

Exploring Supervised Techniques for Automated Recognition of Intention Classes from Portuguese Free Texts on Agriculture

Explorando Técnicas de Aprendizado de Máquina Supervisionadas para o Reconhecimento de Classes de Intenções em Textos Livres sobre Agricultura em Português

Daniel Felix de Brito¹, Jarbas Lopes Cardoso Júnior¹, Júlio Cesar dos Reis², Guilherme Ruppert¹, Rodrigo Bonacin^{1,3*}

Resumo: Technical and scientific knowledge is vast and complex, particularly in interdisciplinary fields such as sustainable agriculture, which is available in several interrelated, geographically dispersed and interdisciplinary online textual information sources. In this context, it is essential to support people with computational mechanisms that allow them to retrieve and interpret information in an appropriate way, as communication in these software systems is typically asynchronous and textual. User's intention recognition and analysis in textual documents results in benefits for better information retrieval. However, intentions are expressed implicitly in texts in natural language and the specificities of the domain and cultural aspects of language make it difficult to process and analyze the text by computer systems. This requires the study of methods for the automatic recognition of intention classes in text. In this article, we conduct extensive experimental analyses on techniques based on language models and machine learning to detect instances of intention classes in texts about sustainable agriculture written in Portuguese. In our methodology, we perform a morphological analysis of the sentences and evaluate four *Word Embeddings* techniques (*Word2Vec*, *Wang2Vec*, *FastText* and *Glove*) combined with four machine learning techniques (Support Vector Machine, Artificial Neural Network, Random Forest and Transfer Learning). The results obtained by applying the techniques proposed in a database with textual information on sustainable agriculture indicate promising possibilities in the recognition of intentions in free texts in Portuguese language on sustainable agriculture.

Keywords: Intention Detection; Sustainable Agriculture; Word Embeddings; Illocution Class; Intentions Recognition; Machine Learning

Resumo: O conhecimento técnico e científico é vasto e complexo, particularmente em áreas interdisciplinares tais como a agricultura sustentável, que provém de várias fontes de informações textuais *online* inter-relacionadas, geograficamente dispersas e interdisciplinares. Nesse contexto, é essencial apoiar pessoas com mecanismos computacionais que permitam recuperação e interpretação de informações adequadamente, pois a comunicação nesses sistemas de software é tipicamente assíncrona e textual. O reconhecimento e análise da intenção do usuário em documentos textuais traz benefícios para essas tarefas computacionais. No entanto, intenções são expressas implicitamente em textos em linguagem natural. Especificidades do domínio e aspectos culturais da linguagem dificultam o processamento e análise do texto. Isso exige investigações aprofundadas de métodos de reconhecimento automático de tipos de intenção em texto. Neste artigo, conduzimos análises experimentais extensivas sobre técnicas baseadas em modelos de linguagem e aprendizado de máquina para detectar instâncias de classes de intenções em textos em linguagem natural na língua Portuguesa sobre agricultura sustentável. Nossa metodologia considerou a aplicação de análise morfológica das sentenças e investigou as técnicas de *Word Embeddings* *Word2Vec*, *Wang2Vec*, *FastText* e *Glove* em atributos numéricos, combinadas com as técnicas de aprendizado de máquina *Support Vector Machine* (SVM), Rede Neural Artificial (ANN) e *Random Forest*. Resultados obtidos em uma base de dados com informações textuais sobre agricultura sustentável indicam possibilidades promissoras no processamento de intenções em textos com o uso de métodos supervisionados.

Palavras-Chave: Detecção de Intenção; Agricultura Sustentável; Reconhecimento de Intenção; Classes de Ilocução; Aprendizado de Máquina

¹ Centro de Tecnologia da Informação Renato Archer (CTI), Campinas - São Paulo, Brasil

² Instituto de Computação, Universidade Estadual de Campinas (UNICAMP), Campinas - São Paulo, Brasil

³ Centro Universitário Campo Limpo Paulista (UNIFACCAMP), Campo Limpo Paulista - São Paulo, Brasil

*Corresponding author: rodrigo.bonacin@cti.gov.br

DOI: <http://dx.doi.org/10.22456/2175-2745.117481> • Received: 06/10/2021 • Accepted: 02/03/2022

CC BY-NC-ND 4.0 - This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

1. Introdução

A conjuntura socioeconômica e ambiental mundial proporciona desafios e aumenta o interesse no desenvolvimento de uma agricultura sustentável, com repercussões nas áreas social e econômica. A sustentabilidade é um aspecto fundamental não apenas para aumentar a competitividade dos sistemas de produção agrícola, mas também para garantir que as atividades antrópicas no planeta não ameacem a humanidade no futuro. A formulação de indicadores de sustentabilidade, bem como práticas da agricultura sustentável, inclui vários métodos e técnicas que envolvem colaboração multidisciplinar intensiva. O desenvolvimento de comunidades multidisciplinares de pesquisadores e especialistas é um processo que pode enfrentar questões complexas devido a aspectos de comunicação, tais como o compartilhamento de intenções.

O grande interesse pela agricultura sustentável produziu um enorme volume de dados a serem analisados, disponíveis em várias publicações como revistas científicas, relatórios técnicos, anais de conferências e mídia tradicional. Esses dados não estão em um formato padronizado e, de fato, estão principalmente em formato de texto livre. Assim, podemos identificar vários desafios relacionados à computação, como os ligados ao compartilhamento de informações, análise de dados e Recuperação de Informação (RI). O que torna a agricultura sustentável um domínio rico para o estudo e avanço dos sistemas de informações que visam apoiar o compartilhamento, análise e RI sobre conteúdo técnico e científico.

O avanço em análises automatizadas e RI sobre texto livres depende muitas vezes em se ter meios de rotular texto via o reconhecimento e classificação de intenções e entidades. A literatura em RI apresenta benefícios ao considerarmos intenções em mecanismos de busca, por fornecer resultados de acordo com a intenção de quem executa a busca e de quem produziu o conteúdo [1]. Estudos com aprendizado de máquina podem, por exemplo, apoiar a definição de indicadores de sustentabilidade com base na enorme quantidade de dados disponíveis, e refinar metodologias de RI para retornar resultados que realmente tenham uma alta correlação com o que se está buscando.

Técnicas de aprendizado de máquina e mineração de texto vem sendo largamente utilizadas em diversas aplicações relacionadas a análise e classificação de textos em linguagem natural, incluindo conteúdos de redes sociais (e.g., [2, 3]), conteúdo de notícias (e.g., [4]), entre outros. Os trabalhos existentes para a detecção e classificação automática de intenções em textos livres apresentam desde soluções simples que visam distinguir aspectos diretos (e.g., se é uma pergunta ou uma afirmação) a soluções mais complexas utilizando aprendizado de máquina para classificações multi-classes baseadas em teorias da linguagem [3]. Nesse contexto, existem numerosos estudos na literatura cujo aspecto principal está na análise de sentimentos [5] (um tópico relacionado), sendo menos numerosos e menos conclusivos os estudos sobre intenções.

O objetivo deste trabalho consiste em propor, aplicar e analisar experimentalmente resultados de técnicas de aprendi-

zado de máquina para classificar automaticamente intenções em textos em linguagem natural de agricultura e sustentabilidade escritos na Língua Portuguesa. Este artigo contribui no estudo e avaliação de um método inovador de reconhecimento de intenção em texto livre, com base nas teorias dos Atos de Fala (SAT - *Speech Act Theory*) [6] e na Semiótica [7, 8]. Aplicamos nossa técnica de reconhecimento em dados reais e cenários de RI na agricultura sustentável e em Língua Portuguesa. Nossos resultados contribuem, entre outros aspectos, na melhora do processo automatizado de anotação de texto, que pode ser utilizado em mecanismos avançados de RI.

De acordo com a SAT [6], um ato ilocucionário/ilocutório carrega a intenção de quem pronunciou um enunciado (ato locutório); esses atos ilocucionários podem ser distinguidos de acordo com os *frameworks* de classificação. No *framework* adotado neste trabalho [9, 7], fundamentado pelas teorias da SAT e da Semiótica, uma intenção presente em uma frase pode ser classificada em uma das seguintes oito classes de atos ilocucionários (intenções): *Proposal*, *Inducement*, *Forecast*, *Wish*, *Assertion*, *Valuation*, *Palinode* e *Contribution*. De um modo geral, as classes de intenção são caracterizadas da seguinte forma: *Proposal* - quando solicitamos ou prometemos; *Inducement* - quando ameaçamos, advertimos ou tentamos que outra pessoa realize determinada ação; *Forecast* - quando predizemos ou assumimos algo; *Wish* - quando desejamos ou temos esperança; *Assertion* - quando afirmamos, denunciamos ou notificamos; *Valuation* - quando julgamos, reclamamos ou acusamos; *Palinode* - quando retraímos, anulamos ou revogamos; *Contribution* - quando lamentamos ou pedimos desculpas. Essas classes podem, por exemplo, distinguir entre frases que fazem previsões sobre algo, daquelas que expressam julgamentos ou desejos. Tal distinção pode ser explorada para aprimorar a recuperação e o compartilhamento de informações.

Para associar automaticamente uma frase a uma classe de intenção específica, investigamos quatro técnicas de aprendizado de máquina: *Support Vector Machine* (SVM) [10], *Artificial Neural Network* (ANN) [10], *Random Forest* [11] e *Transfer Learning* (*CNN - Convolutional Neural Network*) [7]. Em nossa metodologia, visando uma melhor efetividade, utilizamos *Word Embeddings* pré-treinadas para efetuar a conversão dos elementos textuais que compõem uma frase para uma representação numérica. As *Word Embeddings* utilizadas foram geradas por Hartmann *et. al* [12], utilizando os seguintes algoritmos: *Word2Vec* [13], *Wang2Vec* [14], *FastText* [15] e *Glove* [16]. Avaliamos experimentalmente nosso estudo utilizando uma base de dados com 700 frases extraídas de artigos científicos relacionados à agricultura sustentável. Resultados obtidos revelam a efetividade do reconhecimento de diversas classes de ilocuições nas sentenças processadas em Língua Portuguesa.

As principais contribuições desenvolvidas deste artigo incluem:

1. Arquitetura computacional para processamento e estudo de classificação de intenções em textos livres de

- agricultura e sustentabilidade escritos em português;
2. Implementação de uma ferramenta de software desenvolvida para reconhecimento de intenções segundo classes de ilocuções com uso de modelos de aprendizado de máquina;
 3. Concepção de uma base de dados anotada por três pessoas possibilitando a extensão do trabalho para o uso de outras técnicas (ex: *Recurrent Neural Networks*) e expansão da base de dados;
 4. Resultados de um estudo comparativo abrangente envolvendo diferentes combinações e configurações de técnicas de *Word Embedding* (Word2Vec, FastText, Wang2Vec e Glove) com métodos de aprendizado de máquina (SVM, ANN, *Random Forest* e *Transfer Learning* (CNN)) na tarefa investigada.

O restante deste artigo está organizado da seguinte maneira: a Seção 2 apresenta uma revisão teórica sobre classes de intenções, bem como trabalhos correlatos deste estudo. A Seção 3 apresenta a metodologia conduzida descrevendo detalhes dos métodos e técnicas empregadas. A Seção 4 apresenta os resultados experimentais obtidos assim como desenvolve as discussões dos resultados. Por fim, a Seção 5 efetua a conclusão do trabalho e apresenta sugestões de trabalhos futuros.

2. Fundamentação e Trabalhos Relacionados

Nesta seção primeiramente apresentamos a fundamentação teórica sobre classes de intenções (Subseção 2.1). Destacamos diferentes *frameworks* teóricos para a classificação de intenções. A Subseção 2.2 apresenta trabalhos relacionados ao uso de aprendizado de máquina para classificação automática de intenções incluindo aplicações computacionais de dados rotulados com as intenções. Em particular, destacamos estudos que utilizam classificação de intenções em RI, e trabalhos relacionados com abordagens para a classificação automática de intenções. Nesse estágio, enfatizamos os diferenciais do nosso estudo em relação à literatura existente.

2.1 Classes de Intenções

Este trabalho adota o *framework* proposto por Liu [9, 7] para a classificação de intenções. Este *framework* é baseado na SAT e na Semiótica. A Semiótica é entendida como a ciência que estuda os signos, seus significados e processos de criação de sentido [17]. Na teoria Peirciana [17] signos são entendidos como qualquer coisa que significa algo para alguém, por exemplo, figuras, palavras, termos, gestos, som, entre outros.

Ambos, SAT e Semiótica possuem métodos úteis para identificar e analisar aspectos relacionados às intenções em atos de comunicação. Austin [18] propôs a SAT. Nela, a linguagem tem uma dimensão de ação e um enunciado em uma função performativa. Essa teoria possibilita construir *frameworks* para analisar as intenções das pessoas ao se expressarem [19].

De acordo com Austin [18], um ato da fala pode ser analisado em três níveis, sendo: locucionário, ilocucionário e perlocucionário.

Um ato locucionário se refere a realização de um enunciado real (e seu significado ostensivo). Já um ato ilocucionário é a tentativa de comunicar o sentido ao expressar uma atitude (*i.e.*, conteúdos proposicionais com intenções). Uma ilocução carrega as intenções do falante de acordo com uma “força ilocucionária” (o que é pretendido no ato). Por exemplo, os enunciados “... as chuvas vão aumentar nos próximos anos.” e “... gostaria que as chuvas aumentassem nos próximos anos.” dizem respeito ao aumento de chuva nos próximos anos. Contudo, a força ilocucionária nessas sentenças são diferentes, uma vez que intenções distintas são definidas nos atos. Enquanto o primeiro ato expressa uma previsão, o segundo expressa um desejo. Já o nível perlocucionário refere-se ao efeito produzido no endereçado. Neste trabalho, o foco está nos níveis locucionário e ilocucionário. Ou seja, o estudo de técnicas e algoritmos para se classificar automaticamente ilocuções (que carrega intenções) em frases livres descritas em linguagem natural.

Searle [20] propôs uma classificação de atos da fala com base na força ilocucionária, que inclui as seguintes classes: (1) *Assertives* (para expressar a posição do falante perante algo), (2) *Directives* (para fazer o destinatário executar uma ação); (3) *Commissives* (comprometer-se a fazer algo no futuro); (4) *Expressives* (para expressar as emoções e atitudes do falante) e (5) *Declarations* (para mudar um estado do mundo).

