornada, além de Ele sempre me dar forças para continuar trilhando o caminho que escolhi, mesmo nos momentos mais difíceis e quando mais tive dúvidas.

Agradeço aos meus familiares, especialmente aos meus pais, Carlos e Fátima, pelo apoio, paciência, compreensão e incentivo em minhas decisões ao longo da vida. Agradeço aos meus avós, Milton e Dijanira, por todo amor e carinho, pelos finais de semana que passamos juntos, viagens, passeios e quermesses. Sei que se não fosse por seus esforços, eu jamais sonharia em chegar aqui.

Agradeço também aos meus familiares de Bebedouro, meus tios João e Carmen, primas Débora e Dayane e primos Mateus, Leonardo e Luca, por todos churrascos de final de semana, feriados, finais de ano e natais que passamos juntos, momentos que se tornaram uma fonte de incontáveis momentos felizes e histórias para o resto da vida.

Agradeço ao professor João Paulo, meu orientador, sempre presente (mesmo que fisicamente distante), paciente, prestativo e fonte de inspiração pessoal e profissional desde os primeiros contatos durante minha graduação. Agradeço a ele pelas várias oportunidades, projetos, ensinamentos e conversas que tivemos ao longo deste período. Aproveito para estender este cumprimento aos companheiros de trabalho do Grupo Recogna, em especial ao Luisão, Clayton e Lélis pela amizade e ajuda em todos os momentos.

Agradeço à CAPES pela viabilização deste projeto de pesquisa através de seu financiamento, ainda mais em tempos onde o fomento à ciência tem se tornado cada vez mais escasso em detrimento de outros interesses, prejudicando assim toda a sociedade brasileira. O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Agradeço também a todos os professores, tanto da UNESP, como do Departamento de Computação, do Colégio Técnico Industrial e da *Queensland University of Technology*, que de alguma forma influenciaram minha visão sobre o mundo. Não posso deixar de mencionar os professores Humberto e Nilceu, minhas orientadores de iniciação científica, Adriana e Andréa e o professor Guido Zuccon, quem permitiu que eu vislumbrasse a área de Processamento de Linguagem Natural e percebesse que a ciência é feita de questionamentos.

"É isso que os nossos sonhos se tornaram?"
(Christopher John Cornell)

Resumo

A disponibilidade cada vez maior de dados em domínio textual tem motivado o desenvolvimento de técnicas baseadas em Processamento de Linguagem Natural para extrair informações estruturadas desse meio. Particularmente, técnicas de Análise de Sentimento permitem identificar a emoção presente em um fragmento de texto e podem ser utilizadas para diferentes fins, seja priorizar o atendimento de clientes insatisfeitos ou aferir o satisfação do interlocutor durante uma conversa. No que concerne ao uso desse tipo de técnica em diálogos, trabalhos na literatura mostram que considerar informações extraídas de mensagens antecessoras para classificar a atual leva a melhores resultados, seja para a identificação de interlocutores ou intenção das mensagens. Todavia, essa abordagem ainda não tem sido largamente empregada para a Análise de Sentimento e, quando utilizada, a mesma não alia a robustez dos *word embeddings*, técnica desenvolvida recentemente, com os rótulos preditos no passado, mas considera apenas o histórico de características extraídas anteriormente. O presente trabalho propõe o desenvolvimento de um modelo baseado em aprendizado de máquina para a Análise de Sentimento em conversas no domínio textual ao levar em consideração seu contexto. Essa fonte de informação pode ser explorada ao considerar rótulos de mensagens anteriores, suas características, a identidade dos interlocutores e como as palavras são combinadas em cada mensagem. Resultados experimentais mostram que estes aspectos permitem superar o estado-da-arte em quatro bases de dados diferentes.

Palavras-chave: Aprendizado de Máquina, Processamento de Linguagem Natural, Análise de Sentimento em diálogos, Análise de Sentimento baseado em contexto, Floresta de Caminhos Ótimos.

Abstract

The increasing availability of data in the textual domain has motivated the development of techniques based on Natural Language Processing to extract structured information from this domain. More specifically, Sentiment Analysis allows identifying the emotion present in a fragment of text and can be used with different goals, for instance, prioritizing the service of dissatisfied customers and assessing the interlocutor satisfaction in a conversation. Regarding the use of this type of technique in dialogues, works in the literature show that considering information extracted from previous messages when classifying the current sample leads to better results, either for identification of interlocutors or for message intent classification. However, this approach has not been widely adopted on Sentiment Analysis and when used it does not exploit the robustness of the recently developed word embeddings representation along with the labels predicted in the past but only the history of features previously extracted. The present work proposes the development of a machine learning model for Sentiment Analysis on textual conversations that considers their context. This source of information can be exploited by considering labels from previous messages and their features, the identity of the speakers, and how words are combined in each message. Experimental results show that these aspects allow outperforming the current state of the art on four different datasets.

Keywords: Machine Learning, Natural Language Processing, Dialogue Sentiment Analysis, context-based Sentiment Analysis, Optimum-Path Forest

Lista de Ilustrações

Lista de Tabelas

Lista de Abreviaturas e Siglas

ADAM	Adaptative Moment Estimation
AS	Análise de Sentimento
BiLSTM	*Bidirectional Long-Short Term Memory*
BIO	*Begin*, *Inside*, *Outside*
BOW	*Bag of Words*
BPTT	*Backpropagation Through Time*
CNN	*Convolutional Neural Network*
CRF	*Conditional Random Field*
DA	*Dialogue Act*
DSTC	*Dialog System Technology Challenges*
E-RNN	*Elman Recurrent Neural Network*
GRU	*Gated Recurrent Unit*
HCRC	*Human Comunication Research Centre*
HDBSCAN	*Hierarchical Density-Based Spatial Clustering of Applicatinos with Noise*
HMM	*Hidden Markov Model*
ICSI	*International Computer Science Institute*
J-RNN	*Jordan Recurrent Neural Networks*
LSTM	*Long-Short Term Memory*
MLP	*Multilayer Perceptron*
M-OPF	*Majority Optimum-Path Forest*
NER	*Named Entity Recognition*
NPS	*Naval Postgraduate School*
OPF	*Optimum-Path Forest*
OPT	*Optimum-Path Trees*
PLN	Processamento de Linguagem Natural
POS	*Part of Speech*
RNN	*Recurrent Neural Network*
SGD	*Stochastic Gradient Descent*
SVM	*Support Vector Machine*
SWDA	*Switchboard Dialogue Act Corpus*
TF-IDF	*Term Frequency-Inverse Document Frequency*

Sumário

1 Introdução

O crescente uso da Internet pela sociedade para os mais variados fins faz com que a quantidade de dados disponíveis para análise neste meio torne-se cada vez maior. Todavia, como as interações ali ocorrem majoritariamente de forma textual, torna-se necessário utilizar métodos apropriados para a extração de informações deste meio. A fim de atingir tal objetivo, são utilizadas técnicas de Processamento de Linguagem Natural (PLN) que, por sua vez, buscam converter a linguagem humana em uma representação formal tal que esta seja facilmente manipulável por computadores (COLLOBERT; WESTON, 2008).

Atualmente, diferentes *websites*, como Amazon.com e IMDb, disponibilizam espaços para que consumidores relatem suas experiências sobre um determinado produto ou serviço. A partir desses textos, torna-se possível utilizar técnicas de Análise de Sentimento (AS) com o objetivo de extrair informações relevantes. Neste sentido, Pang, Lee e Vaithyanathan (PANG; LEE; VAITHYANATHAN, 2002) determinam se uma dada análise de filme possui sentimento positivo ou negativo. De maneira mais detalhada, McAuley, Leskovec e Jurafsky (MCAULEY; LESKOVEC; JURAFSKY, 2012) identificam quais aspectos de um produto são discutidos em uma análise, além da forma como são avaliados. Tais técnicas podem ser utilizadas, por exemplo, na identificação dos atributos mais relevantes para um grupo de consumidores.

O estreitamento no relacionamento entre empresas e clientes por meio de redes sociais também pode ser visto como outra oportunidade para o emprego de técnicas de AS. Amora et al. (AMORA et al., 2017), por sua vez, priorizam o atendimento de clientes em uma central de relacionamento de acordo com os comentários que mencionam a empresa de forma mais severa no Twitter[1]. Inserido nesse contexto, existe uma crescente adoção de agentes conversacionais, ou *chatbots*, em canais como Facebook Messenger[2] e Whatsapp[3] com o objetivo de auxiliar humanos a realizarem algum tipo de tarefa, como compras ou agendamento de serviços. Técnicas de AS podem ser utilizadas neste tipo de interação a fim de identificar traços de sentimentos, ou emoções humanas, e o agente pode levar em consideração tais informações durante a elaboração de suas respostas, tornando-o mais empático e enriquecendo a interação entre as partes (SKOWRON, 2010). Essas características também podem ser empregadas para determinar o posicionamento do interlocutor humano sobre um determinado assunto, bem como inferir a eficácia de uma certa abordagem utilizada pelo agente a fim de atingir seu objetivo. Um *chatbot* para cobrança, por exemplo, pode utilizar o sentimento da conversa para inferir como um ser humano reage ao receber diferentes propostas de negociação. Ainda neste contexto, técnicas de AS também podem ser utilizadas em uma etapa após o atendimento

[1] <https://www.twitter.com>
[2] <https://www.messenger.com>
[3] <https://www.whatsapp.com>

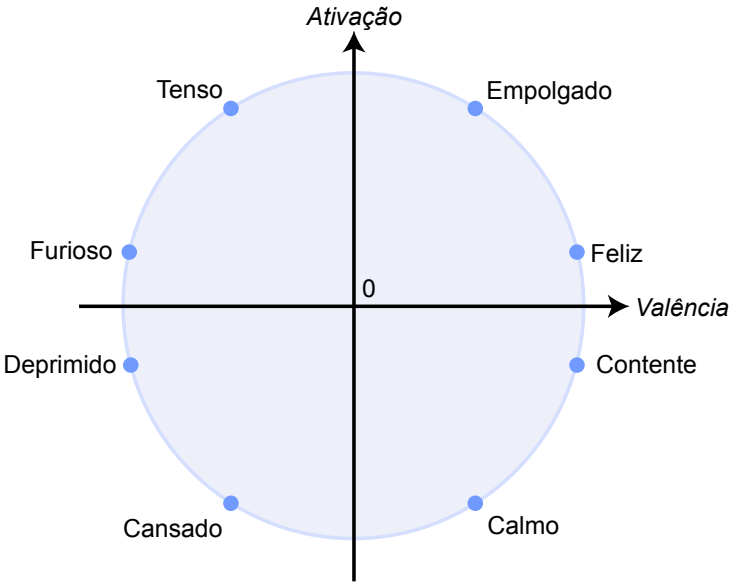
realizado por um humano com o objetivo de analisar as variações na polaridade das respostas emitidas pelo cliente no decorrer da conversa. Apesar das potenciais aplicações existentes, pouquíssimos trabalhos na literatura abordam o problema de AS em diálogos, de modo que maior foco é dado a análise de produtos, e mais recentemente, a mensagens do Twitter.

A fim de sistematizar o estudo em questão, é interessante definir uma conversa entre dois ou mais indivíduos como uma sequência de turnos (isto é, uma rodada de pergunta-resposta) alternados na cada qual um dos participantes produz uma ou mais mensagens de texto. Apesar desta dinâmica, de acordo com Lee e Dernoncourt (LEE; DERNONCOURT, 2016), a maioria dos estudos para a classificação de turnos em diálogos considera apenas a fala em questão. Estes mesmos autores obtiveram melhores resultados na classificação de turnos quanto à sua intenção ao considerar o histórico da conversa em conjunto com diferentes redes neurais. Conclusões semelhantes são traçadas por Ma, Xiao e Choi (MA; XIAO; CHOI, 2017) ao utilizarem Redes Neurais Convolucionais para a determinação da identidade de indivíduos em transcrições de conversas entre múltiplos participantes. Ademais, ao enxergar uma conversa como uma sequência de turnos, tal aspecto também pode ser explorado durante a classificação por meio do uso de Campos Aleatórios Condicionais (LAFFERTY; MCCALLUM; PEREIRA, 2002).

Apesar dos problemas de classificação de texto em tópicos e de AS se assemelharem em certos pontos, a segunda tarefa possui diversas particularidades, já que no primeiro caso é comum poder identificar o assunto abordado pelo documento com base em palavras-chave, ao passo que a polaridade pode ser inserida em um texto de forma sutil (PANG; LEE; VAITHYANATHAN, 2002). Recursos semânticos como a negação, ironia e a presença de diferentes sentimentos na mesma sentença também dificultam esta tarefa.

Em relação à sua abordagem, a Análise de Sentimento pode ser vista tanto como um problema de classificação como de regressão. No primeiro caso é possível rotular amostras apenas como positivas ou negativas (PANG; LEE; VAITHYANATHAN, 2002), além de ser possível considerar uma classe neutra adicional (KOPPEL; SCHLER, 2006). O problema também pode ser expandido ao considerar mais rótulos, como as seis emoções básicas de Ekman (EKMAN, 1992): raiva, aversão, medo, felicidade, tristeza e susto, conforme explorado por Yasmina et al. (YASMINA; HAJAR; HASSANA, 2016). Para o problema de regressão, é possível adotar uma escala com várias intensidades de sentimento entre os extremos negativo e positivo, similar a classificação de análises de filmes entre uma e cinco estrelas (PANG; LEE, 2005). Por outro lado, o sentimento também pode ser visto como uma combinação de valores numéricos contínuos em diferentes dimensões, como no espaço circular de valência-ativação ilustrado na Figura 1, onde o primeiro eixo representa diferentes valores de satisfação e o segundo, graus de empolgação (RUSSELL, 1980).

Figura 1 – Espaço de valência-ativação com alguns sentimentos correspondentes à combinação de valores de ambos eixos destacados.



Fonte: Elaborado pelo autor.

1.1 Hipótese

Tendo em vista os ganhos obtidos ao abordar outros problemas de classificação em PLN de forma sequencial, como AS a nível de aspectos e identificação de entidades nomeadas, espera-se que endereçar o problema de AS em diálogos da mesma forma também leve a melhores resultados, ao observar que uma conversa é formada por uma sequência de turnos. Além do histórico de mensagens daquela interação e rótulos preditos anteriormente, outras informações contextuais podem ser consideradas, como a identidade dos interlocutores, ao supor que cada indivíduo possui um perfil inclinado a utilizar certas emoções com maior frequência em suas falas.

Já em um nível hierárquico menor, é possível explorar características contextuais de cada turno. Isso pode ser feito ao considerar que o significado de uma sentença depende da forma como suas palavras são combinadas e da influência de umas sobre as outras. Tal dinâmica pode ser modelada ao dar mais relevância para algumas palavras em detrimento de outras dado sua vizinhança.

De maneira geral, quatro fontes de informação contextual podem ser exploradas: (i) o histórico de mensagens do diálogo; (ii) a sequência de sentimentos emitidos para cada turno; (iii) a identidade dos interlocutores envolvidos; e (iv) a forma como as palavras são combinadas em cada fala. Essas decisões se baseiam na hipótese de que uma conversa evolui de forma incremental e em função de interações passadas. Sob estas condições, é esperado que o sentimento transmitido por cada um de seus interlocutores dependa do que já foi dito anteriormente, além de transitar gradativamente para outras emoções.

1.2 Objetivos

1.2.1 Objetivo Geral

Desenvolver uma técnica baseada em aprendizado de máquina para realizar a classificação de turnos de diálogos em domínio textual quanto ao seu sentimento ao incorporar informação de contexto da conversa, a qual é representada por rótulos emitidos no passado, turnos anteriores e também por seus interlocutores.

1.2.2 Objetivos Específicos

a) Estudar as principais técnicas utilizadas para a tarefa de Análise de Sentimento no domínio textual com o objetivo de entender seu funcionamento, vantagens e desvantagens;

b) Avaliar outras formas de incorporar o contexto do diálogo no modelo, além dos turnos anteriores da conversa;

c) Identificar bases de dados utilizadas para a AS em diálogos, tendo em vista que a maioria dos trabalhos desenvolvidos considera análises de produtos ou, mais recentemente, mensagens em redes sociais;

d) Propor e implementar um classificador baseado em aprendizado de máquina que permita realizar a Análise de Sentimento Contextual;

e) Analisar, por meio de medidas estatísticas, os resultados obtidos ao aplicar o modelo desenvolvido nas bases de dados identificadas com o objetivo de gerar resultados claros, reprodutíveis e comparáveis com outras abordagens que porventura sejam desenvolvidos no futuro;

f) Comparar os resultados obtidos com as técnicas já existentes para AS em diálogos.

1.3 Estrutura da Dissertação

Os demais capítulos da presente dissertação estão organizados da seguinte maneira:

- Capítulo 2: Apresenta uma revisão sobre as abordagens mais comumente utilizadas para a Análise de Sentimento;
- Capítulo 3: Introduz as principais redes neurais utilizadas para a realização de AS baseada em aprendizado de máquina de acordo com a literatura;
- Capítulo 4: Apresenta os Campos Aleatórios Condicionais Linearmente Encadeados, comumente designados apenas como Campos Aleatórios Condicionais e motiva sua utilização na tarefa de classificação sequencial;

- Capítulo 5: Revisa métodos que levam em consideração o contexto para a determinação de descritores de segmentos de textos, bem como técnicas para a classificação contextual de documentos e de turnos de diálogos quanto ao seu sentimento e intenção;
- Capítulo 5.3: Apresenta a metodologia utilizada para treinar e avaliar os modelos propostos, além de discutir características sobre a base de dados e as técnicas de pré-processamento empregadas;
- Capítulo 7: Discute os resultados experimentais obtidos, além de analisar as principais diferenças e vantagens de cada variante do modelo proposto;
- Capítulo 8: Sintetiza o conteúdo apresentado ao contextualizar a proposta, embasada em técnicas de aprendizado de máquina, com a tarefa de Análise de Sentimento. Com base nos experimentos conduzidos é possível concluir que a hipótese apresentada na dissertação é válida, tendo em vista que o contexto, representado pelo histórico de mensagens e identidade dos interlocutores, leva a melhores resultados de classificação;
- Apêndice A: Discute o trabalho elaborado durante a execução do projeto de mestrado para a classificação não-supervisionada de turnos de um diálogo quanto à sua intenção e apresentado no 31º Simpósio Brasileiro em Computação Gráfica e Processamento de Imagens (SIBGRAPI).

2 Abordagens para Análise de Sentimento

Técnicas de Análise de Sentimento, por vezes referida como mineração de opiniões ou análise de emoções, têm por objetivo analisar opiniões, sentimentos e avaliações a respeito de produtos, tópicos e serviços. Segundo Liu (LIU, 2012) os métodos de análise podem ser divididos em três grupos de acordo com o escopo do texto considerado para a classificação:

- **A nível de documento:** o texto é examinado de forma global ao assumir que o mesmo aborda apenas um tema para o qual deseja-se atribuir apenas um rótulo, como geralmente ocorre com a análise de produtos.
- **A nível de sentenças:** cada sentença do texto é classificada individualmente. Nessa configuração é comum a adoção de uma classe neutra para contemplar frases objetivas, as quais não possuem nenhuma informação de sentimento. Outra possibilidade consiste inicialmente em determinar se a amostra sob análise é subjetiva (possui informação de sentimento) ou é objetiva.
- **A nível de aspectos e entidades:** técnicas desse tipo buscam identificar o relacionamento entre o sentimento emitido e o aspecto sob julgamento a fim obter informações sobre o que mais agrada e aborrece o autor do texto. Apesar de ser o método de avaliação mais complexo, o mesmo permite extrair informações estruturadas do texto original, as quais podem ser utilizadas para avaliações mais detalhadas.

Em uma análise realizada com base nos trabalhos sobre AS entre os anos de 2000 e 2015, Piryani, Madhavi e Singh (PIRYANI; MADHAVI; SINGH, 2017) categorizam os métodos de AS de acordo com a maneira como as informações são extraídas do texto e utilizadas para a classificação, as quais podem ser baseadas em dicionário (ou palavras-chave), aprendizado de máquina ou até mesmo híbridas, combinando ambas as metodologias. Cada uma destas abordagens é descrita nas próximas seções. Ainda sobre este aspecto, os autores constatam que a maioria dos estudos desenvolvidos recentemente, mais especificamente 67%, utilizam técnicas baseadas em aprendizado de máquina, ao passo que 27% dos trabalhos catalogados empregam dicionários, enquanto que apenas 6% dos trabalhos analisados utilizam métodos híbridos.

2.1 Técnicas Baseadas em Dicionário

As técnicas baseadas em dicionário, também denominado léxico ou conhecimento prévio, adotam uma base de dados com anotações que mapeiam palavras ao seu sentimento relacionado.

A partir dessa informação é possível determinar com qual frequência cada sentimento é utilizado em um texto de interesse e empregar uma heurística para a classificação ou extração de informações mais detalhadas. A estrutura da sentença também pode ser explorada nesta etapa ao considerar sua árvore sintática, bem como a classe gramatical de suas palavras, denominada informação morfossintática – do inglês, *Part of Speech* (POS). Normalmente, substantivos, verbos, adjetivos e advérbios são considerados, já que estas são as categorias de palavras mais prováveis a serem empregadas para introduzir emoção no texto (SAILUNAZ et al., 2018).

Neste tipo de abordagem, o léxico desempenha um papel fundamental, tendo em vista que o mesmo é utilizado tanto para identificar quais palavras possuem informação subjetiva, bem como seu sentimento específico (LU et al., 2011). Apesar disso, não existe uma base genérica capaz de desempenhar bem esta tarefa em todos os casos, já que a mesma palavra pode corresponder a diferentes sentimentos de acordo com o domínio e aspecto sob análise (PANG; LEE et al., 2008). O adjetivo *"imprevisível"*, por exemplo, representa um atributo positivo ao descrever um filme, mas possui sentido contrário quando se refere ao funcionamento de um eletrônico.

A construção do dicionário pode ser realizada de forma manual, processo laborioso e que exige conhecimento prévio do domínio em questão, ou utilizar técnicas semi-supervisionadas. Neste caso, palavras iniciais, denominadas "sementes", para as quais seu sentimento correspondente é conhecido, são comparadas com outros vocábulos presentes em um conjunto de documentos ao levar em consideração seu contexto de co-ocorrência e a partir dessa informação é possível determinar o sentimento da nova palavra e adicioná-la ao léxico (TABOADA et al., 2011). As relações de sinônimo e antônimo, as quais podem ser obtidas em um tipo específico de lista de palavras, denominado tesauro, também podem ser exploradas nessa etapa. Lu et al. (LU et al., 2011) apresentam um algoritmo que permite a criação de um dicionário dependente de domínio e de contexto, ou seja, é possível aprender diferentes anotações de sentimento para a mesma palavra de acordo com o aspecto sob análise. Outra possibilidade consiste no uso de dicionários previamente compilados, como WordNet–Affect (STRAPPARAVA; VALITUTTI et al., 2004) ou SentiWordNet (BACCIANELLA; ESULI; SEBASTIANI, 2010).

Com relação a estudos desenvolvidos, Turney (TURNEY, 2002) classifica análises de produtos como positivas ou negativas ao considerar a média do sentimento obtido a partir de cada sentença do texto. Para isso, são extraídas informações de POS das palavras que compõem a sentença, e um conjunto de regras é utilizado com o objetivo de identificar o aspecto sob análise e seu parecer correspondente, o qual carrega informação de polaridade e pode ser classificado ao considerar seu contexto. O sentimento da frase de interesse é por sua vez computado com base em uma medida derivada da co-ocorrência entre a amostra e os termos *"excellent"* (excelente) e *"poor"* (ruim), os quais são obtidos a partir de consultas no buscador AltaVista[4]. De acordo com o autor, tais termos são empregados devido à sua frequência em

[4] Atualmente este sistema foi incorporado pelo buscador Yahoo! <https://search.yahoo.com/>.

pareceres positivos e negativos, respectivamente. Note que neste caso o dicionário não é fixo, mas sim derivado dos documentos catalogados pelo motor de busca.

Nasukawa e Yi (NASUKAWA; YI, 2003) realizam a análise de sentimento em sentenças a nível de aspecto e entidades. Os autores procedem de maneira similar ao trabalho anterior para a identificação do aspecto sob análise bem como seu parecer. Nesse caso, o sentimento emitido, o qual é utilizado para rotular a sentença, é extraído ao buscar a palavra de interesse em um dicionário construído manualmente. Apesar do modelo ser capaz de reconhecer negações simples, o mesmo falha ao identificar este recurso de linguagem em construções mais complexas. Já Bobicev, Sokolova, e Oakes (BOBICEV; SOKOLOVA; OAKES, 2015) classificam postagens em fóruns médicos de acordo com seu sentimento ao utilizar um léxico especialmente elaborado para este domínio.

A dificuldade em contemplar negações evidencia a limitação das abordagens baseadas em dicionário para incorporar recursos de linguagem na etapa de classificação como negação, intensificação, comparação, conjunção, dentre outros, os quais consistem no princípio de composicionalidade semântica. Sob essa perspectiva, o significado de uma expressão longa, seja uma sentença ou documento, depende do significado de cada um de seus termos constituintes, da forma como são combinados e de acordo com seu papel sintático (LIU, 2012). Na sentença *"Esse filme não é muito ruim"*, por exemplo, o princípio composicional surge por meio das palavras *"ruim"*, a qual confere o aspecto negativo à análise, *"muito"*, que amplifica esse sentimento e *"não"*, responsável por atenuar a desqualificação do produto.

Taboada et al. (TABOADA et al., 2011) descrevem o processo de elaboração de um conjunto de dicionários, bem como regras que abordam diferentes tipos de construções semânticas para a tarefa de AS. Inicialmente, os autores definem que cada palavra, frase, ou texto, possui uma orientação semântica, a qual é composta por sua polaridade, neste caso, positiva ou negativa e intensidade. Este último atributo varia no intervalo inteiro $[-5, 5]$, também definido pelos autores, de modo que as partes positiva e negativa estão relacionadas aos sentimentos correspondentes. Segmentos de texto com polaridade zero são ditos neutros e, consequentemente, desconsiderados.

A princípio, é criado um dicionário para cada classe gramatical considerada: verbos, adjetivos e advérbios, cada qual com sua lista de *stop words*, ou seja, palavras irrelevantes e que não devem ser consideradas. A ideia motivadora por trás desta divisão é a possibilidade do mesmo vocábulo ter diferentes orientações semânticas de acordo com sua classe gramatical. Os autores argumentam que a palavra em inglês *"novel"*, por exemplo, é um adjetivo positivo, ao passo que a mesma palavra torna-se neutra quando empregada como um substantivo. Este processo de desambiguação, o qual leva em consideração apenas a informação morfossintática, fica ainda mais evidente ao traduzir a palavra para o português, já que a mesma corresponde a *"inédito"*, no primeiro caso e *"romance"* no segundo. Formados os dicionários a partir

de um conjunto de documentos, denominado *corpus*[5], os termos encontrados são rotuladas manualmente de acordo com sua orientação semântica com base em seu significado mais comum. Especificamente, informações de contexto não são consideradas neste processo. A seguir, as anotações realizadas são revisadas por um comitê formado por três indivíduos que não participam da tarefa anterior. É importante salientar que os dicionários podem contemplar sequências de palavras, e não apenas termos individuais.

Após esta etapa inicial, as regras utilizadas para a análise de sentimento dos textos de interesse são estabelecidas para determinar a orientação semântica de uma amostra a partir de suas palavras constituintes e das construções semânticas utilizadas. Quatro aspectos principais são considerados pelos autores: intensificação, negação, *irrealis* e características a nível do texto como um todo.

De acordo com Quirk et al. (QUIRK et al., 1985), o processo de intensificação pode ocorrer por meio do uso de palavras que amplificam ou degradam a intensidade das palavras em sua vizinhança, por exemplo, *"bastante"* e *"pouco"*, respectivamente. Dado um fragmento de texto para análise, primeiramente são identificados quais termos constituintes carregam informação de sentimento ao procurá-los nos dicionários compilados. Para cada um, é realizada uma busca em sua vizinhança por palavras intensificadoras, as quais estão relacionados a uma margem de variação de intensidade percentual, que é aplicada de forma recursiva ao termo original a partir da palavra mais próxima. Com o objetivo de ilustrar este processo, considere a frase *"Este filme é realmente muito bom"*, onde a palavra *"bom"* tem intensidade $+3$ e os modificadores *"muito"* e *"realmente"*, por sua vez, possuem margens de variação de $1,15$ e $1,25$, respectivamente[6] . Utilizando a estratégia proposta, a intensidade final da sentença será $(1,15 \cdot (1,25 \cdot (+3))) = +4,3$.

Já as regras para a identificação de negações dependem da POS do termo sob análise. A grosso modo, é realizada uma busca nos antecessores da palavra atual até que um delimitador de oração, como uma conjunção ou pontuação seja encontrado, ou enquanto os termos percorridos pertencerem a uma lista de palavras válidas. Se alguma negação for encontrada, ao invés de simplesmente inverter a intensidade do segmento sob análise, é adicionada uma constante ao valor de intensidade atual com sinal oposto. Como exemplo, considere a frase *"Ela não é uma pessoa incrível"* com intensidade $+1$, obtida ao somar -4 (constante de negação) à intensidade $+5$, advinda do adjetivo *"incrível"*. É importante mencionar que outras estratégias comumente utilizadas para lidar com a negação, além da atenuação empregada por estes autores, incluem ignorar ou inverter a polaridade da palavra subjetiva.

Já a *irrealis* consiste em um recurso que permite ao autor do texto expressar uma opinião não factual sobre algo, por exemplo, uma expectativa que não foi atendida. Em português isso pode ocorrer através de tempos verbais no modo subjuntivo, como na frase *"Se o filme* `fosse`

[5] A título de curiosidade, no plural, *corpora*.
[6] O sinal positivo enfatiza o sentimento relacionado à palavra.

bom eu não teria dormido". Apesar deste modo não ser comumente empregado em inglês, conforme argumentado pelos autores, é apresentada uma série de regras específicas para o idioma em questão.

Algumas medidas são empregadas para levar em consideração características gerais da amostra sob análise, como a possibilidade de ponderar a intensidade das palavras acordo com sua posição no texto. A fim de penalizar a repetição de termos, a contribuição da intensidade de polaridade de um vocábulo para a amostra é dividida pelo número de vezes que o mesmo termo já foi utilizado no passado. Adicionalmente, amostras com orientação semântica negativa têm sua intensidade aumentada em 50%, a fim de compensar a tendência humana de favorecer termos positivos, de acordo com os autores.

Uma avaliação empírica também conduzida por Taboada et al. (TABOADA et al., 2011) em três bases de dados não observadas durante a elaboração dos dicionários mostrou que as regras desenvolvidas mais importantes são, respectivamente, análise de negação e intensificação, aplicadas em conjunto, seguidas pelo fator de correção para sentenças negativas e penalização de termos repetidos.

Por fim, apesar de factível, é possível concluir que abordagens baseadas em dicionário demandam esforço considerável tanto para a criação dos próprios dicionários, como no que diz respeito à elaboração das regras utilizadas na identificação do aspecto e de seu parecer correspondente, as quais devem ser compreensivas a fim de identificar a maior quantidade possível de palavras de interesse (YASMINA; HAJAR; HASSANA, 2016). Todavia, utilizar regras criadas com base em um conjunto de textos torna o classificador dependente do domínio em questão, de modo que este apresentará bom desempenho apenas para amostras que possuam o estilo de escrita esperado. De maneira similar, a ausência de palavras pertinentes ao domínio no léxico utilizado faz com que informações que poderiam ser exploradas durante a classificação sejam desprezadas. Este último aspecto se mostra bastante relevante no que diz respeito à análise de textos extraídos de redes sociais, característico pelo tom informal, além da presença de gírias, erros de digitação e vocabulário em constante evolução (HU et al., 2013).

2.2 Técnicas Baseadas em Aprendizado de Máquina

Essa abordagem normalmente envolve uma etapa preliminar de extração de características, na qual as amostras de interesse, sejam estas sentenças, ou mesmo documentos completos, passam a ser representadas por um vetor descritor que é apresentado a algum método de aprendizado de máquina. Em relação a estes, é possível treinar um classificador para aprender a rotular novas amostras de acordo com seu sentimento, mas também pode ser relevante gerar um modelo de regressão, onde dado um fragmento de texto, como uma análise de produto ou uma mensagem de texto, deseja-se atribuir ao mesmo uma nota numérica dentro de uma escala pré-definida, a qual reflete um contínuo de sentimentos que varia entre os extremos

negativo e positivo (PANG; LEE, 2005; ROSENTHAL; FARRA; NAKOV, 2017). Por outro lado, ao saber de antemão o sentimento da amostra também é possível treinar um regressor que determine sua intensidade (MOHAMMAD; BRAVO-MARQUEZ, 2017).

Em relação às representações, a abordagem mais simples é a bolsa de palavras – do inglês *bag of words* (BoW). Neste regime, inicialmente é construído um vocabulário $\mathcal{V}$ com as palavras relevantes para o problema a partir do *corpus*, o qual é formado pelas amostras de treinamento. A seguir, cada palavra $w \in \mathcal{V}$ é representada por um vetor $\boldsymbol{v}_w \in \mathbb{R}^{|\mathcal{V}|}$, onde $|\mathcal{V}|$ é o tamanho do vocabulário, de modo que cada coordenada (ou dimensão) deste vetor está relacionada a um único termo do vocabulário. A palavra *"tese"*, por exemplo, é descrita pelo vetor $\boldsymbol{v}_{tese}$, ilustrado na Figura 2. Tal representação é comumente denominada *one-hot encoding*, já que apenas um elemento do vetor é "ligado" para descrever cada palavra. A representação em bolsa de palavras também pode ser estendida para descrever documentos, independente de seu tamanho, ao somar os vetores que descrevem cada uma de suas palavras constituintes. Uma outra forma de enxergar esta estratégia consiste atribuir a cada coordenada do vetor descritor a frequência de seu termo correspondente no documento de interesse.

Figura 2 – Representação da palavra *"tese"* como bolsa de palavras.

$$v_{tese} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \sim \begin{bmatrix} \text{a} \\ \text{aba} \\ \vdots \\ \text{tese} \\ \vdots \\ \text{zebra} \end{bmatrix} \in \mathbb{R}^{|\mathcal{V}|}$$

Fonte: Elaborado pelo autor.

Adicionalmente, ao invés de empregar apenas palavras na construção do vocabulário (representação em unigramas), duplas (bigramas) ou sequências de n palavras (n-gramas) também podem ser consideradas. Esta estratégia permite preservar, mesmo que parcialmente, a ordem em que os vocábulos surgem no texto, informação importante a fim de considerar construções semânticas como negação e comparação, apesar de aumentar significativamente o tamanho de $\mathcal{V}$. Pang, Lee e Vaithyanathan (PANG; LEE; VAITHYANATHAN, 2002) utilizam essa estratégia em conjunto com diferentes métodos supervisionados (Classificador Bayesiano, Máxima Entropia e Máquina de Vetores de Suporte) para a rotulação de análises de filmes quanto ao seu sentimento.

Na representação em Bolsa de Palavras cada vocábulo da amostra possui a mesma importância ao gerar seu vetor descritor, característica que pode ser alterada ao utilizar a representação denominada TF-IDF – do inglês, *Term Frequency–Inverse Document Frequency*. Neste caso, cada elemento do vetor é ponderado de acordo com sua frequência na amostra

de interesse e no *corpus* de maneira geral (ROBERTSON, 2004). Muito embora esta seja uma abordagem comumente utilizada para descrever documentos na área de Recuperação de Informação, a mesma também acaba sendo utilizada não apenas para a tarefa de Análise de Sentimento, mas na área de Processamento de Linguagem Natural de maneira geral.