Outros estudos propõem alternativas para classificar atos da fala, por exemplo as categorias propostas por D’Andrade and Wish [21], a classificação de Olney *et al.* [22], a taxonomia de modos de respostas de Stiles [23] e a hierarquia de diálogo proposta por Alexandersson *et al.* [24]. Hawizy, Phillips, and Connolly [25] destacaram que estudos da comunicação, tal como Semiótica, podem ser incorporados em modelos de intenções humanas.

Do ponto de vista da Semiótica, as pessoas se comunicam usando múltiplos meios de comunicação para compartilhar signos. Assim, nesta perspectiva, é importante entender como os signos influenciam o processo de comunicação. A Semiótica Organizacional (SO) estuda uma organização como um sistema de signos [8], incluindo como eles influenciaram o processo de comunicação.

A pragmática, que pode ser entendida como uma subárea da Semiótica que foca em aspectos sobre o uso dos signos e seus efeitos em um contexto social, tais como intenções, comunicações, conversações e negociações. Liu [9, 7] propôs um *framework* para o entendimento da pragmática da comunicação baseado na SO e na SAT. De acordo com Liu [9, 7], uma comunicação é bem sucedida quando um signo significativo é utilizado com uma intenção entendida apropriadamente pelo falante e ouvinte.

Enquanto a SAT adota uma abordagem linguística para a comunicação de intenções, a SO foca na comunicação mediada por signos. De acordo com De Moor [26], SO pode

ser combinada com a SAT para garantir um processo de comunicação significativo (utilizando definições ontológicas) e aceitável (utilizando definições normativas).

No *framework* definido por Liu [9, 7], um ato de comunicação é representado por uma tripla composta por um executor (ou falante), um destinatário (ou ouvinte) e uma mensagem. A mensagem por sua vez é organizada em duas partes: (1) o conteúdo, que especifica o significado da mensagem que é dependente do executor e endereçado; e (2) a função, que corresponde a intenção do falante. Uma classe de ilocução representa o tipo de ilocução que, por sua vez, reflete a intenção do falante.

O *framework* de Liu estrutura as classes utilizando três dimensões. A primeira dimensão se refere ao “tempo”, *i.e.*, se o efeito do ato de comunicação é no futuro ou presente/passado. As classes *Forecast*, *Proposal*, *Inducement* e *Wish* tem efeito no futuro, enquanto *Assertion*, *Valuation*, *Palinode* e *Contribution* efeito no presente ou passado.

A segunda dimensão se refere a “invenção”, que distingue entre atos com conteúdos descritivos daqueles com conteúdos prescritivos. Isto é, diferencia um ato que prescreve algo (para a própria pessoa ou outros) de atos que descrevem algo. Se a ilocução em um ato de comunicação tiver um efeito inventivo ou instrutivo, será denominada prescritiva, caso contrário, ela é descritiva. As classes *Proposal*, *Inducement*, *Palinode* e *Contribution* caracterizam atos prescritivos, enquanto *Assertion*, *Valuation*, *Forecast* e *Wish* são atos descritivos.

A última dimensão é o “modo” (se refere ao estado de humor pessoal), que distingue entre atos denotativos e afetivos. Essa dimensão diferencia atos de comunicação que se referem a sentimentos e julgamentos (afetivos) dos atos denotativos (impessoais). As classes *Assertion*, *Proposal*, *Palinode* e *Forecast* são denotativas, enquanto *Valuation*, *Contribution*, *Wish* e *Inducement* são afetivas.

Cada classe de ilocução é determinada de acordo com essas três dimensões, por exemplo: a frase “A temperatura da terra vai aumentar nos próximos anos.” é uma *Forecast* (*i.e.*, o falante teve a intenção de prever algo), e se refere a algo no futuro (dimensão tempo), é descritiva (dimensão invenção) e é denotativa (dimensão modo). Já a frase “Eu gosto muito de chocolate.” é uma *Valuation* (*i.e.*, o falante teve a intenção de julgar algo), e se refere a algo no presente (dimensão tempo), é descritiva (dimensão invenção) e é afetiva (dimensão modo).

No *framework* de Liu, essas classes se referem a função da mensagem, enquanto o conteúdo é dado pelo significado. Por exemplo, “aumento da temperatura” na primeira frase supracitada e “gosto por chocolate” na segunda frase se referem ao significado. Esta ligação possibilita explorar a classificação de ilocuições em várias situações, tais como cenários de RI. Por exemplo, pode-se distinguir entre textos que fazem previsão sobre o “aumento da temperatura” (classe *Forecast*) de textos que relatam o “aumento da temperatura” (classe *Assertion*) e de textos que fazem algum julgamento sobre o “aumento da temperatura” (classe *Valuation*).

Trabalhos anteriores focaram a descrição de modelos para

representar a ligação entre função e conteúdo [27]. O *framework* foi explorado em sistemas de RI médicas [1], na detecção de intenções criminosas em redes sociais [3], bem como mecanismos para comunicar intenções em interfaces interativas [28, 29]. O nosso presente estudo se diferencia dos demais ao apresentar uma investigação profunda sobre o uso de técnicas de aprendizado de máquina na classificação automatizada de intenções em textos na Língua Portuguesa.

2.2 Classificação Automática de Intenções e suas Aplicações

A literatura apresenta diversos estudos que exploram intenções em RI em vários contextos e domínios. Hwang, Kim e Choi [30], por exemplo, propuseram um modelo para RI no qual as intenções dos usuários são detectadas com o uso de sensores em um ambiente de computação pervasiva. A busca então é realizada com base nas intenções detectadas. Já Gupta, Garg e Gupta [31] apresentaram um modelo de busca utilizando agentes inteligentes e filtros com o objetivo de capturar as intenções dos usuários por meio de resultados de busca no Google[®]. Na abordagem deles, os usuários assinalam *checkboxes* associados a resultados do Google[®], que são utilizados em novas pesquisas.

Alguns pesquisadores consideram a diminuição do esforço do usuário uma questão central na RI com base em intenções; Mendoza e Baeza-Yates [32], por exemplo, propuseram um método automático para categorizar o comportamento dos usuários em buscas. O algoritmo tem como objetivo compreender as intenções dos usuários pela análise das palavras-chaves digitadas. Noor e Martinez [33] exploraram dados de rede sociais por meio de um modelo de interesse que serve como uma interpretação indireta da intenção dos usuários. Zinglé [34] desenvolveram um mecanismo de busca baseado em grafos conceituais para representar o domínio. A concatenação de expressões léxico-sintáticas, descrições semânticas e representação de aspectos pragmáticos permite aos usuários expressarem suas intenções de maneira explícita nos parâmetros de busca.

Embora existam diversas abordagens que levam em consideração a intenção de quem realiza a busca, poucos estudos enfatizam e consideram as intenções de quem produz o conteúdo na indexação e recuperação. Por exemplo, Asai e Yamana [35] apresentaram um *framework* que detecta intenções por meio de anotações em documentos, para então destacar partes do conteúdo (para auxiliar a leitura), extrair termos e prover comentários. Enquanto Montes-y-Gómez, Gelbukh e López [36] definiram um mecanismo de RI que tem o objetivo de determinar a intenção pela análise de títulos e sumários em uma abordagem independente de domínio. Este mecanismo é baseado em casamento de padrões e na identificação de um conjunto de construções e verbos pré-definidos. Mustakim *et al.* [37] apresentaram um mapeamento sistemático sobre o uso de intenções em RI. Este mapeamento aponta que há necessidade de pesquisas adicionais para detectar intenções. Tal aspecto é abordado por trabalhos relacionados a detecção

e classificação automática de intenções em textos livres.

A detecção e classificação automática de intenções em textos livres (foco deste artigo) continua sendo um problema difícil. A literatura apresenta vários estudos e avanços em temas relacionados, tais como análise de opinião e de sentimentos (a referência [5] apresenta um *survey* sobre este tema). Já os trabalhos sobre detecção de intenções são menos frequentes, embora também apresentem avanços recentes. Os trabalhos em detecção e classificação de intenções utilizam diferentes abordagens de áreas relacionadas tais como aprendizado de máquina, processamento de linguagem natural e linguística. Algumas investigações iniciais (e.g., [19]) exploram modelos linguísticos, tais como modelos de discurso como uma abordagem para analisar conversações e intenções do usuário.

De acordo com Teh, Cheng e Chee [38], a detecção de discursos (mais especificamente discurso de ódio) baseada exclusivamente em uma abordagem léxica é falha. Já Agarwal e Sureka [39] destacaram que o uso de técnicas limitadas a dicionários léxicos não são suficientes para construir mecanismos de detecção automática de intenções, necessitando, portanto, de investigar por alternativas, tais como a adoção de descritores semânticos e o uso de técnicas de aprendizado de máquina.

O uso de aprendizado de máquina é uma abordagem de pesquisa promissora para implementar soluções para reconhecimento de intenções dos usuários. O uso de classificadores como *Random Forest*, *Naive Bayes*, *SVM* e *ANN* é uma das alternativas estudadas na literatura [40]. Cohen, Carvalho e Mitchell [41] apresentaram um estudo sobre classificação de atos da fala de acordo com a intenção de quem escreveu emails. Os atos são classificados em sete classes: *Request*, *Proposal*, *Delivery*, *Commitment*, *Directive*, *Commissive* e *Meet*. Os resultados apontaram o uso de árvores de decisão e *SVM* como promissores, mas com F1-Score entre 0.44 e 0.85 ao considerar individualmente as classes na validação cruzada com $K=5$.

Lambert *et al.* [42] apresentaram resultados promissores com o uso de *SVM* e *Bayes Networks* ao apresentar uma acurácia de 79.8% em um corpus de 1368 enunciados anotados. O objetivo dos autores foi anotar automaticamente atos da fala de acordo com a taxonomia VRM (*Verbal Response Modes*). A VRM classifica atos da fala em: *Disclosure*, *Edification*, *Advisement*, *Confirmation*, *Question*, *Acknowledgement*, *Interpretation* e *Reflection*.

O'Shea, Bandar e Crockett [43] exploraram o uso de palavras funcionais da língua inglesa (e.g., preposições, conjunções, artigos, verbos auxiliares, adjetivos possessivos e adjetivos demonstrativos) no treinamento de técnicas de aprendizado de máquina para diferenciar enunciados que são questões, daqueles que não são questões. Os autores concluíram que as palavras funcionais são suficientes para distribuir questões de não questões, resultando na acurácia de 0.89 com o uso de árvores de decisão na validação cruzada. Kim, Seon e Seo [44] apresentaram uma revisão sobre o uso de técnicas de

aprendizado de máquina na detecção de intenções em língua coreana. Uma análise de um corpus sobre gestão de agendas com 800 diálogos para treinamento e 100 diálogos para testes (não foi usada validação cruzada) apresentou o melhor resultado com a técnica de *SVM* com 0.88 de F1-score.

Moldovan *et al.* [45] investigaram o uso de *Naive Bayes* e árvores de decisão (J48) para classificar atos da fala de um corpus de chats do *Linguistic Data Consortium*¹. Os autores enfatizaram que as primeiras palavras de um enunciado podem ser utilizadas para prever uma categoria de atos da fala com acurácia razoável em postagens de chats online. Os resultados apresentaram F1-score ao redor de 70% para várias classificações de atos da fala. Borhan *et al.* [46] exploraram o papel do contexto na classificação de atos da fala em sistemas tutores inteligentes utilizando *Naive Bayes* e árvores de decisão (J48). Foram incluídas 10 características referentes ao contexto na classificação. Segundo os autores, o contexto teve um papel modesto na previsão automática dos atos de fala. Os resultados gerais obtidos na classificação de sete classes de ilocuções (*ExpressiveEvaluation*, *Greeting*, *MetaStatements*, *Statement*, *Question*, *Reaction* e *Request*) foram de 0.56 de acurácia utilizando árvores de decisão e 0.55 com *Naive Bayes*, ambos na validação cruzada. O ganho com a inclusão das 10 características referentes ao contexto foi inferior a 0.01.

O estudo apresentado por Hu e Wang [47] utilizou SAT [18] como referencial teórico para construção de um sistema de pré-alarme de crimes com base em conteúdo da internet. Ele detalha a construção de um modelo matemático para utilização em conjunto com a técnica de *Naive Bayes*. Um comparativo alterando o algoritmo para árvores de decisão evidencia a superioridade da técnica de *Naive Bayes*.

Estudos recentes fazem uso de técnicas como *CNN* (*Convolutional Neural Network*) e *RNN* (*Recurrent Neural Network*) para classificação de atos da fala. Por exemplo, Koo, Kim e Hong [48] apresentaram o uso de *SVM* e *CNN* na classificação automática de textos em coreano, ao analisar diálogos entre um professor e um estudante; e uma chamada telefônica entre um estudante de graduação e um de pós-graduação. A classificação foi realizada em dois níveis de intenções, atos da fala representativos (primeiro nível) e atos da fala concretos (segundo nível). Foram obtidos resultados preliminares com medidas próximas para *SVM* e *CNN*, com acurácia por volta de 0.83 no primeiro nível e 0.84 no segundo nível, ambos os casos utilizando validação cruzada.

Vários trabalhos têm desenvolvido técnicas de reconhecimento de intenções em textos para o contexto de chatbots. Por exemplo, Nigam *et al.* [49] desenvolveram uma metodologia para detecção de intenção e entidades nomeadas em um chatbot que responde consultas relacionadas à carreira. Foi adotada uma abordagem em vários estágios, em que os processos de classificação de intenção e de entidades nomeadas informam a tomada de decisão. O modelo divide o problema em estágios, resolvendo um problema de cada vez e passando

¹<https://www.ldc.upenn.edu/> - Acessado em Março de 2020.

os resultados relevantes do estágio atual para o próximo.

Nessa linha, Wang *et al.* [50] apresentaram um modelo de rede neural codificador-decodificador baseado em atenção para a detecção conjunta de intenção e entidades. Essa proposta codifica a representação de sentenças com CNN híbridas e redes de memória de longo prazo bidirecional (CNN-BLSTM). A proposta efetua a decodificação com base em uma RNN baseada em mecanismo de atenção e com entradas alinhadas.

Em literatura recente, Chen *et al.* [51] propuseram um modelo de classificação de intenção conjunta a reconhecimento de entidades com base no BERT (*Bidirectional Encoder Representations for Transformers*) [52], com o objetivo de abordar a capacidade limitada de generalização dos modelos tradicionais de NLU. BERT é um modelo de aprendizado profundo que tem fornecido resultados promissores em uma ampla variedade de tarefas de processamento de linguagem natural [52].

Em Pandey *et al.* [53], os autores apontaram que a categorização de intenções maliciosas ligadas a agressões sexuais pode ser realizada com base em técnicas de aprendizado de máquina apoiadas por recursos semânticos. Três categorias de intenções foram formuladas pelos autores: acusação, confirmação e sensacionalismo. A classificação foi feita usando recursos de semântica distribucional e redes neurais. Os autores destacaram o uso de modelos lineares de regressão logística e redes CNN.

Kim e Kim [54] propuseram uma arquitetura de CNN integrada para classificação simultânea de atos da fala, predicadores² e sentimento. De acordo com os autores, esta classificação conjunta pode melhorar a classificação dos atos da fala. Já Algotiml e Elmadany [55] apresentaram um estudo sobre a classificação de atos da fala em mensagens de tweets escritos em árabe. Esse trabalho comparou resultados de uso de SVM com métodos de aprendizado profundo. O trabalho utilizou a técnica *word2vec_skip-gram* (cf. Seção 3.5) CNN, LSTM-RNN (*Long Short Term Memory - RNN*), BiLSTM, CNN sobre LSTM, CNN sobre BiLSTM, LSTM sobre CNN e BiLSTM sobre CNN. Os melhores resultados foram obtidos com o uso da técnica de BiLSTM para a classificação em cinco categorias de atos ilocucionários, *Assertion* (0.90 - F1-Score), *Expression* (0.87 - F1-Score), *Question* (0.57 - F1-Score), *Request* (0.20 - F1-Score) e outras/diversas (0.53 - F1-Score); e 0.615 de F1-Score considerando as cinco classes na validação cruzada (k=5).

Trabalhos recentes destacam a dificuldades na classificação de intenções em textos livres. De acordo com Liu e Xu [56], essas dificuldades são ainda mais agravadas quando se lidar com textos curtos, como a classificação na granularidade de frases que é abordada em nosso artigo. Entre as dificuldades apontadas por Liu e Xu estão no fato das mensagens curtas geralmente conterem características esparsas (o texto curto contém apenas uma dúzia de palavras com significado prático); mensagens curtas são geradas igualmente em grande

volume e não seguem um formato padrão. Em Liu [56] foram obtidos melhores resultados ao utilizar uma abordagem (proposta pelos autores) que explora múltiplas técnicas de aprendizado profundo. Nas três bases de dados utilizadas foram atingidas números de acurácia variando de 46.3% a 93.4%. Os resultados são promissores, entretanto, o artigo não apresenta números de F1-Score em validação cruzada. As bases possuem um número limitado de classes (de 2 a 6). O trabalho não explora categorias de atos ilocucionários.

Em nossa pesquisa, além de lidarmos com a classificação em uma granularidade pequena, temos 8 categorias fundamentadas na SAT e Semiótica. Algumas dessas categorias tem diferenças sutis na escrita, mas que podem trazer interpretações e consequências muito distintas para o ouvinte. É muito difícil determinar individualmente se uma frase como “O Brasil poderia investir mais em energias renováveis.” é uma proposta, um desejo ou uma indução. Em uma conversa face a face teríamos elementos adicionais tais como tonalidade da voz e expressões corporais. Assim, como o efeito no ouvinte pode ser consideravelmente diferente, o que leva a literatura destacar a importância de classificar os atos da fala. Em um estudo recente, por exemplo, Argyris *et al.* [57] destacaram o feito dos atos de fala nas emoções em mídias sociais.

Para lidar com o difícil problema de classificar atos ilocucionários, exploramos várias técnicas de aprendizado de máquina e aprendizado profundo. Tais técnicas são dominantes nos estudos sobre classificação de textos, tais como análise de sentimento [58], bem como estão presentes nos trabalhos supracitados.

Embora os trabalhos relacionados (discutidos nesta subseção) apresentem avanços em técnicas para detecção e classificação automática de intenções em textos livres, bem como seu uso em RI, nossa investigação se diferencia e contribui para o avanço da pesquisa no tema por abordar aspectos como: (1) análise de um conjunto considerável de combinações de técnicas e parâmetros para mapeamento de palavras em vetores (*word embedding*) com várias combinações de parâmetros em múltiplos classificadores (SVM, ANN, *Random Forest* e *Transfer Learning/CNN*); (2) aplicado a um corpus na Língua Portuguesa no domínio de agricultura e sustentabilidade; (3) comparação de resultados da classificação de base anotada por três pessoas, com resultados individualizados; e (4) utilização do *framework* em oito dimensões com foco no uso posterior em mecanismos de RI.

3. Metodologia

Esta seção descreve a metodologia desenvolvida neste estudo. A Figura 1 apresenta uma visão geral dessa metodologia, destacando o fluxo de dados desde as suas fontes (canto superior esquerdo) até a aplicação dos classificadores (canto inferior direito). Nosso estudo é organizado em cinco passos (ver Figura 1) descritos a seguir:

- **Passo 1 - Elaboração da Base de Dados:** Este passo consiste em criar e preparar uma base de dados para

²verbo/ação principal de um sentença.

treinamento e teste com frases relacionadas à agricultura sustentável, extraídas de diferentes fontes [59, 60, 61, 62, 63]. Subseção 3.1 apresenta detalhes sobre a criação do conjunto de dados explorado.

- **Passo 2 - Análise Morfológica:** Neste passo é realizada a análise morfológica das palavras que compõem cada frase e a identificação das classes gramaticais: substantivos, adjetivos, verbos, advérbios, conjunções e preposições. O resultado é uma estrutura textual composta apenas com verbos, adjetivos, advérbios e conjunções. Palavras pertencentes a outras classes gramaticais, não listadas aqui, foram descartadas (cf. Subseção 3.2).
- **Passo 3 - Word Embedding:** por meio da aplicação de técnicas de *Word Embeddings*, cada estrutura textual foi convertida para um vetor numérico, denominado de *Word Embedding*. Quatro técnicas foram aplicadas e avaliadas: Word2Vec, FastText, Wang2Vec e Glove (cf. Subseção 3.3).
- **Passo 4 - Balanceamento de Classes:** Neste passo, efetuamos o balanceamento de classes de intenção no conjunto de treinamento utilizando a técnica *Synthetic Minority Over-sampling Technique (SMOTE)* (cf. Subseção 3.4), uma técnica comumente utilizada por diversos trabalhos que necessitam realizar balanceamento de classes;
- **Passo 5 - Treinamento e Testes:** Treinamento e teste do algoritmo com os classificadores SVM, ANN, *Random Forest* e *Transfer Learning*. Subseção 3.5 apresenta detalhes dos classificadores investigados na tarefa. A seção 4 descreve os procedimentos e resultados experimentais das análises conduzidas considerando diferentes configurações experimentais.

3.1 Elaboração da Base de dados

A base de dados desenvolvida é composta por 700 frases extraídas de artigos indexados e relacionados à agricultura, sendo que, em sua maior parte, as frases foram extraídas do Boletim Técnico-Informativo do Instituto Agrônomo [59]. Três pessoas, neste artigo nomeadas de A, B e C, classificaram, de maneira individual e manual, cada uma das frases nas oito classes de intenção [9, 7]: *Assertion, Valuation, Proposal, Inducement, Forecast, Wish, Palinode* ou *Contrition*. As pessoas A e B possuem experiência prévia de classificação com a teoria do cubo (Seção 2) e análise de textos, ao passo que para a pessoa C esta teoria foi apresentada pela primeira vez.

A Tabela 1 apresenta exemplos de frases que compõem a base de dados e suas respectivas classificações de acordo com cada pessoa. A base de dados está disponível para consulta mediante solicitação. Cada frase foi associada a exclusivamente apenas uma classe de intenção.

Para classificar cada frase em sua respectiva classe de intenção, duas pessoas (A e B) utilizaram sistematicamente o

modelo de referência descrito na Seção 2, ao passo que uma pessoa (C), de maneira proposital, não o seguiu estritamente (*i.e.*, atribuiu classes de acordo com seus nomes e seu entendimento próprio sobre as mesmas). A Tabela 2 apresenta o resultado da classificação das frases nas classes de intenções de acordo com cada pessoa.

Calculamos o percentual de concordância³ entre as três pessoas, e os resultados são apresentados na Tabela 3. As avaliações foram executadas em paralelo sem comunicação entre os avaliadores, também não houve remoção de frases para manter uma taxa mínima de concordância.

3.2 Análise morfológica de palavras

Esta etapa tem como objetivo dividir cada frase em palavras individuais e determinar a classe gramatical de cada palavra, com ambos procedimentos efetuados de forma automática. Na literatura, o processo de dividir uma frase em palavras individuais é normalmente conhecido como *tokenization*, sendo cada palavra denominada de *token*.

Foram estabelecidos dois critérios independentes para definir um *token*. Considera-se um *token* como o conjunto de caracteres, o qual: (1) está entre dois espaços vazios, ou (2) está entre um espaço vazio e um carácter de pontuação de início ou fim de linha.

Uma vez determinado cada *token* da frase, é definida sua classe gramatical. Para atingir este objetivo, foi utilizado um Hunspell⁴: um corretor ortográfico e analisador morfológico. Especificamente, foi utilizado o Hunspell disponível em Python⁵. O Hunspell escolhido precisa ser configurado com dois arquivos: um conteúdo afixos (*.aff*) e outro arquivo contendo um dicionário (*.dic*). O arquivo de dicionário contém palavras da Língua Portuguesa classificados com suas respectivas classes gramaticais. Quanto maior o número de palavras disponíveis no arquivo de dicionário, mais robusto ele é considerado. Foram utilizados os arquivos de afixo e de dicionário disponíveis no OnlyOffice⁶.

Após o processo de *tokenization* e de verificação da classe gramatical de cada palavra, mantivemos na frase apenas as palavras pertencentes às classes gramaticais verbo, adjetivo e advérbio. Palavras de outras classes gramaticais definidas pelo Hunspell foram descartadas da frase.

3.3 Mapeamento de palavras em vetores numéricos

Em Processamento de Linguagem Natural, frequentemente trabalha-se com representação textual de palavras. No entanto, diversos algoritmos de aprendizado de máquina não consideram representação textual, mas sim representações numéricas, como é o caso dos classificadores SVM e ANN. Por esse

³Numero de frases em que os revisores atribuíram a mesma classe de intenção dividido pelo número total de frases

⁴Disponível em: <<https://github.com/hunspell/hunspell>>, Último acesso em Janeiro de 2020.

⁵Disponível em: <<https://pypi.org/project/hunspell/>>, Último acesso em Janeiro de 2020.

⁶Disponível em: <https://github.com/ONLYOFFICE/dictionaries/tree/master/pt_PT>, Último acesso em Janeiro de 2020.

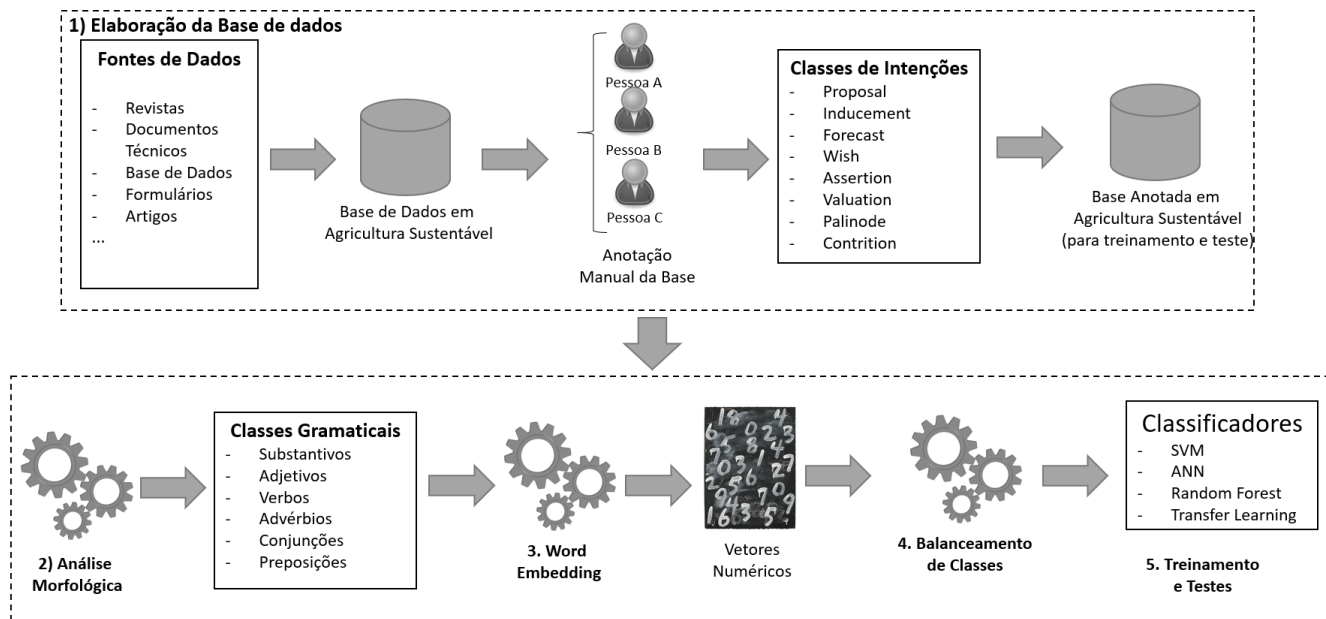


Figura 1. Visão Geral da Metodologia Empregada.

Tabela 1. Exemplos de frases que compõem a base de dados desenvolvida e suas respectivas classificações nas classes de intenções de acordo com cada avaliador.