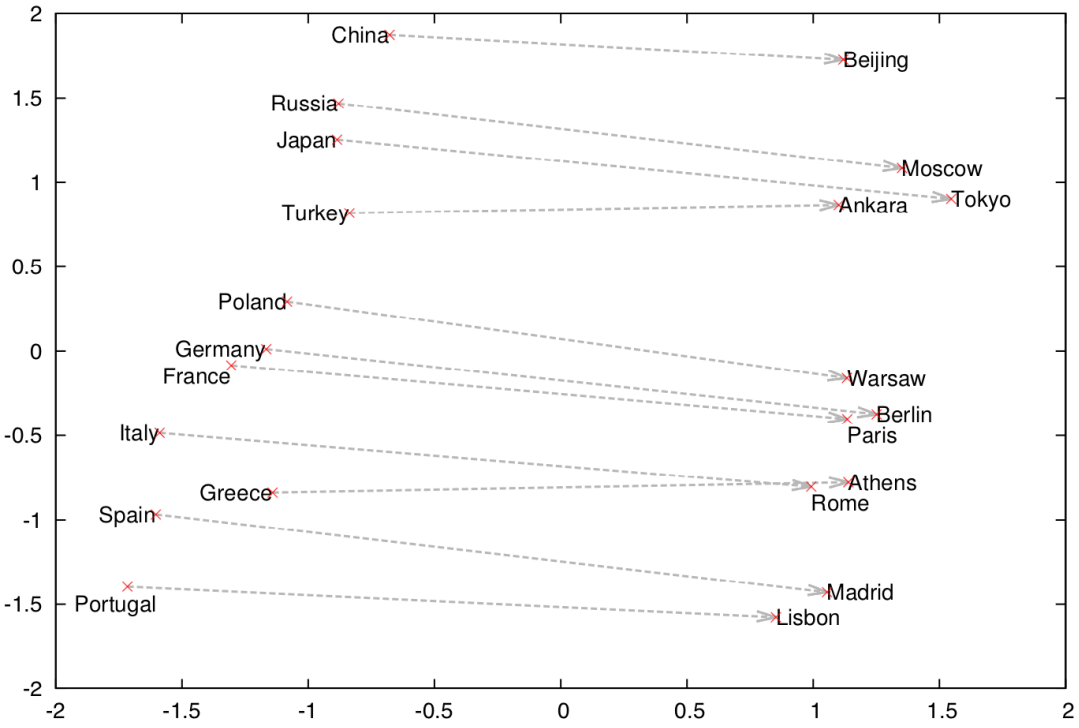
Apesar das possibilidades apresentadas, os descritores baseados em BoW fazem com que a informação sintática e semântica dos termos originais sejam perdidas, tendo em vista que cada vocábulo w é simplesmente mapeado para uma dimensão diferente de um vetor $\boldsymbol{v}_w$ sem qualquer critério definido. Logo, apesar da palavra *"lar"* ser semanticamente mais similar a *"casa"* que a *"navio"*, não existe nenhuma característica compartilhada entre $\boldsymbol{v}_{lar}$ e $\boldsymbol{v}_{casa}$ nesta situação. Além disso, ambos vetores são equidistantes de $\boldsymbol{v}_{navio}$, apesar deste último termo não estar relacionado de maneira alguma aos outros dois[7]. Outra dificuldade diz respeito ao uso de vocabulários extensos, que se traduz em vetores cada vez maiores e mais esparsos, dificultando sua manipulação e armazenamento (MIKOLOV et al., 2013a). Uma forma de contornar tais problemas consiste em utilizar *word embeddings*, os quais são normalmente obtidos a partir de modelos treinados de maneira não-supervisionada. A utilização desta abordagem por meio de técnicas desenvolvidas recentemente como word2vec (MIKOLOV et al., 2013b), fastText (BOJANOWSKI et al., 2016) e GloVe (PENNINGTON; SOCHER; MANNING, 2014), dentre outras, permite o treinamento eficiente de vetores densos, os quais eliminam o problema de esparsidade, além de preservar, mesmo que parcialmente, informações sintáticas e semânticas das palavras. Este último aspecto é devido aos modelos empregados buscarem mapear palavras que co-ocorrem com frequência no conjunto de treinamento em vetores com maior similaridade de cosseno, apresentada na Equação 1:

$$\text{sim}\left(\boldsymbol{v}_a, \boldsymbol{v}_b\right) = \cos\left(\boldsymbol{v}_a, \boldsymbol{v}_b\right) = \frac{\boldsymbol{v}_a\, \boldsymbol{v}_b}{||\boldsymbol{v}_a||\ ||\boldsymbol{v}_b||}. \tag{1}$$

Dados pares de palavras com seu grau de similaridade sintática ou semântica atribuído por juízes humanos, a retenção de informação nos vetores correspondentes é avaliada ao comparar sua similaridade de cosseno com a primeira métrica por meio do Coeficiente de Correlação de Spearman ou de Pearson. Mikolov et al. (MIKOLOV et al., 2013b) também mostram que esta representação é capaz de aprender relações entre palavras de forma implícita ao projetá-las em um espaço bidimensional, conforme ilustrado na Figura 3, onde é possível observar que os segmentos de reta que conectam as diferentes capitais a seus países correspondentes possuem inclinações próximas, e consequentemente, valores de cosseno similares. É importante salientar que a função cosseno é par, ou seja, ângulos com a mesma inclinação nos sentidos horário e anti-horário levam a valores iguais. Adicionalmente, note que cada conceito forma um grupo isolado de palavras.

[7] Seria razoável esperar que palavras semanticamente relacionadas estivessem mais próximas no espaço vetorial.

Figura 3 – Projeção de vetores dos *word embeddings* em um espaço bidimensional por meio da Análise de Componentes Principais.



Fonte: Mikolov et al. (MIKOLOV et al., 2013b)

Abordagens baseadas em *word embeddings* têm se tornado a principal forma de representar vocábulos na área de Processamento de Linguagem Natural, fenômeno que pode ser atribuído à diversos aspectos, seja a sua robustez, conforme apresentado anteriormente, à disponibilidade de modelos pré-treinados em diversos idiomas (GRAVE et al., 2018)[8], inclusive em português (HARTMANN et al., 2017)[9], além da facilidade em gerar novos modelos, tendo em vista que tratam-se de técnicas não-supervisionadas e que consequentemente não incorrem o alto custo de rotulação das amostras. Adicionalmente, a comunidade tem dedicado esforço considerável buscando investigar e explorar tais técnicas.

Word embeddings, por sua vez, consistem apenas em uma forma de representação robusta de palavras, restando ainda abordar o problema principal, o qual consiste no treinamento de algum modelo de aprendizado de máquina para a realização de uma tarefa de interesse. Neste caso, tais vetores são normalmente incorporados dentro de modelos maiores como uma etapa preliminar para a extração de características das amostras, processo que pode ocorrer de inúmeras formas, sendo as mais comuns adotar uma representação pré-treinada ou aprendê-la em conjunto com a técnica de interesse, a qual acaba por exigir um conjunto de treinamento maior para que vetores robustos sejam obtidos. Estas e outras alternativas, como

[8] Modelos pré-treinados do FastText em 157 idiomas podem ser obtidos em <https://fasttext.cc/docs/en/crawl-vectors.html>.

[9] Modelos em português podem ser obtidos em <http://www.nilc.icmc.usp.br/nilc/index.php/repositorio-de-word-embeddings-do-nilc>.

o ajuste fino dos vetores pré-treinados durante a aprendizagem do modelo são estudadas por Kim (KIM, 2014) nas tarefas de classificação de amostras quanto a seu sentimento ou seu assunto principal. Goldberg (GOLDBERG, 2017) apresenta outras estratégias mais elaboradas para realizar o ajuste fino de representações aprendidas previamente com o objetivo de preservar as características originais dos *word embeddings*.

Como cada vocábulo agora corresponde a um vetor denso e não apenas a uma de suas coordenadas, é necessário obter uma representação da amostra de interesse, como uma sentença, a partir dessas informações. Matematicamente, cada palavra w é representada por um *word embedding* $\boldsymbol{v}_w \in \mathbb{R}^{d_{emb}}$, onde d_{emb} é o tamanho do vetor descritor, com $d_{emb} \ll |\mathcal{V}|$, de modo que enquanto um vocabulário é potencialmente composto por milhares de elementos, *word embeddings* têm, normalmente, entre 50 e 300 dimensões.

Joulin et al. (JOULIN et al., 2016) utilizam uma rede neural simples para classificar sentenças quanto a seu sentimento, as quais são descritas por meio da média dos *word embeddings* de seus vocábulos constituintes. Apesar da facilidade em computar tal representação, a mesma não leva em consideração a ordem das palavras, aspecto relevante para este tipo de tarefa, conforme discutido anteriormente. Arora, Liang e Ma (ARORA; LIANG; MA, 2017) chegam a esta mesma conclusão a respeito da importância em preservar a ordem das palavras ao utilizar uma abordagem similar, onde a média dos *word embeddings* da sentença é ponderada de acordo com probabilidade de ocorrência de cada termo.

Vale a pena mencionar que o sucesso de técnicas para obter *word embeddings* tem motivado o desenvolvimento recente de trabalhos que investigam formas de estender tais abordagens para gerar representações de sentenças também de forma não-supervisionada. Le e Mikolov (LE; MIKOLOV, 2014) generalizam a idéia subjacente do método word2vec para representar segmentos de texto de comprimento variável em um método denominado vetores de parágrafo – do inglês *paragraph vectors*, outros trabalhos relevantes incluem Skip-Thought (KIROS et al., 2015) e sent2vec (PAGLIARDINI; GUPTA; JAGGI, 2017).

Com base na crescente variedade de métodos para representação de palavras e de segmentos de textos, é fácil concluir que essa tarefa não consiste apenas em um subproblema para a tarefa de Análise de Sentimento, ou classificação de textos de maneira geral, mas sim uma sub-área dentro do domínio de PLN. Consequentemente, uma discussão mais ampla sobre descritores de texto está fora do escopo deste trabalho, tendo em vista que o tema é bastante amplo e está em constante evolução. Todavia, isso não impacta a compreensão da maior parte da literatura sobre Análise de Sentimento baseada em aprendizado de máquina, já que a maioria dos trabalhos desenvolvidos utilizam os *word embeddings* discutidos anteriormente.

Métodos de classificação mais complexos, além daqueles apresentados até então, também dependem de algum tipo específico de rede neural, aspecto que se torna ainda mais evidente ao levar em consideração sequências de amostras, um dos tópicos centrais da presente dissertação de mestrado. A fim de facilitar a análise destes modelos sua discussão é adiada

para o Capítulo 5, após a apresentação das redes neurais de interesse no Capítulo 3. Apesar de ser possível utilizar outras técnicas de aprendizado de máquina, como Máquinas de Vetores de Suporte – do inglês *Support Vector Machines* (SVM), ou classificadores Bayesianos, Moraes, Valiati e Neto (MORAES; VALIATI; NETO, 2013) mostram empiricamente que redes neurais apresentam resultados superiores na tarefa de Análise de Sentimento.

Por fim, as vantagens do uso da abordagem baseada em aprendizado de máquina se traduzem na independência de um léxico e do desenvolvimento de regras para a identificação de palavras de interesse, já que o modelo aprende a identificar termos relevantes para a classificação de sentimento durante a etapa de treinamento.

2.3 Técnicas Híbridas

Os métodos dessa categoria combinam aspectos das técnicas baseadas em dicionário e aprendizado de máquina. Normalmente, regras são empregadas na identificação de características relevantes para a classificação, as quais são utilizados por um método supervisionado para a rotulação dos documentos de interesse.

Spertus (SPERTUS, 1997) utiliza esse tipo de abordagem a fim de identificar comentários nocivos em grupos de discussão na Internet ao treinar uma árvore de decisões baseada em vetores de características, os quais são obtidos ao aplicar um conjunto de regras criadas manualmente para identificar aspectos da sentença em questão, como a presença de insultos, elogios e tom de condolescência, dentre outros. Já Seol, Kim e Kim (SEOL; KIM; KIM, 2008) utilizam um método de classificação paralelo onde a abordagem baseada em dicionário é utilizada se a frase de interesse possuir palavras com informação subjetiva; caso contrário, uma rede neural baseada em conhecimento é empregada. Por fim, Mohammad e Bravo-Marquez (MOHAMMAD; BRAVO-MARQUEZ, 2017) representam mensagens no Twitter (*tweets*) não apenas por meio de *word embeddings*, mas também ao utilizar uma combinação de diferentes léxicos para treinar um modelo de regressão de sentimento baseado em Máquinas de Vetores de Suporte.

Enquanto abordagens híbridas trazem os benefícios fornecidos pelas técnicas baseadas em aprendizado de máquina, os aspectos relacionados aos métodos baseados em léxico, como adoção de dicionário e desenvolvimento de regras para a extração de informações, devem ser endereçados propriamente.

2.4 Considerações Finais

Por fim, este capítulo apresentou as principais abordagens para a realização da tarefa de AS. Na presente dissertação de mestrado visamos desenvolver um método baseado em aprendizado de máquina, tendo em vista sua popularização para a realização da tarefa em

questão. Ademais, dada uma base de dados representativa de um novo domínio de interesse, torna-se relativamente fácil treinar novos modelos após estabelecida uma arquitetura relevante.

3 Redes Neurais

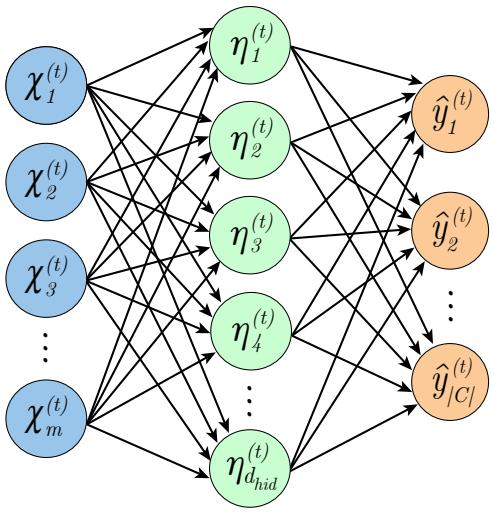
Este capítulo apresenta as redes neurais mais comumente utilizadas na tarefa de classificação de textos. Os conceitos são elaborados sobre uma arquitetura com apenas uma camada oculta, mas podem ser facilmente generalizados para múltiplas camadas desse tipo.

3.1 Redes Neurais *Perceptron* Multicamadas

Dado um conjunto de amostras para as quais deseja-se aprender um classificador, $\left\{\left(\boldsymbol{\chi}^{(t)}, y^{(t)}\right)\right\}_{t=1}^{T}$, onde $\boldsymbol{\chi}^{(t)}$ e $y^{(t)}$ correspondem ao descritor e ao rótulo da t-ésima amostra, respectivamente, uma rede neural Perceptron multicamadas – do inglês *Multilayer Perceptron* (MLP), tem por objetivo aprender uma função $\mathcal{H}_\theta$ tal que a mesma seja capaz de calcular probabilidade da amostra em questão pertencer a cada uma das classes possíveis.

Neste processo, cada amostra, representada por seu descritor $\boldsymbol{\chi}^{(t)} \in \mathbb{R}^m$, em que m é quantidade de características do vetor[10], é apresentada individualmente à camada de entrada da rede. A seguir, seus componentes são recombinados na camada oculta com o objetivo de criar uma nova representação $\boldsymbol{\eta}^{(t)} \in \mathbb{R}^{d_{hid}}$, denominada estado oculto. Por fim, a função softmax é aplicada sobre tal vetor na camada de saída a fim de obter $\hat{\boldsymbol{y}}^{(t)} \in \mathbb{R}^{|\mathcal{C}|}$, o qual representa a probabilidade de $\boldsymbol{\chi}^{(t)}$ pertencer a cada uma das $|\mathcal{C}|$ classes possíveis e finalmente a amostra recebe o rótulo da classe com maior probabilidade. Este procedimento é ilustrado por meio da Figura 4:

Figura 4 – Rede neural MLP composta por uma camada oculta.



Fonte: Elaborado pelo autor.

[10] Se as amostras correspondem a *word embeddings*, então $m = d_{emb}$.

Cada conexão entre os neurônios de camadas adjacentes está associada a um peso $W_{ij}^{[\ell]}$, que tem por objetivo controlar o fluxo de valores entre as estruturas conectadas, onde i é o índice do neurônio de chegada e j o de saída, que por sua vez está situado na ℓ-ésima camada da rede. Consequentemente $W^{[\ell]} \in \mathbb{R}^{\varphi(\ell+1)\times\varphi(\ell)}$, de forma que $\varphi(\ell)$ representa a quantidade de neurônios da camada ℓ. A fim de uniformizar a notação utilizada, os índices superescritos entre colchetes representam a camada em questão, já aqueles entre parêntese indicam a qual amostra uma variável está relacionada. Adicionalmente, os neurônios que não fazem parte da camada de entrada, onde as amostras são apresentadas ao modelo, estão sujeito a um fator de *bias* $\boldsymbol{b}_i^{[\ell]}$ com $\boldsymbol{b}^{[\ell]} \in \mathbb{R}^{\varphi(\ell)}$. Matematicamente o processo apresentado é descrito por meio das seguintes equações:

$$\boldsymbol{\eta}^{(t)} = \Phi\left(W^{[1]}\boldsymbol{\chi}^{(t)} + \boldsymbol{b}^{[1]}\right), \tag{2}$$

$$\hat{\boldsymbol{y}}^{(t)} = \operatorname{softmax}\left(W^{[2]}\boldsymbol{h}^{(t)} + \boldsymbol{b}^{[2]}\right), \tag{3}$$

$$\mathcal{H}_\theta\left(\boldsymbol{\chi}^{(t)}\right) = \hat{\boldsymbol{y}}^{(t)}, \tag{4}$$

em que $\Phi(\cdot)$ é uma função de ativação, usualmente não linear, como a sigmoide[11] ou tangente hiperbólica, por exemplo. Já $\boldsymbol{h}^{(t)}$ representa o estado oculto da rede neural, nesse caso $\boldsymbol{h}^{(t)} = \boldsymbol{\eta}^{(t)}$ e os parâmetros a serem otimizados são $\theta = \left\{W^{[1]}, W^{[2]}, \boldsymbol{b}^{[1]}, \boldsymbol{b}^{[2]}\right\}$. Uma discussão mais detalhada sobre o funcionamento da função softmax é apresentada no Capítulo 4. O processo de aprendizado de uma rede neural consiste em realizar a predição das amostras no conjunto de treinamento e computar seu erro ao comparar os rótulos gerados aos reais por meio de uma função de custo, como a entropia cruzada, onde $\hat{\boldsymbol{y}}_{y^{(t)}}^{(t)}$ representa a probabilidade atribuída pela rede neural da t-ésima amostra pertencer à classe correta $y^{(t)}$:

$$S\left(\boldsymbol{\chi}^{(t)}, y^{(t)}\right) = -\log\left(\hat{\boldsymbol{y}}_{y^{(t)}}^{(t)}\right). \tag{5}$$

A partir desta informação é possível ajustar os pesos (ou parâmetros) do modelo com base nas derivadas de suas funções utilizando o algoritmo de retropropagação – do inglês *backpropagation*, em conjunto com algum método de otimização, como o gradiente descendente estocástico – do inglês, *Stochastic Gradient Descent* (SGD), a fim de que o classificador apresente uma baixa taxa de erro tanto para as amostras de treinamento, quanto para exemplares não vistos nesta etapa. Uma revisão compreensiva sobre o funcionamento de redes neurais, bem como técnicas de treinamento é apresentado por Goodfellow, Bengio e Courville. (GOODFELLOW; BENGIO; COURVILLE, 2016).

[11] A função sigmoide é calculada por meio da expressão $\sigma(x) = [1 + \exp(-x)]^{-1}$.

3.2 Redes Neurais Recorrentes

Apesar de ser empregada com sucesso em uma vasta gama de tarefas, o uso de redes neurais MLP torna-se limitado ao analisar dados sequenciais, onde as características de amostras anteriores podem influenciar a classificação atual. Considere, por exemplo, que dado apenas um *word embedding* deseja-se determinar a classe morfossintática de sua palavra correspondente. Em outras palavras, deseja-se treinar um etiquetador morfissintático, mais comumente denominado *POS tagger*. Neste caso, utilizar um classificador MLP parece ser uma possibilidade plausível, já que cada amostra será um *word embedding* diferente, mais especificamente $\boldsymbol{\chi}^{(w)} = \boldsymbol{v}_w$ e $y^{(t)}$ sua classe gramatical.

Por outro lado, o mesmo problema pode se tornar mais interessante ao modificá-lo para determinar a classe de cada palavra dentro de uma sentença ao levar em consideração seu contexto, representado pelos vocábulos em sua vizinhança. De maneira geral, cada uma das k amostras agora passam a ser compostas por uma sequência de τ_k episódios, que por sua vez são representados por meio de vetores de características $\boldsymbol{x}^{(k,t)}$:

$$\mathcal{X}^{(k)} = \left(\boldsymbol{x}^{(k,1)}, \boldsymbol{x}^{(k,2)}, \ldots, \boldsymbol{x}^{(k,\tau_k)}\right). \tag{6}$$

Com o objetivo de simplificar a notação utilizada no restante do texto, o índice k é omitido, de modo que fica subentendido que cada elemento $\boldsymbol{x}^{(t)}$ está associado a uma sequência correspondente, a qual possui tamanho τ_k, conforme ilustrado por meio da Figura 5.

Figura 5 – Exemplo de uma tarefa de rotulação sequencial.

Amostra	José	gosta	de	viagens	noturnas	longas
Descritores	$\boldsymbol{x}^{(1)}$	$\boldsymbol{x}^{(2)}$	$\boldsymbol{x}^{(3)}$	$\boldsymbol{x}^{(4)}$	$\boldsymbol{x}^{(5)}$	$\boldsymbol{x}^{(6)}$
Rótulos	$y^{(1)}$	$y^{(2)}$	$y^{(3)}$	$y^{(4)}$	$y^{(5)}$	$y^{(6)}$

Fonte: Elaborado pelo autor.

A fim de levar em consideração o aspecto da sequencialidade inerente deste tipo de problema, a forma de computar o estado oculto da rede neural pode ser modificada de acordo com a Equação 7, a qual considera as informações extraídas em etapas de classificação anteriores por meio de uma relação de recorrência, conforme segue:

$$\boldsymbol{\xi}^{(t)} = \Phi\left(W^{[1]}\boldsymbol{x}^{(t)} + V^{[1]}\boldsymbol{\xi}^{(t-1)} + \boldsymbol{b}^{[1]}\right), \tag{7}$$

em que $V^{[1]} \in \mathbb{R}^{\varphi(\ell)\times\varphi(\ell)}$ representa os pesos entre as conexões de cada neurônio da camada oculta consigo mesmo, permitindo que seu estado oculto, computado durante a classificação do episódio $(t-1)$, também seja considerado nos cálculos da camada oculta para o t-ésimo episódio. Note que quando o primeiro evento da sequência é examinado, ou seja $t = 1$,

como não existe estado oculto anterior, $\boldsymbol{\xi}^{(0)} = \mathbf{0}$. Essa formulação recebe o nome de Rede Neural Recorrente – do inglês *Recurrent Neural Network* (RNN) (ELMAN, 1990), e é menos comumente denominada Rede Neural Recorrente de Elman (E-RNN). A mesma é ilustrada de forma detalhada na Figura 6a e tem sua relação de recorrência expandida em função do tempo na Figura 6b.

Durante o treinamento, uma rede neural recorrente normalmente aprende a utilizar seu estado oculto, que também é encaminhado para as camadas mais profundas do modelo, como um resumo capaz de armazenar informações relevantes para a tarefa de interesse. Naturalmente, esta forma de representar dados vistos no passado incorre perda de informação, já que uma amostra formada por uma sequência arbitrária de episódios deve ser sintetizada em um único vetor de tamanho fixo $\boldsymbol{\xi}^{(t)}$. De acordo com a tarefa, o modelo pode inclusive aprender a armazenar alguns aspectos de forma mais persistente, ou com maior precisão, que outros (GOODFELLOW; BENGIO; COURVILLE, 2016). Esta característica é interessante do ponto de vista da Análise de Sentimento para contemplar construções semânticas, por exemplo.

Uma outra formulação que permite a rede neural ter acesso ao histórico de classificações passadas consiste em realimentar os neurônios da camada oculta não com seu estado anterior, mas sim com a distribuição de probabilidades de classificação gerada para a amostra antecessora. Nesse caso, a Equação 7 é substituída pela Equação 8, formando a Rede Neural Recorrente de Jordan (JORDAN, 1997) (J-RNN):

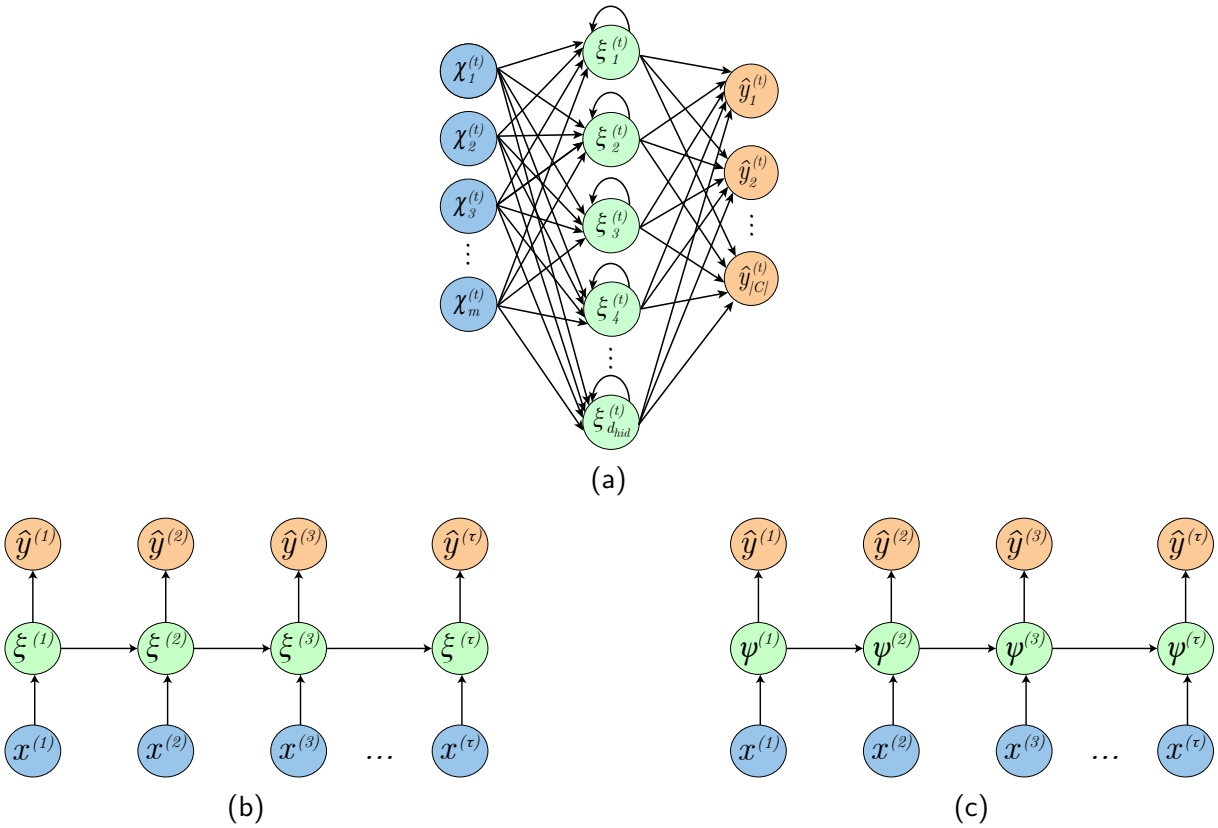
$$\boldsymbol{\psi}^{(t)} = \Phi\left(W^{[1]}\boldsymbol{x}^{(t)} + U^{[1]}\hat{\boldsymbol{y}}^{(t-1)} + \boldsymbol{b}^{[1]}\right), \tag{8}$$

em que, $U \in \mathbb{R}^{\varphi(\ell)\times C}$ pondera as conexões de retroalimentação entre os neurônios das camadas de saída e oculta. A Figura 6c ilustra essa configuração.

De acordo com Goodfellow, Bengio e Courville (GOODFELLOW; BENGIO; COURVILLE, 2016) a J-RNN é estritamente menos poderosa que uma E-RNN justamente devido à impossibilidade de compartilhar informações entre estados ocultos diretamente. O vetor recuperado de análises anteriores neste caso é o resultado da camada de saída, a qual tem seus valores ajustados de modo a produzir os rótulos no conjunto de treinamento, inviabilizando o uso desta representação como um resumo dinâmico da sequência. A independência entre estados ocultos, por sua vez, se traduz na possibilidade de paralelizar o treinamento deste modelo, já que todas informações para o cálculo do gradiente no instante t são previamente conhecidas. Esta característica não se verifica para as E-RNNs, onde o estado oculto do instante t depende das $(t-1)$ análises anteriores.

Por fim, independente do tipo de rede utilizada, as probabilidades de classificação são calculadas por meio da Equação 3 ao substituir $\boldsymbol{h}^{(t)}$ pela expressão apropriada para determinar o estado oculto da rede. Todas formulações apresentadas têm por objetivo aprender os parâmetros $\boldsymbol{b}^{[1]}$, $\boldsymbol{b}^{[2]}$, $W^{[2]}$ e $W^{[1]}$, $V^{[1]}$ ou $U^{[1]}$, de acordo com o tipo de rede em questão. As J-RNN

Figura 6 – Arquitetura das Redes Neurais Recorrentes de Elman e Jordan. (a) E-RNN; (b) E-RNN expandida em função do tempo; (c) J-RNN expandida em função do tempo.



Fonte: Elaborado pelo autor.

podem ser treinadas ao utilizar o mesmo algoritmo de retropropagação para redes do tipo MLP. Já nas redes que possuem conexões diretas entre os estados ocultos, como é o caso da E-RNN, a determinação do estado oculto atual depende de seus $(t-1)$ antecessores, e como o cálculo do gradiente é feito na direção contrária, da camada de saída para a de entrada, a determinação deste valor no t-ésimo instante depende de todos seus sucessores. Devido a esta sutil modificação, seu treinamento ocorre por meio do algoritmo de retropropagação através do tempo – do inglês *Backpropagation Through Time* (BPTT). O mesmo consiste em uma alteração no algoritmo de retropropagação, onde o cálculo dos gradientes também leva em consideração o aspecto sequencial das amostras.

Justamente devido a esta dependência, os sinais de correção de pesos da rede podem se tornar excessivamente grandes ou pequenos ao retroceder suas camadas, introduzindo o problema conhecido como explosão ou dissipação do gradiente, em ambos os casos impossibilitando a otimização dos parâmetros do modelo[12] (BENGIO; SIMARD; FRASCONI, 1994). A fim de contornar essa situação, técnicas como corte do gradiente – do inglês *gradient clipping* (PASCANU; MIKOLOV; BENGIO, 2013), a qual consiste em reescalar esse vetor quando o mesmo excede um determinado limite, além de outros tipos de neurônios como a unidade de

[12] O mesmo problema também surge em redes neurais MLP com múltiplas camadas ocultas, ditas profundas.

Longa Memória de Curto Prazo – do inglês *Long Short-Term Memory* (LSTM) (HOCHREITER; SCHMIDHUBER, 1997) e a Unidade Recorrente Bloqueável – do inglês *Gated Recurrent Unit* (GRU) (CHO et al., 2014) foram desenvolvidos.

3.2.1 Redes Neurais com Longa Memória de Curto Prazo

As unidades LSTM, desenvolvidas por Hochreiter e Schmidhuber (HOCHREITER; SCHMIDHUBER, 1997), possuem uma memória interna, de forma que as mesmas são capazes de "lembrar" características extraídas durante classificações anteriores por um período de tempo maior devido a seus "portões", responsáveis por regular o fluxo de informação em seu interior.

As expressões utilizadas no cálculo de cada componente são apresentadas a seguir, acompanhadas de sua descrição. A notação $[\boldsymbol{a};\boldsymbol{b}]$ indica a concatenação de vetores, de modo que se $\boldsymbol{a} \in \mathbb{R}^p$ e $\boldsymbol{b} \in \mathbb{R}^q$, então $[\boldsymbol{a};\boldsymbol{b}] \in \mathbb{R}^{p+q}$, ao passo que o símbolo '$\circ$' representa o produto de Hadamard, já $\sigma(\cdot)$ é a função sigmóide, $A_{\{f,i,o,g\}}$ são as matrizes de pesos e $\boldsymbol{d}_{\{f,i,o,g\}}$ são os vetores de *bias*.

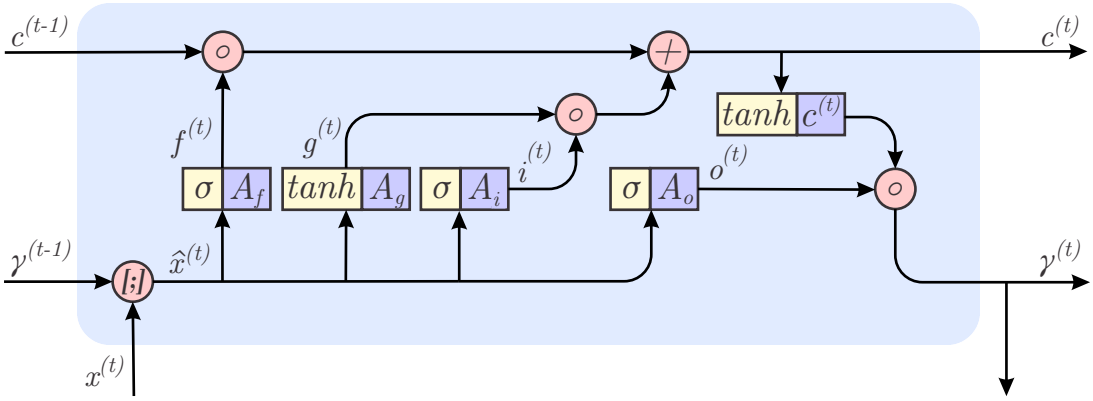
$$
\begin{aligned}
\hat{\boldsymbol{x}}^{(t)} &= \left[\boldsymbol{x}^{(t)};\boldsymbol{\gamma}^{(t-1)}\right] && \text{(concatenação da entrada)},\\
\boldsymbol{f}^{(t)} &= \sigma\left(A_f\hat{\boldsymbol{x}}^{(t)} + \boldsymbol{d}_f\right) && \text{(portão de esquecimento)},\\
\boldsymbol{i}^{(t)} &= \sigma\left(A_i\hat{\boldsymbol{x}}^{(t)} + \boldsymbol{d}_i\right) && \text{(portão de entrada)},\\
\boldsymbol{o}^{(t)} &= \sigma\left(A_o\hat{\boldsymbol{x}}^{(t)} + \boldsymbol{d}_o\right) && \text{(portão de saída)},\\
\boldsymbol{g}^{(t)} &= \tanh\left(A_g\hat{\boldsymbol{x}}^{(t)} + \boldsymbol{d}_g\right) && \text{(memória intermediária)},\\
\boldsymbol{c}^{(t)} &= \boldsymbol{f}^{(t)} \circ \boldsymbol{c}^{(t-1)} + i^{(t)} \circ \boldsymbol{g}^{(t-1)} && \text{(memória interna)},\\
\boldsymbol{\gamma}^{(t)} &= \boldsymbol{o}^{(t)} \circ \tanh\left(\boldsymbol{c}^{(t)}\right) && \text{(estado oculto)}.
\end{aligned}
\tag{9}
$$

A dinâmica de interação entre os componentes é apresentada na Figura 7, onde as setas indicam o fluxo dos vetores, os quais interagem entre si ao atravessar um círculo por meio da operação em seu interior. Os retângulos representam a aplicação de uma função de ativação (em amarelo) em conjunto com uma matriz de pesos (em roxo), produzindo assim uma nova representação. Apesar de uma unidade LSTM apresentar relações de recorrência, é possível analisar seu funcionamento interno de maneira sequencial.

Inicialmente, o vetor de características do episódio a ser classificado $\boldsymbol{x}^{(t)}$ é concatenado com o estado oculto $\boldsymbol{\gamma}^{(t-1)}$ produzido pela mesma unidade durante a análise do exemplar anterior, gerando assim $\hat{\boldsymbol{x}}^{(t)}$. Tal representação será utilizada para compor os vetores que representam cada um dos portões da unidade, bem como sua nova memória, que por sua vez é empregada na determinação do novo estado oculto $\boldsymbol{\gamma}^{(t)}$.

O portão de esquecimento $\boldsymbol{f}^{(t)}$ é computado por meio da interação entre $\hat{\boldsymbol{x}}^{(t)}$ e sua matriz de pesos e vetor de *bias* correspondentes. Por conseguinte, a função sigmóide é aplicada

Figura 7 – Dinâmica de interação entre os elementos em uma unidade LSTM.



Fonte: Elaborado pelo autor.

sob o resultado obtido, garantindo que todos componentes deste novo vetor pertençam ao intervalo $[0, 1]$. Ao fazer com que a memória da unidade, $\boldsymbol{c}^{(t-1)}$, interaja com $\boldsymbol{f}^{(t)}$ por meio do produto de Hadamard, cada um de seus componentes é modulado com diferentes intensidades. Desta forma, é possível regular a importância dada as informações advindas de classificações anteriores, presentes no vetor que representa a memória da unidade. Em casos extremos onde $\boldsymbol{f}^{(t)} = \mathbf{0}$, todas características extraídas anteriormente serão "esquecidas" pela unidade, ao passo que se todos elementos desse portão forem 1, a memória permanece intacta. Essa capacidade de modular os componentes de um vetor se estende a todos portões da unidade e é uma característica fundamental para garantir que diferentes informações sejam mantidas a cada episódio de classificação.

Em seguida, a memória intermediária $\boldsymbol{g}^{(t)}$ é computada ao aplicar a função tangente hiperbólica em $\hat{\boldsymbol{x}}^{(t)}$. Note que este passo é idêntico àquele realizado em uma rede neural convencional em sua camada oculta. O vetor obtido é modulado através do portão de entrada $\boldsymbol{i}^{(t)}$, o qual regula qual proporção deste vetor será adicionada ao vetor modulado $\boldsymbol{c}^{(t-1)}$ a fim de produzir a nova memória da unidade $\boldsymbol{c}^{(t)}$.

De posse de $\boldsymbol{c}^{(t)}$, que depende das memórias anterior e intermediária, resta agora produzir o novo estado oculto da rede $\boldsymbol{\gamma}^{(t)}$, o qual é calculado ao aplicar a função de ativação tangente hiperbólica sobre a nova memória da unidade em conjunto com sua matriz de pesos e vetor de *bias*. A representação produzida é, por sua vez, modulada pelo portão de saída $\boldsymbol{o}^{(t)}$, o qual regula a quantidade de informação que será propagada para o restante da rede por meio do novo estado oculto. Tal vetor será utilizado para a classificação dessa amostra por camadas mais profundas da rede, bem como para a extração de características de $\boldsymbol{x}^{(t+1)}$.