Frases	Classificação		
	Pessoa A	Pessoa B	Pessoa C
A bioeconomia pressupõe aumentar bem estar humano e a equidade social.	Valuation	Valuation	Assertion
O Brasil poderia investir mais em energias renováveis.	Inducement	Proposal	Wish
Os indicadores indicam que a produção agrícola diminuirá drasticamente.	Forecast	Forecast	Forecast
A cana-de-açúcar tem se consolidado como uma cadeia produtiva competitiva e em constante expansão.	Assertion	Assertion	Assertion
Eu gostaria de indicar novos indicadores de sustentabilidade.	Wish	Wish	Wish

motivo, para se trabalhar com linguagem natural e as técnicas SVM e ANN, é necessária a conversão da representação textual para vetores numéricos. Para fazer essa conversão de representações, técnicas atuais utilizam *Word Embeddings*.

Word Embedding pode ser definido como um conjunto de vetores de números reais que representam palavras em um espaço n -dimensional. Esses vetores são gerados a partir da aplicação de um algoritmo sobre uma base textual, de forma que os vetores numéricos sejam capazes de representar os aspectos morfológico, sintático e semântico desse conjunto de dados textuais [12].

Cada *Word Embedding* pré-treinada representa uma palavra por um conjunto composto por d números reais. No presente trabalho, tem-se $d \in \{50, 100, 300, 600\}$ ⁷. Para produzir um vetor de características, cada palavra de uma frase é convertida para o conjunto numérico de d valores reais de determinada *Word Embedding*.

Tipicamente, as técnicas de aprendizado de máquina re-

querem um número fixo de atributos para todas as amostras. Em nosso contexto, as frases da base de dados em nosso estudo possuem um número variável de total de palavras. Portanto, definimos a dimensão do vetor de características de acordo com a frase com o maior número de palavras multiplicado por d , cujos resultados são apresentados na Tabela 4. Frases menores tiveram seus vetores de características completados com o valor zero. Caso uma palavra da frase não esteja disponível na *Word Embedding* pré-treinada, essa palavra foi convertida para o conjunto numérico da letra e ⁸.

Diferentes algoritmos têm sido propostos para gerar *Word Embeddings*. Desses, foram escolhidos: Global Vectors (Glove), Word2Vec, Wang2Vec e FastText. A escolha desses algoritmos foi motivada por dois aspectos: (1) por estarem entre os mais recentemente utilizados para gerar *Word Embeddings*; e (2) pela existência de um repositório de armazenamento e compartilhamento de vetores de palavras gerados para a Língua Portuguesa [12]. Esse repositório é composto por 35

⁷Não foi utilizado o tamanho $d = 1000$ pois, assim como os resultados apresentados em [12], em nossos testes preliminares esta dimensão apresentou resultados piores, além de demandar mais recursos computacionais.

⁸Escolhemos a letra e porque a conjunção e é removida da frase na etapa de análise morfológica, portanto, é uma letra que não deverá criar um viés na classificação da frase nas classes de intenções.

Tabela 2. Classificação das frases da base de dados de acordo com cada pessoa.

	Classes de intenção							
	Proposal	Inducement	Forecast	Wish	Assertion	Valuation	Palinode	Contribution
Pessoa A	33	76	108	22	232	202	14	13
Pessoa B	14	78	124	25	197	231	16	15
Pessoa C	61	92	63	56	285	109	20	14

Tabela 3. Taxa de concordância em relação às classificações das frases em classes de intenções.

Pessoas	Percentual de concordância (%)
A/B	77,00
A/C	46,43
B/C	42,86
A/B/C	38,42

Tabela 4. Número total de atributos do vetor de características de acordo com a dimensão da *Word Embedding*.

Dimensão da <i>Word Embedding</i>	Dimensão do vetor de características
50	1250
100	2500
300	7500
600	15000

vetores de palavras, com dimensões de 50, 100, 300, 600 e 1000. Para gerar esses vetores foram utilizados dezessete conjuntos de dados diferentes, totalizando 1.395.926.282 tokens⁹. Os algoritmos são descritos a seguir.

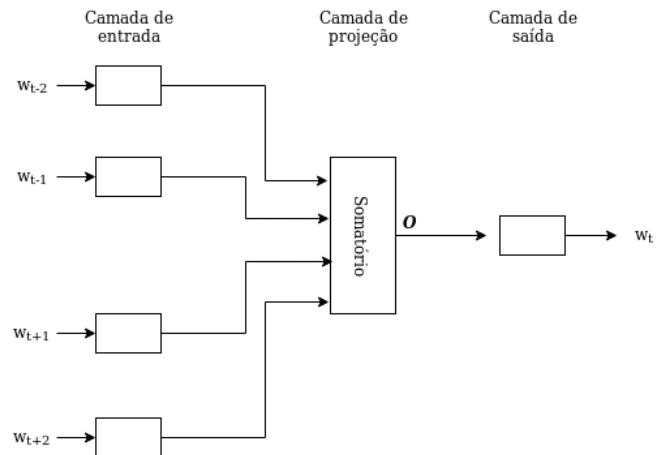
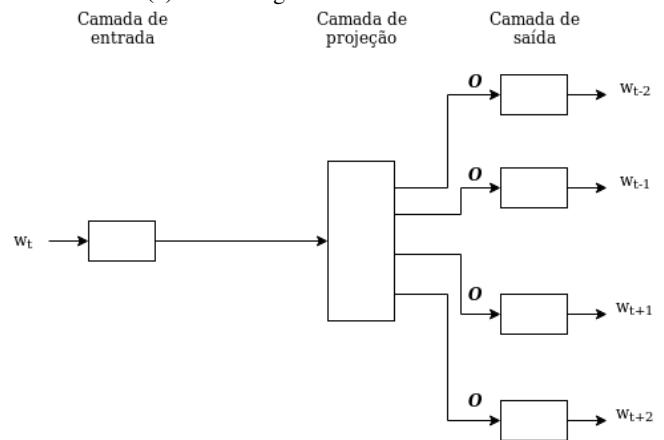
3.3.1 Word2Vec

Word2Vec é um método para gerar *word embedding* tendo como base uma rede neural composta por três camadas: entrada, intermediária (ou projeção) e saída. O Word2Vec pode utilizar um de dois modelos de arquitetura para produzir uma representação numérica de palavras: *Continuous Bag-of-Words* (CBOW) e *Skip-Gram*.

No modelo CBOW, o objetivo é prever uma palavra a partir de um grupo de palavras vizinhas, conforme ilustrado na Figura 2a, em que a palavra a ser predita é representada por w_t (camada de saída) e o grupo de palavras vizinhas é representado por w_{t-i} . No exemplo da Figura 2a, tem-se $-2 \leq i \leq 2$. Entretanto, outros valores podem ser testados para se obter melhores resultados.

No modelo Skip-Gram o objetivo é, a partir de uma determinada palavra, prever o grupo de palavras vizinhas, conforme ilustrado na Figura 2b. Para ambos os casos, CBOW e Skip-Gram, o Word2Vec consiste em ajustar os valores (pesos) de uma matriz de predição $O \in \mathbb{R}^{|V| \times d}$ (Figura 2), sendo d a dimensão do vetor numérico que representa uma palavra e V o total de palavras únicas do dicionário utilizado para

treinar a rede neural. Ao final do treinamento, cada linha da matriz O representa a *word embedding* de cada palavra única do dicionário. Uma análise formal da matemática envolvida pode ser consultada em [13, 14, 64].

**CBOW****(a) Estrutura geral do modelo CBOW.****Skip-gram****(b) Estrutura geral do modelo Skip-Gram.****Figura 2.** Visão geral da estrutura dos dois modelos disponíveis no Word2Vec. Fonte: Imagens adaptadas de [14].

3.3.2 Wang2Vec

Wang *et al.* [14] ressaltaram que o algoritmo Word2Vec considera o aspecto semântico do texto, mas não considera o aspecto sintático, ou seja, a ordem das palavras em uma frase

⁹<<http://nilc.icmc.usp.br/embeddings>> - Último acesso em Janeiro de 2020.

não tem relevância na forma em que os vetores numéricos são gerados. O algoritmo conhecido como Wang2Vec é uma variação do Word2Vec, que considera a ordem das palavras na frase, ou seja, seu aspecto sintático.

Wang *et al.* [14] propuseram duas alterações, sendo uma para o modelo CBOW e outra para o modelo Skip-Gram. Os modelos modificados são ilustrados na Figura 3, na qual: w_t é uma palavra central de um grupo de palavras w_{t-i} ; O , $O_t \in \mathbb{R}^{|V| \times d}$ são matrizes de predição, sendo d a dimensão do vetor numérico que representa uma palavra e V o total de palavras únicas do dicionário. No exemplo da Figura 3, tem-se $-2 \leq t \leq 2$, mas outros valores podem ser utilizados. Os autores não indicam qual seria a palavra central caso a frase contivesse um número pares de palavras. Resumidamente, as seguintes alterações em relação a técnica Word2Vec foram propostas:

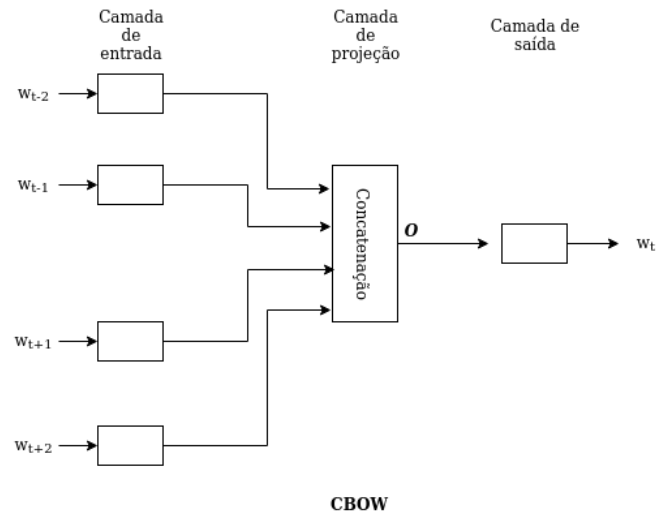
- Modelo *CBOW*: ao invés de preencher a matriz de predição O com a soma das *word embeddings* das palavras vizinhas a palavra central (Word2Vec), a matriz de predição é formada pela concatenação das *word embeddings* das palavras vizinhas na ordem em que elas ocorrem. Desse modo, a ordem torna-se importante, já que uma matriz de predição diferente será formada para a mesma palavra dependendo de sua posição, o que não ocorre no Word2Vec original, no qual, independentemente da posição da palavra, a mesma matriz de predição é gerada;
- Modelo Skip-Gram: ao invés de ser gerada uma única matriz de predição O (Figura 2b) para a palavra central w_t , são geradas diversas matrizes de predição O_t , cada uma de acordo com a posição relativa da palavra no grupo de palavras (Figura 3b, com $-2 \leq t \leq 2$).

3.3.3 FastText

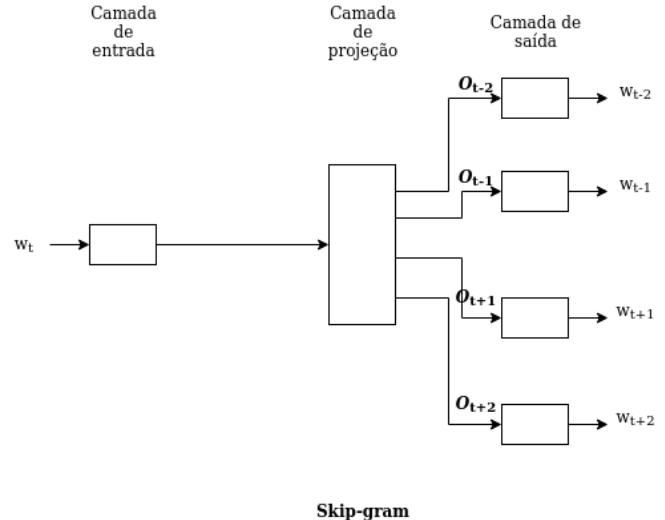
FastText é um método mais recente proposto em [15], no qual a *word embedding* é formada a partir dos *n-grams* (n-gramas) das palavras. Este método tem como base o *Word2Vec*, e tem como objetivo capturar a informação morfológica da palavra. É um dos mais indicados para idiomas que possuam uma estrutura morfológica de formação de palavras bem definida.

Formalmente, os autores citados definem a formação da *word embedding* do seguinte modo: seja um dicionário de n-gramas de tamanho G . Dada uma palavra w e $G_w \in \{1, \dots, G\}$ um conjunto com G n-gramas gerados a partir de w , associa-se um vetor z_g para cada $g \in G_w$. A representação da palavra w é a soma das representações vetoriais de seus n-gramas.

Na prática, os autores utilizam apenas n-gramas compostos de 3 a 6 letras. Além disso, a implementação ocorre utilizando uma função *hash*: cada n-grama é mapeado para um inteiro entre 1 e K , sendo $K = 2 \times 10^6$. Deste modo, a representação numérica final de uma palavra é composta pelo seu índice no dicionário de palavras e pelo conjunto numérico resultante do mapeamento dos n-gramas.



(a) Estrutura geral do modelo CBOW.



(b) Estrutura geral do modelo Skip-Gram.

Figura 3. Visão geral da estrutura dos dois modelos disponíveis no Wang2Vec. Fonte: Imagens adaptadas de [14].

3.3.4 Glove

O GloVe (*Global Vector*) é um modelo de regressão log-bilinear para aprendizado não supervisionado com a finalidade de obter representações vetoriais para palavras. O modelo captura diretamente as estatísticas globais do conjunto de frases (Corpus) [16]. Nesse sentido, as estatísticas das ocorrências de palavras em um Corpus são as principais fontes de informação disponíveis para todos os métodos não supervisionados para aprender representações vetoriais de palavras.

A vantagem do GloVe em relação aos métodos mencionados anteriormente está em demandar menor esforço computacional. Especificamente, GloVe é uma maneira de resolver a complexidade computacional de otimização da função de probabilidade de co-ocorrência de palavras em um Corpus.

3.4 Balanceamento de classes

Entre os aspectos chave que podem influenciar a efetividade de um modelo de classificação gerado por um sistema de aprendizado de máquina supervisionado está na diferença entre o número de exemplos pertencentes a cada uma das classes. Quando há uma diferença considerável de dados pertencentes a cada classe, o método de aprendizado de máquina adotado pode encontrar dificuldades em prever ou classificar dados referentes à classe minoritária, podendo apresentar uma tendência de classificação para a classe majoritária. No entanto, em muitas aplicações práticas, a desproporção no número de casos pertencentes a cada uma das classes é uma característica intrínseca do problema. No caso do nosso domínio, por exemplo, observamos a frequência muito maior de *Assertions* e *Valuations* do que *Palinodes* e *Contritions*.

Na literatura, encontram-se algumas técnicas para efetuar o balanceamento das classes, de modo que todas as classes possuam a mesma ou uma quantidade próxima de amostras. Atualmente, uma das principais técnicas é o balanceamento artificial, que divide-se em dois casos: (1) retirar exemplos de uma ou mais classes; ou (2) adicionar exemplos artificiais para uma ou mais classes. No caso da classificação binária, são retirados exemplos da classe majoritária ou adicionados exemplos à classe minoritária.

Como o problema real em foco neste trabalho possui classes desbalanceadas, é recomendável a utilização de uma técnica de balanceamento. Para tanto, adotamos a técnica SMOTE [65], sendo essa uma técnica amplamente utilizada por diversos métodos na literatura para o balanceamento das classes. No caso deste projeto, utilizou-se a implementação do SMOTE apresentada em [66] e disponível em Python¹⁰. As seguintes configurações de parâmetros foram adotadas: *sampling_strategy = 'auto'* e *k_neighbors = 5*. O restante dos parâmetros foram adotados os valores padrões da implementação. Foram testados valores para *k_neighbors* ∈ {2, 3, 4, 5}, não sendo possível testar valores superiores a 5 devido ao número reduzido de amostras disponíveis para a classe minoritária.

É importante salientar que a técnica SMOTE foi aplicada somente no conjunto de treinamento, não sendo aplicada em nenhum momento no conjunto de testes para não gerar viés positivo nos resultados. No caso da aplicação da técnica de validação cruzada, a técnica SMOTE foi aplicada aos *folds* que formariam o conjunto de treino, não sendo aplicada ao *fold* com os dados de teste. Sendo assim, para cada nova combinação de *folds*, a técnica SMOTE foi aplicada novamente sobre o conteúdo dos *folds* originais, garantindo-se sempre que o *fold* de conjunto de testes fosse utilizado sem qualquer alteração.

3.5 Classificadores

Em nosso estudo comparativo da tarefa em investigação, avaliamos classificadores bem conhecidos na literatura. Objeti-

vamos averiguar a efetividade das técnicas de classificação SVM, ANN, *Random Forest* na classificação de frases relacionadas à agricultura sustentável nas oito classes de intenções previamente definidas. Para esse fim, foram utilizadas as implementações das respectivas técnicas disponíveis no pacote *scikit-learn*¹¹. No caso da *Transfer Learning*, foi utilizada a implementação disponível no *Keras*¹².

Cada uma das técnicas averiguadas possui suas particularidades e configurações de parâmetros que permitem ser ajustados durante o treinamento para produzir um modelo de classificação o mais generalista possível. Assim, nas seções seguintes, para cada técnica de aprendizado de máquina utilizada, efetuamos preliminarmente uma revisão teórica sucinta da respectiva técnica. Após a revisão teórica preliminar, apresentamos um detalhamento de como a técnica foi utilizada em específico em nosso estudo, bem como os valores de parâmetros utilizados durante o treinamento e avaliação para a escolha do melhor modelo.

3.5.1 Rede Neural Artificial multicamadas

As redes neurais apresentam várias características que justificam sua aplicação em tarefas de classificação de texto, considerando sua alta efetividade e a capacidade de generalização. Uma ANN tradicional de múltiplas camadas, usada nesta investigação, é um modelo em camadas composto por vários elementos de processamento chamados neurônios. Esse modelo apresenta uma camada de entrada, na qual os dados são recebidos, uma ou mais camadas intermediárias, e uma camada de saída, que fornece a resposta do classificador.

Cada neurônio tem uma função chamada Função de Ativação, usada para determinar o valor de saída do respectivo neurônio a partir do valor de entrada. A Figura 4 ilustra a estrutura comumente utilizada em uma ANN multicamadas.

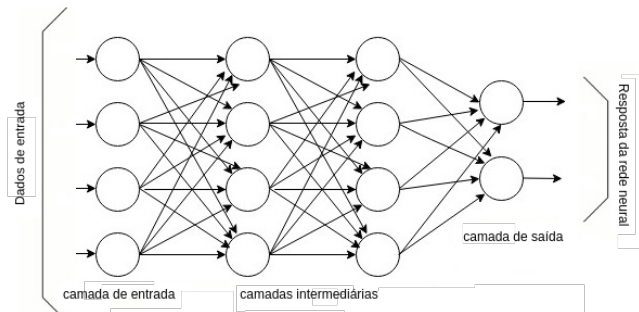


Figura 4. Estrutura tradicional de uma rede neural artificial multicamadas.

Pesos sinápticos são associados às conexões entre neurônios (sinapses) de diferentes camadas. Os valores de entrada para os neurônios da camada de entrada correspondem aos dados a serem utilizados pela rede neural. Para os neurônios de outras camadas, o valor de entrada no neurônio, isto é, o valor de entrada da função de ativação de um dado neurônio, é igual

¹⁰<https://imbalanced-learn.readthedocs.io/en/stable/generated/imblearn.over_sampling.SMOTE.html> - Último acesso em Janeiro de 2020.

¹¹<<https://scikit-learn.org/stable/>> - Último acesso em Janeiro 2020.

¹²<<https://keras.io/>> - Acessada em Janeiro de 2020.

à soma das saídas dos neurônios da camada anterior, multiplicados pelos respectivos pesos sinápticos. Formalmente, a entrada x_j^l de uma função de ativação $f(x_j^l)$ de um neurônio j pertencente à camada l pode ser descrita de acordo com a Equação 1:

$$x_j^l = \sum_{i=1}^J w_{i,j} f(x_i^{l-1}) + c, \quad (1)$$

na qual J é o número total de neurônios na camada $l-1$; $f(x_i^{l-1})$ é o valor de saída da função de ativação do neurônio i pertencente à camada $(l-1)$; $w_{i,j}$ é o peso sináptico da conexão entre os neurônios i e j ; c é uma constante chamada *bias* e seu objetivo é permitir um ajuste na saída da função de ativação, permitindo produzir um modelo melhor.

Neste trabalho foi utilizado um caso especial de ANN multicamadas, denominado de MLP (*Multilayer Perceptron*). Treinamos a rede utilizando o algoritmo *Backpropagation*. Este algoritmo realiza uma comparação entre os resultados atingidos e os resultados esperados na camada de saída da rede neural e, a partir desta última camada, ajusta os pesos sinápticos da rede. Detalhes adicionais sobre a teoria de Redes Neurais, rede MLP e o algoritmo *Backpropagation* podem ser consultados em [10].

Para a implementação prática da ANN do tipo MLP, utilizamos a classe *MLPClassifier*¹³. Os principais parâmetros da classe *MLPClassifier* são: *alpha*, *hidden_layer_sizes* e a função de ativação. O parâmetro *alpha* é um valor escalar. O parâmetro *hidden_layer_sizes* indica o número de neurônios nas camadas intermediárias, sendo necessário especificar o número total de neurônios para cada camada intermediária. A função de ativação selecionada foi a tangente hiperbólica (cf. Equação 2),

$$f(x_j) = \tanh(x_j) = \frac{e^{x_j} - e^{-x_j}}{e^{x_j} + e^{-x_j}}, \quad (2)$$

na qual x_j é o valor de entrada do neurônio j , sendo $1 \leq j \leq N$, e N o total de neurônios da rede neural.

Preliminarmente, treinamos a ANN com uma, duas e para três camadas intermediárias. Por meio da análise preliminar, com base nos resultados obtidos, decidimos efetuar a avaliação completa para uma e para três camadas, uma vez que forneceram os melhores resultados.

Para o caso de uma camada intermediária a seguinte configuração da ANN é utilizada:

- **Camada de entrada:** o total de nós é igual a dimensão do vetor de características;
- **Camada intermediária:** composta por L nós, sendo *hidden_layer_sizes*=(L);
- **Camada de saída:** composta por 8 nós.

Para o caso de 3 camadas intermediárias, adiciona-se duas camadas a estrutura mencionada acima: a segunda camada intermediária, composta por $L/2$ nós e a terceira camada intermediária, composta por $L/2$ nós, sendo assim, *hidden_layer_sizes* = ($L, L/2, L/2$). Treinamos e avaliamos os seguintes valores para L : $20 \leq L \leq 300$, para todo L múltiplo de 20. Avaliamos *alpha* $\in \{0.01, 0.1\}$.

3.5.2 Support Vector Machine - SVM

SVM é uma técnica de aprendizado de máquina supervisionado tipicamente utilizada para a classificação de dados, sendo utilizada também para a previsão de novos valores, processo esse denominado regressão. A técnica de SVM divide-se em SVM Lineares e SVM Não-Lineares. Um classificador SVM é considerado linear caso consiga separar os dados de duas classes por um hiperplano definido pela Equação 3. Porém, o SVM linear não busca apenas um hiperplano, mas busca àquele que proporciona a maior distância entre os elementos de classes distintas que estejam mais próximos ao hiperplano buscado [67],

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b, \quad (3)$$

na qual:

- $\mathbf{w} \cdot \mathbf{x}$: é o produto escalar entre os vetores \mathbf{w} e \mathbf{x} ;
- $\mathbf{w} \in \mathbb{R}$: é um vetor de pesos e normal ao hiperplano descrito;
- b : é um escalar, sendo que $b \in \mathbb{R}$.

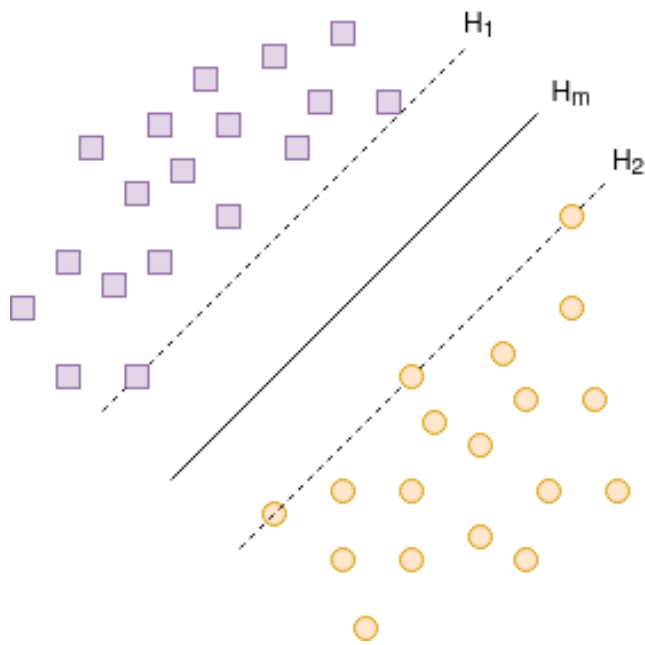
A Figura 5a ilustra a representação de duas classes e o hiperplano (H_m) buscado pela SVM linear. Como na Figura 5a a representação pressupõe um espaço de características bidimensional (dimensionalidade dos elementos igual a 2), H_m é uma reta. Se a dimensionalidade dos elementos fosse 3, H_m seria um plano e assim por diante.

O SVM pode ser usado para solucionar de modo linear um problema com dados não linearmente separáveis. No entanto, são necessários ajustes, já que o hiperplano separa apenas dados linearmente separáveis. Para isto, o conjunto de dados é mapeado de seu espaço original para um novo espaço de maior dimensão, denominado espaço de características, e a separação de dados é feita por uma SVM linear nesse novo espaço expandido, conforme ilustrado na Figura 5b.

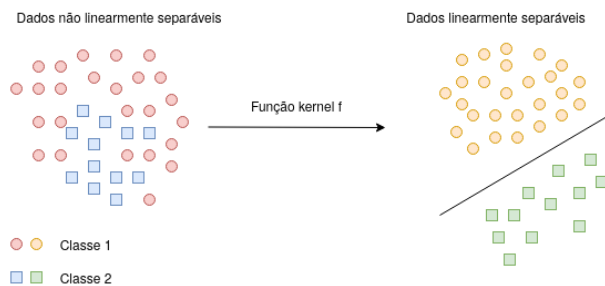
O mapeamento é feito por uma função denominada de *kernel*. Uma função *kernel* é uma função que recebe dois vetores \mathbf{x}_i e \mathbf{x}_j do espaço de entradas e computa o produto escalar desses dados no espaço de características [67]. Detalhes sobre a teoria de SVM, bem como de funções *kernel* podem ser consultados em [10].

Neste trabalho, os dados em análise são não-linearmente separáveis, o que resulta da necessidade do uso de uma SVM não linear. Em termos práticos, utilizamos uma função *kernel* que efetua o mapeamento dos dados para um espaço linear.

¹³<http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html>, Último acesso em Janeiro de 2020.



(a) Representação de duas classes em uma SVM linear e o hiperplano de margem máxima (representado por uma reta).



(b) Representação gráfica dos dados não linearmente separáveis mapeado para um problema linear utilizando uma função de mapeamento f .

Figura 5. Ilustrações gráficas de conceitos referentes a SVM. Imagens próprias.

Para fins de implementação, utilizamos a classe *SVC*¹⁴. Esta classe possui funções que implementam o treinamento e a classificação de dados não linearmente separáveis combinando uma função *kernel* e uma SVM linear. Em particular, os métodos *fit* e *predict* da classe *SVC* são responsáveis pelo treinamento e pela classificação, respectivamente. Deste modo, a partir dos parâmetros recebidos, a classe efetua o mapeamento dos dados para um espaço abstrato utilizando a função *kernel* e separa, nesse novo espaço, os dados linearmente, buscando o hiperplano de margem máxima. Nós utilizamos a versão multi-classes *one-versus-one*.

Os principais parâmetros da classe *SVC* são: a função *kernel* e C (parâmetro de penalidade de erro). Nós escolhemos a função *kernel* RBF (*Radial Basis Function*), definida pela Equação 4. Nesse caso, é necessário determinar o parâmetro

γ da função:

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2}, \quad (4)$$

na qual \mathbf{x}_i e \mathbf{x}_j são dois vetores do conjunto de dados e $\gamma > 0$ é o parâmetro cujo valor deve ser ajustado durante o treinamento. Avaliamos os seguintes conjuntos de parâmetros: $10^{-3} \leq C \leq 10^2$, para todo C múltiplo de 10 (não houve variações significativas em testes com intervalos menores). Para cada valor C , $10^{-3} \leq \gamma \leq 10^2$, para todo γ múltiplo de 10.