Apesar de conseguir modelar dependências de longo alcance e não sofrer do problema relacionado ao desaparecimento e explosão do gradiente, esse tipo de rede possui uma quantidade significativamente maior de parâmetros a serem treinados, representado pelas matrizes A_f, A_i, A_o, A_g e seus vetores de *bias* correspondentes. Ainda que este aspecto permita extrair

características mais complexas das amostras, é necessário utilizar mais dados para o treinamento a fim de evitar que o modelo memorize as amostras ou passe a dar importância para correlações espúrias observadas durante esta etapa, situação característica do super-treinamento, regime que impacta negativamente o modelo ao analisar novas amostras.

3.2.2 Redes Neurais com Unidades Recorrentes Bloqueáveis

As unidades LSTM apresentadas na Seção 3.2.1 permitem que uma rede neural consiga extrair características, ou mesmo classificar, uma sequência sem sofrer com os problemas de dissipação e desaparecimento do gradiente em situações normais. Além disso, unidades desse tipo conseguem modelar sequências de tamanhos razoáveis graças à sua memória interna. Todavia, estas vantagens surgem com o aumento significativo dos parâmetros a serem aprendidos pela camada quando comparada a uma RNN tradicional. Isso ocorre basicamente devido à introdução dos portões que modulam os vetores internos na unidade, além da matriz de pesos em sua não-linearidade adicional.

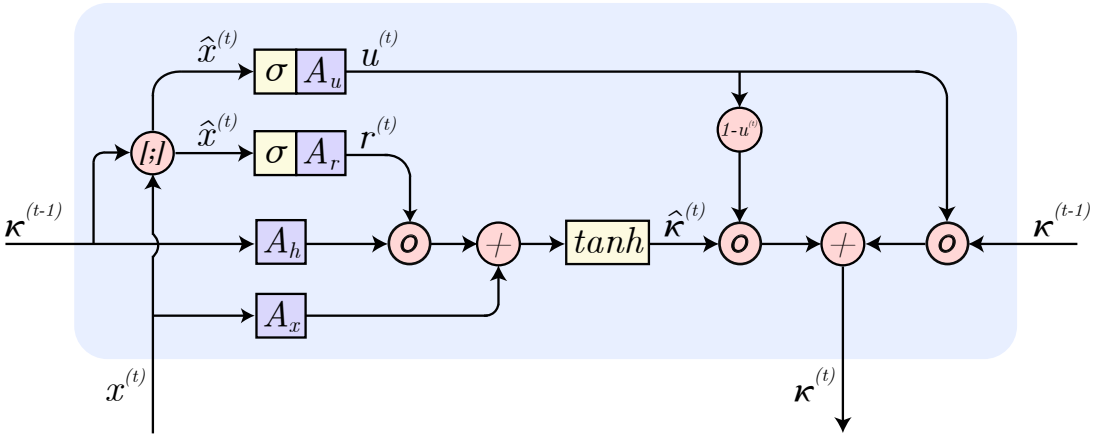
A fim de diminuir a quantidade de parâmetros, mas ainda assim explorar a dinâmica da LSTM, um tipo de unidade menos complexo foi proposto por Cho et al. (CHO et al., 2014), a GRU. A principal diferença deste mecanismo quando comparado à LSTM reside no fato de que a memória interna $\boldsymbol{c}^{(t)}$ deixa de existir, além de que um único portão passa a ser responsável por determinar quanto do estado oculto anterior é apagado e quanto da memória intermediária é preservada neste vetor durante o instante t. A seguir são apresentadas as equações que regem este tipo de unidade e seu funcionamento é esquematizado na Figura 8:

$$
\begin{aligned}
\hat{\boldsymbol{x}}^{(t)} &= \left[\boldsymbol{x}^{(t)}; \boldsymbol{\kappa}^{(t-1)}\right] && \text{(concatenação da entrada)},\\
\boldsymbol{u}^{(t)} &= \sigma\left(A_u \hat{\boldsymbol{x}}^{(t)} + \boldsymbol{d}_u\right) && \text{(portão de atualização)},\\
\boldsymbol{r}^{(t)} &= \sigma\left(A_r \hat{\boldsymbol{x}}^{(t)} + \boldsymbol{d}_r\right) && \text{(portão de esquecimento)},\\
\hat{\boldsymbol{\kappa}}^{(t)} &= \tanh\left(\boldsymbol{r}^{(t)} \circ A_h \boldsymbol{\kappa}^{(t-1)} + A_x \boldsymbol{x}^{(t)}\right) && \text{(memória intermediária)},\\
\boldsymbol{\kappa}^{(t)} &= \left(1 - \boldsymbol{u}^{(t)}\right) \circ \hat{\boldsymbol{\kappa}}^{(t)} + \boldsymbol{u}^{(t)} \circ \boldsymbol{\kappa}^{(t-1)} && \text{(estado oculto)}.
\end{aligned}
\tag{10}
$$

De forma mais detalhada, assim como na LSTM, inicialmente o descritor da amostra atual $\boldsymbol{x}^{(t)}$ é concatenado com o estado oculto anterior $\boldsymbol{\kappa}^{(t-1)}$, gerando $\hat{\boldsymbol{x}}^{(t)}$, que é utilizado no cálculo dos dois portões da unidade: o de esquecimento $\boldsymbol{r}^{(t)}$ e o de atualização $\boldsymbol{u}^{(t)}$. O primeiro determina quanto do estado oculto anterior será utilizado para computar a memória intermediária $\hat{\boldsymbol{\kappa}}^{(t)}$ da unidade. Já o segundo é utilizado para produzir o novo estado oculto $\boldsymbol{\kappa}^{(t)}$ ao interpolar a memória intermediária com o estado oculto anterior.

Apesar de diminuir a complexidade da camada, fazendo com que menos dados sejam necessários para treiná-la, além de acelerar seus cálculos, ainda assim, em PLN a maior parte dos trabalhos têm utilizado camadas LSTM. Talvez um dos fatores que expliquem esta adoção

Figura 8 – Dinâmica de interação entre os elementos em uma GRU.



Fonte: Elaborado pelo autor.

esteja relacionado à observação feita por Goodfellow, Bengio e Courville (GOODFELLOW; BENGIO; COURVILLE, 2016) em que os autores apontam que inicializar o vetor de *bias* do portão de esquecimento da LSTM com $\boldsymbol{d}_f = \mathbf{1}$, conforme advogado por Gers, Schmidhuber e Cummins (GERS; SCHMIDHUBER; CUMMINS, 2000), torna os resultados obtidos com esta camada um *baseline* muito forte, mesmo quando comparada a outras formulações recorrentes e não apenas à GRU. Ainda assim, a camada GRU normalmente requer menos dados para treinamento que uma LSTM.

3.2.3 Redes Neurais Bidirecionais

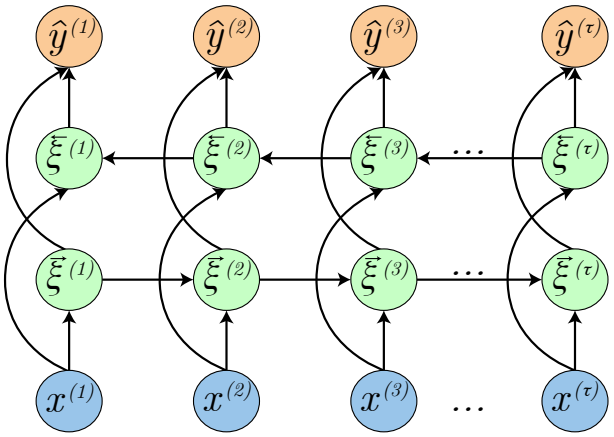
As redes neurais recorrentes apresentadas até o momento são capazes de aprender a sintetizar características relevantes de episódios vistos no passado por meio de seu estado oculto com o objetivo de levar em conta tal informação para a classificação de amostras que ainda serão observadas. Entretanto, em algumas aplicações pode ser interessante levar em consideração a sequência completa, composta por τ_k vetores de características, para predizer o rótulo de um de seus episódios.

Considere a tarefa de reconhecimento de caracteres em imagens, por exemplo. Neste caso o modelo pode se beneficiar de informações do futuro e do passado para realizar predições mais precisas para o t-ésimo episódio. Neste caso, esse tipo de informação pode ser utilizada para eliminar rótulos pouco prováveis ou mesmo auxiliar na tomada de decisões. Por exemplo, se o modelo encontra-se indeciso em relação à classe de $\boldsymbol{x}^{(t)}$ quanto às letras *"m"* e *"n"*, mas o mesmo tem acesso a características de sua vizinhança, as quais apresentam fortes indícios de ser formada pelas letras *"a"* e *"p"*, respectivamente, logo é bastante provável que a primeira opção de predição esteja correta, já que em português a letra *"n"* não antecede *"p"* e *"b"*.

Informações sobre episódios que antecedem e sucedem o atual podem ser incorporadas aos modelos por meio de redes neurais recorrentes bidirecionais. Neste caso cada episódio $\boldsymbol{x}^{(t)}$ é

apresentado a duas redes neurais recorrentes isoladas, por exemplo, E-RNNs. A primeira, $\overrightarrow{RNN}$, lê a sequência da esquerda para a direita, enquanto a segunda, $\overleftarrow{RNN}$, lê os episódios em sentido contrário. A cada amostra analisada, dois estados ocultos distintos são gerados, $\overrightarrow{\boldsymbol{\xi}}^{(t)}$ e $\overleftarrow{\boldsymbol{\xi}}^{(t)}$, os quais podem ser concatenados para que sejam encaminhados para camadas mais profundas do modelo, permitindo que uma análise contextual, que contemple toda a vizinhança do episódio em questão, seja realizada, conforme exemplificado anteriormente. A Figura 9 ilustra este arranjo, onde é possível observar que as redes recorrentes para a esquerda e direita são independentes entre si. Note que múltiplas camadas de redes não-recorrentes, recorrentes ou bidirecionais podem ser empilhadas para formar modelos mais robustos, e consequentemente cada vez mais difíceis de serem treinados, tendo em vista o aumento significativo dos parâmetros a serem aprendidos.

Figura 9 – Rede neural recorrente de Elman bidirecional.



Fonte: Elaborado pelo autor.

3.2.4 Mecanismo de Atenção

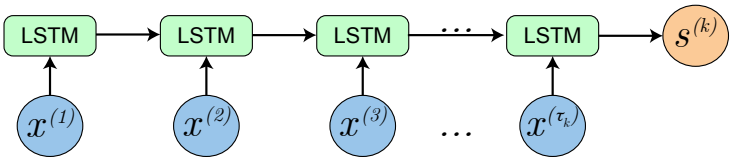
Conforme tem sido exposto, a análise de uma amostra $\mathcal{X}^{(k)}$ com característica sequencial é normalmente realizada ao utilizar uma RNN a fim de considerar a possibilidade de seus elementos influenciarem seus sucessores, ou vizinhos, no caso de redes bidirecionais. Neste regime, os vetores descritores são apresentados um a um ao modelo que, por sua vez, aprende a combinar as características do exemplar atual com informações extraídas de outros episódios em um primeiro nível e então encaminha esta nova representação contextualizada a camadas mais profundas da rede neural, normalmente dedicadas à classificação.

Este arranjo mostra-se de grande valia para situações onde cada um dos episódios da sequência possui um rótulo correspondente, conforme ilustrado na Figura 9. Entretanto, ao invés de rotular cada um de seus componentes individualmente, também pode ser interessante classificar a sequência como um todo. Neste caso a RNN é empregada como um extrator de características ao aprender a acumular aspectos de interesse da sequência ao analisar seus

episódios e sumarizar as informações mais relevantes por meio de seu estado oculto. Todavia, a compressão de uma sequência de tamanho variável em um estado oculto de dimensão fixa é um processo com perda de informação, conforme já discutido na Seção 3.2. A fim de contornar este problema, Bahdanau, Cho e Bengio (BAHDANAU; CHO; BENGIO, 2014) propuseram representar $\mathcal{X}^{(k)}$ também como uma sequência de estados ocultos de comprimento variável e utilizar um mecanismo de atenção para gerar o descritor da amostra, o qual possui tamanho fixo. Esta ideia foi inicialmente proposta para o desenvolvimento de modelos neurais para a tradução de sentenças mas atualmente tem sido utilizada em diferentes aplicações.

Dado os vetores $\left\{\boldsymbol{x}^{(t)}\right\}_{t=1}^{\tau_k}$, os quais descrevem cada um dos episódios da sequência de interesse $\mathcal{X}^{(k)}$, os mesmos são apresentados um a um a uma RNN, como uma LSTM, a fim de gerar sua representação contextualizada $\left\{\boldsymbol{\gamma}^{(t)}\right\}_{t=1}^{\tau_k}$, que consiste nos estados ocultos desta camada. Se o mecanismo de atenção não for utilizado, $\mathcal{X}^{(k)}$ pode ser descrito através de $\boldsymbol{\gamma}^{(\tau_k)}$, que é o último estado oculto da camada recorrente. A Figura 10 ilustra este processo.

Figura 10 – Utilização de uma rede LSTM para extração de características de uma sequência.



Fonte: Elaborado pelo autor.

O mecanismo de atenção, por sua vez, permitirá o cálculo de um descritor da sequência $\boldsymbol{s}^{(k)}$ como a soma ponderada de cada um de seus estados ocultos intermediários, dando ao modelo a liberdade de determinar quais vetores contextualizados, e consequentemente quais características, são mais importantes para o desempenho de sua tarefa. Este aspecto é interessante pois mesmo devido à possibilidade de preservar informações de longa distância dentro de uma sequência por meio de redes recorrentes, o estado oculto no instante t enfatiza mais as características dos episódios locais, que foram vistas recentemente. O peso do t-ésimo estado oculto $\left\{\boldsymbol{\gamma}^{(t)}\right\}_{t=1}^{\tau_k}$ da amostra, ou seja, seu coeficiente de atenção é computado por meio da expressão a seguir:

$$\alpha^{(t)} = \frac{\exp\left(e\left(\boldsymbol{h}^{(t)}\right)\right)}{\sum_{i=1}^{\tau}\exp\left(e\left(\boldsymbol{h}^{(i)}\right)\right)}, \tag{11}$$

em que $\boldsymbol{h}^{(t)} = \boldsymbol{\gamma}^{(t)}$ devido ao exemplo considerar uma LSTM e a função $e\left(\cdot\right)$ tem a seguinte formulação:

$$e\left(\boldsymbol{h}^{(t)}\right) = \boldsymbol{\iota}^{T}\tanh\left(A_a\boldsymbol{h}^{(t)} + \boldsymbol{d}_a\right), \tag{12}$$

tal que A_a é uma matriz de pesos, enquanto $\boldsymbol{\iota}$ e $\boldsymbol{d}_a$ são vetores que devem ser treinados em conjunto com a rede neural, enquanto que $e : \mathbb{R}^{d_{hid}} \mapsto \mathbb{R}$. Esta formulação é comumente modificada, especialmente a respeito de seus parâmetros, de acordo com a finalidade do modelo. Aproveitando o ensejo para exemplificar brevemente esta situação, considere que a camada LSTM que vem sendo discutida seja sucedida por uma camada recorrente de Elman com estado oculto $\boldsymbol{\xi}^{(t)}$. Devido à natureza do problema abordado, talvez fosse interessante também parametrizar a função $e(\cdot)$ com esta informação, almejando que modelo seja capaz de mover sua atenção entre diferentes episódios da sequência conforme a realização da tarefa de interesse progride. Neste caso é possível reescrever a equação do mecanismo de atenção da seguinte forma:

$$e'\left(\boldsymbol{h}^{(t)}, \boldsymbol{\xi}^{(t-1)}\right) = \boldsymbol{\iota}^T \tanh\left(A_1\boldsymbol{h}^{(t)} + A_2\boldsymbol{\xi}^{(t-1)} + \boldsymbol{d}_a\right). \tag{13}$$

Após o cálculo dos pesos para os estados ocultos, o descritor da sequência $\boldsymbol{s}^{(k)}$ é computado por meio de uma soma ponderada, conforme apresentado na Equação 14, e logo após é encaminhado para as camadas mais profundas da rede:

$$\boldsymbol{s}^{(k)} = \sum_{t=1}^{\tau_k} \alpha^{(t)}\boldsymbol{h}^{(t)}. \tag{14}$$

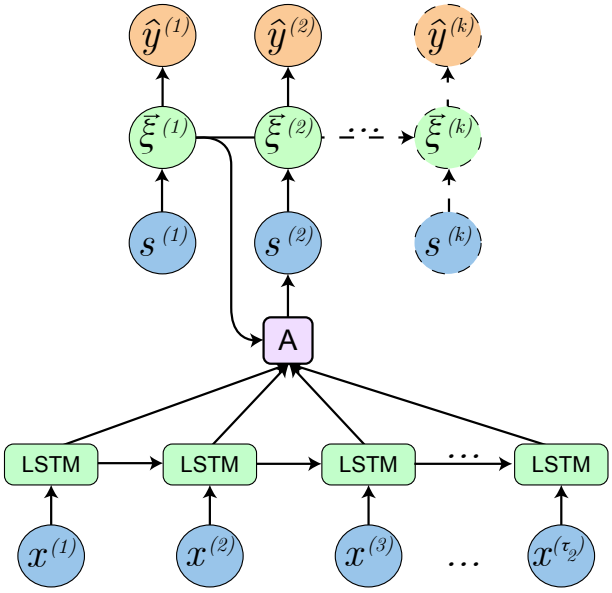
Novamente, $\boldsymbol{h}^{(t)}$ deve ser substituído pelo estado oculto do tipo de rede neural de interesse. É importante notar que a Equação 11 consiste na função softmax, garantindo que todos coeficientes de atenção sejam positivos e que o valor de sua soma seja 1. Isso implica no fato de que se função $e\left(\cdot\right)$ for uma constante, então todos os estados ocultos da sequência passarão a ter importância $1/\tau_k$ e o mecanismo de atenção deixará de ser uma soma ponderada para se tornar uma média simples de vetores, degenerando-se para uma operação de amostragem média – do inglês *average pooling*, que consiste em uma outra estratégia para gerar $\boldsymbol{s}^{(k)}$. Matematicamente:

$$\boldsymbol{s}'^{(k)} = \frac{1}{\tau_k}\sum_{t=1}^{\tau_k} \boldsymbol{h}^{(t)}. \tag{15}$$

Em suma, esta técnica consiste em uma forma mais elaborada para compor o vetor descritor de uma sequência arbitrária, como uma sentença (que nada mais é que uma sequência de palavras), dentro de um documento (que é uma sequência de sentenças). O modelo ilustrado na Figura 11 utiliza a função $e'\left(\cdot\right)$ em seu mecanismo de atenção, de modo que a camada LSTM gera o descritor de cada sentença enquanto que os estados ocultos $\boldsymbol{\xi}^{(t)}$ da E-RNN modelam a interação entre as sentenças a nível de documento. Nesse arranjo cada palavra é ponderada pelo mecanismo de atenção de acordo com a formulação da sentença em questão e ao considerar a interação entre as sentenças predecessoras.

Como os coeficientes de atenção $\alpha^{(t)}$ variam no intervalo $[0, 1]$, é possível inspecioná-los com o objetivo de tentar entender o que a rede neural considera mais importante para realizar sua tarefa, contribuindo, inclusive para a explicação de suas decisões. A Figura 12 destaca com cores de maior intensidade as palavras mais relevantes utilizadas para a classificação de frases de acordo com seu objetivo no modelo desenvolvido por Tran, Zukerman e Haffari (TRAN; ZUKERMAN; HAFFARI, 2017). No tocante às possíveis alterações neste mecanismo, ainda é possível empilhar múltiplas camadas recorrentes e adicionar a cada uma um mecanismo de atenção, formando assim uma mecanismo de atenção hierárquico (YANG et al., 2016).

Figura 11 – Mecanismo de atenção A utilizado para descrever uma sequência ao levar em consideração o estado anterior da camada oculta subsequente. As linhas pontilhadas indicam amostras futuras a serem analisadas.



Fonte: Elaborado pelo autor.

3.3 Redes Neurais Convolucionais

Redes Neurais Convolucionais – do inglês *Convolutional Neural Networks* (CNNs), as quais foram propostas para abordar problemas de processamento de imagens, também podem ser adotadas para endereçar problemas no domínio textual, por exemplo, determinar o vetor descritor de uma sentença. Ao considerar que cada palavra corresponde a um vetor $\boldsymbol{x}^{(t)} \in \mathbb{R}^m$, uma sentença de n palavras pode ser formada por meio da concatenação de seus *word embeddings* transpostos, formando a matriz $X \in \mathbb{R}^{n \times m}$, conforme apresentado na Equação 16:

$$X = \left[\boldsymbol{x}^{(1)^T}; \boldsymbol{x}^{(2)^T}; \ldots; \boldsymbol{x}^{(n)^T}\right]. \tag{16}$$

Figura 12 – Visualização dos coeficientes de atenção para a classificação de frases. Os símbolos <s> e </s> indicam início e fim das sequências de palavras.

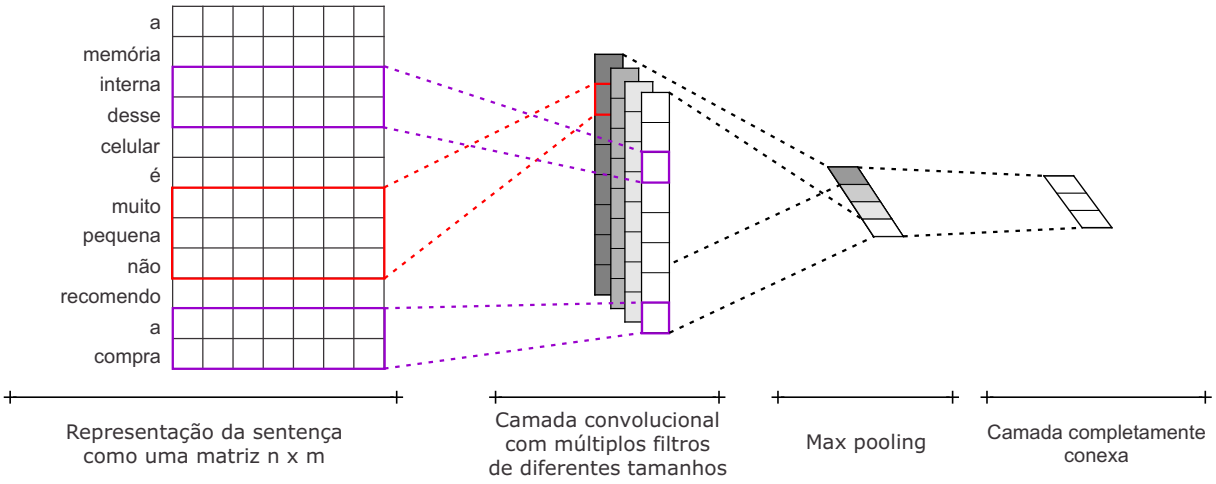
Fonte: Adaptado de (TRAN; ZUKERMAN; HAFFARI, 2017).

De forma análoga, é possível enxergar a sentença como uma imagem, onde cada linha corresponde a um *word embedding*, e a partir da qual se deseja extrair descritores para realizar sua classificação, já que a presença da combinação de algumas palavras em uma frase pode ser utilizada como evidência para sua classificação. Entretanto, uma diferença entre as CNNs empregadas no processamento de imagens e PLN diz respeito às dimensões dos filtros e a área sobre a qual a operação de convolução é realizada. Enquanto que no primeiro caso essa região é normalmente quadrada, cobrindo pequenas áreas da imagem, no segundo a mesma é retangular e abrange linhas inteiras (ou palavras) da matriz com o objetivo de extrair características de n-gramas, conforme ilustrado na Figura 13.

Uma limitação dessa abordagem diz respeito ao tamanho da sentença, que deve conter no máximo n palavras a fim de manter as restrições impostas sobre as dimensões de X. Caso a amostra de interesse possua menos vocábulos que o necessário, é possível completar essa representação por meio da concatenação de vetores nulos, os quais podem ser vistos como palavras vazias. Por outro lado, se o tamanho da sentença for maior que n, os termos excedentes são normalmente desprezados. Outra possibilidade consiste em treinar diferentes CNNs para amostras com diferentes tamanhos.

A representação matricial gerada X, é dividida com sobreposição em regiões com h palavras contíguas, as quais consistem nos n-gramas e são representadas pela expressão $\boldsymbol{x}^{(t:t+h-1)} = \left[\boldsymbol{x}^{(t)}, \boldsymbol{x}^{(t+1)}, \ldots, \boldsymbol{x}^{(t+h-1)}\right]$. Sobre cada uma dessas janelas é aplicada uma convolução ao adotar um filtro $\Lambda^{(j)} \in \mathbb{R}^{h \times k}$, uma unidade de bias $\boldsymbol{b}^{(j)} \in \mathbb{R}$ e uma função de ativação

Figura 13 – Utilização de uma CNN para classificação de texto. Os retângulos representam a convolução realizada por um filtro de bigramas e trigramas.



Fonte: Adaptado de Kim (KIM, 2014).

$\Phi(\cdot)$. Logo, cada elemento do vetor resultante $\boldsymbol{q}^{(j)} \in \mathbb{R}^{(n-h+1)}$ é determinado de acordo com a seguinte equação, onde "$*$" representa a operação de convolução:

$$\boldsymbol{q}_i^{(j)} = \Phi\left(\Lambda^{(j)} * \boldsymbol{x}^{(i:i+h-1)} + \boldsymbol{b}^{(j)}\right). \tag{17}$$

Múltiplos filtros com tamanhos variados podem ser utilizados para aprender diferentes combinações de palavras de interesse que ocorrem nas sentenças, assim como na classificação de imagens, onde deseja-se aprender filtros capazes de detectar diferentes padrões de interesse. Para cada um desses j filtros um vetor de características $\boldsymbol{q}^{(j)}$ é gerado, assim é possível concatená-los novamente, formando uma nova matriz sobre a qual o procedimento de convolução pode ser repetido; outra possibilidade consiste em utilizar uma camada de amostragem, como a *max-pooling*, com o objetivo de manter apenas o maior elemento de cada um dos descritores e formar um novo vetor de características mais compacto apenas com os padrões mais proeminentes na sentença. Tal procedimento pode ser repetido sucessivamente até que a representação final da amostra seja gerada. Este último vetor é, por fim,encaminhado a uma rede neural completamente conexa para a classificação, conforme descrito na Seção 3.1. A etapa de treinamento, por sua vez, segue o mesmo procedimento utilizado por CNNs tradicionais.

3.4 Considerações Finais

Neste capítulo foram discutidos os funcionamentos das redes neurais, partindo de um modelo simples até alcançar variantes recorrentes, as quais serão utilizadas na presente dissertação a fim de classificar falas dentro de um diálogo quanto ao seu sentimento. Essa decisão vem do fato de que uma conversa pode ser vista como uma sequência de turnos, cada

qual correspondente a uma sequência de palavras. Atualmente, as principais redes recorrentes utilizadas em PLN são as LSTMs e GRUs, normalmente com caráter bidirecional, as quais serão avaliadas para o modelo proposto. Além disso, o mecanismo de atenção pode ser empregado com dois objetivos principais: melhorar os resultados do modelo e entender seus critérios de tomada de decisão a partir dos coeficientes de atenção.

4 Campos Aleatórios Condicionais

O presente capítulo tem por objetivo discutir o funcionamento dos Campos Aleatórios Condicionais – do inglês *Conditional Random Fields* (CRFs) (LAFFERTY; MCCALLUM; PEREIRA, 2002) para a classificação de elementos dentro de uma sequência. Inicialmente, é apresentada sua motivação para a área de PLN, seguida pela formulação matemática do modelo, o processo de inferência e aprendizagem dos parâmetros. Uma revisão mais abrangente sobre o assunto é apresentada por Sutton e McCallum (SUTTON; MCCALLUM, 2012).

4.1 Motivação

As redes neurais introduzidas para a classificação de sequências no Capítulo 3 têm por objetivo combinar as características dos episódios que formam a amostra para gerar descritores contextualizados, almejando melhores resultados de classificação. A seguir, as representações obtidas são rotuladas na camada de saída, normalmente por meio da função softmax, conforme reproduzido na Equação 18. Entretanto, este passo toma decisões independentes em função do tempo, ou seja, a classificação de um episódio não é influenciada diretamente pelo rótulo de seus antecessores ou sucessores, diferentemente da extração de características ao empregar uma RNN. Esse processo é evidenciado por meio da Figura 6b no Capítulo 3, onde apenas a camada oculta está interconectada em função do tempo.

Assim como as RNNs tornam a extração de características um processo contextual, também seria interessante parametrizar a camada de saída em função de rotulações realizadas no passado. Isso permite não só que o modelo passe a dar preferência para algumas sequências de rótulos para certas ocasiões, mas também torne-se capaz de analisar amostras com um grau maior de complexidade. Tal situação pode ser novamente motivada por meio da tarefa de rotulação das palavras de uma sentença de acordo com sua informação morfossintática, como na sentença em inglês *"the old man the boat"*, a qual pode ser rotulada como {artigo, adjetivo, substantivo, artigo, substantivo} e implica na tradução incorreta "o velho homem o barco". O erro pode ser detectado seja por meio da frase traduzida, que não faz sentido, ou ao observar o par de rótulos de POS adjacentes {substantivo, artigo}, formando uma transição bastante improvável tanto em inglês quanto em português, onde um substantivo normalmente não é seguido de um artigo. Com esta restrição em mente e a possibilidade de considerar rotulações de episódios anteriores na classificação atual, característica fundamental do CRF, o exemplo pode ser classificado novamente como {artigo, substantivo, verbo, artigo, substantivo}, levando à tradução correta *"o velho tripula o barco"*[13] (EISEINSTEIN, 2019).

[13] Essas frases são denominadas *garden paths*. Foi utilizado um exemplo em inglês pois a maioria dos exemplos em português não permitem a desambiguiação concisa ao substituir o papel sintático apenas de uma palavra.

4.2 Formulação

A função softmax empregada na camada de saída dos modelos vistos até o momento foi apresentada da seguinte forma, onde $\boldsymbol{h}^{(t)}$ é o estado oculto computado, independente do tipo de arquitetura utilizada:

$$\hat{\boldsymbol{y}}^{(t)} = \text{softmax}\left(W^{[2]}\boldsymbol{h}^{(t)} + \boldsymbol{b}^{[2]}\right). \tag{18}$$

O argumento desta função pode ser reescrito ao considerar que $\boldsymbol{z}^{(t)} = \left(W^{[2]}\boldsymbol{h}^{(t)} + \boldsymbol{b}^{[2]}\right) \in \mathbb{R}^{|\mathcal{C}|}$, onde $\boldsymbol{z}_c^{(t)}$ corresponde à afinidade entre o t-ésimo episódio e a c-ésima classe, de modo que o objetivo da função em questão consiste exclusivamente em converter estas pontuações em probabilidades $\hat{\boldsymbol{y}}^{(t)}$ por meio da seguinte expressão:

$$\text{softmax}\left(\boldsymbol{z}_c^{(t)}\right) = \frac{\exp\left(\boldsymbol{z}_c^{(t)}\right)}{\sum_{d=1}^{|\mathcal{C}|}\exp\left(\boldsymbol{z}_d^{(t)}\right)} = p\left(\hat{\boldsymbol{y}}_c^{(t)}|\boldsymbol{x}^{(t)};\theta\right), \tag{19}$$

em que o operador de exponenciação é responsável por tornar a pontuação atribuída pelo modelo positiva. Já o denominador soma todas pontuações convertidas e tem caráter normalizador, garantindo que a soma das probabilidades seja sempre 1. Este elemento é por vezes denominado função de partição e representado pela função $Z(\cdot)$[14]. Por fim, o último termo da Equação 19 representa a probabilidade $\hat{\boldsymbol{y}}^{(t)}$ da amostra $\boldsymbol{x}^{(t)}$ pertencer à c-ésima classe, parametrizada por θ, de modo que durante o treinamento deseja-se que:

$$y^{(t)} = \underset{c}{\text{argmax}}\, p\left(\hat{\boldsymbol{y}}_c^{(t)}|\boldsymbol{x}^{(t)};\theta\right). \tag{20}$$

A seguir serão apresentadas alterações incrementais na Equação 19 com o objetivo de derivar a formulação dos Campos Aleatórios Condicionais, mais especificamente, Campos Aleatórios Condicionais Linearmente Encadeados – do inglês *Linear-Chain Conditional Random Fields*, a partir da função softmax. Inicialmente, considere que deseja-se determinar a probabilidade de uma sequência formada por dois episódios $\mathcal{X} = \left(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}\right)$ ser classificada com dois rótulos quaisquer $\mathcal{Y} = (y_1, y_2)$ a partir da formulação anterior, a qual pode ser reescrita como:

$$\begin{aligned} p\left(\mathcal{Y} \mid \mathcal{X};\theta\right) = p\left(y_1, y_2 \mid \boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)};\theta\right) &= \text{softmax}\left(\boldsymbol{z}_{y_1}^{(1)}\right)\ \text{softmax}\left(\boldsymbol{z}_{y_2}^{(2)}\right) \\ &= \frac{\exp\left(\boldsymbol{z}_{y_1}^{(1)}\right)\ \exp\left(\boldsymbol{z}_{y_2}^{(2)}\right)}{\sum_{c=1}^{|\mathcal{C}|}\exp\left(\boldsymbol{z}_c^{(1)}\right)\ \sum_{d=1}^{|\mathcal{C}|}\exp\left(\boldsymbol{z}_d^{(2)}\right)}. \end{aligned} \tag{21}$$

[14] Apesar de letras maiúsculas corresponderem a matrizes no texto, esta regra é violada para a função de partição a fim de manter conformidade com a literatura.

Observe que $\boldsymbol{z}^{(t)}$ é um vetor que descreve a afinidade entre uma amostra e todas as classes, mas para o cálculo em questão é relevante apenas a coordenada a para o primeiro episódio e b para o segundo. Desta forma, tal expressão pode ser reescrita genericamente como:

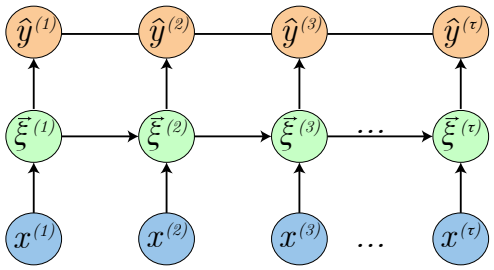
$$\boldsymbol{z}_{y^{(t)}}^{(t)} = W_{y^{(t)}.}^{[2]} \boldsymbol{h}^{(t)} + \boldsymbol{b}_{y^{(t)}}^{[2]} = f_1\left(\mathcal{X}, y^{(t-1)}, y^{(t)}, t\right), \tag{22}$$

em que o primeiro termo da multiplicação no segundo membro corresponde à linha da matriz de pesos $W^{[2]}$ relacionada à classe $y^{(t)}$. Apesar de aparentemente complexa, a ideia por trás da expressão é simples e consiste apenas em selecionar a $y^{(t)}$-ésima coordenada do vetor $\boldsymbol{z}^{(t)}$ para a amostra atual, ou seja, dadas todas as afinidades entre a amostra em questão e os rótulos possíveis, deseja-se recuperar apenas a pontuação referente à classe $y^{(t)}$ para o t-ésimo instante.

As funções de características $f_i(\cdot)$ podem ser vistas como uma extensão da pontuação $\boldsymbol{z}^{(t)}$ computada pelas redes neurais, já que estas também produzem valores reais, os quais correspondem à afinidade episódio-classe, mas além de permitir que toda a sequência de entrada $\mathcal{X}$ possa ser utilizada para computar $\boldsymbol{z}^{(t)}$, como ocorre com as RNNs, as $f_i(\cdot)$ também possibilitam considerar transições entre pares de rótulos adjacentes para o cálculo das afinidades. Comumente, as funções de características são divididas em dois tipos: aquelas que consideram apenas um par episódio-rótulo e são denominadas *"funções características de emissão"*, como é o caso de $\boldsymbol{z}^{(t)}$ e aquelas que analisam pares de rótulos adjacentes e são chamadas *"funções características de transição"*, que motivam o uso dos CRFs.

A função de emissão $f_1(\cdot)$ apresentada na Equação 22 determina o grau de afinidade episódio-rótulo calculado por meio de uma rede neural recorrente, conforme ilustrado na Figura 14. Entretanto, novas funções podem ser projetadas aliando múltiplas fontes de informação e inclusive geradas manualmente. São exemplos válidos de funções *"$\mathcal{X}$ começa com uma letra maiúscula"*, ou $\left(\boldsymbol{x}^{(t)} = \text{'pássaro'} \wedge y^{(t-1)} = \text{'verbo'}\right)$, ou ainda $\left(y^{(t-1)} = \text{'substantivo'} \wedge y^{(t)} = \text{'adjetivo'}\right)$, para ilustrar uma função de transição.

Figura 14 – Classificação sequencial ao empilhar uma E-RNN e um CRF na camada de saída para a classificação contextual.



Fonte: Elaborado pelo autor.

A formulação dos CRFs pode ser vista de duas formas: a primeira como uma função

capaz de calcular a probabilidade de se observar uma sequência de rótulos arbitrária $\mathcal{Y}$ dado $\mathcal{X}$, e a segunda como um método para determinar a sequência de rótulos mais provável para uma amostra. Para realizar a primeira tarefa, basta somar as pontuações do "caminho" definido pelos rótulos em $\mathcal{Y}$ ao longo de $\mathcal{X}$ para cada função característica $f_i(\cdot)$, conforme ilustrado na Figura 15:

$$F_i(\mathcal{X}, \mathcal{Y}) = \sum_t f_i\left(\mathcal{X}, y^{(t-1)}, y^{(t)}, t\right). \tag{23}$$

Com esta formulação em mente, a Equação 21 pode ser reescrita como um CRF para dois episódios com apenas uma função característica:

$$\begin{aligned} p(\mathcal{Y} \mid \mathcal{X}; \theta) &= \frac{\exp\left(\boldsymbol{z}_{y_1}^{(1)}\right)\ \exp\left(\boldsymbol{z}_{y_2}^{(2)}\right)}{\sum_{c=1}^{|\mathcal{C}|}\exp\left(\boldsymbol{z}_c^{(1)}\right)\ \sum_{d=1}^{|\mathcal{C}|}\exp\left(\boldsymbol{z}_d^{(2)}\right)} \\ &= \frac{\exp(\boldsymbol{z}_{y_1}^{(1)} + \boldsymbol{z}_{y_2}^{(2)})}{Z(\mathcal{X}, \boldsymbol{w})} \\ &= \frac{\exp\left[\boldsymbol{w}_1 \sum_t f_1\left(\mathcal{X}, y^{(t-1)}, y^{(t)}, t\right)\right]}{Z(\mathcal{X}, \boldsymbol{w})} \\ &= \frac{1}{Z(\mathcal{X}, \boldsymbol{w})}\exp\left[\boldsymbol{w}_1 F_1(\mathcal{X}, \mathcal{Y})\right]. \end{aligned} \tag{24}$$

Esta expressão pode ser generalizada para sequências de tamanhos arbitrários com uma quantidade qualquer de funções características, onde $\boldsymbol{w}_j$ é um parâmetro que regula a importância de cada uma das j funções e é aprendido durante o treinamento do modelo:

$$p(\mathcal{Y} \mid \mathcal{X}; \theta) = \frac{1}{Z(\mathcal{X}, \boldsymbol{w})}\exp\left[\sum_i \boldsymbol{w}_i F_i(\mathcal{X}, \mathcal{Y})\right], \tag{25}$$

sendo que a função de partição consiste na soma da pontuação de todas possíveis sequências de rótulos $\mathcal{Y}'$ que podem ser geradas para $\mathcal{X}$, representadas por meio do conjunto $\mathbb{Y}(\mathcal{X})$:

$$Z(\mathcal{X}, \boldsymbol{w}) = \sum_{\mathcal{Y}' \in \mathbb{Y}(\mathcal{X})}\left[\exp\sum_i \boldsymbol{w}_i F_i(\mathcal{X}, \mathcal{Y}')\right]. \tag{26}$$

Por fim, a camada de saída de uma rede neural formada por um CRF pode derivar pontuações de afinidade episódio-episódio ao considerar o rótulo atribuído ao exemplar anterior durante a classificação atual por meio de uma matriz de afinidade de transição $P \in \mathbb{R}^{|\mathcal{C}|\times|\mathcal{C}|}$, a qual também é aprendida pelo modelo, onde cada um de seus elementos corresponde à possibilidade de transitar dois estados (ou classes) consecutivos, ou seja:

$$f_2\left(\mathcal{X}, y^{(t-1)}, y^{(t)}, t\right) = P_{y^{(t-1)}, y^{(t)}}. \tag{27}$$

É justamente esse tipo de função que é responsável por aprender que é bastante improvável um artigo preceder um verbo na tarefa de POS. Em outras palavras, a pontuação de transição entre as classes "artigo" e "verbo" deve ser pequena ao final do treinamento do modelo.

A mesma notação matricial também pode ser utilizada em f_1. Para isso considere $Q = \left[\boldsymbol{z}^{(1)}; \boldsymbol{z}^{(2)}; \ldots; \boldsymbol{z}^{(\tau)}\right]^T \in \mathbb{R}^{\tau \times |\mathcal{C}|}$, ou seja, a concatenação de $\boldsymbol{z}^{(t)}$ para toda a sequência, logo:

$$f_3\left(\mathcal{X}, y^{(t-1)}, y^{(t)}, t\right) = Q_{t,y^{(t)}}. \tag{28}$$

Para justificar o fato das funções características de transição serem restritas a observar apenas o estado anterior ao atual, considere a função de partição $Z(\mathcal{X}, \boldsymbol{w})$, a qual deve somar as pontuações de todas as sequências de rótulos possíveis para a amostra em questão. Devido a este papel, a mesma cresce exponencialmente em função do tamanho da sequência. Enquanto que para amostras com apenas um episódio existem $|\mathcal{C}|$ rótulos possíveis (como é o caso da função softmax), para uma sequência de tamanho τ existem $|\mathcal{C}|^\tau$ possibilidades a serem examinadas, tornando o problema computacionalmente intratável. Todavia, se as observações feitas aos rótulos forem restritas apenas ao antecessor de t, então apenas a decisão do rótulo anterior influenciará a pontuação para a classificação atual. Isso faz com que o problema possa ser resolvido de maneira eficiente através de Programação Dinâmica utilizando o Algoritmo de Viterbi (VITERBI, 1967), o qual será apresentado na Seção 4.3.

Uma última modificação a ser realizada para a utilização eficiente dos CRFs diz respeito às amostras a serem analisadas. Considere que deseja-se treinar um modelo para rotular as palavras de uma frase quanto a sua informação de POS utilizando a arquitetura apresentada na Figura 14. Neste cenário, os descritores de uma sequência a ser analisada $\mathcal{X}$ são extraídos por meio da E-RNN, gerando uma nova sequência $\mathcal{Z} = \left(\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \ldots, \boldsymbol{z}^{(\tau)}\right)$, que será apresentada à camada de saída, formada agora pelo CRF. Antes desta etapa ser iniciada, a sequência intermediária pode ser alterada da seguinte forma: $\mathcal{Z} := \left(\triangle, \boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \ldots, \boldsymbol{z}^{(\tau)}, \triangledown\right)$, onde os novos vetores especiais indicam o início e fim da sequência. Consequentemente o conjunto de classes $\mathcal{C}$ também deve ser modificado a fim de incluir dois novos rótulos "▲" e "▼", cada qual relacionado apenas a um dos símbolos especiais. Proceder desta forma permite ao CRF aprender quais classes são mais comuns no início e fim das sequências por meio da matriz de transição P.

4.3 Inferência

Apesar da formulação do CRF permitir determinar a probabilidade de gerar uma sequência de rótulos qualquer dada uma amostra, o real interesse consiste em determinar a sequência de rótulos mais provável para realizar a tarefa de classificação. Para facilitar os cálculos e evitar instabilidade numérica, as operações a seguir consideram o logaritmo da

probabilidade, que por ser uma função monotonicamente crescente não influencia o resultado final obtido:

$$\begin{aligned}\mathcal{Y}^{\star} &= \underset{\mathcal{Y}}{\operatorname{argmax}} \log p\left(\mathcal{Y} \mid \mathcal{X}; \theta\right) \\ &= \underset{\mathcal{Y}}{\operatorname{argmax}} \log \left[\frac{1}{Z(\mathcal{X}, \boldsymbol{w})} \exp \sum_{i} \boldsymbol{w}_{i} F_{i}\left(\mathcal{X}, \mathcal{Y}\right)\right] \\ &= \underset{\mathcal{Y}}{\operatorname{argmax}} \left[\log \exp \sum_{i} \boldsymbol{w}_{i} F_{i}\left(\mathcal{X}, \mathcal{Y}\right) - \log Z\left(\mathcal{X}, \boldsymbol{w}\right)\right] \\ &= \underset{\mathcal{Y}}{\operatorname{argmax}} \left[\sum_{j} \boldsymbol{w}_{i} F_{i}\left(\mathcal{X}, \mathcal{Y}\right) - \log Z\left(\mathcal{X}, \boldsymbol{w}\right)\right].\end{aligned} \tag{29}$$

Como a função de partição é constante para cada amostra, a mesma pode ser desprezada para resolver o problema de maximização, de modo que a sequência mais provável pode ser determinada da seguinte forma:

$$\begin{aligned}\mathcal{Y}^{\star} &= \underset{\mathcal{Y}}{\operatorname{argmax}} \sum_{i} \boldsymbol{w}_{i} F_{i}\left(\mathcal{X}, \mathcal{Y}\right) \\ &= \underset{\mathcal{Y}}{\operatorname{argmax}} \sum_{t=1}^{\tau+1} \sum_{i} \boldsymbol{w}_{i} f_{i}\left(\mathcal{X}, y^{(t-1)}, y^{(t)}, t\right).\end{aligned} \tag{30}$$

Note que a indexação em y varia entre 0 e $(\tau + 1)$ devido à inclusão dos símbolos especiais "$\triangle$" e "$\triangledown$". Assim, podemos definir a seguinte função auxiliar que representa a pontuação de transitar entre os rótulos $y^{(t-1)}$ e $y^{(t)}$ no instante t da sequência ao considerar todas as funções características:

$$g_{t}\left(\mathcal{X}, y^{(t-1)}, y^{(t)}\right) = \sum_{i} \boldsymbol{w}_{i} f_{i}\left(\mathcal{X}, y^{(t-1)}, y^{(t)}, t\right). \tag{31}$$

Ao substituir esta definição na Equação 30 é possível reduzir o problema de inferência à formulação da Equação 32, a qual consiste basicamente em somar a pontuação gerada ao transitar entre os rótulos de $\mathcal{Y}$ para cada episódio da sequência:

$$\begin{aligned}\mathcal{Y}^{\star} &= \underset{\mathcal{Y}}{\operatorname{argmax}} \sum_{t=1}^{\tau+1} \sum_{i} \boldsymbol{w}_{i} f_{i}\left(\mathcal{X}, y^{(t-1)}, y^{(t)}, t\right) \\ &= \underset{\mathcal{Y}}{\operatorname{argmax}} \sum_{t=1}^{\tau+1} g_{t}\left(\mathcal{X}, y^{(t-1)}, y^{(t)}\right).\end{aligned} \tag{32}$$

Por fim, o problema da Equação 32 pode ser resolvido por meio de Programação Dinâmica. Para isso considere por um momento que deseja-se determinar apenas a pontuação

da melhor sequência de rótulos, já que a função de partição foi desprezada:

$$s\left(\mathcal{X}, \mathcal{Y}\right) = \max_{\mathcal{Y}_{1:\tau}} \sum_{t=1}^{\tau+1} g_t\left(\mathcal{X}, y^{(t-1)}, y^{(t)}\right), \tag{33}$$

em que $\mathcal{Y}_{1:\tau}$ representa os rótulos entre os episódios 1 e τ. Por construção, como o último termo de $\mathcal{Y}$ agora é o rótulo "▼" a expressão anterior pode ser decomposta em:

$$s\left(\mathcal{X}, \mathcal{Y}\right) = \left[\max_{\mathcal{Y}_{1:\tau-1}} \sum_{t=1}^{\tau} g_t\left(\mathcal{X}, y^{(t-1)}, y^{(t)}\right) + \max_{y^{(\tau)}} g_{(\tau+1)}\left(\mathcal{X}, y^{(\tau)}, \blacktriangledown\right)\right]. \tag{34}$$

Devido ao segundo termo de $g_{(\tau+1)}\left(\cdot\right)$ ser fixo no rótulo final, torna-se necessário determinar apenas o último rótulo não-artificial da sequência $\mathcal{Y}$ que ao transitar para "▼" gere a maior pontuação possível. Como essa decisão não influencia o restante do espaço de busca, a mesma pode ser tomada de forma gananciosa, ou seja, escolher o maior valor atual levará ao resultado ótimo no futuro. Uma outra maneira de enxergar a formulação é a seguinte: a primeira parte da soma consiste no caminho ótimo do início da sequência até o instante $(\tau - 1)$. Resta agora escolher o rótulo no instante t com maior pontuação que conecta ambas partes do caminho, da esquerda para a direita, formando uma sequência completa.

O mesmo raciocínio pode ser aplicado recursivamente ao primeiro termo da Equação 34 até que o problema seja decomposto em uma série de decisões individuais. Assim, todo o problema pode ser definido por meio da função recorrente $\alpha\left(t, y^{(t)}\right)$, que retorna a pontuação da melhor sequência de rótulos entre os episódios 1 e t, terminando com o rótulo $y^{(t)}$:

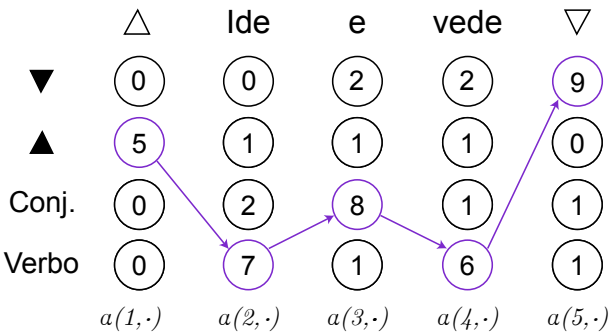
$$\alpha\left(t, y^{(t)}\right) = \begin{cases} g_1\left(\blacktriangle, y^{(t)}\right) & \text{, se } t = 1 \\ \alpha\left(t-1, y^{(t-1)}\right) + \max\limits_{\boldsymbol{y}^{(t-1)}} \left[g_t\left(\mathcal{X}, y^{(t-1)}, y^{(t)}\right)\right] & \text{, caso contrário.} \end{cases} \tag{35}$$

Por fim, o problema para determinar a maior pontuação de $\mathcal{X}$ se reduz a:

$$s\left(\mathcal{X}, \mathcal{Y}\right) = \alpha\left(\tau + 1, \blacktriangledown\right). \tag{36}$$

Para resolvê-lo, inicialmente a pontuação ótima de todas sequências com comprimento 1 é determinada por meio da primeira sentença da Equação 35, com $\alpha\left(1, \cdot\right)$. Isso possibilita então computar $\alpha\left(2, \cdot\right)$, já que esta expressão depende apenas do cálculo anterior. O mesmo procedimento é repetido sucessivamente até que $\alpha\left(\tau + 1, \blacktriangledown\right)$ seja alcançado. O caminho ótimo, que é o objetivo de real interesse, também pode ser acompanhado em conjunto com o cálculo das pontuações ao substituir o operador $\max$ por argmax na Equação 35. Com isso, basta percorrer a sequência na ordem inversa e ignorar os símbolos de início e fim da sentença para determinar a rotulação ótima $\mathcal{Y}^{\star}$. Este processo é ilustrado na Figura 15 por meio de uma estrutura gráfica comumente denominada *treliça*.

Figura 15 – Aplicação do Algoritmo de Viterbi para determinar a melhor rotulação de POS em uma sentença por meio do caminho ótimo.



Fonte: Elaborado pelo autor.

4.4 Treinamento

Os parâmetros $\boldsymbol{w}_j$ do CRF são aprendidos ao otimizar uma determinada função de custo, assim como ocorre com as redes neurais. Neste caso o logaritmo negativo da verossimilhança regularizada – do inglês *regularized negative log-likelihood*, é minimizado:

$$\begin{aligned} L(\mathcal{X}, \mathcal{Y}) &= -\log p(\mathcal{Y} \mid \mathcal{X}; \theta) + \frac{\lambda}{2}||\theta||_2^2 \\ &= -\log \left[\frac{1}{Z(\mathcal{X}, \boldsymbol{w})} \exp \sum_i \boldsymbol{w}_i F_i(\mathcal{X}, \mathcal{Y}) \right] + \frac{\lambda}{2}||\theta||_2^2 \\ &= -\sum_j \boldsymbol{w}_i F_i(\mathcal{X}, \mathcal{Y}) + \log Z(\mathcal{X}, \boldsymbol{w}) + \frac{\lambda}{2}||\theta||_2^2, \end{aligned} \tag{37}$$

em que a notação $||\cdot||_2$ corresponde à norma euclidiana, ou L2 e λ controla a severidade da regularização. Para otimizar o objetivo em questão $L(\cdot)$ é derivada em relação à cada uma de suas variáveis e os parâmetros do modelo têm seus valores atualizados por meio do método do gradiente descendente até a convergência, assim como ocorre durante o treinamento de uma rede neural. A seguir o processo para obter as derivadas de cada parâmetro $\boldsymbol{w}_i$ do modelo é apresentado:

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{w}_i} L(\mathcal{X}, \mathcal{Y}) &= \frac{\partial}{\partial \boldsymbol{w}_i} \left[\log Z(\mathcal{X}, \boldsymbol{w}) - \sum_j \boldsymbol{w}_j F_j(\mathcal{X}, \mathcal{Y}) \right] + \frac{\partial}{\partial \boldsymbol{w}_j} \frac{\lambda}{2}||\theta||_2^2 \\ &= \frac{\partial}{\partial \boldsymbol{w}_i} \log Z(\mathcal{X}, \boldsymbol{w}) - \frac{\partial}{\partial \boldsymbol{w}_i} \sum_j \boldsymbol{w}_j F_j(\mathcal{X}, \mathcal{Y}) + \lambda \boldsymbol{w}_j \\ &= \frac{\partial}{\partial \boldsymbol{w}_i} \log Z(\mathcal{X}, \boldsymbol{w}) - F_i(\mathcal{X}, \mathcal{Y}) + \lambda \boldsymbol{w}_i. \end{aligned} \tag{38}$$

A derivada da função de partição é calculada separadamente para simplificar o processo:

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{w}_i} \log\left[Z\left(\mathcal{X}, \boldsymbol{w}\right)\right] &= \frac{1}{Z(\mathcal{X}, \boldsymbol{w})} \frac{\partial}{\partial \boldsymbol{w}_i} Z\left(\mathcal{X}, \boldsymbol{w}\right) \\
&= \frac{1}{Z\left(\mathcal{X}, \boldsymbol{w}\right)} \frac{\partial}{\partial \boldsymbol{w}_i} \left[\sum_{\mathcal{Y}' \in \mathbb{Y}(\mathcal{X})} \exp \sum_j \boldsymbol{w}_j F_j\left(\mathcal{X}, \mathcal{Y}'\right)\right] \\
&= \frac{1}{Z\left(\mathcal{X}, \boldsymbol{w}\right)} \sum_{\mathcal{Y}' \in \mathbb{Y}(\mathcal{X})} \left[\frac{\partial}{\partial \boldsymbol{w}_i} \exp \sum_j \boldsymbol{w}_j F_j\left(\mathcal{X}, \mathcal{Y}'\right)\right] \\
&= \frac{1}{Z(\mathcal{X}, \boldsymbol{w})} \sum_{\mathcal{Y}' \in \mathbb{Y}(\mathcal{X})} \left[\exp \sum_j \boldsymbol{w}_j F_j\left(\mathcal{X}, \mathcal{Y}'\right)\right] F_i\left(\mathcal{X}, \mathcal{Y}'\right) \\
&= \sum_{\mathcal{Y}' \in \mathbb{Y}(\mathcal{X})} \frac{1}{Z\left(\mathcal{X}, \boldsymbol{w}\right)} \left[\exp \sum_j \boldsymbol{w}_j F_j\left(\mathcal{X}, \mathcal{Y}'\right)\right] F_i\left(\mathcal{X}, \mathcal{Y}'\right) \\
&= \sum_{\mathcal{Y}' \in \mathbb{Y}(\mathcal{X})} \frac{\exp \sum_j \boldsymbol{w}_j F_j\left(\mathcal{X}, \mathcal{Y}'\right)}{\sum_{\mathcal{Y}'' \in \mathbb{Y}(\mathcal{X})} \left[\exp \sum_k \boldsymbol{w}_k F_k\left(\mathcal{X}, \mathcal{Y}''\right)\right]} F_i\left(\mathcal{X}, \mathcal{Y}'\right) \\
&= \sum_{\mathcal{Y}' \in \mathbb{Y}(\mathcal{X})} p\left(\mathcal{Y}' | \mathcal{X}; \theta\right) F_i\left(\mathcal{X}, \mathcal{Y}'\right).
\end{aligned} \tag{39}$$

Ao substituir a derivada da função de partição na Equação 38 é possível definir a derivada da função de custo a respeito de cada parâmetro como:

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{w}_i} L\left(\mathcal{X}, \mathcal{Y}\right) &= \frac{\partial}{\partial \boldsymbol{w}_i} \log Z\left(\mathcal{X}, \boldsymbol{w}\right) - F_i\left(\mathcal{X}, \mathcal{Y}\right) + \lambda \boldsymbol{w}_i \\
&= \sum_{\mathcal{Y}' \in \mathbb{Y}(\mathcal{X})} p\left(\mathcal{Y}' | \mathcal{X}; \theta\right) F_i\left(\mathcal{X}, \mathcal{Y}'\right) - F_i\left(\mathcal{X}, \mathcal{Y}\right) + \lambda \boldsymbol{w}_i.
\end{aligned} \tag{40}$$

Em palavras, a derivada de $L(\cdot)$ em relação ao i-ésimo parâmetro consiste na diferença entre o valor médio da mesma função de característica para todas as sequências possíveis $\mathcal{Y}'$ e a pontuação gerada pela função de característica $F_i(\cdot)$ para a sequência de rótulos real $\mathcal{Y}$ (ELKAN, 2008). Esta expressão pode ser computada manualmente por meio do Algoritmo Frente-Trás – do inglês *Forward-Backward Algorithm*, conforme detalhado por Gupta (GUPTA, 2006), o qual também apresenta outros regimes de treinamento para o modelo. Alternativamente é possível calcular diretamente a função de custo da Equação 37 e depois utilizar técnicas de diferenciação automática, presentes em bibliotecas como `PyTorch` e `TensorFlow`, para que suas derivadas, ou seja, a Equação 40 seja avaliada automaticamente (EISNER, 2016).

Independente da estratégia utilizada, torna-se necessário avaliar a função de partição na Equação 37. Entretanto devido à restrição imposta de que as funções de características podem observar apenas rótulos adjacentes para emitir sua pontuação, esta operação pode ser realizada eficientemente novamente ao utilizar Programação Dinâmica por meio de uma

variante do Algoritmo de Viterbi, denominado Algoritmo de Propagação, o qual corresponde à primeira fase do Algoritmo Frente-Trás, conforme apresentado por Elkan (ELKAN, 2008).

A vantagem do método da diferenciação automática se traduz na facilidade de acoplamento entre o CRF e as redes neurais para a extração de características e classificação, conforme ilustrado na Figura 14. Nesta situação θ não é formado apenas pelos pesos w_j que determinam a importância de cada função característica do CRF, mas também por todos os parâmetros que formam a rede neural subjacente. Consequentemente, $L(\mathcal{X}, \mathcal{Y})$ deve ser derivada em relação aos demais parâmetros a fim de permitir que o modelo seja treinado de ponta a ponta.

4.5 Considerações Finais

As redes neurais recorrentes permitem combinar características de episódios passados durante a análise atual, seja para gerar o descritor de uma sequência de palavras, por exemplo, em uma fala, ou também para classificar cada um de seus episódios, como no caso do etiquetador morfossintático. O CRF pode ser utilizado com o objetivo de permitir a contextualização de informações não apenas na parte oculta das redes neurais, onde ocorre a extração de características, mas também na camada de saída.

5 Análise contextual

O presente capítulo discute como as redes neurais apresentadas podem ser utilizadas com o objetivo de extrair representações robustas de sentenças para a tarefa de Análise de Sentimento, de modo que cada palavra, ou sequência de palavras, possa influenciar de maneira diferente a composição do descritor. Proceder desta maneira permite que os recursos de linguagem sejam refletidos, mesmo que indiretamente, no vetor final calculado. As representações obtidas podem ser utilizadas para a classificação individual de sentenças ou combinadas para que esta mesma tarefa seja realizada a nível de documento ou em turnos de um diálogo, conforme será apresentado a seguir.

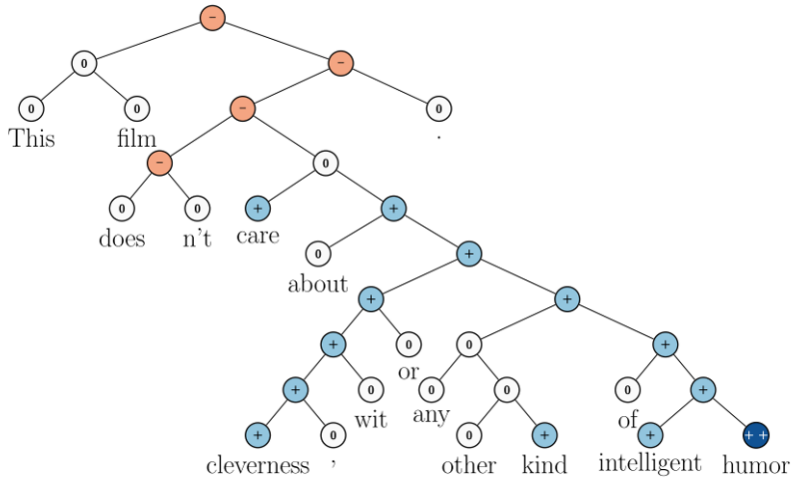
A segunda parte do capítulo tem por início a discussão de trabalhos que realizam AS em diálogos. Tendo em vista a existência de poucos estudos que abordam o problema de interesse, são apresentadas brevemente algumas outras tarefas realizadas em transcrições de diálogos com o objetivo de traçar paralelos com a tarefa em questão. A seguir, são examinados trabalhos que incorporam informação de contexto para outras aplicações baseadas em conversas, como a classificação de atos de diálogo. Por fim, a seção se encerra ao discutir a natureza sequencial do problema de AS tanto em documentos como em diálogos, motivando a utilização dos Campos Aleatórios Condicionais, seja no problema de interesse ou para outras aplicações em PLN.

5.1 Representação de Sentenças

A forma mais simples de computar o descritor de uma sentença consiste na média de seus *word embeddings*, seja esta ponderada (ARORA; LIANG; MA, 2017) ou não (BOJANOWSKI et al., 2016). Entretanto, representações mais robustas podem ser obtidas ao levar em consideração a ordenação das palavras na sentença, bem como sua árvore sintática, através de diferentes tipos de redes neurais.

A fim de contemplar o segundo aspecto, Socher et al. (SOCHER et al., 2013) utilizam uma Rede Neural Tensorial Recursiva, a qual permite combinar o sentimento relacionado a cada palavra, ou segmento da sentença, seguindo a composição de sua árvore sintática. Ao proceder dessa maneira é possível identificar negações de sentimentos tanto positivos quanto negativos, bem como mudanças de polaridade, conforme ilustrado na Figura 16. Na ausência da árvore sintática das amostras, Socher et al. (SOCHER et al., 2011) propõem uma abordagem semelhante, a qual consiste em utilizar Auto-Codificadores Recursivos de maneira não supervisionada para que esses sejam capazes de reconstruir a estrutura da sentença, a qual não necessariamente corresponde a sua árvore sintática, e consequentemente gerar sua representação vetorial. Essa última informação é utilizada em uma camada softmax para realizar a classificação quanto a seu sentimento.

Figura 16 – Anotação de sentimento em cada nível da árvore sintática de uma sentença.



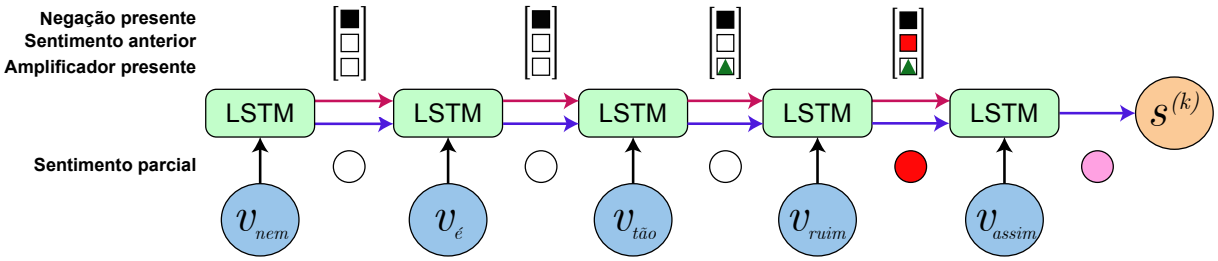
Fonte: (SOCHER et al., 2013).

Apesar das representações obtidas por ambos modelos serem robustas, torna-se difícil determinar a estrutura hierárquica de mensagens curtas e irregulares (WANG et al., 2015), situação presente em turnos de uma conversa conduzida no domínio textual. Para obter vetores de sentenças de *tweets* utilizados na Análise de Sentimento, Wang et al. (WANG et al., 2015) utilizam uma rede neural com unidades LSTM. Nesta arquitetura, cada palavra é mapeada para seu *word embedding* e apresentada de forma sequencial à rede. Como cada novo estado é produzido ao considerar uma nova palavra em conjunto com a memória interna da unidade e seu estado oculto anterior, os quais sintetizam a composição de palavras vistas no passado, é possível modelar construções que utilizem sequências de vocábulos graças a natureza desse tipo de unidade.

Para exemplificar essa formulação, considere que deseja-se obter a representação vetorial do trecho de uma frase, por exemplo, *"nem é tão ruim assim"*, através de uma rede formada por apenas uma camada LSTM para a classificação quanto a seu sentimento, conforme ilustrado na Figura 17. Inicialmente a palavra *"nem"* é lida, a qual indica que uma negação está por vir. Esta informação pode ser codificada de maneira conveniente na memória da unidade $\boldsymbol{c}^{(1)}$, representada pelo quadrado preto no vetor da figura, enquanto seu estado oculto $\boldsymbol{\gamma}^{(1)}$ ainda não armazenou nenhum aspecto relacionado à polaridade, representado pelo círculo branco. A seguir, a palavra "é", a qual não é de grande relevância para a tarefa em questão, é lida, os estados internos da unidade LSTM são atualizados e a rede observa a palavra *"tão"*. Como esta palavra corresponde a um intensificador, tal informação é mantida na memória $\boldsymbol{c}^{(3)}$ (seta verde para cima no vetor), já que trata-se de um termo relevante para a tarefa em questão, apesar de nenhum sentimento ter sido detectado ainda (repare que o círculo continua branco). Esta situação muda ao ler a próxima palavra, *"ruim"*, a qual tem seu caráter negativo combinado com a informação de intensificação presente na memória da rede neural, gerando o

sentimento "muito ruim", representado pelo círculo vermelho na imagem. Durante o próximo passo, a última palavra, *"assim"*, é lida e o sentimento da sentença é atualizado ao combinar a negação com o sentimento anterior, atenuando-o para *"ligeiramente ruim"* e representado pelo círculo rosa. Finalmente, o último estado oculto da rede, $\gamma^{(5)}$ é utilizado como descritor da sentença por levar em consideração seus traços semânticos composicionais, no caso, negação e amplificação, e pode ser encaminhando ou para as camadas mais profundas do modelo ou utilizado diretamente para a classificação da amostra.

Figura 17 – Arquitetura de uma rede utilizada para classificação contextual de documentos. As setas coloridas representam a memória da rede e seu estado oculto.



Fonte: Elaborado pelo autor.

Ainda no que diz respeito à obtenção do descritor de amostras por redes recorrentes, suas variantes discutidas no Capítulo 3 também podem ser empregadas, como é o caso de redes bidirecionais, seu empilhamento e mecanismos de atenção. Neste âmbito, Baziotis, Pelekis e Doulkeridis (BAZIOTIS; PELEKIS; DOULKERIDIS, 2017) utilizam as três abordagens em conjunto para treinar duas redes neurais para competição sobre Análise de Sentimento em mensagens do Twitter organizada pela Oficina Internacional em Avaliação Semântica (ROSENTHAL; FARRA; NAKOV, 2017). O primeiro modelo é utilizado para a tarefa de AS a nível de sentença (no caso, *tweet*) e o segundo a nível de aspecto.

Apesar de não possuírem memória para identificar construções maiores que os filtros utilizadas, além de restringir o tamanho das sentenças, CNNs também podem ser utilizadas para computar descritores de sentenças e até mesmo serem combinadas com suas variantes recorrentes. Cliche (CLICHE, 2017) e Wang (WANG et al., 2016a) exploram esta abordagem híbrida, onde inicialmente a CNN identifica sequências de palavras relevantes para o problema, que a seguir são combinadas por meio de uma LSTM para AS em *tweets*.

É interessante notar que nas técnicas de AS que utilizam representações baseadas em BoW, ou mesmo em dicionário, existe uma distinção bastante clara entre as etapas de extração de características e desenvolvimento do modelo. Esta margem torna-se menos evidente ao utilizar técnicas de aprendizado de máquina baseadas em redes neurais, onde as primeiras camadas dos modelos, responsáveis pela extração de características, já são desenvolvidas com a atividade final em mente. Além disso, as mesmas são treinadas em conjunto com o modelo de forma geral, tornando-as especialistas naquela tarefa de interesse. Devido às possibilidades

relacionadas aos tipos de camada, direção de leitura da sequência, utilização de mecanismos de atenção, bem como sua hierarquia, existe uma diversidade muito maior de possibilidades a serem exploradas ao realizar AS, bem como outras tarefas, utilizando redes neurais.

Um exemplo desse caso é a LSTM em árvore – do inglês *Tree-LSTM*, proposta por Tai, Socher e Manning (TAI; SOCHER; MANNING, 2015), a qual consiste em uma generalização das unidades LSTM, com natureza sequencial, para estrutura em árvores, tornando-as plausíveis, do ponto de vista linguístico devido à sua relação com a árvore sintática de sentenças. Os autores observam ganhos de acurácia ao utilizá-las tanto para a Análise de Sentimento em sentenças como para determinar a similaridade entre sentenças. Outro modelo extremamente complexo para a realização de AS a nível de sentença, bem como uma série de outras tarefas relacionadas ao PLN é apresentado por McCann et al. (MCCANN et al., 2017), ao combinar *word embeddings*, um modelo de tradução neural, redes bidirecionais, mecanismos de atenção e quantização.

5.2 Utilização de Informação Sobre o Contexto

A maioria dos métodos apresentados na literatura realizam a Análise de Sentimento contextual a partir de textos, como análises de produtos e filmes, tendo em vista a disponibilidade de bases de dados desse tipo (TANG; QIN; LIU, 2015; DAI; LE, 2015). No tocante aos diálogos, a tarefa mais comumente abordada na literatura é a classificação contextual de turnos em termos de seu objetivo ou intenção (STOLCKE et al., 2000; LEE; DERNONCOURT, 2016), denominada classificação de atos de diálogo – do inglês *Dialogue Act classification* (DA). Um exemplo deste tipo de problema é apresentado na Tabela 1. Outras tarefas que abordam diálogos incluem determinar a identidade do participante que emitiu uma fala dentro de um diálogo (MA; XIAO; CHOI, 2017) e, mais recentemente, a correspondência entre menção e indivíduo em transcrições de diálogos a partir do contexto em que surgem (CHOI; CHEN, 2018), conforme exemplificado na Tabela 2.

Tabela 1 – Exemplo de DA em um diálogo entre dois participantes.

Participante	**Fala**	**Rótulo**
B	Então eu tenho que andar para frente no início?	Verificação
A	Mmhmm.	Confirmação
B	E então virar para a direita...	Verificação
A	Isso.	Confirmação
A	Siga em frente apenas alguns passos.	Clarificação
B	Mmhmm.	Confirmação
B	Eu tenho que passar pelo poço?	Pergunta

Zahiri e Choi (ZAHIRI; CHOI, 2018) argumentam que uma das dificuldades em realizar Análise de Sentimento em diálogos está relacionada à escassez de bases de dados suficientemente grandes e anotadas para tal finalidade. Com o intuito de remediar esse problema, os autores

Tabela 2 – Exemplo de ligação entidade-menção em trechos de conversas. As falas não pertencem ao mesmo diálogo. Nomes entre parêntese devem ser ligados à palavra anterior a partir de seu contexto.

Personagem	**Fala**
Ross	Eu (Ross) disse à mamãe (Judy) e ao papai (Jack) ontem a noite e eles pareceram aceitar bem.
Joey	Ok Ross (Ross), olha você (Ross) está sentindo muita dor agora. Você (Ross) está bravo. Você (Ross) está sofrendo. Eu (Joey) posso te (Ross) dizer a resposta?
Ross	Me desculpe.

apresentam um conjunto de dados obtido a partir da transcrição de diálogos em inglês entre múltiplos participantes da série *Friends* anotado com sete emoções: triste, furioso, susto, eufórico, tranquilo, alegre e neutro. Devido à especificidade do esquema de anotação empregado, a base de dados pode ser utilizada em dois níveis de granularidade: um menor, onde o problema de classificação é formado pelas sete classes originais, e outro maior, ao agrupar os sentimentos em três rótulos mais abrangentes: positivo (formado pelas classes eufórico, tranquilo e alegre), negativo (formado pelas classes triste, furioso e susto) e neutro. A Tabela 3 apresenta um trecho de uma das cenas anotadas. Na análise realizada no período entre 2000 e 2015 sobre os trabalhos que abordam a tarefa de AS por Piryani, Madhavi e Singh (PIRYANI; MADHAVI; SINGH, 2017) não são reportadas bases de dados com tal fim. Todavia, ainda mais recentemente um segundo conjunto de dados, denominado *Emotionlines* (CHEN et al., 2018), foi disponibilizado publicamente aos moldes do trabalho desenvolvido pelos autores anteriores. Como ambas bases foram consideradas nos experimentos realizados, uma discussão mais aprofundada sobre as mesmas é reservada à Seção 6.1.

Tabela 3 – Transcrição da conversa em parte de uma cena da série *Friends*.