3.5.3 Random Forest

Random Forest (ou floresta aleatória de decisão) é um algoritmo de aprendizado de máquina supervisionado que pode ser utilizado para classificação e regressão de dados. Este algoritmo efetua a decisão final com base no voto de diversas árvores de decisão, todas com o mesmo peso de voto. Um aspecto importante deste algoritmo é o fato de que cada árvore de decisão trabalha independentemente das demais, mas todas contribuem para a decisão final. Detalhes adicionais sobre o algoritmo *Random Forest* são descritos em [11].

Para fins práticos, na solução adotada foi utilizada a implementação disponibilizada na classe *RandomForestClassifier*¹⁵. O principal parâmetro dessa classe é o $n_estimators$ (número de árvores de decisão utilizadas). Para tanto, foi efetuado o treinamento variando $n_estimators$ entre 20 e 300, em múltiplos de 20. Optamos por limitar o $n_estimators$ em 300, pois próximo deste número não houve indicativos de ganhos nos resultados e houve um aumento nos recursos computacionais necessários para execução.

Em [68] é apresentado um estudo sobre o número de árvores necessários em uma *Random Forest*. Para os demais parâmetros foram utilizados os valores padrões da biblioteca, por considerarmos adequados ao problema em foco.

3.5.4 Transfer Learning

Com a técnica de *Transfer Learning* é possível reaproveitar soluções aplicadas para resolver um problema (domínio de origem) em outros problemas (domínio destino). Tais soluções incluem algoritmos, arquiteturas, modelos treinados entre outros artefatos. Modelos pré-treinados e arquiteturas CNN vem sendo reutilizadas com sucesso por meio de *Transfer Learning* (e.g., [69]). Por exemplo, redes CNN treinadas para o reconhecimento de imagens com o uso de grandes bases de dados anotadas (e.g., ImageNet¹⁶) são reutilizadas para solucionar outros problemas relacionados com o reconhecimento de imagens ou não (i.e., reconhecimento de padrões em outros domínios destino) [70, 71]. Particularmente, essa abordagem é útil quando o domínio destino não possui grande bases anotadas, como no domínio de origem [72]. Detalhes teóricos e metodológicos podem ser consultados em [69, 72, 73].

¹⁴<<https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>> - Último acesso em Janeiro de 2020.

¹⁵<<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>> - Último acesso em Janeiro de 2020.

¹⁶<http://www.image-net.org/>

Para fins práticos, utilizamos a implementação disponível na documentação do *Keras*¹⁷. Escolhemos esta implementação por fornecer resultados promissores em outras aplicações em Machine Learning [74]. Foram utilizados os modelos pré-treinados VGG19, Xception e MobileNetV2 entre os disponíveis no *Keras*, pois são três modelos entre aqueles com maior acurácia durante a validação para a geração desses modelos. Tais modelos foram treinados com a base ImageNet, que inclui mais de 14 milhões de imagens organizadas/anotadas pela WordNet¹⁸.

Para viabilizar a reutilização desses modelos é necessário mapear descritores de nosso domínio destino (*i.e.*, vetores de características gerados pelos algoritmos de *Word Embeddings*) em imagens que possam ser submetidas aos classificadores CNN treinadas no domínio de origem. Assim, mapeamos os vetores de características (cf. Tabela 4) para uma imagem de dimensão $N \times N$, preenchendo sequencialmente cada valor de cada vetor de características na matriz de tamanho $N \times N$. Após este procedimento, a imagem resultante é a entrada do modelo pré-treinado. Os dados de saída do modelo são usados como entrada de um classificador ANN, SVM ou *Random Forest*, com as mesmas características descritas nas Seções 3.5.1, 3.5.2 e 3.5.3, respectivamente. A saída do classificador ANN/SVM/Random Forest é utilizada como a resposta final da classificação.

A Tabela 5 apresenta os valores de N mínimos (segunda coluna) de acordo com a dimensão do vetor de características (primeira coluna). A quarta coluna apresenta o número de píxeis da imagem (*i.e.*, N^2 pois são imagens quadradas). Já a terceira coluna apresenta o maior número inteiro divisível entre o número de píxeis e a dimensão do vetor de características (primeira coluna). Por exemplo, para uma imagem com 1296 píxeis (segunda linha) é possível armazenar 25 palavras com vetores de 50 valores por imagem, resultando em 1250 preenchidos. Neste caso, os 46 píxeis restantes (que não completam uma palavra inteira) são preenchidos com zeros.

Tabela 5. Valores mínimos de N que define o tamanho da imagem ($N \times N$) a ser utilizada como dado de entrada da CNN.

Dimensão da <i>Word Embedding</i>	Valor de N	Dimensão do vetor de caract.	Número de píxeis por imagem
50	36	1250	1296
100	50	2500	2500
300	88	7500	7744
600	124	15000	15376

A Tabela 6 indica os valores mínimo e *default* de *width* e *height* exigido por cada modelo utilizado. Avaliamos as dimensões mínima e *default* para cada modelo. Caso a imagem $N \times N$ tenha dimensão inferior/superior ao exigido pelo modelo (mínimo/*default*), a imagem é redimensionada para

o valor exigido utilizando a função *resize*¹⁹, que tem como base a interpolação de dados. Para a interpolação, utilizamos a opção bi-cúbica.

Tabela 6. Valores mínimos/*default* de *width* e *height* para cada modelo pré-treinado avaliado.

Modelo	Valor mínimo/ <i>default</i> de <i>width</i> e <i>height</i>
VGG19	32/224
Xception	71/299
MobileNetV2	32/224

4. Avaliação Experimental

Esta seção apresenta os resultados experimentais do nosso estudo comparativo das técnicas de aprendizado de máquina na tarefa de classificação de intenções em textos livres. A Subseção 4.1 apresenta os procedimentos experimentais conduzidos, incluindo as métricas de avaliação usadas. A Subseção 4.2 apresenta os resultados experimentais obtidos, ao passo que, a subseção 4.3 apresenta as discussões destes resultados, bem como limitações, lições aprendidas e desafios de pesquisa a serem mais profundamente abordados.

4.1 Procedimentos

Nosso estudo experimental avaliou individualmente as técnicas SVM, ANN, *Random Forest* e *Transfer Learning*. Cada técnica de aprendizado de máquina foi investigada utilizando o banco de dados classificado por cada uma das três pessoas (A, B e C) individualmente. As técnicas de aprendizado de máquina (SVM, ANN e *Random Forest*) foram combinadas com sete *Word Embeddings* (*Word2Vec CBOW*, *Word2Vec Skip-Gram*, *FastText CBOW*, *FastText Skip-Gram*, *Wang2Vec CBOW*, *Wang2Vec Skip-Gram* e *Glove*).

Adicionalmente, conduzimos duas análises específicas: 1) uma utilizando a técnica de validação cruzada *k-fold*; e 2) outra dividindo-se os dados disponíveis em dois conjuntos aleatórios, sendo 80% para o treinamento e 20% para os testes. Os conjuntos de parâmetros avaliados para cada técnica foram descritos na subseção 3.5.

Para a avaliação das técnicas *Random Forest*, ANN e SVM foram utilizados simultaneamente três computadores com as seguintes arquiteturas computacionais: (1) servidor com dois Intel Xeon E5-2620 de 2.00 GHz com 6 cores e 96 Gb RAM; (2) servidor com dois Intel Xeon E5620 de 2.00 GHz com 4 cores e 64 Gb RAM; e (3) computador pessoal com um Intel Core i5-3570 de 3.40 GHz e 16 Gb RAM. A alocação de recursos para cada computador teve como critério a disponibilidade de cada máquina. Para a avaliação da técnica de *Transfer Learning* foram utilizadas duas GPUs Nvidia TITAN Xp 12 Gb instaladas no servidor (1).

¹⁷<<https://keras.io/applications/>> - Acessado em Janeiro de 2020.

¹⁸<<https://wordnet.princeton.edu/>> - Último acesso em Maio de 2020

¹⁹<<https://scikit-image.org/docs/0.12.x/api/skimage.transform.html>> - Último acesso em Julho de 2021.

Para uma avaliação objetiva, utilizamos como métrica o F1-Score [75], pois trata-se de uma métrica comumente utilizada em vários estudos de aprendizado de máquina, bem como foi adotada pelos principais trabalhos relacionados atuais sobre classificação automática de intenções (cf., Subseção 2.2).

A métrica F1-Score é calculada de acordo com a Equação 5:

$$F1 - Score = 2 * \frac{precision * recall}{precision + recall} \quad (5)$$

sendo,

$$precision = \frac{True\ Positives}{True\ Positives + False\ Positives}, \quad (6)$$

$$recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}. \quad (7)$$

Calculamos a F1-Score para cada classe de intenção e a F1-Score ponderada geral, denominada no decorrer da seção experimental de macro F1-Score, considerando todas as classes. A partir dos resultados experimentais gerados por meio da avaliação dos parâmetros (cf. Seção 3.5), verificamos os melhores resultados para cada combinação **classificador/ algoritmo de Word Embedding/tamanho do vetor**, tendo como critério de escolha o maior valor de macro F1-Score. Em caso de valores iguais de macro F1-Score, escolhemos na ordem: menor dimensão de *Word Embedding* (50, 100, 300 e 600); ANN; RF; SVM; menor número de recursos necessários (número de árvores para a RF e o número de camadas/nós no caso da ANN). A ordem de escolha dos classificadores ANN, RF e SVM foi decidida após uma análise global dos melhores resultados obtidos. A seguir, a subseção 4.3 apresenta os melhores resultados experimentais obtidos.

4.2 Resultados

As Tabelas 7 e 8 apresentam os melhores resultados experimentais obtidos com a validação cruzada e o sorteio aleatório de 80% para treino e 20% para teste, respectivamente. Nessas tabelas são apresentados resultados com a melhor configuração para o macro F1-Score ponderado. Os resultados são apresentados e organizados da seguinte maneira: por pessoa que classificou os dados (primeira coluna); por técnica de aprendizado de máquina (segunda coluna); por algoritmo de *Word Embedding* e dimensões (terceira coluna); F1-Score de cada classe de intenção (quarta coluna, subdividida em oito), sendo o valor médio no caso da validação cruzada; e o macro F1-Score ponderado de acordo com as classes (última coluna), sendo o valor médio no caso da validação cruzada. Para o cálculo do macro F1-Score ponderado, foi utilizado o valor *weighted* para o parâmetro *average* da função *f1_score*²⁰.

²⁰<https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html> - Último acesso em março de 2020

A coluna **Algoritmo de Word Embedding** apresenta o algoritmo de *Word Embedding* que obteve o melhor resultado para o respectivo classificador, a dimensão do vetor de *Word Embedding* (50, 100, 300 ou 600), seguido da estrutura detalhada a seguir:

- *Random Forest*: número de árvores;
- ANN: número de camadas intermediárias, número de nós (valor de *L*), e o valor do parâmetro *alpha* (Seção 3.5.1), respectivamente;
- SVM: *g* e *c*, respectivamente;
- *Transfer Learning*: a estrutura do classificador utilizado como última camada (ANN, *Random Forest* ou SVM), modelo pré-treinado e a dimensão da imagem de entrada na CNN.

Os melhores resultados de macro F1-Score ponderado, utilizando a validação cruzada com *k* = 5, de acordo com a Tabela 7, por pessoa foram: macro F1-Score ponderado igual a 0,55 com os dados da Pessoa A, obtidos com o classificador ANN e o algoritmo FastText Skip-Gram; 0,54 para a Pessoa B, com o classificador ANN e o algoritmo FastText Skip-Gram; 0,43 para os dados da Pessoa C com o classificador *Random Forest* e o algoritmo FastText Skip-Gram.

Na Tabela 7, apresentamos os resultados de macro F1-Score ponderado obtidos com a técnica de *Transfer Learning*, sendo: 0,38 com a Pessoa A, 0,42 com a Pessoa B e 0,35 com a Pessoa C. Esses resultados experimentais foram obtidos com o modelo *MobileNetV2* e imagens de 224 x 224. No caso da *Transfer Learning*, foram avaliadas apenas as configurações dos melhores resultados obtidos para cada pessoa, de acordo com as Tabelas 7 e 8. Em outras palavras, para cada modelo pré-treinado (*VGG19*, *Xception* e *MobileNetV2*), foram avaliados os três classificadores (*Random Forest*, ANN e SVM) e suas duas respectivas configurações de algoritmos de *Word Embedding*, dimensões e parâmetros apresentados na terceira coluna das Tabelas 7 e 8.

Os melhores resultados de macro F1-Score ponderado, utilizando o sorteio aleatório de 80% para treino e 20% para teste, de acordo com a Tabela 8, por pessoa foram: 0,67 com os dados da Pessoa A, obtidos com o classificador ANN e o algoritmo FastText Skip-Gram; 0,65 para a Pessoa B, com o classificador ANN e o algoritmo Wang2Vec CBOW; 0,51 para os dados da Pessoa C com o classificador *Random Forest* e o algoritmo Word2Vec Skip-Gram. A Tabela 8 apresenta igualmente os resultados de macro F1-Score ponderado obtidos com a técnica de *Transfer Learning*, sendo: 0,44 com a Pessoa A, 0,44 com a Pessoa B e 0,44 com a Pessoa C.

As Tabelas 9-11 apresentam os melhores resultados de F1-Score para cada classe de intenção, obtidos por meio das avaliações com a validação cruzada e com o sorteio de 80% para treino e 20% para teste. Os dados apresentados na terceira coluna indicam o algoritmo de *Word Embedding* / dimensão do vetor / classificador / configurações do classificador. As

Tabela 7. Melhores resultados experimentais por pessoa e classificador, utilizando validação cruzada com k=5.

Pessoa	Classificador	Algoritmo de Word Embedding	Classe de Intenção								F1-Score médio balanceado
			Proposal	Inducement	Forecast	Wish	Assertion	Valuation	Palinode	Contribution	
A	Random Forest	FastText Skip-Gram / 300 / 280	0,37	0,33	0,44	0,59	0,66	0,58	0,26	0,43	0,54
	ANN	FastText Skip-Gram / 300 / 1 / 80 / 0.1	0,27	0,36	0,57	0,42	0,63	0,59	0,34	0,42	0,55
	SVM	Word2Vec CBOW / 600 / 0.01 / 100	0,13	0,33	0,45	0,43	0,67	0,55	0,20	0,33	0,52
	Transfer Learning	FastText Skip-Gram / 300 / SVM / 0.01 / 100 / MobileNetV2 / 224	0,14	0,22	0,30	0,33	0,50	0,40	0,33	0,07	0,38
B	Random Forest	Wang2Vec CBOW / 300 / 220	0,56	0,42	0,50	0,41	0,51	0,59	0,46	0,35	0,52
	ANN	FastText Skip-Gram / 300 / 1 / 80 / 0.1	0,61	0,42	0,63	0,45	0,49	0,56	0,61	0,48	0,54
	SVM	FastText Skip-Gram / 50 / 0.1 / 10	0,24	0,39	0,57	0,35	0,53	0,60	0,44	0,34	0,53
	Transfer Learning	FastText Skip-Gram / 50 / SVM / 0.01 / 10 / MobileNetV2 / 224	0,21	0,24	0,45	0,26	0,44	0,48	0,38	0,20	0,42
C	Random Forest	Glove / 100 / 240	0,27	0,28	0,36	0,35	0,64	0,21	0,29	0,45	0,43
	ANN	FastText Skip-Gram / 300 / 3 / 220 / 0.1	0,30	0,25	0,43	0,34	0,61	0,18	0,31	0,38	0,42
	SVM	Wang2Vec Skip-Gram / 100 / 0.01 / 10	0,27	0,20	0,31	0,28	0,63	0,20	0,16	0,18	0,40
	Transfer Learning	Word2Vec Skip-Gram / 50 RF / 260 / MobileNetV2 / 224	0,15	0,23	0,25	0,22	0,54	0,27	0,13	0,08	0,35

Tabela 8. Melhores resultados experimentais por pessoa e classificador, utilizando 80% do conjunto para treinamento de 20% para teste.

Pessoa	Classificador	Algoritmo de Word Embedding	Classe de Intenção								F1-Score médio balanceado
			Proposal	Inducement	Forecast	Wish	Assertion	Valuation	Palinode	Contribution	
A	Random Forest	Wang2Vec / 300 / 40	0,60	0,62	0,59	0,86	0,65	0,64	0,50	0,67	0,63
	ANN	FastText Skip-Gram / 600 / 1 / 40 / 0.01	0,31	0,53	0,70	0,60	0,75	0,70	0,50	0,50	0,67
	SVM	FastText Skip-Gram / 300 / 0.01 / 100	0,22	0,41	0,68	0,67	0,63	0,70	0,50	0,33	0,60
	Transfer Learning	FastText Skip-Gram / 300 / SVM / 0.01 / 100 / MobileNetV2 / 224	0,17	0,26	0,42	0,44	0,55	0,43	0,50	0,40	0,44
B	Random Forest	Word2Vec Skip-Gram / 300 / 120	1,00	0,36	0,53	0,44	0,68	0,66	0,50	0,80	0,61
	ANN	Wang2Vec CBOW / 600 / 3 / 240 / 0.1	0,50	0,60	0,68	0,67	0,60	0,69	0,80	0,67	0,65
	SVM	FastText Skip-Gram / 600 / 0.01 / 10	0,80	0,67	0,62	0,55	0,55	0,62	0,40	0,80	0,61
	Transfer Learning	FastText Skip-Gram / 50 / SVM / 0.01 / 10 / MobileNetV2 / 224	0,40	0,18	0,51	0,25	0,50	0,47	0,50	0,40	0,44
C	Random Forest	Word2Vec Skip-Gram / 50 / 260	0,30	0,36	0,25	0,40	0,67	0,46	0,67	1,00	0,51
	ANN	FastText CBOW / 300 / 3 / 180 / 0.01	0,48	0,36	0,61	0,38	0,67	0,20	0,40	0,50	0,50
	SVM	FastText Skip-Gram / 50 / 0.01 / 1	0,43	0,21	0,50	0,36	0,68	0,17	0,40	0,33	0,46
	Transfer Learning	Word2Vec Skip-Gram / 50 RF / 260 / MobileNetV2 / 224	0,21	0,24	0,48	0,27	0,61	0,37	0,33	0,40	0,44

configurações do classificador apresentam a mesma estrutura descrita anteriormente nesta subseção. Note que as Tabelas 9-11 apresentam os melhores resultados por classe independentemente do resultado de macro F1-Score ponderado, isto as diferenciam dos resultados por classe apresentados nas Tabelas 7 e 8. Em outras palavras, nas Tabelas 7 e 8 selecionamos os melhores macro F1-Score ponderado e apresentamos também os resultados das classes que produziram conjuntamente este macro F1-Score. Já nas Tabelas 9-11, dentre todas as avaliações efetuadas e resultados produzidos, escolhemos o melhor valor de F1-Score de cada classe.

Para pessoa A (Tabela 9), o valor de F1-Score (na configuração 80/20) teve melhor resultado para *Wish*, *Palinode* e *Contribution* com 1,00 e o pior resultado foi 0,75 para a classe *Valuation*. Já na validação cruzada foi obtido 0,68 para *Assertion* como melhor resultado e 0,41 para *Proposal* como o resultado menos satisfatório.

Para pessoa B (Tabela 10), o valor de F1-Score (na configuração 80/20) teve melhor resultado para *Proposal*, *Palinode*

e *Contribution* com 1,00 e o pior resultado foi 0,69 para classe *Assertion*. Já na validação cruzada foi obtido 0,79 para *Proposal* como melhor resultado; e 0,50 para *Inducement* como o resultado menos satisfatório.

Para pessoa C (Tabela 11), o valor de F1-Score (na configuração 80/20) teve melhor resultado para *Palinode* e *Contribution* com 1,00, sendo que o pior resultado foi 0,51 para a classe *Valuation*. Na validação cruzada foi obtido F1-Score de 0,64 para *Assertion* como melhor resultado e F1-Score de 0,33 para *Inducement*.

4.3 Discussão

A classificação de intenções em texto livre na Língua Portuguesa é um desafio de pesquisa ainda pouco explorado. No melhor de nosso conhecimento, o estudo com textos na área agrícola não foi encontrado na literatura. Nesta investigação, conduzimos uma análise comparativa extensiva para averiguar a efetividade em se aplicar diferentes modelos de linguagem que vetorizam palavras e técnicas de aprendizado de máquina nesta tarefa.

Tabela 9. Melhores resultados de cada classe de intenção, classificado pela **peessoa A.**

Classe de Intenção	80/20		Validação cruzada	
	F1-Score	Classificador / Configurações	F1-Score	Classificador / Configurações
Proposal	0,91	Word2Vec Skip-Gram / 600 / ANN / 3 / 100 / 0.1	0,41	Word2Vec CBOW / 100 / RF / 160
Inducement	0,77	Word2Vec CBOW / 100 / RF / 180	0,49	FastText CBOW / 300 / RF / 220
Forecast	0,79	FastText CBOW / 100 / ANN / 1 / 160 / 0.01	0,64	FastText Skip-Gram / 600 / ANN / 1 / 220 / 0.1
Wish	1,00	Glove / 600 / ANN / 1 / 160 / 0.1	0,66	FastText Skip-Gram / 600 / ANN / 1 / 240 / 0.1
Assertion	0,80	Word2Vec Skip-Gram / 600 / ANN / 1 / 220 / 0.01	0,68	Word2Vec Skip-Gram / 600 / ANN / 1 / 140 / 0.1
Valuation	0,75	Word2Vec CBOW / 600 / ANN / 3 / 280 / 0.01	0,60	Word2Vec Skip-Gram / 300 / RF / 260
Palinode	1,00	Wang2Vec CBOW / 50 / ANN / 1 / 40 / 0.1	0,62	FastText Skip-Gram / 50 / ANN / 3 / 140 / 0.01
Contribution	1,00	Wang2Vec CBOW / 50 / ANN / 1 / 60 / 0.1	0,60	FastText Skip-Gram / 600 / RF / 160

Tabela 10. Melhores resultados de cada classe de intenção, classificado pela **peessoa B.**

Classe de Intenção	80/20		Validação cruzada	
	F1-Score	Classificador / Configurações	F1-Score	Classificador / Configurações
Proposal	1,00	Word2Vec CBOW / 50 / RF / 20	0,79	Wang2Vec Skip-Gram / 600 / ANN / 1 / 180 / 0.01
Inducement	0,83	Glove / 600 / ANN / 3 / 80 / 0.01	0,50	Wang2Vec Skip-Gram / 600 / RF / 120
Forecast	0,83	FastText Skip-Gram / 300 / 1 / 80 / 0.01	0,65	FastText Skip-Gram / 300 / ANN / 1 / 60 / 0.1
Wish	0,91	Word2Vec Skip-Gram / 300 / 1 / 80 / 0.01	0,58	Wang2Vec CBOW / 50 / RF / 60
Assertion	0,69	Glove / 50 / RF / 160	0,56	Wang2Vec CBOW / 600 / ANN / 3 / 220 / 0.01
Valuation	0,73	Word2Vec Skip-Gram / 50 / RF / 280	0,62	Wang2Vec CBOW / 600 / ANN / 1 / 220 / 0.01
Palinode	1,00	Word2Vec CBOW / 50 / ANN / 1 / 80 / 0.1	0,78	Glove / 600 / ANN / 1 / 180 / 0.01
Contribution	1,00	Word2Vec CBOW / 50 / RF / 20	0,73	Word2Vec Skip-Gram / 300 / RF / 120

Ao analisar os resultados da Tabela 7, há um indicativo de que o algoritmo de *Word Embedding* que tem melhor resultado para contexto de agricultura sustentável é o FastText Skip-Gram, uma vez que esteve presente em 5 dos 9 melhores resultados obtidos (considerando os classificadores *Random Forest*, ANN e SVM). Em relação as estruturas Skip-Gram e CBOW, a primeira está em 6 dos 9 melhores resultados, sendo um indicativo de ser a mais indicada no contexto estudado. Em relação a dimensão do vetor de *Word Embedding*, a dimensão igual a 300 está em 5 de 9 melhores cenários/configurações experimentais, indicando ser o melhor valor de dimensão entre as 4 dimensões avaliadas.

Ao analisar os resultados presentes na Tabela 8, também há um indicativo de que o algoritmo de *Word Embedding* que apresenta o melhor resultado para o contexto avaliado é o FastText Skip-Gram, sendo o mais frequente com os melhores resultados. As dimensões 300 e 600 foram as mais frequentes nos melhores resultados, embora, ao considerarmos o melhor resultado para cada pessoa, três das quatro dimensões avaliadas foram encontradas.

Ao compararmos os resultados de macro F1-Score ponderado das Tabelas 7 e 8, observamos resultados superiores quando utilizamos a divisão de 80% para treino e 20% para teste em comparação com a validação cruzada. Por não possuímos uma base anotada com um número muito alto de sentenças (milhares ou milhões), na validação cruzada a divisão aleatória das frases entre os *folds* pode resultar em *folds* de teste (que não é balanceado) nos quais determinadas classes possuam poucas amostras pouco relacionadas com as utilizadas no treinamento²¹. Invariavelmente, isso reduz o

F1-Score geral da respectiva rodada, impactando o cálculo do macro F1-Score ponderado médio.

Com base na teoria de cada *Word Embedding* apresentada na Seção 3.3, esperava-se como resultado das avaliações experimentais que as técnicas FastText e Wang2Vec apresentassem os melhores resultados. Isto, pois, duas características fortemente presentes na Língua Portuguesa são consideradas por essas técnicas, sendo: o aspecto morfológico na formação das palavras (considerado pela técnica FastText); e a forma sintática bem definida de uma frase (considerado na técnica Wang2Vec). Os resultados obtidos corroboram com a teoria, uma vez que FastText e Wang2Vec representam aproximadamente 77,78% dos melhores resultados de macro F1-Score ponderado nas Tabelas 7 e 8; e a maioria dos casos nas Tabelas 9 e 11 (aproximadamente 54%).

Para efetuar uma análise mais ampla dos resultados, selecionamos 5% dos melhores resultados do total de cada avaliação (validação cruzada e 80% para treino / 20% para teste). Para esta seleção, consideramos como melhores resultados os maiores valores de macro F1-Score. Após selecionar 5% do total, calculamos a média e o desvio padrão dos valores de macro F1-Score desses dados. Por fim, mantivemos entre os dados selecionados apenas aqueles cujo macro F1-Score fossem maiores do que o valor da média somado com o valor do desvio padrão. Após esse procedimento, contamos o número de ocorrências entre os dados restantes agrupando por *Word Embedding* e sua dimensão, estrutura (CBOW ou Skip-Gram) e a técnica de classificação. Os resultados são apresentados nas Tabelas 12 e 13 (Apêndice A), referentes as avaliações de validação cruzada e de 80/20, respectivamente.

Ao analisar a Tabela 12, observamos que as técnicas com o maior número de ocorrências são Wang2vec CBOW (dimensão 600), FastText Skip-Gram (dimensão 600) e FastText

²¹ As classes *Palinode* e *Contribution* são as menos frequentes no domínio e que foram suscetíveis a esta distribuição, com situações em que há apenas duas amostras.

Tabela 11. Melhores resultados de cada classe de intenção, classificado pela **pessoa C**.

Classe de Intenção	80/20		Validação cruzada	
	F1-Score	Classificador / Configurações	F1-Score	Classificador / Configurações
Proposal	0,67	Glove / 600 / RF / 120	0,38	Word2Vec Skip-Gram / 600 / ANN / 1 / 180 / 0.01
Inducement	0,53	Wang2Vec Skip-Gram / 600 / ANN / 3 / 40 / 0.01	0,33	Wang2Vec CBOW / 300 / ANN / 3 / 160 / 0.01
Forecast	0,80	FastText Skip-Gram / 50 / ANN / 1 / 200 / 0,01	0,51	FastText Skip-Gram / 50 / ANN / 1 / 60 / 0.1
Wish	0,74	Wang2Vec Skip-Gram / 300 / RF / 240	0,47	FastText CBOW / 600 / RF / 140
Assertion	0,72	Wang2Vec Skip-Gram / 300 / ANN / 3 / 220 / 0.01	0,64	Glove / 100 / RF / 240
Valuation	0,51	Wang2Vec CBOW / 100 / ANN / 1 / 240 / 0.01	0,34	Glove / 50 / ANN / 1 / 180 / 0.01
Palinode	1,00	Wang2Vec Skip-Gram / 600 / ANN / 3 / 80 / 0.01	0,44	Wang2Vec Skip-Gram / 50 / ANN / 3 / 260 / 0.01
Contribution	1,00	Glove / 50 / ANN / 1 / 120 / 0.01	0,63	Wang2Vec Skip-Gram / 300 / ANN / 1 / 120 / 0.1

Skip-Gram (dimensão 300), e o classificador ANN. Porém, entre estas três melhores, apenas a combinação FastText Skip-Gram, dimensão 300 e classificador ANN, está presente nos resultados das três pessoas (A, B e C), sendo, portanto, esta a combinação a mais indicada para ser considerada como a melhor quando a análise é efetuada na avaliação com a validação cruzada.