Personagem	**Fala**	**Sentimento**
Mônica	Ele é tão bonito. Então, onde vocês cresceram?	Alegre
Angela	Brooklyn Heights.	Neutro
Bob	Cleveland.	Neutro
Mônica	Como, como isso aconteceu?	Neutro
Joey	Ah, meu Deus!	Susto
Mônica	O que foi?	Neutro
Joey	Eu tive a sensação repentina de que eu estava caindo. Mas eu não estou.	Susto

Fonte: Adaptado de Zahiri e Choi (ZAHIRI; CHOI, 2018).

Também é importante mencionar a existência do conjunto de dados utilizado no *Audio/Visual Emotion Challenge and Workshop (AVEC) 2017* (RINGEVAL et al., 2017), o qual é formado por gravações de vídeo, áudio e a transcrições dos diálogos entre pares de participantes que discutem sobre um certo vídeo exibido a eles. Diferentemente da abordagem estudada na presente dissertação de mestrado, onde cada turno de uma conversa é visto como um evento atômico e que deve ser atribuído a uma classe, que por sua vez também é um elemento discreto, neste cenário a conversa é vista como uma interação contínua. Adicionalmente, a informação de

sentimento é decomposta em três elementos, também contínuos: valência, excitação, afinidade, abordagem análoga àquela apresentada na Figura 1. O objetivo da tarefa, por sua vez, consiste em treinar um regressor com base nas informações dos três domínios disponíveis de modo que o mesmo produza uma série temporal para cada uma das dimensões de sentimento. Em outras palavras, deseja-se produzir sequências de valores contínuos e ordenados cronologicamente para todo o decorrer da conversa para os três domínios separadamente.

No que concerne a trabalhos relacionados à Analise de Sentimentos em diálogos, Murray e Carenini (MURRAY; CARENINI, 2011), dentre outras tarefas, classificam turnos de um diálogo como subjetivos ou objetivos ao empregar uma extensão da técnica *bag of words* que também considera informação morfossintática, denominada Instanciação Variacional de n-gramas, em conjunto com outros aspectos extraídos das palavras, totalizando mais de 200 mil características. Adicionalmente são consideradas informações obtidas a partir dos turnos no diálogos, por exemplo, sua posição relativa na conversa, extensão em termos de palavras e predominância de cada um dos participantes na conversa. Por meio dos experimentos realizados, os autores observam que as 24 características extraídas dos turnos desempenham papel fundamental para a obtenção de bons resultados de classificação, sendo quase tão efetivas quanto as mais de 200 mil restantes. É importante mencionar que os autores não treinam um único classificador capaz de rotular amostras como positivas, negativas ou neutras, mas consideram dois cenários distintos: no primeiro o classificador é projetado para segmentar as amostras positivas das demais, e no segundo caso o mesmo cenário se repete para as amostras negativas.

Muralidhar (MURALIDHAR, 2013) realiza esta tarefa em segmentos de transcrições formados por múltiplos turnos de diálogos ao considerar dois *corpora*: um composto por diálogos entre pais e filhos e outro por pares de adultos. Cada amostra tem seu descritor computado a partir da soma ponderada de seus *word embeddings*, que são obtidos a partir de uma base de conhecimento baseada em grafos, denominada ConceptNet (SPEER; HAVASI, 2012). A autora observa que existe pouca variação de sentimento ao longo do diálogo na primeira base, provavelmente devido à simplicidade das conversas. Já no segundo caso, uma variação um pouco maior é identificada. Apesar disso, como ambas bases não foram rotuladas com o propósito de Análise de Sentimento, torna-se difícil realizar uma análise mais profunda sobre os resultados obtidos. Ademais, o objetivo do trabalho está mais direcionado à exploração e análise de dados, de forma que nenhum estudo experimental é realizado a fim de comparar esta abordagem com outras alternativas. Por outro lado, Ojamaa, Jokinen e Muischenk (OJAMAA; JOKINEN; MUISCHENK, 2015) realizam AS em transcrições de diálogos em estoniano ao utilizar uma abordagem baseada em dicionário para classificar cada turno individualmente, ou seja, a informação de contexto não é considerada.

O método de Análise de Sentimento contextual em conversas mais recente encontrado na literatura, desenvolvido por Zahiri e Choi (ZAHIRI; CHOI, 2018), utiliza diferentes tipos de

Redes Neurais Convolucionais Sequenciais, propostas pelos autores, as quais permitem utilizar descritores extraídos em classificações passadas em conjunto com um mecanismo de atenção para a classificação do sentimento na fala atual. Os dois níveis de granularidade da base *Friends* são considerados, de modo que os autores obtém acurácia de 37.9% ao considerar sete classes, e 54.0% ao considerar 3 classes, indicando a existência de espaço para a obtenção de melhores resultados.

Tran, Zukerman e Haffari (TRAN; ZUKERMAN; HAFFARI, 2017) exploram o contexto das conversas para a tarefa de classificação quanto aos atos de diálogo de seus turnos por meio de uma rede neural hierárquica, onde cada camada é responsável por extrair informações de um nível diferente da interação. Inicialmente, cada fala tem seu descritor computado a partir de seus *word embeddings* por uma camada LSTM bidirecional acrescida de um mecanismo de atenção, o qual também considera a distribuição de probabilidades de classificação emitida durante a análise do turno anterior. Isso pode ser feito ao conectar os nós $\hat{\boldsymbol{y}}^{(\cdot)}$ na Figura 11 ao mecanismo de atenção A. O descritor obtido é utilizado na próxima camada, formada por LSTMs unidirecionais a nível de conversa, encarregada de sintetizar o contexto formado por turnos anteriores. Além do modelo desenvolvido obter melhores resultados quando comparado a outras abordagens que não consideram o contexto, os autores realizam uma série de testes de ablação, onde partes da arquitetura gerada são removidas para verificar como isso impacta seu desempenho, permitindo evidenciar componentes desnecessários. A análise realizada permitiu identificar que considerar o rótulo anterior para a classificação atual trouxe ganhos de acurácia estatisticamente significativos ao modelo.

Lee e Dernoncourt (LEE; DERNONCOURT, 2016) abordam o mesmo problema também incorporando informação de contexto, porém através de uma forma mais simples. Os autores computam o descritor de uma fala ao apresentar os *word embeddings* de suas palavras a uma rede CNN ou LSTM, que é então propagado para uma segunda rede neural completamente conexa de duas camadas (vista no Capítulo 3.1) em conjunto com as representações obtidas para as p interações anteriores. Proceder dessa maneira permite que informações de contexto, presentes em turnos anteriores, sejam levadas em consideração durante a classificação com pesos diferentes, fazendo com que os autores alcancem resultados melhores. Ganhos de acurácia também foram obtidos por Ma et al. (MA; XIAO; CHOI, 2017) ao utilizarem uma abordagem bastante similar baseada em CNN para a identificação de personagens em conversas entre múltiplos participantes a partir das transcrições de fala da série *Friends*. As melhorias nos resultados são obtidas ao considerar duas sentenças antecessoras e uma sucessora a atual, e também ao agrupar todas emitidas pelo mesmo personagem em um único turno na mesma amostra para análise.

Apesar de levar em consideração representações extraídas de sentenças anteriores, os estudos examinados desprezam rótulos preditos anteriormente. Todavia, este histórico pode ser explorado no sentido de preservar a continuidade do diálogo, além de permitir extrair informações

relevantes para a classificação atual. Tal aspecto é explorado por Mao e Lebanon (MAO; LEBANON, 2006) ao utilizar uma variante do CRF para classificar o sentimento de sentenças que compõem a análise de filmes. Todavia, o método proposto pelos autores é baseado em dicionário. Uma modelagem bastante parecida é realizada por Yang e Cardie (YANG; CARDIE, 2014), entretanto o CRF é acrescido de Regularização Posterior (GANCHEV et al., 2010), na qual um conjunto de funções de penalização, desenvolvidas manualmente, são adicionadas à função de custo do modelo com o objetivo de influenciar suas decisões. Assim, o classificador é penalizado, por exemplo, ao atribuir duas sentenças conectadas por um termo comparativo à mesma classe, já que neste cenário normalmente tem-se sentimentos contrastantes. Contudo, as características utilizadas pelo CRF ainda são baseadas em dicionário e as regras devem ser projetadas manualmente.

Abordagens similares são utilizadas em trabalhos que realizam AS a nível de aspecto, ou seja, consideram sentenças individualmente e buscam identificar quais palavras são utilizadas na expressão de sentimentos. Como cada vocábulo deve ser classificado, o problema é visto como uma rotulação sequencial. Um exemplo desse tipo de tarefa é apresentado na Tabela 4 ao utilizar o esquema de rotulação BIO (RAMSHAW; MARCUS, 1999), onde cada palavra é identificada como início (*begin*), dentro (*inside*) ou fora (*outside*) de sequências relacionadas a opinião, ao passo que o marcador de expressão (*expression*) indica o sentimento, enquanto o marcador de alvo (*target*) sinaliza o aspecto sob análise. Yang e Cardie (YANG; CARDIE, 2012) relaxam a restrição de que cada palavra deve ser rotulada individualmente e passam a classificar segmentos obtidos a partir da árvore sintática da sentença em uma abordagem denominada Campos Aleatórios Condicionais Semi-Markovianos. Todavia, as características consideradas para a classificação também são baseadas em dicionário.

Tabela 4 – Rotulação a nível de aspecto ao utilizar o esquema de anotação BIO.

Esse	**disco**	**rígido**	é	*bastante*	*barulhento*
O	B-*target*	I-*target*	O	O	O
O	O	O	O	B-*expression*	I-*expression*

Adaptado de (LIU; JOTY; MENG, 2015).

A utilização de técnicas que levam em consideração rótulos vistos anteriormente para classificar novas amostras é frequente no tratamento de outros problemas na área de PLN, como no desenvolvimento de etiquetadores morfossintáticos, ou para o Reconhecimento de Entidades Nomeadas – do inglês *Named Entity Recognition* (NER), em que dado uma frase, deseja-se identificar palavras que correspondem a nomes próprios e lugares, por exemplo. Com o objetivo de se beneficiar das vantagens trazidas pelos métodos baseados em aprendizado de máquina e considerar informação sequencial a nível de rótulos, a qual é explorada por métodos baseados em CRF, ambas técnicas podem ser combinadas (LAMPLE et al., 2016; LIU et al., 2017). Ma e Hovy (MA; HOVY, 2016) reportam ganhos de acurácia nas tarefas de POS e NER ao adicionar uma camada CRF à sua rede neural, baseada em CNN e LSTM. Ao realizar

essa combinação para o reconhecimento de entidades, Huang et al. (HUANG; XU; YU, 2015) argumentam que este tipo de arranjo permite o uso eficiente de informações passadas por meio da camada LSTM, ao passo que aspectos a nível de sentença são combinadas pela camada CRF. Logo, estender essa mesma combinação a um nível hierárquico maior, onde as sentenças são classificadas quanto a seu sentimento e características de contexto são exploradas pelo CRF pode ser uma área de estudo promissora.

5.3 Considerações Finais

Por fim, o presente capítulo apresentou algumas técnicas utilizadas para o cálculo dos descritores de sentença contextuais, principalmente aqueles baseadas em redes neurais recorrentes, os quais consideram além das palavras na frase, sua ordem e vizinhança, a fim de contemplar o princípio da composicionalidade semântica, mesmo que parcialmente. Posteriormente foram analisados alguns trabalhos que, de alguma forma, tentam incorporar informações contextuais para a classificação de interesse, seja a nível de características, por meio de uma segunda camada recorrente, ou durante a etapa de classificação, através do CRF na camada de saída. A partir da análise dos trabalhos é possível concluir que explorar este tipo de informação tem beneficiado técnicas desenvolvidas para outros problemas na área de PLN, o que motiva a decisão por incorporar tais aspectos na tarefa de AS em diálogos, tanto a nível de extração de características, como de classificação.

6 Metodologia

O capítulo em questão tem por objetivo descrever as bases de dados utilizadas nos experimentos, bem como apresentar análises realizadas sobre estes conjuntos de dados e as das técnicas de pré-processamento utilizadas. Posteriormente, o modelo proposto e cada um de seus módulos são discutidos detalhadamente de forma incremental. Por fim, o protocolo de avaliação empregado para comparar os modelos é exposto.

6.1 Bases de Dados

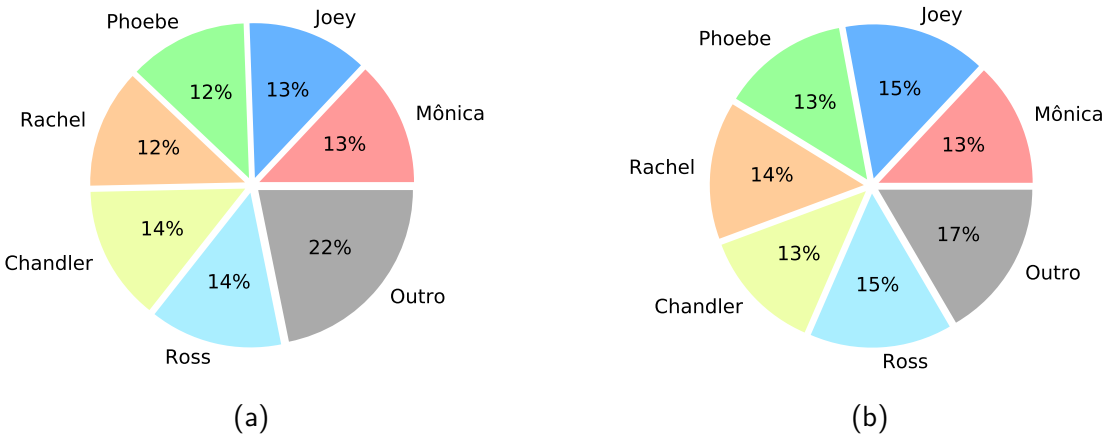
Devido ao problema de AS em diálogos ser recente na literatura, existem poucas bases de dados formadas por transcrições de conversas e anotadas devidamente com esta finalidade em mente. Assim, foram considerados dois conjuntos para validar a abordagem proposta na presente dissertação de mestrado: *Emory* e *Emotionlines*. Em todos os casos, as bases consistem ou em transcrições de diálogos da série *Friends* entre um (monólogo) ou mais participantes, ou em fragmentos de conversas entre pares de indivíduos, as quais foram obtidas por meio da plataforma *Facebook Messenger*. Neste último caso, os autores da base de dados se responsabilizaram por obter o consentimento dos participantes para registrar seus diálogos, além de anonimizar suas identidades. Todas as interações ocorreram em inglês.

Além das particularidades de cada conjunto de dados, a presença de recursos de linguagem, como gírias e metáforas, contribuem para aumentar a dificuldade do problema em questão. Adicionalmente, vale lembrar que a tarefa de categorizar o sentimento presente em fragmentos de texto é subjetiva, tendo em vista que diferentes indivíduos podem atribuir sentimentos distintos à mesma amostra de acordo com critérios pessoais. Essa característica se torna ainda mais proeminente devido à ausência de outras fontes de informação, como um contexto mais amplo do diálogo, ou, mais notavelmente na AS em transcrições, a impossibilidade de considerar o tom de voz do interlocutor. Tais conclusões são apresentadas pelos autores das bases de dados ao observar um grau de concordância baixo entre os rotuladores por meio das estatísticas Kappa de Cohen (COHEN, 1960) e de Fleiss (FLEISS, 1971), onde o primeiro coeficiente permite calcular a concordância apenas entre dois avaliadores, ao passo que o segundo estende esta análise a um grupo maior de indivíduos. De forma geral, todos autores observam que aumentar o tamanho do comitê de anotação melhora a concordância observada.

Como metade dos conjuntos consideram diálogos extraídos da série *Friends*, ressaltamos aqui alguns detalhes relevantes para os experimentos realizados. Devido ao nosso modelo completo considerar o interlocutor de cada fala, consideramos as identidades dos seis personagens principais do *show*, Mônica, Chandler, Rachel, Ross, Phoebe e Joey, ao passo que os demais participantes são mapeados a uma identidade genérica denominada "Outro". Tal decisão foi

tomada com base em uma análise realizada no conjunto de dados *Emory*, onde foi possível observar que o personagem principal com o menor número de falas (Phoebe) contribui com 365 amostras (12% do total da base de dados), ao passo que o personagem secundário mais frequente tem apenas 30 falas (menos de 1% do total de turnos). A mesma tendência também foi observada na segunda base de dados, conforme exibido na Figura 18.

Figura 18 – Participação dos interlocutores da série *Friends* em diferentes bases de dados: (a) *Emory*, (b) *Emotionlines Friends*.



(a) (b)

Fonte: Elaborado pelo autor.

6.1.1 Emory

Até onde pudemos identificar, a primeira base de dados que permite realizar AS em diálogos disponibilizada publicamente foi gerada por Zahiri e Choi (ZAHIRI; CHOI, 2018), a qual denominamos *Emory*[15]. Este conjunto consiste na anotação quanto ao sentimento das transcrições dos diálogos que formam as cenas das primeiras quatro temporadas da série *Friends*. Cada fala, ou turno, de um personagem é rotulado com uma das seis emoções primárias de Willcox (WILLCOX, 1982): tristeza, fúria, assustado, eufórico, tranquilidade e alegria, além de um rótulo adicional correspondente ao sentimento neutro. As mesmas emoções também são agrupados em três classes maiores: positivo (formada pelas classes eufórico, tranquilo e alegre), negativo (formada pelas classes triste, furioso e susto) e neutro, permitindo realizar a mesma tarefa em um nível macroscópico. A base de dados é formada por 12.606 turnos divididos entre 879 cenas, as quais devem ser distribuídas entre os conjuntos de treinamento, validação e teste, conforme detalhado na Tabela 5.

A Tabela 6, por sua vez, apresenta a frequência dos rótulos, tornando possível observar que existe um desbalanceamento de classes considerável a nível microscópico (sete classes), ao passo que a situação é atenuada no nível macroscópico (três classes), o que dificulta o

[15] Disponível em <https://github.com/emorynlp/emotion-detection>.

Tabela 5 – Distribuição de turnos para o *Emory Dataset* em treinamento, validação e teste.

7 emoções	**Treinamento**	**Validação**	**Teste**	**3 emoções**	**Treinamento**	**Validação**	**Teste**
Eufórico	784	134	145				
Tranquilidade	899	132	159	Positivo	3.867	555	586
Alegria	2.184	289	282				
Susto	1.286	178	182				
Fúria	1.076	143	113	Negativo	3.033	396	393
Tristeza	671	75	98				
Neutro	3.034	393	349	Neutro	3.034	393	349
Total	**9.934**	**1.344**	**1.328**	**Total**	**9.934**	**1.344**	**1.328**

problema de classificação. Por fim, cada interlocutor é representado de forma razoavelmente igual em termos de turnos nas duas bases de dados, conforme ilustrado na Figura 18.

Tabela 6 – Frequência de rótulos para o *Emory Dataset*.

7 emoções	**Turnos**	**Frequência**	**3 emoções**	**Turnos**	**Frequência**
Eufórico	1.063	8,43%			
Tranquilidade	1.191	9,45%	Positivo	5.009	39,74 %
Alegria	2.755	21,86%			
Susto	1.645	13,05%			
Fúria	1.332	10,57%	Negativo	3.821	30,31%
Tristeza	844	6,69%			
Neutro	3.776	29,95%	Neutro	3.776	29,95%

6.1.2 *Emotionlines*

A base de dados *Emotionlines*[16] (CHEN et al., 2018) é formada por diálogos advindos de duas fontes distintas: conversas obtidas por meio da plataforma *Facebook Messenger* com o consentimento e anonimização de seus interlocutores, as quais foram cedidas por Wang et al. (WANG et al., 2016b); e um conjunto de transcrições da série *Friends* diferente daquele empregado no *Emory Dataset*. Assim, optamos por designar o primeiro conjunto por *Emotionlines Facebook* e o segundo por *Emotionlines Friends*. Os autores realizaram a anotação de cada turno em ambos casos quanto ao seu sentimento utilizando, ao invés dos rótulos propostos por Willcox (WILLCOX, 1982), as emoções apresentadas por Ekman (EKMAN et al., 1987): felicidade, tristeza, medo, fúria, susto e aversão, além de uma classe adicional correspondente ao sentimento neutro. Por fim, amostras com empate quanto ao seu rótulo por parte dos anotadores foram atribuídas a uma sétima classe, denominada "não-neutro". Com a finalidade de comparar o modelo proposto com outros resultados obtidos a partir desta base de dados, consideramos os experimentos submetidos à competição EmotionX (HSU; KU, 2018), a qual é mais recente que o trabalho original. Ao invés de considerar as sete classes iniciais, os modelos competidores devem classificar apenas sentenças correspondentes aos sentimentos alegria, fúria, tristeza e neutro, os quais têm suas frequências exibidas nas Tabelas 7 e 8 referente aos diálogos extraídos da série *Friends* e do *Facebook Messenger*, respectivamente. No que diz respeito a considerar a identidade dos interlocutores em nosso modelo, realizamos os mesmos procedimentos utilizados nos diálogos do conjunto *Emory* e a quantidade de turnos por participante é apresentada na Figura 18b. Por outro lado, optamos por não mapear nenhum interlocutor do conjunto *Emotionlines Facebook* a uma identidade genérica, já que existem mais de 300 participantes na base de dados e todos são representados de maneira razoavelmente uniforme no conjunto de diálogos.

Tabela 7 – Distribuição de turnos no *EmotionLines Friends Dataset*.

Emoções	Treinamento	Validação	Teste	Total	Frequência
Alegria	1.283	123	304	1.710	18,01%
Fúria	513	85	161	759	7,99%
Tristeza	351	62	85	498	5,24%
Neutro	4.752	491	1.287	6.530	68,76%
Total	**6.899**	**761**	**1.837**	**9.497**	**100,00%**

Tabela 8 – Distribuição de turnos no *EmotionLines Facebook Dataset*.

Emoções	Treinamento	Validação	Teste	Total	Frequência
Alegria	1.482	160	458	2.100	16,65%
Fúria	94	9	37	140	1,11%
Tristeza	389	38	87	514	4,08%
Neutro	7.148	825	1.882	9.855	78,16%
Total	**9.113**	**1.032**	**2.464**	**12.609**	**100,00%**

[16] Disponível em <http://doraemon.iis.sinica.edu.tw/emotionlines/download.html>.

É importante mencionar que os diálogos em *Emotionlines Friends* foram recentemente expandidos em uma nova base de dados para a realização da tarefa de Análise de Sentimento Multimodal, onde são consideradas características extraídas das transcrições, áudios e vídeos gerados a partir de cada diálogo, dando origem ao *Multimodal Emotionlines Dataset* (MELD) (PORIA et al., 2018). A diferença entre os dois conjuntos de dados se dá no agrupamento de sentenças que formavam um mesmo diálogo, mas estavam divididas em várias cenas não contíguas, além da remoção de algumas conversas e da aglutinação de rótulos nas classes de sentimento macroscópicas positivo, negativo e neutro, assim como ocorre no conjunto de dados *Emory*. Devido as conversas serem bastante parecidas, optamos por não considerar esta base em nossos experimentos.

6.2 Configuração experimental

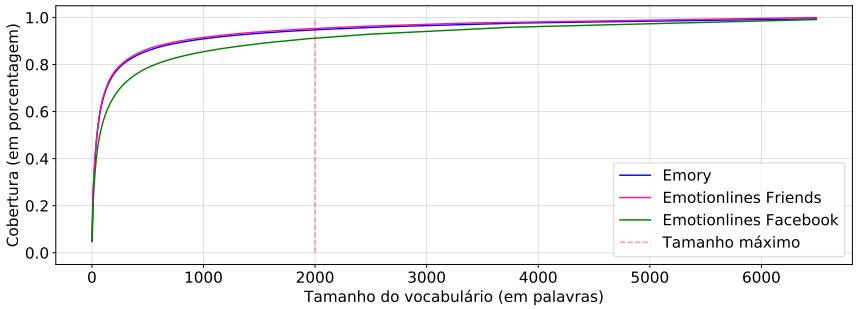
Antes de realizar os experimentos, as amostras devem passar por uma etapa de pré-processamento com o objetivo de remover características indesejadas e simplificar o processo de treinamento dos modelos. O primeiro passo diz respeito à *tokenização*, ou seja, transformação de cada fala (*strings* contíguas) em uma sequência de palavras. Enquanto a base de dados *Emory* já é disponibilizada com este procedimento realizado, a biblioteca `spaCy`[17] foi utilizada nos conjuntos *Emotionlines Facebook* e *Emotionlines Friends* devido a mesma gerar resultados bastante parecidos com aqueles observados no primeiro conjunto de dados. Em seguida, todas as palavras são mapeadas para letras minúsculas para diminuir o tamanho do vocabulário a ser aprendido e tornar o modelo invariante a esta característica. As pontuações também são removidas, com exceção das reticências e exclamações, as quais são mantidas a fim de tentar preservar disfluências (pausas na fala) e hesitações no primeiro caso, e ao observar que a segunda pontuação surge em uma quantidade significativa de amostras com sentimento não-neutro. No tocante à base de dados *Emotionlines Facebook*, dois passos adicionais são empregados: um conjunto de expressões regulares é utilizado para filtrar repetições excessivas de caracteres e unificar alguns emojis, que então são substituídos por uma única palavra que os descreve. Por exemplo, a representação de uma face sorrindo é trocada pela palavra "*happy*" (feliz).

Com o objetivo de diminuir o vocabulário de cada conjunto de dados para facilitar o treinamento dos modelos, foi realizada uma análise sobre a porcentagem do vocabulário preservada ao manter apenas as palavras mais frequentes para cada base e o resultado é apresentado na Figura 19a. É possível observar que considerar apenas as primeiras 2.000 palavras, ou seja, menos de $1/3$ do vocabulário quantidade original, garante que mais de 90% dos termos sejam conservados em todos os casos. A mesma análise é feita sobre tamanho de cada turno, em termos de palavras, e o resultado é exibido na Figura 19b, a partir da qual se

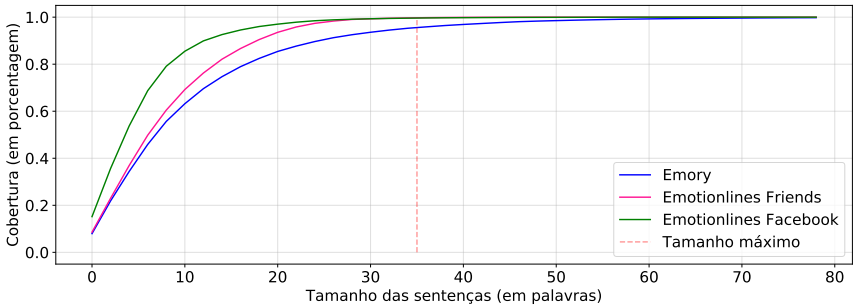
[17] Disponível em <https://spacy.io>.

conclui que manter apenas as 35 primeiras palavras de cada sentença garante que pelo menos 95% de todas amostras sejam conservadas integralmente.

Figura 19 – Cobertura da base de dados (a) em função do tamanho do vocabulário (b) em função do comprimento das sentenças.



(a)



(b)

Fonte: Elaborado pelo autor.

A abordagem proposta na presente dissertação de mestrado defende o emprego de características extraídas de turnos adjacentes durante a classificação atual. Entretanto, torna-se infactível treinar o modelo utilizando diálogos inteiros devido a sua extensão, além das dificuldades já envolvidas no treinamento de rede neurais recorrentes sofisticadas. A fim de contornar tais problemas, além de aumentar o tamanho das bases de dados em termos da quantidade de conversas disponíveis para treinamento, cada diálogo é dividido em segmentos formados por uma sequência de q turnos consecutivos, que são utilizados como amostras para treinar os modelos contextuais. A Tabela 9 apresenta um exemplo onde uma única cena, ou diálogo, origina três segmentos distintos formados por $q = 3$ falas. Por outro lado, note que aumentar a quantidade de turnos em cada segmento diminui as bases de dados à razão de $1/q$, fator que acaba limitando a complexidade dos modelos que poderão ser treinados, tendo em vista que, conforme já mencionado anteriormente, aumentar a sofisticação do modelo e, consequentemente, a quantidade de parâmetros a serem aprendidos demanda mais amostras para realizar um treinamento adequado.

Por fim, a base de dados *Emory* é utilizada a fim de treinar dois modelos, um a nível

Tabela 9 – Divisão de um diálogo em segmentos com 3 turnos para classificação sequencial.

	Personagem	**Fala**	**Rótulo**
Segmento 1	Mônica	Solte isso!	Furioso
	Ross	Não! Solte você!	Furioso
	Mônica	Não!	Furioso
Segmento 2	Ross	Por que sempre sobramos só nós no campo segurando a bola?	Neutro
	Mônica	Eu não sei. Acho que as outras pessoas não se importam tanto.	Furioso
	Ross	Olha! Está começando a nevar.	Alegre
Segmento 3	Ross	Me dê isso!	Furioso
	Mônica	Solte!	Furioso
	–	–	–

macroscópico, responsável por rotular cada turno com os sentimentos positivo, negativo e neutro, e outro microscópico, abrangendo as sete classes originalmente propostas. Por outro lado, um modelo é treinado a partir do conjunto de dados *Emotionlines Facebook* e outro a partir do *Emotionlines Friends*, em ambos os casos considerando apenas os sentimentos alegria, fúria, tristeza e neutro, conforme discutido anteriormente. Os turnos das classes desconsideradas ainda são mantidos nos segmentos a fim de preservar ao máximo a dinâmica dos diálogos, permitindo que a rede neural consiga extrair e combinar suas características e contexto com as demais falas do segmento. Todavia, os modelos não rotulam nem consideram as amostras desprezadas em sua função de custo.

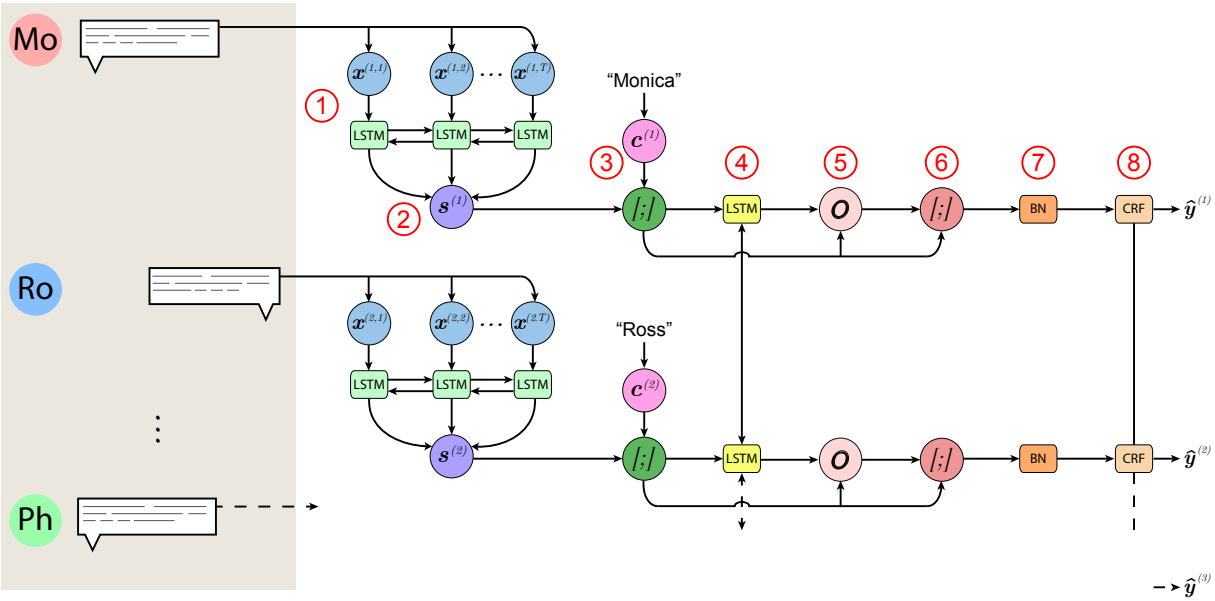
6.3 Modelo Proposto

O modelo proposto para a realização de AS contextual em transcrições de diálogos se baseia na hipótese de que uma conversa evolui de forma incremental, de modo que o sentimento de turnos futuros dependa do que já foi dito anteriormente. A partir desta premissa, e motivado pelas melhorias nos resultados de classificação em outras tarefas sequenciais na área de PLN, o modelo em questão foi proposto. É importante notar que além de considerar a informação de contexto, decidimos também analisar se considerar o perfil de cada interlocutor durante a classificação de seu turno também resulta em melhores resultados. Esta segunda decisão busca identificar se alguns indivíduos possuem uma certa predileção por determinados sentimentos em suas falas. De forma geral, a arquitetura utilizada é ilustrada na Figura 20, onde cada etapa do modelo, identificada por um número em vermelho, é detalhada a seguir.

Cada segmento, ou trecho, de diálogo $\mathcal{S}$ a ser classificado pode ser visto como uma sequência de turnos (falas) $\mathcal{S} = \left(\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, \ldots, \mathcal{X}^{(q)}\right)$. Da mesma forma, cada uma das falas $\mathcal{X}^{(k)}$ também pode ser vista como uma sequência de palavras, as quais são inicialmente mapeadas a seus *word embeddings* correspondentes, conforme ilustrado em (1). Detalhadamente, cada turno passa a ser representado por $\mathcal{X}^{(k)} = \left(\boldsymbol{x}^{(k,1)}, \boldsymbol{x}^{(k,2)}, \ldots, \boldsymbol{x}^{(k,T)}\right)$, onde $\boldsymbol{x}^{(k,t)} \in \mathbb{R}^{d_{emb}}$ e $T = 35$ é o tamanho máximo de todos os turnos, conforme a análise apresentada na Seção 6.2.

Para cada $\mathcal{X}^{(k)}$, o próximo passo, ilustrado em (2), é responsável por gerar um descritor

Figura 20 – Modelo proposto.



Fonte: Elaborado pelo autor.

de turno $\boldsymbol{s}^{(k)}$. Com este objetivo, os *word embeddings* são apresentados a uma BiLSTM, onde a primeira camada LSTM lê a sequência da esquerda para a direita e segunda em sentido contrário[18]. Os estados ocultos intermediários produzidos em ambas direções são concatenados e então apresentados ao mesmo mecanismo de atenção descrito na Seção 3.2.4. Matematicamente, o processo é detalhado a seguir, de modo que a função LSTM corresponde à formulação da rede neural recorrente de mesmo nome, discutida no Seção 3.2.1, enquanto a seta indica a direção de leitura da sequência. Adicionamente, $\overrightarrow{\boldsymbol{\xi}}^{(k,t)} \in \mathbb{R}^{d_{hid}}$, assim como $\overleftarrow{\boldsymbol{\xi}}^{(k,t)}$, e $\boldsymbol{s}^{(k)} \in \mathbb{R}^{(2 \cdot d_{hid})}$:

$$
\begin{aligned}
\overrightarrow{\boldsymbol{\xi}}^{(k,t)} &= \overrightarrow{\text{LSTM}}\left(\boldsymbol{x}^{(k,t)}, \overrightarrow{\boldsymbol{\xi}}^{(k,t-1)}\right) \quad \forall t \in [1, T], \\
\overleftarrow{\boldsymbol{\xi}}^{(k,t)} &= \overleftarrow{\text{LSTM}}\left(\boldsymbol{x}^{(k,t)}, \overleftarrow{\boldsymbol{\xi}}^{(k,t-1)}\right) \quad \forall t \in [T, 1], \\
\boldsymbol{\xi}^{(k,t)} &= \left[\overrightarrow{\boldsymbol{\xi}}^{(k,t)}; \overleftarrow{\boldsymbol{\xi}}^{(k,t)}\right], \\
\boldsymbol{s}^{(k)} &= \text{Atenção}\left(\boldsymbol{\xi}^{(k,1)}, \boldsymbol{\xi}^{(k,2)}, \ldots, \boldsymbol{\xi}^{(k,T)}\right).
\end{aligned}
\tag{41}
$$

Em seguida, o descritor de sentença $\boldsymbol{s}^{(k)}$ é concatenado com o descritor do interlocutor do turno em questão $\boldsymbol{c}^{(k)} \in \mathbb{R}^{d_{int}}$, conforme representado em (3), onde cada personagem é mapeado a um vetor correspondente, o qual é inicializado aleatoriamente e deve ser aprendido pelo modelo. Desta forma, os turnos passam a ser representados por $\boldsymbol{u}^{(k)} = \left[\boldsymbol{s}^{(k)}; \boldsymbol{c}^{(k)}\right] \in \mathbb{R}^{(2 \cdot d_{hid} + d_{int})}$, que é então aplicado à uma camada tangente hiperbólica, permitindo que as características do

[18] A camada recorrente é formada por unidades LSTM, ao invés de GRU, pois este é o modelo mais comumente utilizado em trabalhos de Processamento de Linguagem Natural.

interlocutor e sua fala interajam entre si, gerando $\hat{\boldsymbol{u}}^{(k)} \in \mathbb{R}^{(2 \cdot d_{hid} + d_{int})}$:

$$\hat{\boldsymbol{u}}^{(k)} = \tanh\left(W^{[3]}\boldsymbol{u}^{(i)} + \boldsymbol{b}^{[3]}\right). \tag{42}$$

O vetor resultante passa por uma segunda camada BiLSTM em (4), a qual tem por finalidade combinar as características do turno atual com informações de falas adjacentes no segmento sob análise e, consequentemente, incorporar informação de contexto na classificação sendo realizada, gerando assim um novo vetor $\boldsymbol{l}^{(k)}$, conforme apresentado na Equação 43, onde $\overrightarrow{\boldsymbol{l}}^{(k)} \in \mathbb{R}^{d_{ctx}}$, assim como $\overleftarrow{\boldsymbol{l}}^{(k)}$, e $\boldsymbol{l}^{(k)} \in \mathbb{R}^{(2 \cdot d_{ctx})}$:

$$\begin{aligned}
\overrightarrow{\boldsymbol{l}}^{(k)} &= \overrightarrow{\text{LSTM}}\left(\hat{\boldsymbol{u}}^{(k)}, \overrightarrow{\boldsymbol{l}}^{(k-1)}\right) \quad \forall k \in [1, q], \\
\overleftarrow{\boldsymbol{l}}^{(k)} &= \overleftarrow{\text{LSTM}}\left(\hat{\boldsymbol{u}}^{(k)}, \overleftarrow{\boldsymbol{l}}^{(k-1)}\right) \quad \forall k \in [q, 1], \\
\boldsymbol{l}^{(k)} &= \left[\overrightarrow{\boldsymbol{l}}^{(k)}; \overleftarrow{\boldsymbol{l}}^{(k)}\right].
\end{aligned} \tag{43}$$

É interessante observar que: (i) a análise de todas as falas até o passo anterior, inclusive dentro de um fragmento de diálogo, ocorre de forma independente e apenas neste instante ocorre a combinação de características de turnos adjacentes; (ii) ainda assim, não existe o compartilhamento de informações entre segmentos distintos, independentemente de ambos estarem originalmente em um único diálogo ou não.

Ao invés de encaminhar o vetor contextualizado $\boldsymbol{l}^{(k)}$ diretamente para as próximas camadas da rede neural, o mesmo interage com o descritor $\hat{\boldsymbol{u}}^{(k)}$ por meio do produto de Hadamard. Isso permite amplificar ou atenuar as características do turno em questão de acordo com o desenvolvimento do diálogo, conforme indicado em (5), processo análogo ao funcionamento dos portões de uma LSTM, por exemplo, e é inspirado no trabalho de McCann et al. (MCCANN et al., 2017) para classificação de sentenças[19]. Em (6) a representação obtida é novamente concatenada com o descritor da sentença antes do processo de contextualização a fim de que o modelo tenha um referencial "antes" e "depois", gerando assim o descritor final da amostra a ser analisado, $\boldsymbol{\Omega}^{(k)} \in \mathbb{R}^{(2 \cdot d_{ctx})}$:

$$\boldsymbol{\Omega}^{(k)} = \left[\hat{\boldsymbol{u}}^{(k)}; \left(\boldsymbol{l}^{(k)} \circ \hat{\boldsymbol{u}}^{(k)}\right)\right]. \tag{44}$$

A representação obtida passa então por uma camada de *Batch Normalization* (IOFFE; SZEGEDY, 2015) em (7), a qual faz com que o treinamento do modelo, em termos de acurácia e função de custo, se torne mais estável. Por fim, a classificação sequencial acontece em (8) através dos Campos Aleatórios Condicionais com o objetivo de permitir que predições

[19] Neste trabalho as sequências são formadas apenas por pares de sentenças. Além do produto de Hadamard, os autores também utilizam a subtração entre descritores, porém experimentalmente foi observado que considerar este terceiro aspecto piorou os resultados obtidos.

adjacentes influenciem a atual dentro de um mesmo segmento. Os mesmos passos descritos são aplicados aos demais turnos dentro de um mesmo segmento. A fim de identificar como cada um dos componentes discutidos influenciam o modelo gerado, diferentes versões do mesmo modelo para cada base de dados são treinadas:

- *Completo*: Considera o modelo em sua totalidade, incluindo todos os aspectos discutidos;
- *-CRF*: Substitui os Campos Aleatórios Condicionais pela função de classificação softmax, de modo que a troca de informação contextual ocorre apenas por meio da segunda camada BiLSTM;
- *-Ctx-CRF*: A segunda camada BiLSTM também é removida, de forma que a única informação contextual compartilhada entre os turnos advém dos descritores de personagem;
- *-Int-Ctx-CRF*: Os descritores dos interlocutores também passam a ser desprezados, de modo que cada turno passa a ser descrito como $\mathbf{\Omega}^{(k)} = \boldsymbol{s}^{(k)}$, em outras palavras, a etapa (2) é imediatamente sucedida pela função softmax;
- *-Att-Int-Ctx-CRF*: A camada de atenção também é removida e cada sentença passa a ser descrita pelo último estado oculto produzido pela BiLSTM, ou seja, $\mathbf{\Omega}^{(k)} = \boldsymbol{\xi}^{(k,T)}$. Esta é a versão mais simples do modelo proposto, onde os rótulos são gerados individualmente a partir da combinação dos *word embeddings* de cada amostra;
- *-Att-Int-Ctx+CRF*: Esta versão é idêntica à anterior, com a diferença de que os rótulos são gerados ao substituir a função softmax da camada de saída pelo CRF. Isso permite comparar o desempenho desta abordagem ao considerar apenas características locais com a versão *Completa*, a qual contempla outros aspectos contextuais.

Por fim, os resultados obtidos são comparados com dois *baselines*, o primeiro é a versão supervisionada do algoritmo fastText, tendo em vista que este modelo é (i) bastante simples de ser treinado; (ii) apresenta bons resultados de classificação em outras tarefas; (iii) já foi utilizado na AS em análises de produtos, gerando bons resultados de classificação (JOULIN et al., 2016). Já o segundo, consiste nos melhores resultados obtidos atualmente para cada conjunto de dados, mais especificamente, o modelo de Zahiri e Choi (ZAHIRI; CHOI, 2018) na base de dados *Emory* e a técnica utilizada por Khosla (KHOSLA, 2018) nos conjuntos *Emotionlines Friends* e *Emotionlines Facebook*.

6.4 Procedimentos de avaliação

A fim de analisar os resultados obtidos a partir de cada versão dos modelos treinados e compará-los, foi utilizada a acurácia média entre as versões aprendidas. Como algumas classes são mais frequentes que outras nas bases de dados consideradas, essa medida deve

ser empregada com cautela, tendo em vista que um modelo que prediz apenas a classe mais frequente apresentará um bom desempenho aparente, apesar de classificar todas as demais amostras incorretamente, tornando-se indesejável.

Com o propósito de contornar este cenário, outras métricas podem ser consideradas, como é o caso da precisão, revocação (mais comumente referida na literatura como *recall*) e F1, as quais são inicialmente definidas em problemas de classificação binários, mas podem ser estendidas para cenários multi-classe por meio da macro precisão, macro revocação e macro-F1, as quais não favorecem as classes mais frequentes (SOKOLOVA; LAPALME, 2009). Suas formulações são apresentadas a seguir, onde tp_c, fp_c e fn_c correspondem, respectivamente, a quantidade de verdadeiros positivos, falso positivos e falso negativos da c-ésima classe:

$$\begin{aligned} Pr_M &= \frac{\sum_{c=1}^{|\mathcal{C}|} \frac{tp_c}{tp_c+fp_c}}{|\mathcal{C}|}, \\ Re_M &= \frac{\sum_{c=1}^{|\mathcal{C}|} \frac{tp_c}{tp_c+fn_c}}{|\mathcal{C}|}, \\ F1_M &= \frac{2}{\frac{1}{Pr_M} + \frac{1}{Re_M}}. \end{aligned} \tag{45}$$

No tocante ao conjunto de dados *Emotionlines*, uma formulação alternativa à acurácia usual, denominada *"Acurácia não-ponderada"* é utilizada por Hsu e Ku (HSU; KU, 2018) para avaliar os modelos, tendo em vista o problema de desbalanceamento das classes. A mesma é computada como a média das acurácias para cada uma das classes do conjunto de dados, conforme a Equação 46, em que a_c é a acurácia, ou taxa de acerto, da c-ésima classe:

$$U_a = \frac{1}{|\mathcal{C}|} \sum_{c=1}^{|\mathcal{C}|} a_c. \tag{46}$$

Cada um dos modelos a ser analisado é treinado 15 vezes com o objetivo de determinar se existe diferença estatística significativa entre os resultados de acurácia e macro-F1 obtidos por meio do teste assinalado de Wilcoxon (WILCOXON, 1945) com $p = 0.05$. Como os modelos que alcançaram o estado-da-arte foram publicados com apenas uma métrica, torna-se impossível comparar estatisticamente os resultados publicados com os nossos, de modo que apenas uma comparação direta pode ser realizada.

6.5 Considerações Finais

Devido ao problema de AS ser recente na literatura, existem poucas bases de dados anotadas para tal finalidade. Assim, dois conjuntos distintos são considerados nos experimentos realizados: o primeiro, denominado *Emory*, é formado exclusivamente por transcrições de

diálogos em inglês da série *Friends*, onde cada turno é anotado com dois níveis diferentes de sentimentos, um macroscópico (formado por três classes) e outro microscópico (formado por sete classes). Já o segundo, *Emotionlines*, é constituído de dois subconjuntos de diálogos, um oriundo da mesma fonte anterior e outro formado por conversas realizadas por meio da troca de mensagens de texto, entretanto apenas um conjunto de rótulos, mais genérico e formado por quatro classes, é considerado.

Todos as bases de dados passam por uma etapa de pré-processamento a fim de reduzir o tamanho do vocabulário e das falas, facilitando assim o treinamento das diferentes versões do modelo recorrente proposto. Cada diálogo também é dividido em segmentos visando facilitar a aprendizagem das redes neurais propostas, além de aumentar a base de dados.

No que concerne à abordagem proposta, os diálogos são modelados de forma incremental e hierárquica, onde conversas (mais especificamente, segmentos) são vistos como sequências de turnos a serem classificadas contextualmente. Da mesma forma, cada fala é inicialmente vista como uma sequência de palavras, as quais precisam ter suas principais características (obtidas a partir de *word embeddings* pré-treinados) combinadas em um descritor de tamanho fixo. Tal vetor é então agregado ao descritor do interlocutor correspondente, permitindo dar início à incorporação do contexto aos descritores através da identidade dos participantes do diálogo. A seguir, os vetores têm suas características combinadas com elementos de turnos vizinhos por meio de uma segunda camada recorrente. Finalmente, a classificação ocorre ao considerar a melhor sequência de rótulos por meio de Campos Aleatórios Condicionais.

Devido ao modelo proposto ser formado por diferentes etapas, múltiplas versões são treinadas ao remover incrementalmente cada módulo a fim de verificar sua contribuição para os resultados obtidos. Cada versão do modelo é então treinada separadamente 15 vezes e as métricas de acurácia e macro-F1 médias são comparadas estatisticamente comparadas. Para cada conjunto de dados, os modelos propostos são comparados com dois *baselines*: o classificador fastText e o modelo que leva ao estado-da-arte atual, respectivamente SCNN$_c^a$ (ZAHIRI; CHOI, 2018) e CNN (KHOSLA, 2018). Todavia, como os resultados publicados consistem apenas em uma métrica, torna-se impossível compará-los estatisticamente, de modo que optamos por comparar diretamente os nossos melhores resultados com aqueles disponíveis na literatura.

7 Experimentos

Este capítulo apresenta o protocolo utilizado para o treinamento dos modelos, escolha dos hiperparâmetros, além dos resultados obtidos e sua análise. Para permitir a comparação direta entre os resultados obtidos e o estado-da-arte na literatura, os mesmos conjuntos de treinamento, avaliação e teste são utilizados.

Tendo em vista que a rede neural proposta na Seção 6.3 é formada por múltiplos elementos que interagem entre si, torna-se relevante estudar como cada uma dessas etapas contribuem para os resultados obtidos. Assim, as diferentes versões do modelo, as quais são obtidas ao remover seus módulos incrementalmente, conforme discutido na Seção 6.3, são treinadas independentemente e os resultados obtidos são comparados. Esta última etapa emprega o teste estatístico assinalado de Wilcoxon (WILCOXON, 1945) com $p = 0.05$ e os melhores valores médios são destacados em negrito. Como os resultados disponíveis na literatura correspondem apenas à uma rodada de treinamento e teste, inviabilizando sua comparação estatística, comparamos nossos melhores resultados para cada versão diretamente com o estado-da-arte.

7.1 Treinamento

Os hiperparâmetros utilizados foram obtidos incrementalmente a partir de cada uma das versões do modelo proposto, da mais simples (*-Att-Int-Ctx-CRF*) à mais complexa (modelo *Completo*), as quais foram treinadas com o conjunto de dados *Emory* a nível microscópico. Para cada iteração, os melhores valores foram estabelecidos ao fixar aqueles já determinados durante treinamentos anteriores e ao realizar uma busca em grade para os demais, de modo que os modelos tiveram suas métricas computadas a partir do conjunto de validação. Mais especificamente, o tamanho da BiLSTM que lê as sentenças, bem como as taxas de *dropout* (SRIVASTAVA et al., 2014) e de aprendizado, além do parâmetro de *gradient clipping* foram determinados a partir da versão "*-Att-Int-Ctx-CRF*" do modelo. Fixados estes valores foi possível determinar a relevância do mecanismo de atenção, e assim sucessivamente até que todos os hiperparâmetros fossem estabelecidos. Finalizada esta etapa, alguns testes foram realizados no modelo completo ao perturbar os valores obtidos, a fim de verificar se seria possível alcançar resultados ainda melhores com os novos hiperparâmetros, todavia tal busca se mostrou infrutífera.

O modelo *Completo* utiliza o CRF para realizar as predições e por isso é treinado ao minimizar o logaritmo negativo da verossimilhança regularizada (Equação 37), ao passo que os demais, por realizarem suas predições por meio da função softmax, são treinados ao minimizar a função de entropia cruzada (Equação 5). Em ambos os casos foi utilizado o otimizador

Adam (KINGMA; BA, 2014) com taxa de aprendizado cíclica (SMITH, 2015), a qual consiste em variar a taxa de aprendizado em ciclos determinados por meio de uma função cossenoidal. Inicialmente a taxa de aprendizado é alta e diminui à medida que o treinamento avança a fim de evitar pontos de sela na função de custo ou ótimos locais ruins. A fim de impedir que o processo de minimização divirja, a duração e a amplitude das oscilações também se tornam cada vez mais sutis, conforme o comportamento descrito na Equação 47:

$$o^{(b)} = o^{(\ell)}_{min} + \frac{1}{2}\left(o^{(\ell)}_{max} - o^{(\ell)}_{min}\right)\left[1 + \cos\left(\frac{\pi \cdot e}{E^{(\ell)}}\right)\right], \tag{47}$$

onde $o^{(b)}$ é a taxa de aprendizado utilizada para o b-ésimo *mini-batch*, $E^{(\ell)}$ é a quantidade de *mini-batches* no ℓ-ésimo ciclo, normalmente formado por múltiplas épocas, ao passo que e corresponde ao número de *mini-batches* desde o último reinício. Quando $e = E^{(\ell)}$, o processo é reiniciado e os limitantes $o^{(\ell+1)}_{max}$ e $o^{(\ell+1)}_{min}$ são atualizados por meio de um fator de decaimento o_{dec}. O limitante superior, por exemplo, se torna $o^{(\ell+1)}_{max} = o_{dec} \cdot o^{(\ell)}_{max}$. Da mesma forma, a duração dos ciclos é atualizada por meio de um fator de ganho: $E^{(\ell+1)} = o_{cic} \cdot E^{(\ell)}$. Todos os modelos são treinados por um número pré-definido de épocas (20 para os não-contextuais e 30 para os contextuais) e aquele que obtém o menor valor na função de custo para o conjunto de validação é utilizado nos testes. Também é interessante observar que em experimentos preliminares outros otimizadores foram cogitados, como o RMSProp (TIELEMAN; HINTON, 2012) e o Gradiente Descendente Estocástico, entretanto os resultados obtidos foram inferiores àqueles gerados através do método escolhido.

Com relação aos *word embeddings*, optamos por utilizar uma versão pré-treinada do modelo GloVe (PENNINGTON; SOCHER; MANNING, 2014) com 300 dimensões[20]. Tal decisão foi baseada no fato de que o modelo possui descritores para a maior quantidade de palavras dos vocabulários quando comparado a outras técnicas pré-treinados. Em experimentos preliminares avaliamos empregar uma outra versão deste modelo, treinada com as transcrições dos diálogos de todas temporadas da série *Friends*[21] mais um conjunto formado por análises de produtos com sentimentos positivo, negativo ou neutro (MAAS et al., 2011), entretanto os resultados obtidos foram ligeiramente inferiores quando comparados à opção anterior.

Por fim, os melhores hiperparâmetros encontrados, bem como seus intervalos de busca correspondentes, são apresentados na Tabela 10, onde p_{drop} é a probabilidade de *dropout* aplicada antes de cada BiLSTM, em suas conexões recorrentes e no vetor computado pelo mecanismo de atenção, λ é a severidade de regularização L2 aplicada à camada CRF (quando utilizada) e ρ corresponde à norma máxima do gradiente antes de aplicar a técnica de *gradient clipping* (PASCANU; MIKOLOV; BENGIO, 2013). Os mesmos hiperparâmetros foram utilizados para as bases de dados *Emory* e *Emotionlines*, exceto para os valores relacionados à taxa de aprendizado cíclica, os quais foram reajustados para o segundo conjunto de dados. Os modelos

[20] Disponível em <https://nlp.stanford.edu/projects/glove/>.
[21] Disponível em <https://github.com/emorynlp/character-mining>.

fastText, por sua vez, foram treinados a partir de sua implementação oficial[22] com taxa de aprendizado de $0,05$, *word embeddings* de 32 dimensões com uni, bi e trigramas por 5 épocas, exceto para as bases de dados *Emotionlines*, que foram treinados por 10 épocas.

Tabela 10 – Hiperparâmetros utilizados no treinamento dos modelos.

Hiperparâmetro	**Intervalo de busca**	**Melhor valor *(Emory)***	**Melhor valor *(Emotionlines)***
Tam. sentença	–	35 palavras	35 palavras
Tam. vocabulário	–	2.000 palavras	2.000 palavras
d_{emb}	$\{100, 200, 300\}$	300	300
d_{hid}	$\{15, 32, 64, 128\}$	32	32
d_{int}	$\{8, 16, 32\}$	16	16
d_{ctx}	$\{36, 40, 48\}$	40	40
p_{drop}	$\{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$	0.5	0.5
ρ	$\{1, 2, 3, 4, 5, \infty\}$	5	5
λ	$\{0, 10^{-4}, 10^{-3}, 5 \cdot 10^{-3}\}$	10^{-3}	10^{-3}
o_{min}	$\{10^{-4}, 5 \cdot 10^{-4}, 10^{-3}\}$	10^{-4}	10^{-4}
o_{max}	$\{10^{-3}, 5 \cdot 10^{-3}, 10^{-2}\}$	10^{-3}	10^{-3}
o_{dec}	$\{1, 0.9, 0.99, 0.999, 0.9999\}$	1	0.99
o_{cic}	$\{1, 2, 3\}$	1	2

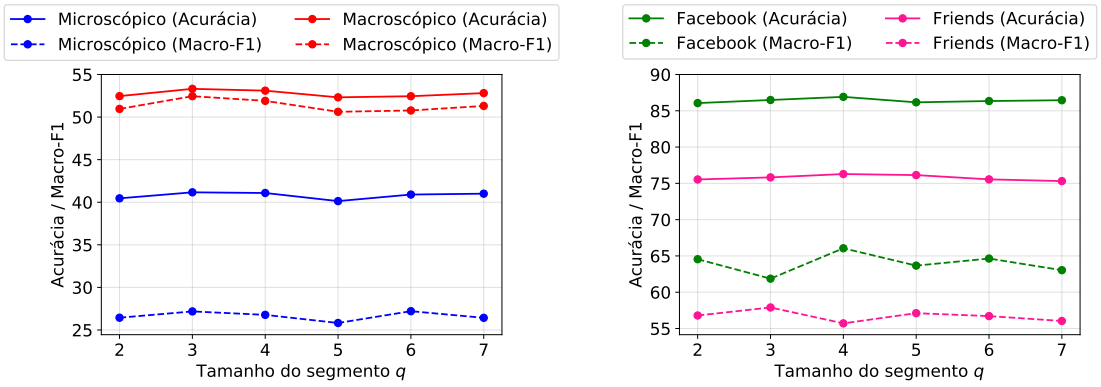
Atualmente não existe um consenso sobre a melhor forma de determinar quais valores de hiperparâmetros utilizar, ou mesmo um conjunto de regras que guie o processo de elaboração de uma nova arquitetura de rede neural. A estratégia empregada permite diagnosticar eventuais problemas de forma mais simples, pois a complexidade é inserida no sistema incrementalmente, apesar de ser difícil introduzir novas variáveis em iterações já finalizadas. Suponha que na iteração um, onde a versão mais simples do modelo é treinada, foram determinados os melhores valores para dois hiperparâmetros h_1 e h_2, de modo que o projeto atualmente se encontre na iteração quatro, onde deseja-se determinar o melhor valor para h_7. Agora, suponha que foi descoberto um novo hiperparâmetro, h'_1, a ser ajustado na iteração um. Como já foram estudados quatro novos hiperparâmetros entre as versões das iterações um e quatro, a única opção existente para avaliar o comportamento das versões do modelo conforme este valor consiste em desprezar todas as métricas obtidas e reiniciar todo o processo. Talvez para este caso faça mais sentido elaborar a arquitetura completa do modelo, ajustar seus hiperparâmetros e então realizar testes de ablação, os quais consistem em sistematicamente remover partes do modelo completo e analisar suas implicações. Independente da estratégia utilizada, seja por testes desta natureza ou pela construção incremental do modelo, é importante determinar quais de seus módulos são relevantes para a tarefa em questão, bem como sua contribuição correspondente para que possamos fundamentar as decisões tomadas em pesquisas futuras.

[22] Disponível em <https://fasttext.cc/>.

7.2 Resultados experimentais

Para treinar as versões do modelo proposto que utilizam informação de contexto, inicialmente é preciso determinar o tamanho dos segmentos formados a partir dos diálogos para cada base de dados. Assim, a versão completa foi treinada três vezes em cada caso ao variar q e as médias dos resultados de acurácia e macro-F1 são apresentadas na Figura 21:

Figura 21 – Valores de acurácia e macro-F1 nas partições de validação de acordo com o tamanho do segmento q para as bases de dados (a) *Emory* (b) *Emotionlines*.



(a) (b)

Fonte: Elaborado pelo autor.

Idealmente é desejável escolher um q tal que o mesmo maximize ambas as métricas simultaneamente, situação factível para os conjuntos *Emory Macroscópico*[23] ($q = 3$), *Emory Microscópico* ($q = 3$) e *Emotionlines Facebook* ($q = 4$). Com relação à base de dados *Emotionlines Friends*, enquanto a métrica macro-F1 é maximizada com $q = 3$, a acurácia é maximizada com $q = 4$, mas como a melhoria da primeira é mais significativa que a degradação da segunda (variações de $+2,19$ e $-0,45$, respectivamente), segmentos com $q = 3$ sentenças foram utilizados para o conjunto em questão.

Os resultados relacionados à base de dados *Emory Macro* são exibidos na Tabela 11, a partir da qual é possível identificar que o fastText é, de fato, um *baseline* bastante forte, já que o mesmo atingiu uma acurácia máxima ligeiramente inferior ao atual estado-da-arte, apesar de ser uma abordagem significativamente mais simples. No tocante à macro-F1 máxima, a mesma técnica também consegue superar os resultados alcançados pela SCNN$_c^a$ em $24,13\%$, enquanto a versão *"-CRF"* melhora em $32,84\%$ o atual estado-da-arte, representando evolução uma expressiva neste aspecto. A diferença de acurácia média entre os modelos completo e sem CRF não apresenta diferença estatisticamente significativa, mas a segunda abordagem possui resultados de macro-F1 estatisticamente melhores quando comparada à versão completa e, em

[23] Optamos por utilizar a nomenclatura *macroscópico* aqui para evitar confusão com o nome da métrica "macro-F1".

média, o classificador obtido é menos enviesado para algumas classes frente aos demais, de acordo com a mesma métrica.

Tabela 11 – Resultados obtidos na base de dados *Emory* no regime macroscópico (3 classes).

Modelo	Acurácia			Macro-F1		
	Média	Mínimo	Máximo	Média	Mínimo	Máximo
completo	52.07 ± 0.84	50.15	53.11	50.70 ± 0.85	49.00	52.14
-CRF	52.37 ± 0.87	50.94	53.90	**51.37 ± 0.71**	50.07	**52.61**
-Ctx-CRF	51.92 ± 0.53	50.90	52.78	50.77 ± 0.57	49.93	51.82
-Int-Ctx-CRF	51.09 ± 0.69	50.22	52.41	50.41 ± 0.65	49.72	51.53
-Att-Int-Ctx+CRF	51.97 ± 0.67	50.67	53.15	50.34 ± 0.79	48.84	51.58
-Att-Int-Ctx-CRF	50.04 ± 0.80	50.01	52.17	48.13 ± 0.43	48.99	48.20
fastText	**52.97 ± 0.29**	52.56	53.69	47.55 ± 0.39	47.12	48.72
SCNN$_c^a$ (ZAHIRI; CHOI, 2018)	–	–	**54.00**	–	–	39.25

No que concerne ao conjunto *Emory Micro*, os resultados obtidos são apresentados na Tabela 12, cenário no qual o fastText deixa de ser um baseline tão forte, provavelmente devido a algumas classes serem muito mais frequentes que outras. Com relação aos valores de acurácia máxima obtidos, todas versões do modelo proposto, exceto a mais simples, conseguem melhorar os resultados obtidos pelo modelo SCNN$_c^a$, de modo que a opção "*-CRF*" supera em $2,34\%$ o estado-da-arte estabelecido anteriormente. Apesar disso, a diferença de acurácia média entre esta versão, o modelo completo e a versão *-Ctx-CRF* não é estatisticamente significativa. Ao analisar os resultados de macro-F1, cada versão proposta também melhora o resultado anterior incrementalmente, tanto em termos de máximos como médias, o que acaba por evidenciar a relevância das características consideradas no modelo proposto.

Tabela 12 – Resultados obtidos na base de dados *Emory* no regime microscópico (7 classes).

Modelo	Acurácia			Macro-F1		
	Média	Mínimo	Máximo	Média	Mínimo	Máximo
completo	**37.57 ± 0.46**	36.58	38.38	23.16 ± 0.74	21.56	23.95
-CRF	**37.87 ± 0.44**	37.07	**38.79**	**24.05 ± 0.44**	23.07	24.84
-Ctx-CRF	**37.90 ± 0.41**	37.20	38.70	23.74 ± 0.52	22.75	24.76
-Int-Ctx-CRF	37.43 ± 0.65	36.44	38.63	23.38 ± 0.77	22.17	24.48
-Att-Int-Ctx+CRF	36.94 ± 0.49	35.70	37.60	23.55 ± 0.43	22.95	24.28
-Att-Int-Ctx-CRF	36.86 ± 0.39	36.14	37.66	21.62 ± 0.36	20.63	22.00
fastText	32.91 ± 0.09	32.68	33.06	13.89 ± 0.07	13.76	14.04
SCNN$_c^a$ (ZAHIRI; CHOI, 2018)	–	–	37.90	–	–	**26.90**

As Tabelas 13 e 14 trazem os resultados obtidos para as bases de dados *Emotionlines Friends* e *Emotionlines Facebook*, respectivamente. É interessante constatar que, novamente, o fastText se apresenta como um *baseline* forte, conseguindo imediatamente superar os resultados estabelecidos anteriormente. Além disso, conforme já foi possível estabelecer a partir de experimentos anteriores, o mesmo pode ser empregado como um limitante inferior bastante robusto para os resultados do modelo proposto.

O modelo completo apresenta resultados de classificação superiores em ambos conjuntos de dados para todas as métricas. Com relação ao estado-da-arte, a acurácia máxima melhora em

Tabela 13 – Resultados obtidos na base de dados *Emotionlines Friends* (4 classes).

Modelo	Acurácia			Macro-F1		
	Média	Mínimo	Máximo	Média	Mínimo	Máximo
completo	**81.57 ± 0.42**	80.90	**82.58**	**61.30 ± 0.94**	59.89	**62.94**
-CRF	81.42 ± 0.25	80.88	81.87	60.51 ± 0.87	58.64	61.60
-Ctx-CRF	81.44 ± 0.26	80.73	81.66	58.90 ± 0.75	57.68	60.47
-Int-Ctx-CRF	81.35 ± 0.30	80.95	81.93	59.17 ± 0.86	57.74	61.00
-Att-Int-Ctx+CRF	81.33 ± 0.29	80.77	81.92	47.54 ± 0.80	45.77	49.21
-Att-Int-Ctx-CRF	80.85 ± 0.36	80.24	81.49	54.96 ± 1.08	53.00	56.39
fastText	78.72 ± 0.13	78.55	78.93	42.73 ± 0.13	42.47	42.92
CNN (KHOSLA, 2018)	–	–	62.50	–	–	–

$32,13\%$ e $37,69\%$ para as bases *Emotionlines Friends* e *Emotionlines Facebook*, respectivamente. Em termos do maior valor para macro-F1, a versão completa melhora os resultados do modelo *"-CRF"* em $2,17\%$ e $1,92\%$ em cada conjunto de dados, respectivamente.

Tabela 14 – Resultados obtidos na base de dados *Emotionlines Facebook* (4 classes).

Modelo	Acurácia			Macro-F1		
	Média	Mínimo	Máximo	Média	Mínimo	Máximo
completo	**85.65 ± 0.23**	85.29	**86.03**	**59.80 ± 1.11**	58.06	**61.68**
-CRF	**85.52 ± 0.18**	85.02	85.76	57.58 ± 1.12	55.39	60.52
-Ctx-CRF	85.36 ± 0.20	85.06	85.75	56.88 ± 0.97	55.46	58.37
-Int-Ctx-CRF	81.40 ± 0.33	80.73	81.82	58.93 ± 0.91	56.04	60.17
-Att-Int-Ctx+CRF	**85.66 ± 0.26**	85.27	86.25	50.15 ± 0.99	47.47	51.81
-Att-Int-Ctx-CRF	80.78 ± 0.23	80.35	81.22	54.93 ± 0.84	53.77	56.91
fastText	84.02 ± 0.09	83.85	84.17	47.20 ± 0.26	46.73	47.52
CNN (KHOSLA, 2018)	–	–	62.48	–	–	–

Com base nos resultados apresentados é possível identificar que cada componente do modelo proposto, de fato, contribuiu para os resultados obtidos. A fim de realizar uma análise mais profunda nesta direção, a Tabela 15 apresenta a evolução da acurácia média para cada versão do modelo conforme novas etapas são consideradas. É possível identificar que os componentes mais relevantes são o mecanismo de atenção e a informação sobre interlocutores, que chega a melhorar em 4.86% a taxa de acerto para o conjunto *Emotionlines Facebook*. Os ganhos trazidos pela adição de informação sobre o contexto, por sua vez, são amplificados com a adição da camada CRF em alguns casos, permitindo que a rede neural não apenas combine características de turnos adjacentes, mas também considere a afinidade de transição de sentimentos entre falas no mesmo segmento a partir dos descritores de turnos contextualizados.

Tabela 15 – Ganhos percentuais para acurácia média em diferentes versões do modelo proposto.

Modelo	*Emory Macro*	*Emory Micro*	*Emotionlines Friends*	*Emotionlines Facebook*
+CRF	-0.57%	-0.79%	0.18%	0.15%
+Contexto	0.87%	-0.08%	-0.02%	0.19%
+Interlocutor	1.62%	1.26%	0.11%	4.86%
+Atenção	2.10%	1.55%	0.62%	0.77%
Básico	–	–	–	–

A mesma análise é repetida em função da métrica macro-F1 na Tabela 16. Assim como no caso anterior, adicionar um mecanismo de atenção para "ler" a transcrição das falas é benéfico em todos os casos, chegando a melhorar os resultados para a base de dados *Emory Micro* em 8.14%. A mesma tendência é observada ao adicionar informação de contexto ao modelo, melhorando a métrica em questão em todos os casos. Já em relação ao demais aspectos, o comportamento observado depende do conjunto de dados em questão. Se por um lado utilizar a informação sobre interlocutor é positivo para os conjuntos *Emory*, o oposto é observado para as bases *Emotionlines* e o mesmo ocorre ao substituir a função softmax na camada de saída pelo CRF. Apesar dessas diferenças sob a perspectiva da métrica macro-F1, em termos de acurácia ambos recursos são positivos. Assim, a decisão de utilizá-los acaba dependendo de qual métrica deseja-se maximizar e também das características da base de dados utilizada. Por fim, com exceção da camada CRF para as bases de dados *Emory*, utilizar todos recursos em conjunto apresentou os melhores resultados de classificação.

Tabela 16 – Ganhos percentuais para macro-F1 média em diferentes versões do modelo proposto.

Modelo	***Emory Macro***	***Emory Micro***	***Emotionlines Friends***	***Emotionlines Facebook***
+CRF	-1.32%	-3.84%	1.31%	3.86%
+Contexto	1.18%	1.31%	2.73%	1.23%
+Interlocutor	0.71%	1.54%	-0.46%	-3.48%
+Atenção	4.74%	8.14%	7.66%	7.28%
Básico	–	–	–	–

Ao contrário das expectativas iniciais, o modelo completo, o qual considera a sequência de rótulos emitidos para o segmento sob análise, apresenta resultados ligeiramente inferiores, ou estatisticamente similares, aos do modelo que não utiliza esta informação para os conjuntos de dados *Emory*. Três hipóteses podem ser levantadas para justificar este fenômeno: (i) a combinação dos demais módulos com esta camada de saída é incompatível; (ii) dividir o diálogo em segmentos para a classificação leva à degradação do CRF ou (iii) existe alguma característica da base de dados que inviabiliza utilizar esta fonte de informação.

Para analisar (i), foram considerados os resultados da versão "*-Att-Int-Ctx+CRF*", a qual consiste em acoplar a versão mais simples do modelo (apenas a camada BiLSTM) diretamente à camada CRF. Apesar deste arranjo levar a resultados bons para o conjunto de dados *Emory Macroscópico*, o mesmo é inferior a outras combinações mais complexas, o que acaba por invalidar esta justificativa. É interessante notar que no conjunto *Emotionlines Friends* este arranjo consegue se equiparar ao modelo completo em termos de acurácia, apesar de apresentar resultados inferiores às versões mais elaboradas sob a métrica macro-F1.

O argumento (ii) pode ser refutado ao considerar os experimentos preliminares para determinar o tamanho do segmento q ideal, o qual considera o modelo completo. Se a estratégia de segmentação fosse efetivamente prejudicial, as métricas apresentadas nos gráficos

da Figura 21 deveriam apenas melhorar conforme os segmentos tornam-se mais extensos, já que cada vez mais informação sequencial é incorporada aos trechos de diálogos.

Resta então a hipótese (iii), a qual assume a existência de alguma característica nesta base de dados que dificulta a aplicação eficiente do CRF. Diferentemente da base de dados *Emory*, o modelo completo apresentou, de fato, melhores resultados no conjunto de dados *Emotionlines*. Ademais, existe uma diferença de desempenho sob ambas as métricas entre os classificadores treinados com a primeira e segunda base de dados, respectivamente.

O baixo desempenho no primeiro conjunto de dados, especialmente no *Emory Microscópico*, pode ser justificado ao considerar a concordância entre anotadores durante a rotulação dos diálogos, medida pelo Kappa de Fleiss (FLEISS, 1971) em 14.34%. Em outras palavras, na maior parte dos casos, os quatro indivíduos que formaram o comitê de anotação discordam entre si[24], contudo em 85.09% dos casos existe um par de anotadores (dentre os quatro) com rotulações iguais (ZAHIRI; CHOI, 2018).

As constantes divergências no comitê implicam em alguns fatores que podem explicar o baixo desempenho nesta tarefa. Se quatro humanos com acesso a todo o diálogo a ser rotulado e um repertório de conhecimento não conseguem concordar quanto ao sentimento de uma fala, é bastante improvável que um classificador que não dispõe destas sofisticações consiga realizar a predição correta. Por outro lado, suponha um grupo de classificadores, dente os quais existe um oráculo, ou seja, um modelo capaz de rotular corretamente qualquer fala quanto ao seu sentimento. Dado que a identidade do oráculo é desconhecida, deseja-se determinar qual o melhor modelo no grupo e para isso é necessário comparar as predições de cada candidato com algum referencial. No caso, o gabarito consiste nas anotações do comitê, que é divergente, conforme já apresentado. Logo, é bastante improvável que o melhor classificador atinja a melhor pontuação neste teste.

Como o conjunto de dados *Emory Macroscópico* foi elaborado ao mapear as classes microscópicas para os sentimentos "positivo" e "negativo", as discrepâncias são propagadas de um nível de anotação para o próximo. Todavia, a melhoria nos resultados obtidos pode ser explicada devido à redução no número de classes e ao fato de que algumas divergências acabam por desaparecer no processo. Predizer a classe de uma amostra como "alegria", por exemplo, representa um erro para uma fala anotada como "tranquilidade", no entanto trata-se de um acerto no regime macroscópico, pois ambas classes agora constituem o rótulo "positivo".

Resta então tentar entender quais motivos causaram a baixa qualidade nas anotações obtidas. Naturalmente, o processo de classificação de turnos quanto ao seu sentimento é uma tarefa subjetiva por depender de critérios pessoais e experiências passadas, logo alguma discordância é esperada. Entretanto, a distinção entre as emoções utilizadas para anotar esta base de dados (que diferem daquelas consideradas no conjunto *Emotionlines*) é bastante sutil,

[24] Grupos formados por indivíduos distintos rotularam diferentes trechos da base de dados

como para as emoções "assustado" e "eufórico". Além disso, é preciso considerar que as amostras foram anotadas por indivíduos em diferentes partes do globo, os quais não necessariamente falam inglês como sua língua materna, por meio do serviço *Amazon Mechanical Turk*[25].

Apesar de resultados melhores terem sido alcançados para a base de dados *Emotionlines*, decidimos repetir a mesma análise para este conjunto de dados, o qual também foi anotado através do *Amazon Mechanical Turk*, por comitês agora formados por 5 indivíduos, os quais também tiveram acesso a todo o diálogo a ser analisado. Como as classificações de cada rotulador não encontram-se disponíveis, reproduzimos o resultado divulgado por Chen et al. (CHEN et al., 2018), os quais reportam que o Kappa de Fleiss (FLEISS, 1971) para os subconjuntos *Facebook* e *Friends* é maior que 33%. Apesar de ser relativamente baixo, o mesmo é significativamente maior que o coeficiente observado no conjunto anterior. Esta característica, em conjunto com a utilização de apenas quatro classes podem servir como justificas para os melhores resultados obtidos pelos mesmos modelos quando comparados às bases de dados *Emory*.

7.3 Considerações Finais

O modelo proposto é formado por diferentes elementos que interagem entre si, o que torna necessária a configuração de múltiplos hiperparâmetros. Para contornar este impasse, tais valores são determinados incrementalmente a partir das variantes mais simples do modelo, partindo para versões mais elaboradas, com novos hiperparâmetros a serem determinados, até atingir a versão completa.

A fim de aumentar a base de dados e facilitar o treinamento da rede neural proposta, formada por camadas recorrentes, os diálogos são divididos em segmentos formados por q turnos consecutivos. O melhor valor é determinado ao treinar a versão completa do modelo sobre cada base de dados e o valor que maximiza as métricas de acurácia e macro-F1 simultaneamente é escolhido, conforme exibido na Figura 21.

No que diz respeito à taxa de aprendizado, as duas abordagens mais comuns consistem em manter este valor constante durante o treinamento ou diminuí-lo monotonicamente em função do tempo. No nosso caso utilizamos uma abordagem alternativa denominada "taxa de aprendizado cíclica" (SMITH, 2015), onde este hiperparâmetro segue um regime cossenoidal. Isso permite que a rede neural não fique estagnada em ótimos locais ruins durante o início do treinamento e explore melhor o espaço de busca, encontrando soluções que apresentem menores valores para a função de custo e sejam mais robustas.

Os resultados obtidos permitem analisar a importância de cada um dos elementos da rede neural proposta. Como cada subconjunto utiliza um regime de anotação ou advém de um

[25] <https://www.mturk.com>.

domínio diferente, é possível realizar uma análise mais ampla sobre a capacidade do modelo se adaptar a diferentes cenários.

De maneira geral, utilizar o mecanismo de atenção trás melhores resultados em todos os casos, já a relevância dos demais elementos depende da métrica em questão, no caso acurácia ou macro-F1. Em relação à primeira, a identidade do interlocutor é uma característica bastante importante, ao passo que a segunda é degradada em alguns casos. Por outro lado, características de turnos adjacentes mostram-se positivas sob a medida de macro-F1, apesar de diminuirem a acurácia de forma bastante sutil em alguns casos. Ainda assim, esta perda de desempenho é menor que os ganhos associados à esta informação. Por fim, no tocante à camada CRF, a mesma apresentou resultados positivos em termos das duas medidas para os conjuntos de dados *Emotionlines* e negativos para as bases *Emory*.

A camada CRF, por sua vez, levou a melhores resultados sob ambas métricas para o conjunto de dados *Emotionlines*, ao passo que valores piores foram atingidos para as bases *Emory*. Muito provavelmente esse comportamento esteja relacionado à baixa qualidade das anotações realizadas, característica inferida a partir do baixo Kappa de Fleiss (FLEISS, 1971), conforme discutido anteriormente.

8 Conclusões

A presente dissertação de mestrado propôs um novo tipo de arquitetura de rede neural para a tarefa de Análise de Sentimentos em transcrições de diálogos ao considerar seu contexto. Esse aspecto toma como base a hipótese de que uma conversa evolui de forma incremental e em função de interações passadas, de modo que o sentimento transmitido por cada um de seus interlocutores dependa do que já foi dito anteriormente. A informação sobre o contexto foi incorporada à rede neural proposta a partir de características extraídas de falas adjacentes àquela a ser classificada; ao contemplar a identidade dos interlocutores no diálogo; ao aprender a afinidade de transição entre sentimentos dentro de um fragmento da conversa; e ao considerar a forma como as palavras são combinadas em cada sentença.

Nesta abordagem o diálogo é modelado de forma hierárquica, onde inicialmente são considerados apenas os aspectos mais relevantes de cada turno, que a seguir são combinados com as características de seu autor, as quais também são aprendidas pelo modelo. Posteriormente, as informações obtidas são amplificadas ou atenuadas de acordo com o teor do diálogo, para que então a amostra seja rotulada ao levar em conta os potenciais sentimentos das demais falas naquele segmento.

Fundamentado nos experimentos realizados em bases de dados anotadas com sentimentos em diferentes níveis de granularidade e oriundas de domínios distintos (transcrições da série *Friends* e conversas casuais realizadas por meio do *Facebook Messenger*), foi possível constatar que, de fato, tais aspectos são realmente relevantes para o problema em questão, levando estatisticamente a melhores resultados de classificação. Com isso, esperamos contribuir com a comunidade científica em uma área de pesquisa que, apesar de recente, possui poucos trabalhos desenvolvidos.

8.1 Trabalhos Futuros

O treinamento de modelos robustos para AS em diálogos, assim como para qualquer tarefa envolvendo aprendizado de máquina, depende de bases de dados suficientemente grandes e bem anotadas. Entretanto, devido ao problema estudado na presente dissertação de mestrado ser relativamente recente, existem poucos conjuntos disponíveis atualmente. Assim, em trabalhos futuros pode ser interessante criar novos conjuntos de dados, seja considerando outros domínios, outros conjuntos de rótulos, ou mesmo outros idiomas. Até onde sabemos, atualmente não existe nenhuma base de dados anotada para AS em diálogos em português, por exemplo. Além dos custos envolvidos, a rotulação de diálogos quanto ao seu sentimento traz desafios adicionais, já que os anotadores devem apresentar um grau de concordância satisfatório, apesar da natureza desta tarefa ser subjetiva.

Já a Tabela 16 mostra que em alguns casos considerar a identidade do interlocutor leva à degradação da métrica macro-F1. Isso levanta a suspeita de que talvez o modelo passe a dar muita importância para a identidade do interlocutor em detrimento das características de sua fala, característica que pode ser melhor explorada em trabalhos futuros.

Conforme estabelecido anteriormente, um diálogo pode ser visto como uma interação formada por uma sequência de turnos emitidos por seus diferentes participantes. Apesar desta dinâmica, o modelo proposto extrai as características do turno atual, emitido por um participante qualquer, e então combina essas informações com o contexto geral do diálogo, formado pelas falas dos demais interlocutores. Entretanto, é possível supor que o sentimento desta fala seja uma função de três parâmetros: suas características, o teor do diálogo e as características de sua fala anterior. A formulação estudada no presente trabalho considera apenas os dois primeiros aspectos, mas pode ser interessante desenvolver um modelo que possua um "canal" exclusivo para cada interlocutor e permita recuperar informações de seus turnos passados.

A estratégia de segmentação, por sua vez, foi utilizada para facilitar o treinamento das redes neurais e para aumentar as base de dados, permitindo que modelos mais complexos pudessem ser treinados adequadamente. Apesar de útil, este artifício acaba dividindo um diálogo em vários blocos, os quais não compartilham informação de contexto. Como esse aspecto é bastante importante para a tarefa em questão, conforme os experimentos demonstraram, pode ser útil retomar o contexto de segmentos anteriores dentro de um mesmo diálogo.

Finalmente, o modelo proposto consegue superar o estado-da-arte em diferentes bases de dados ao propor a utilização de informação contextual sob diferentes perspectivas. Além disso, o mesmo também serve para lançar questionamentos que podem ser utilizados para embasar pesquisas futuras, permitindo entender melhor o problema de Análise de Sentimento em diálogos, o qual é relativamente recente na literatura.

Referências

AMORA, P. R. P.; TEIXEIRA, E. M.; LIMA, M. I. V.; AMARAL, G. M.; CARDOZO, J. R. A.; MACHADO, J. C. A deep learning approach to prioritize customer service using social networks. In: *Proceedings of the 5th Symposium on Knowledge Discovery, Mining and Learning*. 2017. p. 153–160.

ANG, J.; LIU, Y.; SHRIBERG, E. Automatic dialog act segmentation and classification in multiparty meetings. In: *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.* 2005. v. 1, p. 1061–1064. ISSN 1520-6149.

ARORA, S.; LIANG, Y.; MA, T. A simple but tough-to-beat baseline for sentence embeddings. In: *International Conference on Learning Representations 2017*. 2017.

BACCIANELLA, S.; ESULI, A.; SEBASTIANI, F. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta: , 2010.

BAHDANAU, D.; CHO, K.; BENGIO, Y. Neural machine translation by jointly learning to align and translate. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2014.

BAZIOTIS, C.; PELEKIS, N.; DOULKERIDIS, C. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. 2017. p. 747–754.

BENGIO, Y.; SIMARD, P.; FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, IEEE, v. 5, n. 2, p. 157–166, 1994.

BOBICEV, V.; SOKOLOVA, M.; OAKES, M. What goes around comes around: learning sentiments in online medical forums. *Cognitive Computation*, Springer, v. 7, n. 5, p. 609–621, 2015.

BOJANOWSKI, P.; GRAVE, E.; JOULIN, A.; MIKOLOV, T. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

BRYCHCÍN, T.; KRÁL, P. Unsupervised dialogue act induction using gaussian mixtures. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. 2017. p. 485–490.

CAMPELLO, R. J.; MOULAVI, D.; SANDER, J. Density-based clustering based on hierarchical density estimates. In: SPRINGER. *Pacific-Asia conference on knowledge discovery and data mining*. 2013. p. 160–172.

CARLETTA, J.; ISARD, S.; DOHERTY-SNEDDON, G.; ISARD, A.; KOWTKO, J. C.; ANDERSON, A. H. The reliability of a dialogue structure coding scheme. *Computational linguistics*, MIT Press, v. 23, n. 1, p. 13–31, 1997.

CHEN, S.-Y.; HSU, C.-C.; KUO, C.-C.; KU, L.-W. et al. Emotionlines: An emotion corpus of multi-party conversations. *arXiv preprint arXiv:1802.08379*, 2018.

CHO, K.; MERRIËNBOER, B. V.; BAHDANAU, D.; BENGIO, Y. On the properties of neural machine translation: Encoder-decoder approaches. *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, p. 103–111, 2014.

CHOI, J. D.; CHEN, H. Y. Semeval 2018 task 4: Character identification on multiparty dialogues. In: *Proceedings of The 12th International Workshop on Semantic Evaluation*. 2018. p. 57–64.

CLICHE, M. Bb_twtr at semeval-2017 task 4: Twitter sentiment analysis with cnns and lstms. In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, 2017. p. 573–580.

COHEN, J. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, Sage Publications Sage CA: Thousand Oaks, CA, v. 20, n. 1, p. 37–46, 1960.

COLLOBERT, R.; WESTON, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In: ACM. *Proceedings of the 25th International Conference on Machine Learning*. 2008. p. 160–167.

DAI, A. M.; LE, Q. V. Semi-supervised sequence learning. In: *Advances in Neural Information Processing Systems*. 2015. p. 3079–3087.

EISEINSTEIN, J. Natural language processing. Disponível em <https://github.com/jacobeisenstein/gt-nlp-class/tree/master/notes>. 2019.

EISNER, J. Inside-outside and forward-backward algorithms are just backprop (tutorial paper). In: *Proceedings of the Workshop on Structured Prediction for NLP*. 2016. p. 1–17.

EKMAN, P. An argument for basic emotions. *Cognition & emotion*, Taylor & Francis, v. 6, n. 3-4, p. 169–200, 1992.

EKMAN, P.; FRIESEN, W. V.; O'SULLIVAN, M.; CHAN, A.; DIACOYANNI-TARLATZIS, I.; HEIDER, K.; KRAUSE, R.; LECOMPTE, W. A.; PITCAIRN, T.; RICCI-BITTI, P. E. et al. Universals and cultural differences in the judgments of facial expressions of emotion. *Journal of personality and social psychology*, American Psychological Association, v. 53, n. 4, p. 712, 1987.

ELKAN, C. Log-linear models and conditional random fields. *Tutorial notes at Conference on Information and Knowledge Management (CIKM)*, v. 8, p. 1–12, 2008.

ELMAN, J. L. Finding structure in time. *Cognitive Science*, Wiley Online Library, v. 14, n. 2, p. 179–211, 1990.

EUGENIO, B. D.; XIE, Z.; SERAFIN, R. Dialogue act classification, higher order dialogue structure, and instance-based learning. *Dialogue & Discourse*, v. 1, n. 2, p. 1–24, 2010.

EZEN-CAN, A.; BOYER, K. E. Understanding student language: An unsupervised dialogue act classification approach. *Journal of Educational Data Mining (JEDM)*, v. 7, n. 1, p. 51–78, 2015.

FALCÃO, A. X.; STOLFI, J.; LOTUFO, R. A. The image foresting transform: theory, algorithms, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 26, n. 1, p. 19–29, 2004.

FLEISS, J. L. Measuring nominal scale agreement among many raters. *Psychological bulletin*, American Psychological Association, v. 76, n. 5, p. 378, 1971.

FORSYTHAND, E. N.; MARTELL, C. H. Lexical and discourse analysis of online chat dialog. In: IEEE. *Semantic Computing, 2007. ICSC 2007. International Conference on*. 2007. p. 19–26.

GANCHEV, K.; GILLENWATER, J.; TASKAR, B. et al. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, v. 11, n. Jul, p. 2001–2049, 2010.

GERS, F. A.; SCHMIDHUBER, J.; CUMMINS, F. A. Learning to forget: Continual prediction with lstm. *Neural Computation*, v. 12, p. 2451–2471, 2000.

GOLDBERG, Y. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, Morgan & Claypool Publishers, v. 10, n. 1, p. 1–309, 2017.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. : MIT Press, 2016.

GRAVE, E.; BOJANOWSKI, P.; GUPTA, P.; JOULIN, A.; MIKOLOV, T. Learning word vectors for 157 languages. *arXiv preprint arXiv:1802.06893*, 2018.

GUPTA, R. Conditional random fields. *Unpublished report, IIT Bombay*, p. 1–24, 2006.

HARTMANN, N.; FONSECA, E.; SHULBY, C.; TREVISO, M.; RODRIGUES, J.; ALUISIO, S. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. *arXiv preprint arXiv:1708.06025*, 2017.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural Computation*, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.

HSU, C.-C.; KU, L.-W. Socialnlp 2018 emotionx challenge overview: Recognizing emotions in dialogues. In: *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*. 2018. p. 27–31.

HU, X.; TANG, J.; GAO, H.; LIU, H. Unsupervised sentiment analysis with emotional signals. In: ACM. *Proceedings of the 22nd international conference on World Wide Web*. 2013. p. 607–618.

HUANG, Z.; XU, W.; YU, K. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

JANIN, A.; BARON, D.; EDWARDS, J.; ELLIS, D.; GELBART, D.; MORGAN, N.; PESKIN, B.; PFAU, T.; SHRIBERG, E.; STOLCKE, A.; WOOTERS, C. The icsi meeting corpus. In: *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*. 2003. v. 1, p. I–364–I–367 vol.1. ISSN 1520-6149.

JO, Y.; YODER, M. M.; JANG, H.; ROSÉ, C. P. Modeling dialogue acts with content word filtering and speaker preferences. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2017. p. 2169–2179.

JORDAN, M. I. Serial order: A parallel distributed processing approach. *Advances in Psychology*, Elsevier, v. 121, p. 471–495, 1997.

JOULIN, A.; GRAVE, E.; BOJANOWSKI, P.; MIKOLOV, T. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

JURAFSKY, D. Switchboard swbd-damsl shallow-discourse-function annotation coders manual. *Institute of Cognitive Science Technical Report*, 1997.

KALCHBRENNER, N.; BLUNSOM, P. Recurrent convolutional neural networks for discourse compositionality. *arXiv preprint arXiv:1306.3584*, 2013.

KHOSLA, S. Emotionx-ar: Cnn-dcnn autoencoder based emotion classifier. In: *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*. 2018. p. 37–44.

KIM, S.; D'HARO, L. F.; BANCHS, R. E.; WILLIAMS, J. D.; HENDERSON, M.; YOSHINO, K. The fifth dialog state tracking challenge. In: IEEE. *Spoken Language Technology Workshop (SLT), 2016 IEEE*. 2016. p. 511–517.

KIM, S.; D'HARO, L. F.; BANCHS, R. E.; WILLIAMS, J. D.; HENDERSON, M. The fourth dialog state tracking challenge. In: *Proceedings of the 7th International Workshop on Spoken Dialogue Systems (IWSDS)*. : Springer, 2016.

KIM, Y. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

KIROS, R.; ZHU, Y.; SALAKHUTDINOV, R. R.; ZEMEL, R.; URTASUN, R.; TORRALBA, A.; FIDLER, S. Skip-thought vectors. In: *Advances in neural information processing systems*. 2015. p. 3294–3302.

KOPPEL, M.; SCHLER, J. The importance of neutral examples for learning sentiment. *Computational Intelligence*, Wiley Online Library, v. 22, n. 2, p. 100–109, 2006.

KUMAR, H.; AGARWAL, A.; DASGUPTA, R.; JOSHI, S.; KUMAR, A. Dialogue act sequence labeling using hierarchical encoder with crf. *arXiv preprint arXiv:1709.04250*, 2017.

LAFFERTY, J.; MCCALLUM, A.; PEREIRA, F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings of International Conference on Machine Learning*, p. 282–289, 2002.

LAMPLE, G.; BALLESTEROS, M.; SUBRAMANIAN, S.; KAWAKAMI, K.; DYER, C. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

LE, Q.; MIKOLOV, T. Distributed representations of sentences and documents. In: *International Conference on Machine Learning*. 2014. p. 1188–1196.

LEE, D.; JEONG, M.; KIM, K.; RYU, S.; LEE, G. G. Unsupervised spoken language understanding for a multi-domain dialog system. *IEEE Transactions On Audio, Speech, and Language Processing*, IEEE, v. 21, n. 11, p. 2451–2464, 2013.

LEE, J. Y.; DERNONCOURT, F. Sequential short-text classification with recurrent and convolutional neural networks. *arXiv preprint arXiv:1603.03827*, 2016.

LIU, B. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, Morgan & Claypool Publishers, v. 5, n. 1, p. 1–167, 2012.

LIU, L.; SHANG, J.; XU, F.; REN, X.; GUI, H.; PENG, J.; HAN, J. Empower sequence labeling with task-aware neural language model. *arXiv preprint arXiv:1709.04109*, 2017.

LIU, P.; JOTY, S. R.; MENG, H. M. Fine-grained opinion mining with recurrent neural networks and word embeddings. In: *Proceedings of 2015 the Conference on Empirical Methods in Natural Language Processing*. 2015. p. 1433–1443.

LU, Y.; CASTELLANOS, M.; DAYAL, U.; ZHAI, C. Automatic construction of a context-aware sentiment lexicon: an optimization approach. In: ACM. *Proceedings of the 20th International Conference on World Wide Web*. 2011. p. 347–356.

MA, K.; XIAO, C.; CHOI, J. D. Text-based speaker identification on multiparty dialogues using multi-document convolutional neural networks. In: *Proceedings of Association for Computational Linguistics 2017, Student Research Workshop*. 2017. p. 49–55.

MA, X.; HOVY, E. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*, 2016.

MAAS, A. L.; DALY, R. E.; PHAM, P. T.; HUANG, D.; NG, A. Y.; POTTS, C. Learning word vectors for sentiment analysis. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, 2011. p. 142–150. Disponível em: <http://www.aclweb.org/anthology/P11-1015>.

MANNING, C. D.; SURDEANU, M.; BAUER, J.; FINKEL, J.; BETHARD, S. J.; MCCLOSKY, D. The Stanford CoreNLP natural language processing toolkit. In: *Association for Computational Linguistics (ACL) System Demonstrations*. 2014. p. 55–60.

MAO, Y.; LEBANON, G. Sequential models for sentiment prediction. In: *International Conference on Machine Learning Workshop on Learning in Structured Output Spaces*. 2006.

MCAULEY, J.; LESKOVEC, J.; JURAFSKY, D. Learning attitudes and attributes from multi-aspect reviews. In: IEEE. *12th IEEE International Conference on Data Mining (ICDM)*. 2012. p. 1020–1025.

MCCANN, B.; BRADBURY, J.; XIONG, C.; SOCHER, R. Learned in translation: Contextualized word vectors. In: *Advances in Neural Information Processing Systems*. 2017. p. 6294–6305.

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

MIKOLOV, T.; SUTSKEVER, I.; CHEN, K.; CORRADO, G. S.; DEAN, J. Distributed representations of words and phrases and their compositionality. In: *Advances in Neural Information Processing Systems*. 2013. p. 3111–3119.

MOHAMMAD, S. M.; BRAVO-MARQUEZ, F. Emotion intensities in tweets. *arXiv preprint arXiv:1708.03696*, 2017.

MOLDOVAN, C.; RUS, V.; GRAESSER, A. C. Automated speech act classification for online chat. *The 22nd Midwest Artificial Intelligence and Cognitive Science Conference*, v. 710, p. 23–29, 2011.

MORAES, R.; VALIATI, J. F.; NETO, W. P. G. Document-level sentiment classification: An empirical comparison between svm and ann. *Expert Systems with Applications*, Elsevier, v. 40, n. 2, p. 621–633, 2013.

MURALIDHAR, A. *Understanding dialogue: sentiment and topic analysis of dialogue transcripts*. Tese (Doutorado) — Massachusetts Institute of Technology, 2013.

MURRAY, G.; CARENINI, G. Subjectivity detection in spoken and written conversations. *Natural Language Engineering*, Cambridge University Press, v. 17, n. 3, p. 397–418, 2011.

NASUKAWA, T.; YI, J. Sentiment analysis: Capturing favorability using natural language processing. In: ACM. *Proceedings of the 2nd International Conference on Knowledge Capture*. 2003. p. 70–77.

OJAMAA, B.; JOKINEN, P. K.; MUISCHENK, K. Sentiment analysis on conversational texts. In: LINKÖPING UNIVERSITY ELECTRONIC PRESS. *Proceedings of the 20th Nordic Conference of Computational Linguistics, NODALIDA 2015, May 11-13, 2015, Vilnius, Lithuania*. 2015. p. 233–237.

ORTEGA, D.; VU, N. T. Neural-based context representation learning for dialog act classification. *arXiv preprint arXiv:1708.02561*, 2017.

PAGLIARDINI, M.; GUPTA, P.; JAGGI, M. Unsupervised learning of sentence embeddings using compositional n-gram features. *arXiv preprint arXiv:1703.02507*, 2017.

PANG, B.; LEE, L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 43rd annual meeting on association for computational linguistics*. 2005. p. 115–124.

PANG, B.; LEE, L. et al. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, Now Publishers, Inc., v. 2, n. 1–2, p. 1–135, 2008.

PANG, B.; LEE, L.; VAITHYANATHAN, S. Thumbs up? sentiment classification using machine learning techniques. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of 2012 the Conference on Empirical Methods in Natural Language Processing*. 2002. v. 10, p. 79–86.

PAPA, J. P.; FALCÃO, A. X.; SUZUKI, C. T. N. Supervised pattern classification based on optimum-path forest. *International Journal of Imaging Systems and Technology*, John Wiley & Sons, Inc., New York, NY, USA, v. 19, n. 2, p. 120–131, 2009. ISSN 0899-9457.

PASCANU, R.; MIKOLOV, T.; BENGIO, Y. On the difficulty of training recurrent neural networks. In: *Proceedings of the 30th International Conference on Machine Learning*. 2013. p. 1310–1318.

PENNINGTON, J.; SOCHER, R.; MANNING, C. Glove: Global vectors for word representation. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2014. p. 1532–1543.

PERUMAL, K.; HIRST, G. Semi-supervised and unsupervised categorization of posts in web discussion forums using part-of-speech information and minimal features. In: *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. 2016. p. 100–108.

PIRYANI, R.; MADHAVI, D.; SINGH, V. K. Analytical mapping of opinion mining and sentiment analysis research during 2000–2015. *Information Processing & Management*, Elsevier, v. 53, n. 1, p. 122–150, 2017.

PORIA, S.; HAZARIKA, D.; MAJUMDER, N.; NAIK, G.; CAMBRIA, E.; MIHALCEA, R. Meld: A multimodal multi-party dataset for emotion recognition in conversations. *arXiv preprint arXiv:1810.02508*, 2018.

QUIRK, R.; GREENBAUM, S.; LEECH, G.; SVARTVIK, J. *A Comprehensive Grammar of the English Language*. : Longman, 1985.

RAMSHAW, L. A.; MARCUS, M. P. Text chunking using transformation-based learning. In: *Natural Language Processing Using Very Large Corpora*. : Springer, 1999. p. 157–176.

RIBEIRO, L. C. F.; PAPA, J. P. Unsupervised dialogue act classification with optimum-path forest. In: IEEE. *Graphics, Patterns and Images (SIBGRAPI), 2018 31st SIBGRAPI Conference on*. 2018.

RINGEVAL, F.; SCHULLER, B.; VALSTAR, M.; GRATCH, J.; COWIE, R.; SCHERER, S.; MOZGAI, S.; CUMMINS, N.; SCHMITT, M.; PANTIC, M. Avec 2017: Real-life depression, and affect recognition workshop and challenge. In: ACM. *Proceedings of the 7th Annual Workshop on Audio/Visual Emotion Challenge*. 2017. p. 3–9.

ROBERTSON, S. Understanding inverse document frequency: on theoretical arguments for idf. *Journal of Documentation*, Emerald Group Publishing Limited, v. 60, n. 5, p. 503–520, 2004.

ROCHA, L. M.; CAPPABIANCO, F. A. M.; FALCÃO, A. X. Data clustering as an optimum-path forest problem with applications in image analysis. *International Journal of Imaging Systems and Technology*, Wiley Periodicals, v. 19, n. 2, p. 50–68, 2009.

ROSENTHAL, S.; FARRA, N.; NAKOV, P. Semeval-2017 task 4: Sentiment analysis in twitter. In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. 2017. p. 502–518.

RUSSELL, J. A. A circumplex model of affect. *Journal of personality and social psychology*, American Psychological Association, v. 39, n. 6, p. 1161, 1980.

SAILUNAZ, K.; DHALIWAL, M.; ROKNE, J.; ALHAJJ, R. Emotion detection from text and speech: a survey. *Social Network Analysis and Mining*, Springer, v. 8, n. 1, p. 28, 2018.

SEOL, Y.-S.; KIM, D.-J.; KIM, H.-W. Emotion recognition from text using knowledge-based ann. In: *Preceedings of the International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*. 2008. p. 1569–1572.

SHI, J.; MALIK, J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, Washington, DC, USA, v. 22, n. 8, p. 888–905, 2000. ISSN 0162-8828.

SKOWRON, M. Affect listeners: Acquisition of affective states by means of conversational systems. In: *Development of Multimodal Interfaces: Active Listening and Synchrony*. : Springer, 2010. p. 169–181.

SMITH, L. N. No more pesky learning rate guessing games. *CoRR, abs/1506.01186*, 2015.

SOCHER, R.; PENNINGTON, J.; HUANG, E. H.; NG, A. Y.; MANNING, C. D. Semi-supervised recursive autoencoders for predicting sentiment distributions. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2011. p. 151–161.

SOCHER, R.; PERELYGIN, A.; WU, J.; CHUANG, J.; MANNING, C. D.; NG, A.; POTTS, C. Recursive deep models for semantic compositionality over a sentiment treebank. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2013. p. 1631–1642.

SOKOLOVA, M.; LAPALME, G. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, Elsevier, v. 45, n. 4, p. 427–437, 2009.

SPEER, R.; HAVASI, C. Representing general relational knowledge in conceptnet 5. In: *LREC*. 2012. p. 3679–3686.

SPERTUS, E. Smokey: Automatic recognition of hostile messages. In: *Proceedings of Ninth Conference on Innovative Applications of Artificial Intelligence*. 1997. p. 1058–1065.

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014.

STOLCKE, A.; RIES, K.; COCCARO, N.; SHRIBERG, E.; BATES, R.; JURAFSKY, D.; TAYLOR, P.; MARTIN, R.; ESS-DYKEMA, C. V.; METEER, M. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, MIT Press, v. 26, n. 3, p. 339–373, 2000.

STRAPPARAVA, C.; VALITUTTI, A. et al. Wordnet affect: an affective extension of wordnet. In: *Proceedings of the Seventh Language Resources and Evaluation Conference*. 2004. v. 4, p. 1083–1086.

SURENDRAN, D.; LEVOW, G.-A. Dialog act tagging with support vector machines and hidden markov models. In: *Ninth International Conference on Spoken Language Processing*. 2006.

SUTTON, C.; MCCALLUM, A. An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, Now Publishers, Inc., v. 4, n. 4, p. 267–373, 2012.

TABOADA, M.; BROOKE, J.; TOFILOSKI, M.; VOLL, K.; STEDE, M. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, MIT Press, v. 37, n. 2, p. 267–307, 2011.

TAI, K. S.; SOCHER, R.; MANNING, C. D. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015.

TANG, D.; QIN, B.; LIU, T. Document modeling with gated recurrent neural network for sentiment classification. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015. p. 1422–1432.

TIELEMAN, T.; HINTON, G. Lecture 6.5-rmsprop, coursera: Neural networks for machine learning. *University of Toronto, Technical Report*, 2012.

TRAN, Q. H.; ZUKERMAN, I.; HAFFARI, G. A hierarchical neural model for learning sequences of dialogue acts. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. 2017. v. 1, p. 428–437.

TURNEY, P. D. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 40th annual meeting on Association for Computational Linguistics*. 2002. p. 417–424.

VITERBI, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, IEEE, v. 13, n. 2, p. 260–269, 1967.

WANG, J.; YU, L.-C.; LAI, K. R.; ZHANG, X. Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2016. ISBN 9781510827592.

WANG, S.-M.; LI, C.-H.; LO, Y.-C.; HUANG, T.-H. K.; KU, L.-W. Sensing emotions in text messages: An application and deployment study of emotionpush. *arXiv preprint arXiv:1610.04758*, 2016.

WANG, X.; LIU, Y.; SUN, C.; WANG, B.; WANG, X. Predicting polarities of tweets by composing word embeddings with long short-term memory. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. 2015. p. 1343–1353.

WILCOXON, F. Individual comparisons by ranking methods. *Biometrics Bulletin*, v. 1, n. 6, p. 80–83, 1945.

WILLCOX, G. The feeling wheel: A tool for expanding awareness of emotions and increasing spontaneity and intimacy. *Transactional Analysis Journal*, SAGE Publications Sage CA: Los Angeles, CA, v. 12, n. 4, p. 274–276, 1982.

WILLIAMS, J.; RAUX, A.; HENDERSON, M. The dialog state tracking challenge series: A review. *Dialogue & Discourse*, v. 7, n. 3, p. 4–33, 2016.

WILLIAMS, J.; RAUX, A.; RAMACHANDRAN, D.; BLACK, A. The dialog state tracking challenge. In: *Proceedings of the Special Interest Group on Discourse and Dialogue 2013 Conference*. 2013. p. 404–413.

YANG, B.; CARDIE, C. Extracting opinion expressions with semi-markov conditional random fields. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 2012. p. 1335–1345.

YANG, B.; CARDIE, C. Context-aware learning for sentence-level sentiment analysis with posterior regularization. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2014. v. 1, p. 325–335.

YANG, X.; LIU, J.; CHEN, Z.; WU, W. Semi-supervised learning of dialogue acts using sentence similarity based on word embeddings. In: IEEE. *Audio, Language and Image Processing (ICALIP), 2014 International Conference on*. 2014. p. 882–886.

YANG, Z.; YANG, D.; DYER, C.; HE, X.; SMOLA, A.; HOVY, E. 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*. 2016. p. 1480–1489.

YASMINA, D.; HAJAR, M.; HASSANA, A. M. Using youtube comments for text-based emotion recognition. *Procedia Computer Science*, Elsevier, v. 83, p. 292–299, 2016.

ZAHIRI, S. M.; CHOI, J. D. Emotion detection on tv show transcripts with sequence-based convolutional neural networks. In: *Proceedings of the Association for the Advancement of Artificial Intelligence AAAI-18 Workshop on Affective Content Analysis*. 2018.

Apêndices

APÊNDICE A – Classificação de Atos de Diálogo

Este capítulo descreve o trabalho de Ribeiro e Papa (RIBEIRO; PAPA, 2018), elaborado durante o desenvolvimento do presente projeto de mestrado, onde é proposto um método não-supervisionado para a classificação de turnos em um diálogo quanto a sua intenção, tarefa comumente referida na literatura como classificação de atos de diálogo – do inglês *Dialogue Act classification* (DA). O problema de interesse da presente Dissertação de Mestrado diz respeito à Análise de Sentimento em textos estruturados na forma de diálogos. Entretanto, conforme já mencionado anteriormente, existem pouquíssimos trabalhos na literatura que consideram esta modalidade para a realização de AS. Com o objetivo de identificar como diálogos têm sido abordados na literatura, optamos por realizar uma análise dos trabalhos que contemplam o problema de DA, tendo em vista que o objetivo desta tarefa também consiste em classificar cada turno de uma conversa, porém ao invés de considerar o sentimento, cada amostra deve ser rotulada com sua intenção. Além disso, enquanto hipotetizamos que considerar o contexto do diálogo é importante para AS, Lee e Dernoncourt (LEE; DERNONCOURT, 2016) mostram que este aspecto é, de fato, importante para a tarefa de classificação de DA, o que contribui para a relevância de considerar tal tarefa.

Durante a realização das análises foi possível constatar que existem diversas técnicas na literatura para a classificação supervisionada de DA, ao passo poucos estudos baseados em técnicas não-supervisionadas foram desenvolvidos, conforme observado por diferentes autores (BRYCHCÍN; KRÁL, 2017; PERUMAL; HIRST, 2016). O presente trabalho foi realizado com o objetivo de contribuir com a literatura nesta direção, além de introduzir o classificador baseado em Florestas de Caminhos Ótimos – do inglês *Optimum-Path Forest* (OPF) não-supervisionado à área de PLN, comparando os resultados obtidos com outros dois algoritmos: o k-médias e o classificador hierárquico baseado em agrupamento por densidade espacial de aplicações com ruído – do inglês *Hierarchical Density-Based Spatial Clustering of Applications with Noise* (HDBSCAN) (CAMPELLO; MOULAVI; SANDER, 2013).

Um dos principais fatores que motivam a utilização de técnicas não-supervisionadas para a classificação de DA reside no fato de que anotar uma base de dados para então treinar um classificador supervisionado pode se mostrar uma tarefa cara, tanto em termos financeiros quanto em termos de tempo necessário. Isso acaba dificultando explorar a quantidade crescente de dados disponíveis pela popularização do uso da Internet. Ademais, erros de anotação causados pela ambiguidade de algumas sentenças podem ser introduzidos nesta etapa do processo, conforme observado por Stolcke et al. (STOLCKE et al., 2000). Logo, as próximas

seções são dedicadas a apresentar o classificador OPF e as principais bases de dados para a tarefa de DA, a seguir os experimentos realizados e resultados obtidos são discutidos, e então as considerações finais são apresentadas.

A.1 Aprendizado não-supervisionado com Floresta de Caminhos Ótimos

Seja $\mathcal{D} = \left\{\boldsymbol{\chi}^{(1)}, \boldsymbol{\chi}^{(2)}, \ldots, \boldsymbol{\chi}^{(T)}\right\}$ uma base de dados, onde cada um de seus elementos é um descritor $\boldsymbol{\chi}^{(i)} \in \mathbb{R}^m$. A partir deste conjunto é possível gerar um grafo $\mathcal{G} = (\mathcal{D}, \mathcal{A}_k)$, de modo que cada elemento de $\mathcal{D}$ é representado por um nó que está conectado apenas à seus k vizinhos mais próximos através da relação de adjacência $\mathcal{A}_k$.

O algoritmo OPF consiste em eleger alguns nós de $\mathcal{D}$ como "protótipos" e então fazê-los competirem entre si com o objetivo de conquistar as demais amostras do grafo. Isso faz com que $\mathcal{G}$ seja particionado em diferentes árvores de caminhos ótimos – do inglês *optimum-path trees* (OPTs), cada qual enraizada em um protótipo, o qual conquistou os demais nós. Consequentemente as amostras de uma dada OPT apresentem maior similaridade entre si quando comparadas ao restante da base de dados, formando assim um *cluster*. Este processo pode ser brevemente dividido em três fases:

1. Cálculo de um tamanho de vizinhança adequado k, permitindo a construção de $\mathcal{A}_k$;
2. Determinação dos nós protótipos;
3. Execução do processo de competição.

Para o primeiro passo, diferentes estratégias podem ser utilizadas para determinar o valor ótimo de k, designado k^*. Rocha, Cappabianco e Falcão (ROCHA; CAPPABIANCO; FALCÃO, 2009) propuseram calcular o tamanho da vizinhança como o valor que minimiza o corte do grafo $\mathcal{G}$, já que esta medida considera a dissimilaridade entre as partições geradas, bem como a similaridade interna entre as amostras conquistadas pelo protótipo da OPT (SHI; MALIK, 2000).

Em relação ao segundo passo, na versão supervisionada do algoritmo OPF, proposto por Papa, Falcão e Suzuki (PAPA; FALCÃO; SUZUKI, 2009), os protótipos são eleitos como os nós de classes distintas mais próximos entre si, situados em uma região de fronteira entre classes. Neste cenário tais amostras podem ser identificadas por meio do algoritmo da Árvore Geradora Mínima, entretanto tal abordagem mostra-se infactível para o cenário não-supervisionado, já que os rótulos de $\mathcal{D}$ não estão disponíveis, tornando impossível determinar de antemão a classe de cada amostra e, consequentemente, as regiões de fronteira. Devido a esta limitação, Rocha, Cappabianco e Falcão (ROCHA; CAPPABIANCO; FALCÃO, 2009) utilizam uma abordagem

probabilística e atribuem a cada amostra em $\mathcal{D}$ uma pontuação $\rho(\boldsymbol{\chi}^{(i)})$ por meio de uma função densidade de probabilidade (fdp) ao modelar sua vizinhança através de uma função gaussiana:

$$\rho\left(\boldsymbol{\chi}^{(t)}\right) = \frac{1}{\sqrt{2\pi\sigma^2}k} \sum_{\forall \boldsymbol{\chi}^{(j)} \in \mathcal{A}_k(\boldsymbol{\chi}^{(i)})} \exp\left(\frac{-d(\boldsymbol{\chi}^{(i)}, \boldsymbol{\chi}^{(j)})}{2\sigma^2}\right), \tag{48}$$

onde $i \neq j$ e $\sigma = d_{max}/3$, sendo que d_{max} representa comprimento do maior arco em $\mathcal{G}$. Sob esta formulação $\rho\left(\boldsymbol{\chi}^{(i)}\right)$ considera todos nós adjacentes para o cálculo da probabilidade, já que a função gaussiana abrange 99.7% de todas amostras com $d(\cdot) \in [0, 3\sigma]$ (propriedade da distribuição Normal) e $3\sigma = d_{max}$. Por fim, $\mathcal{A}_k\left(\boldsymbol{\chi}^{(i)}\right)$ retorna as k amostras mais próximas de $\boldsymbol{\chi}^{(i)}$ de acordo com a relação de adjacência utilizada.

Após computar a Equação 48 para todos os nós em $\mathcal{D}$ é possível determinar as amostras nos centros de cada *cluster*, que são as regiões de maior densidade, e então promovê-las a protótipos para que o terceiro passo seja iniciado. Os valores obtidos são utilizados para popular uma fila de prioridades (comumente implementada por meio de uma *heap*) onde a ideia do algoritmo OPF não-supervisionado consiste em particionar o grafo ao maximizar o custo de cada amostra, o qual é baseado nos caminhos do grafo. Em outras palavras, o custo depende de uma sequência de amostras conectadas em $\mathcal{A}_k$ sem formar ciclos.

Seja $\pi_{\boldsymbol{\chi}^{(i)}}$ um caminho que termine na amostra $\boldsymbol{\chi}^{(i)}$, denominado "terminal", e que tenha início em alguma raíz $\mathbb{Q}\left(\boldsymbol{\chi}^{(j)}\right)$, onde $\mathbb{Q}$ é o conjunto formado por todos protótipos. Além disso, considere $\pi_{\boldsymbol{\chi}^{(i)}} = \langle \boldsymbol{\chi}^{(i)} \rangle$ um caminho trivial, ou seja, formado por um único nó e a operação $\pi_{\boldsymbol{\chi}^{(i)}} \cdot \langle \boldsymbol{\chi}^{(i)}, \boldsymbol{\chi}^{(j)} \rangle$, que corresponde à concatenação do caminho $\pi_{\boldsymbol{\chi}^{(i)}}$ e o arco $\left(\boldsymbol{\chi}^{(i)}, \boldsymbol{\chi}^{(j)}\right)$, com a restrição de que $i \neq j$, evitando o surgimento de ciclos.

O Algoritmo OPF atribui um custo real $f\left(\pi_{\boldsymbol{\chi}^{(i)}}\right)$ a cada caminho $\pi_{\boldsymbol{\chi}^{(i)}}$ com base em uma função de conectividade. Desta forma, um caminho é dito ótimo se $f\left(\pi_{\boldsymbol{\chi}^{(i)}}\right) \geq f\left(\hat{\pi}_{\boldsymbol{\chi}^{(i)}}\right)$ para qualquer outro caminho $\hat{\pi}_{\boldsymbol{\chi}^{(i)}}$ que também termine em $\boldsymbol{\chi}^{(i)}$. Este tipo de função é dita suave e garante que as restrições fundamentais para a exatidão teórica do algoritmo OPF sejam verificadas (FALCÃO; STOLFI; LOTUFO, 2004). Ainda assim, diferentes funções de custo podem ser consideradas, dentre as quais o OPF não-supervisionado emprega a seguinte definição para $\forall \boldsymbol{\chi}^{(i)}, \boldsymbol{\chi}^{(j)} \in \mathcal{D}$, novamente com $i \neq j$:

$$f\left(\langle \boldsymbol{\chi}^{(i)} \rangle\right) = \begin{cases} \rho\left(\boldsymbol{\chi}^{(i)}\right) & \text{se } \boldsymbol{\chi}^{(i)} \in \mathbb{Q} \\ \rho\left(\boldsymbol{\chi}^{(i)}\right) - \delta & \text{caso contrário,} \end{cases} \tag{49}$$

e

$$f\left(\pi_{\boldsymbol{\chi}^{(i)}} \cdot \langle \boldsymbol{\chi}^{(i)}, \boldsymbol{\chi}^{(j)} \rangle\right) = \min\left\{f\left(\boldsymbol{\chi}^{(i)}\right), \rho\left(\boldsymbol{\chi}^{(j)}\right)\right\}, \tag{50}$$

onde $\delta = \min_{\forall(\boldsymbol{\chi}^{(t)},\boldsymbol{\chi}^{(s)})\in\mathcal{A}_k|\rho(t)\neq\rho(s)} |\rho(t) - \rho(s)|$ e corresponde à menor quantidade necessária para evitar platôs nas regiões próximas aos protótipos, as quais correspondem a regiões de maior densidade.

Dentre todos possíveis caminhos que se originam nos pontos máximos da fdp (protótipos) e terminam em $\pi_{\boldsymbol{\chi}^{(t)}}$, o algoritmo associa à amostra $\boldsymbol{\chi}^{(t)}$ o caminho ótimo com menor valor de densidade entre as densidades máximas que o formam. Tal caminho é representado pelo seguinte mapa de custos:

$$C\left(\boldsymbol{\chi}^{(i)}\right) = \max_{\forall\pi_{\boldsymbol{\chi}^{(j)}}\in(\mathcal{D},\mathcal{A}_k),\, i\neq j} \left\{ f\left(\pi_{\boldsymbol{\chi}^{(j)}} \cdot \langle \boldsymbol{\chi}^{(j)}, \boldsymbol{\chi}^{(i)} \rangle\right)\right\}. \tag{51}$$

O algoritmo OPF maximiza o mapa de custos anterior para todos nós da base de dados ao computar uma floresta de caminhos ótimos sobre o $\mathcal{G}$, consequentemente particionando-o em *clusters* formados pelas amostras mais similares entre si. Tal floresta é representada através de um mapa de predecessores $P\left(\boldsymbol{\chi}^{(i)}\right)$ que retorna o marcador especial nil para protótipos ou o nó pai de $\boldsymbol{\chi}^{(i)}$ na OPT para as demais amostras. Por fim, todo o processo é sumarizado no Algoritmo 1.

ALGORITMO 1: OPF não-supervisionado.

Entrada: grafo $\mathcal{G}(\mathcal{D}, \mathcal{A}_k)$, função de densidade $\rho(\cdot)$ e parâmetro δ.
Saída: mapa de custos C, mapa de predecessores P e mapa de rótulos L.
Auxilizares : fila de prioridades máxima Q, variáveis $temp$ e $l \leftarrow 1$.

1 **para todo** $s \in \mathcal{D}$, **faça**
2 $P(s) \leftarrow nil$
3 $C(s) \leftarrow \rho(s) - \delta$
4 insira s em Q
5 **enquanto** Q *é não vazia,* **faça**
6 Remova de Q uma amostra s tal que $C(s)$ é máximo
7 **se** $P(s) = nil$, **então**
8 $C(s) \leftarrow \rho(s)$
9 $L(s) \leftarrow l$
10 $l \leftarrow l + 1$
11 **para todo** $t \in \mathcal{A}_k(s)$ *com* $C(t) < C(s)$, **faça**
12 $temp \leftarrow \min\{C(s), \rho(t)\}$
13 **se** $temp > C(t)$, **então**
14 $L(t) \leftarrow L(s)$
15 $P(t) \leftarrow s$
16 $C(t) \leftarrow temp$
17 Atualize a posição do nó t na fila Q
18 **retorna** C, P, L

O algoritmo é responsável por identificar um protótipo para cada máximo da fdp na Linha 7 (lembre-se que Q está em ordem decrescente e que protótipos não possuem antecessores

por serem as raízes das OPTs), associá-lo à um novo rótulo l e determinar sua zona de influência ao formar uma árvore, na qual todos seus elementos receberão o mesmo rótulo do protótipo (Linha 14). Por fim, o mapa de custos C, o mapa de rótulos L, responsável por definir a qual *cluster* cada amostra está associada, bem como seu protótipo dominante e o mapa de predecessores P são retornados. Por meio desta última estrutura é possível, se necessário, reconstruir todas OPTs que particionaram o grafo.

Finalmente, a vantagem do algoritmo OPF frente aos demais é o fato de que a quantidade de *clusters* é determinada automaticamente, sendo apenas necessário fixar o intervalo de busca inteiro $[k_{min}, k_{max}]$ para a determinação de $k^{\star}$.

A.2 Bases de Dados

Nesta seção as bases de dados mais comumente utilizadas para classificação de DA são apresentadas em conjunto com seus trabalhos mais recentes. Apesar da existência de diferentes recursos para esta tarefa, a maior parte dos trabalhos baseados em aprendizado não-supervisionado acaba por utilizar dados de domínios específicos não disponíveis publicamente, dificultando a reprodução dos estudos em questão. Por outro lado, algumas das bases públicas utilizadas em trabalhos supervisionados disponibilizam uma divisão especifica para as partições de treinamento, validação e teste. Todavia é difícil reproduzir cada conjunto com exatidão, tendo em vista que antes é necessário pré-processar a base com o objetivo de remover amostras inválidas e agrupar múltiplos rótulos em uma única classe.

Outro aspecto bastante interessante nestes conjuntos de dados diz respeito à distribuição de frequência das classes. Na base de dados do Instituto Internacional de Ciência da Computação – do inglês *International Computer Science Institute* (ICSI) (JANIN et al., 2003), por exemplo, a classe mais comum (*afirmação*) corresponde a 59% de todas amostras, sendo 4,2 vezes maior que a segunda mais comum. O mesmo ocorre na base de dados *Switchboard Dialogue Act Corpus* (SWDA) (JURAFSKY, 1997), onde o rótulo mais comum (*afirmação-não-opinião*) concentra 36% dos dados, ao passo que 10 outras classes juntas representam 10% das amostras, enquanto que outras 25 classes juntas (mais da metade dos 42 possíveis rótulos) correspondem a apenas 5% de toda a base[26]. É importante salientar que o problema de desbalanceamento das classes é amplamente conhecido na área de aprendizado de máquina, tendo em vista que um classificador treinado sob este regime pode se tornar enviesado em favor das classes mais frequentes, já que no caso de dúvida durante a análise de uma amostra, a probabilidade de acerto ao adivinhar seu rótulo como a classe mais frequente é maior.

O trabalho não-supervisionado aqui descrito foi desenvolvido sobre três bases de dados, descritas a seguir: o conjunto *Human Comunication Research Centre (HCRC) Maptask* (CAR-

[26] O código utilizado para esta análise, bem como os resultados podem ser obtidos em <https://github.com/lzfelix/sibgrapi2018_opf>.

LETTA et al., 1997); a base de conversas em salas de bate papo *Naval Postgraduate School (NPS) Internet Chatroom Conversations* (FORSYTHAND; MARTELL, 2007) e a base de dados *ICSI* (a qual já foi brevemente discutida), tendo em vista que as mesmas estão disponíveis publicamente, são formadas por conversas entre humanos apenas e possuem uma quantidade razoável de classes. Visando uma revisão completa, a base de dados SWDA, bem como os múltiplos *corpora* divulgados no *Dialog System Technology Challenges* (DSTC) também são brevemente apresentadas.

A.2.1 HCRC

Esta base é formada pela transcrição de 128 diálogos entre dois participantes, anotados com 12 intenções e uma classe adicional com amostras não rotuladas. Como não existe divisão padrão para os conjuntos de treinamento, avaliação e teste, os autores utilizem-na de maneiras diferente. Em relação aos trabalhos supervisionados, Surendran e Levow (SURENDRAN; LEVOW, 2006) conseguem 59% de acurácia através do uso de SVMs aliadas ao Algoritmo de Viterbi (vide Capítulo 4.3) considerando apenas a metade da base de dados onde os participantes não estabelecem contato visual. Tran, Zukerman e Haffari (TRAN; ZUKERMAN; HAFFARI, 2017) alcançam 63,3% de acurácia ao utilizar LSTMSs hierárquicas (vide Capítulo 3.2.1) em conjunto com um mecanismo de atenção (vide Capítulo 3.2.4). Por fim, Di Eugenio et al. (EUGENIO; XIE; SERAFIN, 2010) obtém 78,76% de taxa de acerto ao utilizar o algoritmo dos k-vizinhos mais próximos ao considerar como características não apenas os turnos dos participantes, mas também outras informações, como a classe da amostra anterior, além da identidade dos interlocutores, com o objetivo de considerar informação de contexto.

A.2.2 NPS *Internet Chatroom*

O conjunto de dados em questão consiste em conversas entre múltiplos participantes extraídos a partir de 15 salas de bate-papo e apresenta características específicas deste domínio, como a presença de erros de digitação, abreviações, gírias e emojis. Neste caso, existem 15 classes, uma das quais corresponde a notificações do sistema alertando sobre ações tomadas pelos participantes da sala. Assim como no caso anterior não existe uma partição oficial para esta base de dados. Em relação à experimentos, Moldovan et al. (MOLDOVAN; RUS; GRAESSER, 2011) conseguem 78,25% de acurácia por meio de um Classificador Bayesiano. No tocante às abordagens não supervisionadas, Jo et al. (JO et al., 2017) utilizam modelos gráficos e alcançam 0,33 pontos na medida V – do inglês *V-measure*, a qual será discutido no Capítulo A.3.2. Os mesmos autores também reproduzem os trabalhos de Brychcín e Král (BRYCHCÍN; KRÁL, 2017), Ezen-Can e Boyer (EZEN-CAN; BOYER, 2015) e de Lee et al. (LEE et al., 2013) na mesma base, obtendo *V-measure* de, respectivamente, 0,28, 0,28 e 0,31.

A.2.3 ICSI

Esta base é formada pela transcrição de 75 conversas entre múltiplos participantes, onde cada turno é anotado com uma das 11 intenções gerais e uma combinação de 39 rótulos específicos, totalizando 2083 combinações, as quais podem ser rearranjadas em apenas 5 classes (ANG; LIU; SHRIBERG, 2005). Adicionalmente, as partições de treinamento, validação e teste são fornecidas. Em relação à classificação de DA supervisionada, Ortega et al. (ORTEGA; VU, 2017) obtém 84,3% de acurácia ao utilizar LSTMs aliadas à um mecanismo de atenção; Lee e Dernoncourt (LEE; DERNONCOURT, 2016) alcançam 84,6% ao considerar fala atual contextualizada com um ou dois turnos anteriores ao utilizar CNNs e LSTMs. Por fim, Kumar et al. (KUMAR et al., 2017) conseguem 90,9% de acurácia por meio de LSTMs bidirecionais com CRFs (vide Capítulo 4).

A.2.4 SWDA

A base de dados *Switchboard*, ou SWDA, é formada pela transcrição de 1155 conversas de telefone casuais também entre pares de participantes anotadas com 220 rótulos, os quais podem ser agrupadas em 42 macro-classes[27] e são comumente utilizadas na literatura para a tarefa de DA. Stolcke et al. (STOLCKE et al., 2000) apresentam uma divisão para as partições de treinamento e teste, sendo a primeira subdividida por Lee e Dernoncourt (LEE; DERNONCOURT, 2016) entre treinamento e validação. É interessante notar que três dos rótulos menos frequentes dentre 42 classes não estão presentes na partição de testes.

Sobre métodos supervisionados, Stolcke et al. (STOLCKE et al., 2000) obtiveram 71% de taxa de acerto com Modelos Ocultos de Markov – do inglês *Hidden Markov Models* (HMM), já Kalchbrenner and Blunsom (KALCHBRENNER; BLUNSOM, 2013) alcançaram 73,9% de acurácia por meio de CNNs Recorrentes, enquanto que Tran et al. (TRAN; ZUKERMAN; HAFFARI, 2017) conseguiram 74,5% de acurácia ao empregar redes LSTM hierárquicas, com o objetivo de capturar informação de contexto. Por fim, Kumar et al. (KUMAR et al., 2017) conseguiram 79,2% de taxa de acerto ao empregar CRFs.

Em relação à abordagens não supervisionadas, Yang et al. (YANG et al., 2014) utilizam o algoritmo das k-médias considerando apenas as dez classes mais frequentes e 50.000 turnos, obtendo acurácia de 78,62%. Já Brychcín e Král (BRYCHCÍN; KRÁL, 2017) conseguem 65,7% de pontuação F1 por meio de HMMs com distribuições gaussianas multivariadas. Apesar destes autores utilizarem a mesma partição de Stolcke et al. (STOLCKE et al., 2000), torna-se difícil comparar estes resultados já que a métrica de acurácia não é disponibilizada.

[27] Utilizamos o processo de agrupamento disponibilizado por Christopher Potts em <http://compprag.christopherpotts.net/swda.html>.

A.2.5 DSTC

Durante suas múltiplas edições o *Dialog System Technology Challenges* divulgou várias tarefas para a extração de informações a partir de turnos de uma conversa com o objetivo de fomentar esta linha de pesquisa (WILLIAMS; RAUX; HENDERSON, 2016). Em suas três primeiras edições, as bases disponibilizadas eram formadas por transcrições de conversas entre um humano e uma máquina. Apesar das mesmas possuírem informação de DA, os participantes não foram solicitados a realizar esta tarefa. No DSTC4 (KIM et al., 2016b) e DSTC5 (KIM et al., 2016a) uma tarefa de classificação de DA multi-rótulo foi introduzida ao utilizar o *corpus* TourSG (WILLIAMS et al., 2013). Apesar de ser formado por conversas apenas entre humanos, a mesma não é disponibilizada publicamente.

A.3 Materiais e Métodos

Como cada base de dados advém de um domínio diferente com características particulares, as mesmas foram pré-processadas conforme necessário. Basicamente esta tarefa consistiu em converter todas palavras para letras minúsculas, desprezar aquelas presentes em uma lista de *stop words* ou com frequência menor que f_{min}, determinada empiricamente, além de remover símbolos especiais, exceto para a base ICSI. Neste conjunto de dados a pontuação foi adicionada durante o processo de transcrição e consiste apenas em interrogações e pontos finais, indicando o fim de um turno, além de um símbolo especial que indica falas incompletas. Menções aos nomes dos usuários na base de dados NPS foram substituídas pela palavra *user* (usuário) durante os experimentos com o objetivo de mapear tais menções para um *word embedding* conhecido. No que diz respeito às classes, a base ICSI foi pré-processada para conter apenas 5 rótulos, ao passo que apenas as 12 classes rotuladas do conjunto HCRC foram consideradas. A Tabela 17 sumariza as características de cada base de dados, onde $|L|$ representa o número de classes e $|V|$ o tamanho do vocabulário, ou seja, a quantidade de palavras únicas consideradas.

Tabela 17 – Características das bases de dados consideradas.

Base	$\|L\|$	$\|V\|$	f_{min}	**# Amostras**	**Tamanho da partição**
NPS	15	3.885	1	10.568	697
HCRC	12	1.052	2	26.158	1.738
ICSI	5	5.004	3	106.047	7.068

Fonte: Ribeiro e Papa (RIBEIRO; PAPA, 2018).

A.3.1 Extração de Características

Cada sentença das bases de dados foi segmentada através do *Stanford Tokenizer* (MANNING et al., 2014) e as palavras obtidas foram mapeadas para seus *word embeddings* ao utilizar um modelo GloVe pré-treinado[28] com 300 dimensões. Esta decisão foi tomada empiricamente

[28] Disponível em <http://nlp.stanford.edu/data/glove.840B.300d.zip>.

com base na observação de que o modelo em questão continha a representação de mais palavras que versões pré-treinadas do word2vec. Palavras sem vetores correspondentes são descartadas. Cada sentença pode ser vista como uma sequência de palavras, $\mathcal{S}^{(i)} = (w^{(1)}, w^{(2)}, \ldots, w^{(\tau_i)})$ que são mapeadas para seus vetores correspondentes e a partir dos quais deseja-se gerar um descritor de sentença de tamanho fixo $\boldsymbol{s}^{(i)} \in \mathbb{R}^d$. Como o trabalho em questão é não-supervisionado, as técnicas apresentadas no Capítulo 3 não podem ser empregadas, tendo em vista que o aprendizado dos parâmetros das redes neurais dependem desta informação.

Devido a esta restrição, a abordagem proposta por Arora et al. (ARORA; LIANG; MA, 2017) é utilizada, já que apesar de simples, a mesma apresenta resultados superiores que alternativas baseadas em RNNs para alguns casos. Sob esta abordagem, uma representação inicial para cada sentença (ou turno) é obtida da seguinte forma:

$$\hat{\boldsymbol{s}}^{(i)} = \frac{1}{|\mathcal{S}^{(i)}|} \sum_{w \in \mathcal{S}^{(i)}} \frac{a}{a + p(w)} \boldsymbol{v}_w, \tag{52}$$

onde $p(w)$ é a probabilidade de ocorrência da palavra w na base de dados utilizada para treinar o modelo GloVe, $|\mathcal{S}^{(i)}|$ é o comprimento do turno em termos de palavras e a é um hiperparâmetro de suavização. Para as configurações em questão $a = 10^{-3}$, observando a análise realizada por Arora et al. (ARORA; LIANG; MA, 2017).

Os vetores obtidos $\hat{\boldsymbol{s}}^{(i)}$ são empilhados, formando uma matriz $E \in \mathbb{R}^{N \times d}$, em que N é o tamanho da base de dados e $d = 300$ neste caso. O primeiro vetor singular de E, designado $\boldsymbol{r}$, é calculado e os descritores de sentenças são atualizados através da seguinte equação:

$$\boldsymbol{s}^{(t)} = \hat{\boldsymbol{s}}^{(i)} - \boldsymbol{r}\boldsymbol{r}^T \hat{\boldsymbol{s}}^{(i)}, \tag{53}$$

para que então sejam normalizados com comprimento unitário, de modo que a função similaridade de cosseno na Equação 1 torne-se proporcional à distância euclidiana.

A.3.2 Procedimento de Avaliação

Com o objetivo de avaliar os resultados obtidos pelos classificadores k-médias, HDBSCAN[29] e OPF[30], um procedimento de validação cruzada foi utilizada considerando 15 partições, das quais 13 foram utilizadas para o treinamento, uma para a validação e a última para os testes. O conjunto de validação foi utilizado para ajustar os seguintes hiperparâmetros: k para o k-médias, β_1 e β_2 para o HDBSCAN e o intervalo de busca $[k_{min}, k_{max}]$ para o tamanho da vizinhança k^* no algoritmo OPF. Apesar da possibilidade de fixar $k_{min} = 1$ e variar apenas o

[29] As implementações disponíveis em `scikit-learn` e `scikit-learn-contrib` foram utilizadas, respectivamente.

[30] Uma implementação modificada da `LibOPF`, disponível em <https://github.com/lzfelix/LibOPF/tree/unsupervised> foi utilizada.

limitante superior do intervalo, foi escolhido utilizar vários intervalos de busca menores com o objetivo de estudar como o valor de k^* influencia na qualidade das partições geradas.

É interessante mencionar que os hiperparâmetros do classificador HDBSCAN β_1 e β_2 controlam, respectivamente, a quantidade mínima de amostras necessárias para formar um *cluster* e quão rigorosamente o algoritmo marca amostras como ruído, ou seja, elementos espúrios que não pertencem a nenhuma das partições, de modo que valores maiores implicam em mais elementos serem atribuídos a esta classe.

Após o particionamento da base de dados por qualquer algoritmo, é necessário determinar a qual classe cada *cluster* corresponde. Como o estudo foi realizado a partir de bases de dados rotuladas, para os algoritmos k-médias e HDBSCAN as partições são determinadas por voto de maioria, ou seja, a classe mais frequente em uma partição acaba dominando todas suas amostras. Por outro lado, na concepção inicial do OPF, o rótulo do protótipo é propagado para todos seus elementos, conforme visto no Algoritmo 1. A mesma abordagem pode ser utilizada neste caso, entretanto a classe real da raiz de cada OPT deve ser propagada para todos seus elementos para a rotulação. Nos experimentos realizados também propomos a abordagem alternativa de dominar cada OPT com o rótulo real da classe mais frequente em suas amostras, denominando esta abordagem *Majority OPF* (M-OPF). Note que as partições formadas pelo OPF e M-OPF são idênticas, tendo em vista que a variante proposta apenas altera a estratégia utilizada para a propagação dos rótulos.

Por fim, para avaliar os resultados obtidos acurácia e *V-measure* são consideradas, sendo a última a média harmônica entre homogeneidade (H) e completude (C), onde $H \in [0, 1]$ e $C \in [0, 1]$. A homogeneidade é máxima quaando cada partição da base de dados é formada apenas por amostras da mesma classe, enquanto que $C = 1$ quando todas amostras de mesmo rótulo são alocadas em um mesmo *cluster*. Por fim, a significância estatística dos resultados obtidos é determinada por meio do teste assinalado de Wilcoxon (WILCOXON, 1945) com $p = 0.05$.

A.4 Resultados Experimentais

Os melhores hiperparâmetros para cada modelo foram determinados por meio de uma busca em grade utilizando os intervalos apresentados na Tabela 18. Para o classificador k-médias o intervalo $[2, 15]$ foi utilizado apenas para a base de dados ICSI, tendo em vista que é sabido de antemaão que a mesma é formada por 5 classes. No tocante ao HDBSCAN, como nenhuma amostra é originalmente rotulada como ruído, todas amostras atribuídas à esta classe pelo algoritmo são agrupadas em uma nova partição, que é rotulada assim como as demais. Por fim, testes preliminares mostraram que aumentar β_2 impacta negativamente no desempenho do classificador, por isso seu valor for fixado em 1.

Os resultados de cada classificador são apresentados na Tabela 19 onde os melhores

Tabela 18 – Intervalos de busca para cada hiperparâmetro.

Modelos	**Hiperparâmetros**
k-médias	$k \in [2, 15]$ ou $k \in [2, 20]$
OPF, M-OPF	$[k_{min}, k_{max}] \in \{[1, 5], [5, 10], [10, 20], [20, 50]$ $[50, 100], [100, 150]\}$
HDBSCAN	$\beta_1 \in \{5, 10, 15, 20, 25, 30, 35, 40\}$; $\beta_2 = 1$

Fonte: Ribeiro e Papa (RIBEIRO; PAPA, 2018).

valores, de acordo com o teste de Wilcoxon, são destacados em negrito. Em relação aos hiperparâmetros foram utilizados $k = 20$ para o k-médias (exceto para a base ICSI, onde $k = 15$), para o HDBSCAN, $\beta_1 = 5$ e $\beta_2 = 1$. Por fim, $k^* = 5$ foi empregado para o OPF e M-OPF.

Tabela 19 – Resultados experimentais para cada classificador e base de dados.

Modelos	**ICSI**	**HCRC**	**NPS**
Homogeneidade			
k-Means	0,49 ± 0,02	0,24 ± 0,01	0,41 ± 0,02
OPF	0,47 ± 0,03	0,21 ± 0,02	**0,51 ± 0,02**
M-OPF	**0,55 ± 0,01**	0,29 ± 0,01	**0,50 ± 0,02**
HDBSCAN	0,42 ± 0,01	**0,34 ± 0,01**	0,44 ± 0,02
Completude			
k-Means	**0,57 ± 0,02**	0,32 ± 0,01	0,62 ± 0,04
OPF	0,47 ± 0,02	0,21 ± 0,01	0,48 ± 0,02
M-OPF	0,55 ± 0,01	0,33 ± 0,02	0,48 ± 0,02
HDBSCAN	0,50 ± 0,01	**0,36 ± 0,01**	**0,66 ± 0,03**
V-Measure			
k-means	0,53 ± 0,02	0,28 ± 0,01	0,49 ± 0,02
OPF	0,47 ± 0,02	0,21 ± 0,01	0,50 ± 0,02
M-OPF	**0,55 ± 0,01**	0,28 ± 0,01	**0,54 ± 0,02**
HDBSCAN	0,46 ± 0,01	**0,30 ± 0,01**	0,53 ± 0,02
Acurácia			
k-means	80,18 ± 0,63	41,20 ± 1,30	68,72 ± 1.63
OPF	78,81 ± 1,57	34,96 ± 1,65	64,25 ± 1,64
M-OPF	**83,09 ± 0,35**	**43,55 ± 1,17**	**71,31 ± 1,59**
HDBSCAN	77,64 ± 0,39	**43,77 ± 1,00**	69,93 ± 2,07

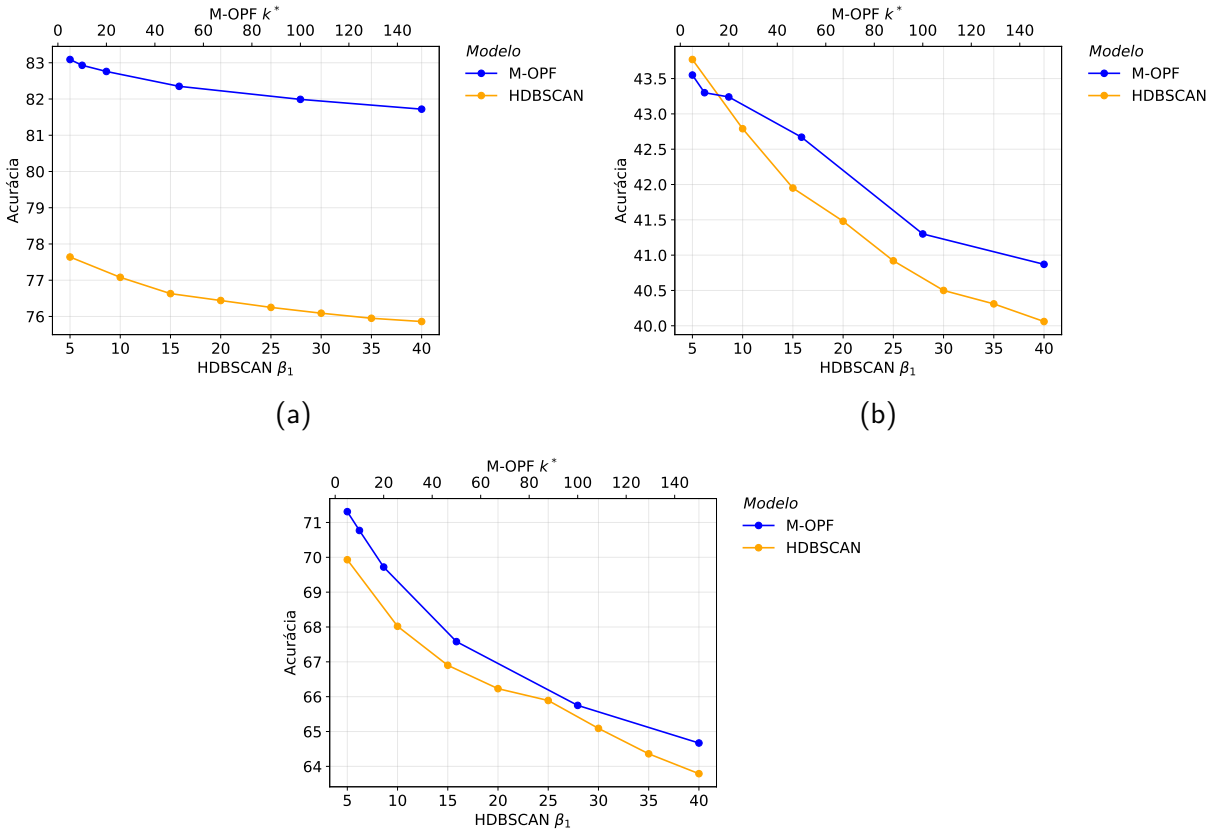
Fonte: Ribeiro e Papa (RIBEIRO; PAPA, 2018).

Ao analizar as medidas de homogeneidade, é possível inferir que o M-OPF favorece a criação de partições dominadas por uma única classe quando comparado às outras abordagens. Novamente, é importante salientar que a diferença nos resultados entre os métodos OPF e M-OPF diz respeito apenas à estratégia de propagação dos rótulos. Ao considerar a completude podemos concluir que o HDBSCAN é melhor em alocar mais amostras da mesma classe no mesmo *cluster*. Tal resultado pode ser atribuído ao fato deste classificador não supor que a base de dados segue uma distribuição gaussiana, sendo capaz de gerar partições com formatos arbitrários. Por fim, quando homogeneidade e completude são consideradas por meio da *V-measure*, é possível observar que o M-OPF oferece o melhor compromisso entre ambos

aspectos em duas bases de dados, sendo apenas ligeiramente pior que o HDBSCAN na base HCRC.

Em termos de acurácia, o M-OPF apresenta desempenho superior aos outros modelos, exceto para a base de dados HCRC, apesar desta diferença não ser estatisticamente significativa. Por outro lado, nos demais casos o M-OPF apresenta o menor desvio padrão, indicando resultados mais estáveis. A seguir a acurácia dos classificadores M-OPF e HDBSCAN é analizada em função de seus hiperparâmetros com o objetivo de obter um melhor entendimento sobre a influência destes valores nos resultados gerados. A Figura 22 mostra a performance de ambos classificadores conforme k^* e β_1 variam. Apenas duas técnicas são comparadas pois as cada hiperparâmetro varia em escalas diferentes. Além disso, o HDBSCAN também apresenta bons resultados sob a *V-measure*.

Figura 22 – Comparação de acurácia para diferentes valores de k^* para o M-OPF e β_1 para o HDBSCAN nas bases de dados (a) ICSI (b) HCRC (c) NPS.



Fonte: Ribeiro e Papa (RIBEIRO; PAPA, 2018)

Apesar do HDBSCAN possuir dois hiperparâmetros (β_2 foi inicialmente fixado em 1), o M-OPF apresenta melhores resultados de acurácia que a melhor versão do primeiro classificador, independente da escolha de seu único hiperparâmetro na base de dados ICSI. Para o conjunto HCRC, ambos classificadores produzem resultados estatisticamente similares para os melhores

hiperparâmetros, todavia quando $k^* > 5$ e $\beta_1 > 5$, o M-OPF passa a apresentar resultados superiores com significância estatística, conforme ilustrado na Figura 22b. De acordo com a Figura 22c, o M-OPF começa a gerar resultados piores que a melhor versão do HDBSCAN com $k^* = 20$, todavia esta diferença não é estatisticamente significativa, portanto o mesmo torna-se pior que o segundo classificador apenas com $k^* = 50$. Assim, é razoável assumir que o M-OPF, e consequentemente o OPF, são menos sensíveis à escolha de seu hiperparâmetro quando comparados aos outros modelos considerados, presumidamente requerindo menor conhecimento sobre o domínio da aplicação para gerar bons resultados.

A.5 Considerações Finais

O estudo discutido foi responsável por introduzir o classificador OPF não-supervisionado ao domínio de PLN, mais especificamente para a tarefa de classificação de Atos de Diálogos. Ademais, foi proposto uma pequena modificação na estratégia utilizada para determinar qual classe domina cada partição gerada pelo algoritmo OPF. Para cada OPT, ao invés de propagar a classe real de sua raiz para os demais elementos, o rótulo real mais frequente em cada partição é utilizado. Apesar de simples e factível apenas quando os rótulos reais das amostras encontram-se disponíveis, tal procedimento auxilia a evidenciar o poder de segmentação do algoritmo OPF, tendo em vista que as partições geradas pelo OPF e M-OPF são idênticas. Sob este novo regime, o M-OPF apresentou bons resultados de acurácia e *V-score* nas três bases de dados consideradas. Além disso, o M-OPF se mostrou menos sensível à escolha de seu único hiperparâmetro quando comparado ao HDBSCAN. Por fim, apesar da existência de diferentes bases de dados publicamente disponibilizadas para a tarefa de DA, a maior parte do trabalho que aborda este problema se baseia em técnicas supervisionadas, o que acaba por deixar oportunidades ainda a serem exploradas por métodos não-supervisionados.