No caso da análise considerando a divisão 80% para treino e 20% para testes, cujos dados são apresentados na Tabela 13, observamos que as técnicas com o maior número de ocorrências são Wang2Vec CBOW, dimensão 600, classificador ANN e FastText Skip-Gram, dimensão 600, classificador ANN, sendo que ambas as combinações estão presentes nos resultados de todas as três pessoas. Além disso, o classificador ANN e a dimensão 600 apresentam-se como o mais indicados. É interessante mencionar que ao analisar a Tabela 12, a combinação FastText Skip-Gram, dimensão 300 e classificador ANN também poderia ser uma escolha a ser considerada, uma vez que é a terceira maior combinação em número de ocorrências, porém com um desvio padrão menor, o que pode indicar que esta combinação atende de forma mais equilibrada a previsão da classe de intenção de frases elaboradas tanto por pessoas com experiência na teoria do cubo (Seção 2) quanto por pessoas que a estudaram pela primeira vez, fato reforçado por ser esta a combinação mais indicada no caso da validação cruzada.

Ressalta-se que a análise mais ampla tanto da validação cruzada quando da avaliação de 80% treino e 20% teste corroboram com a teoria esperada das melhores *Word Embeddings*: *FastText* e *Wang2Vec*.

Em [12] foram conduzidas três avaliações para verificar a efetividade dos modelos de *Word Embeddings* pré-treinados, sendo: (1) uma avaliação intrínseca, com base no trabalho de [76], para avaliar a capacidade dos modelos quanto aos aspectos sintáticos e semânticos; (2) uma avaliação extrínseca utilizando a técnica *POS Tagging* para avaliar a capacidade dos modelos quanto à captura do aspecto morfo-sintático; (3) outra avaliação extrínseca adotando um coeficiente ρ , denominado de *Pearson's Correlation*, com objetivo de verificar a similaridade semântica entre as frases. Considerando os modelos de *Word Embeddings* que apresentaram os melhores resultados nas avaliações conduzidas em [12] (para as dimensões 50, 100, 300 e 600), nota-se que o algoritmo FastText Skip-Gram (dimensão 300) está presente entre os 8 melhores

resultados obtidos nas três avaliações conduzidas. Além disso, a maioria das configurações das Tabelas 7 e 8 também estão presentes entre os 8 melhores resultados apresentados por [12]. Deste modo, embora o contexto de agricultura sustentável tenha suas particularidades quando ao vocabulário, *Word Embeddings* geradas a partir de Corpus com vocabulário não específico também fornecem resultados promissores. É necessário, futuramente, um estudo mais aprofundado para verificar se esses modelos de *Word Embeddings* forem gerados a partir do vocabulário específico da agricultura, haveria melhorias significativas nos resultados. Descartamos os resultados apresentados por [12] referentes a dimensão de *Word Embedding* igual a 1000, pois esse valor de dimensão não foi avaliado em nosso estudo (por apresentar resultados preliminares piores que as dimensões 300 e 600 e necessitar de mais recursos computacionais).

Se observamos os resultados apresentados nas Tabelas 9-11, há a presença de diversos resultados com dimensão de *Word Embeddings* iguais a 50 e 100 para as classes *Palinode* e *Contribution*. Essas classes são as duas com o menor número de frases na base de dados, portanto, que obtiveram o maior número de dados gerados durante o balanceamento de classes durante o treinamento. Sendo assim, este é um indicativo de que uma técnica eficiente de balanceamento de classes pode reduzir a dimensão de *Word Embedding* necessária, um aspecto que merece um estudo mais aprofundado futuramente.

Os resultados obtidos com a técnica de *Transfer Learning*, inferiores aos resultados obtidos com as demais técnicas de aprendizado de máquina, podem ser resultantes principalmente devido aos modelos avaliados: *VGG19*, *Xception* e *MobileNetV2*. Apesar de promissores para outras áreas, conforme mencionado na Seção 3.5.4, esses modelos não são utilizados tradicionalmente em tarefas que envolvam o processamento de linguagem natural. Sendo assim, modelos como *Universal Sentence Encoder* e *Bidirectional Encoder Representations from Transformers* (BERT) [77, 52] podem ser mais apropriados em nosso contexto e serão objeto de estudo futuro.

Se por um lado o tamanho da base de dados investigada pode ser uma limitação do nosso estudo, e deve ser incrementada em longo prazo, uma vez que tal tarefa demanda um esforço muito grande de classificação manual; por outro lado, o fato de obtermos resultados superiores na avaliação 80/20 é um importante indicativo de que com uma base de dados

anotada com um número maior de amostras, possivelmente teremos resultados ainda melhores utilizando o algoritmo as respectivas configurações de *Word Embedding* e classificadores.

Os resultados apresentados nas Tabelas 9-11 indicaram que para uma aplicação prática em que se conhece a priori a classe desejada, pode ser possível obter melhores resultados escolhendo-se com mais refinamento uma das configurações para a classe desejada entre as apresentadas na Tabelas 9-11. Deste modo, embora essa configuração possa resultar em um menor macro F1-Score ponderado, ou um maior número de erros por classe, seria a configuração que, em tese, irá melhor prever as frases corretamente da classe desejada.

Outro ponto investigado neste artigo é a influência do anotador no treinamento e resultados. Enquanto foram obtidos resultados próximos para as pessoas A e B, os resultados para pessoa C foram diferentes dos demais (*e.g.*, o F1-Score ponderado foi igual a 0,55 para Pessoa A e 0,54 para Pessoa B e 0,43 para Pessoa C). Ao mesmo tempo que os resultados ressaltam o viés introduzido por quem realiza a anotação, os resultados também ressaltam que pessoas com maior domínio da técnica tendem a obter resultados médios mais próximos. Embora a subjetividade seja da natureza do problema (interpretar intenções em textos), uma hipótese é que o domínio da técnica pode ter trazido mais consistência nas classificações das pessoas com maior domínio da técnica (A e B). Assim, nossos resultados apontam para a necessidade de ter anotadores treinados, bem como para a necessidade por pesquisas adicionais para explicar o que acontece (tal como, para validar a hipótese supracitada).

Em relação aos principais métodos do Estado da Arte, vale salientar que uma análise comparativa direta seria imprecisa e inadequada neste momento, uma vez que os estudos usam bases diferentes, estão relacionados a diferentes domínios e tarefas, possuem diferentes métodos de construção da base, além de muitas estarem escritas em diferentes línguas. Realizada essa ressalva, discutimos os resultados obtidos perante métodos do estado da arte. Em um trabalho recente, Algotiml e Elmadany [55] obtiveram 0,61 de F1-Score considerando as 5 classes na validação cruzada ($k=5$), ao passo que obtivemos taxas próximas para duas das três pessoas que classificaram a base de dados (0,55 e 0,54, também avaliando com a validação cruzada e $k=5$). Apesar de inferiores, ressaltamos que obtivemos essas taxas para 8 classes de intenções, ao passo que os autores utilizam um número menor de classes (cinco). Outros trabalhos (*e.g.*, [41] e [46]) obtiveram resultados com valores de F1-Score compatíveis com os obtidos nesse trabalho, entretanto em outros domínios, explorando outras classes e utilizando diferentes métodos de análise.

Os autores em [78] avaliaram técnicas de aprendizado de máquina para a classificação de frases em português de três bases de dados. Duas das três bases de dados (EHR e YAHOO!) há classes de intenções que não possuem ao menos uma frase rotulada, o que dificulta uma comparação mais precisa entre os resultados. Para a base *Vilanarede*, os resul-

tados apresentados de macro F1-Score foram 0,45 (técnica proposta pelos autores), 0,37 (LogitBoost), 0,32 (SVM) e 0,27 (Naive Bayes). Em relação à técnica SVM, os valores de macro F1-Score obtidas pelo presente trabalho foram superiores, tanto com a utilização da validação cruzada quanto com a divisão 80/20%. Adicionalmente, em nosso caso, nenhuma classe de intenção teve um F1-Score igual a zero (Tabelas 7-8), conforme resultados experimentais apresentados pelos autores. Já em [3] foram obtidos resultados inferiores ao deste artigo, com F1-Score médio balanceado (considerando 7 das 8 classes) de 0,45 para ANN, 0,47 para SVM e 0,45 para *Random Forest*. Entanto é importante salientar que foram utilizados dados de outra natureza (postagens criminosas em redes sociais), bem como foi realizado um estudo mais limitado no que diz respeito às técnicas de classificação.

No trabalho recente de [79], os autores organizaram a base de dados em 12 domínios distintos e efetuaram a análise de sentimentos (positivo e negativo), utilizando para cada domínio 72% das frases para o treinamento e 18% para os testes. Os autores obtiveram resultados de macro F1-Score que variam de 48,8% a 87,5%, de acordo com cada domínio. Embora uma comparação direta não seja possível, pode-se ressaltar que em nossos resultados obtidos com a avaliação de 80% da base para teste e 20% para teste alcançamos resultados de macro F1-Score entre 43% a 67% (Tabela 8). Assim, embora em contextos diferentes, nossos resultados nesse caso são compatíveis com os apresentados pelos autores, ressaltando que no presente trabalho são oito classes de intenções, ao invés de apenas duas.

Nossa contribuição alcançada é útil para apoiar a execução de mecanismos de recuperação de informação que exploram a anotação de elementos sobre intenção em seu funcionamento. Os resultados no nosso trabalho também servem de base para o desenvolvimento futuro de soluções que explorem outras técnicas de aprendizado de máquina no contexto de análise de intenções no domínio da agricultura.

5. Conclusão

Atividades relacionadas à ciência produzem conhecimento técnico e científico vasto e complexo. Domínios como agricultura sustentável, por exemplo, exigem a concepção de indicadores, que por sua vez, demanda intensiva atividade humana de comunicação e análise textual de diferentes fontes apoiada por sistemas de software. Nesses contextos, a detecção automatizada de intenções em sentenças em texto livre possui um papel chave, pois as anotações explícitas das intenções podem servir como metadados para mecanismos de recuperação, compartilhamento e análise de informações. Neste artigo, conduzimos um estudo para averiguar o uso de técnicas supervisionadas para o reconhecimento de oito classes de ilocução. Construímos uma base de dados experimental considerando sentenças obtidas de textos na área agrícola para análise da tarefa nesse domínio. Nosso estudo identificou que é possível a aplicação de modelos de linguagem no Português que codificam vetorialmente palavras e

técnicas de aprendizado de máquina para a classificação do texto nas diferentes classes de ilocução. Encontramos as técnicas mais efetivas considerando diversificadas configurações experimentais. Nossos resultados alcançados podem ser essenciais para apoiar análise textual no domínio explorado. Trabalhos futuros envolvem expandir e aprimorar nossa base de dados para treinamento e teste. Planejamos igualmente estudar a aplicabilidade de outras técnicas de transferência de conhecimento. Pretende-se também aplicar técnicas não supervisionadas (sem anotação) para pré-treinamento, seguida da aplicação de técnicas supervisionadas, em uma abordagem “semi-supervisionada”. Por fim, planejamos treinar uma rede CNN própria com um conjunto de dados expandidos e propor novas técnicas e configurações de aprendizado profundo baseados no BERT.

Agradecimentos

O presente trabalho foi realizado com apoio do Ministério da Ciência, Tecnologia e Inovações, programa PCI/CNPq #201702325-5. As opiniões, hipóteses e conclusões ou recomendações expressas neste material são de responsabilidade dos autores e não necessariamente refletem a visão das agências financiadoras.

Contribuições dos autores

Daniel Felix de Brito: design da solução, implementação do protótipo, preparação e pré-processamento de dados, treinamento e testes de aprendizado de máquina, redação - rascunho original

Jarbas Lopes Cardoso Júnior: conceituação, preparação e pré-processamento de dados, redação - revisão e edição, supervisão

Júlio Cesar dos Reis: conceituação, redação - rascunho original, redação - revisão e edição

Guilherme Ruppert: design da solução, implementação do protótipo, redação - revisão e edição

Rodrigo Bonacin: conceituação, design de solução, preparação e pré-processamento de dados, treinamento e testes de aprendizado de máquina, redação - rascunho original, redação - revisão e edição, administração do projeto

References

- [1] BONACIN, R. et al. Exploring intentions on electronic health records retrieval. studies with collaborative scenarios. *Ingénierie des Systèmes d'Inf.*, v. 23, n. 2, p. 111–135, 2018. Disponível em: <<https://doi.org/10.3166/isi.23.2.111-135>>.
- [2] IGAWA, R. A. et al. Recognition on Online Social Network by user's writing style. *iSys – Revista Brasileira de Sistemas de Informação*, v. 8, n. 3, p. 64–85, 2015. ISSN 1984-2902.
- [3] MENDONÇA, R. Resende de et al. A framework for detecting intentions of criminal acts in social media: A case study on twitter. *Information*, v. 11, n. 3, 2020. ISSN 2078-2489. Disponível em: <<https://www.mdpi.com/2078-2489/11/3/154>>.
- [4] MORAIS, J. I. de et al. A Multi-label Classification System to Distinguish among Fake, Satirical, Objective and Legitimate News in Brazilian Portuguese. *iSys – Revista Brasileira de Sistemas de Informação*, v. 13, n. 4, p. 126–149, 2020. ISSN 1984-2902.
- [5] RAVI, K.; RAVI, V. A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. *Knowledge-Based Systems*, v. 89, p. 14 – 46, 2015. ISSN 0950-7051. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0950705115002336>>.
- [6] SEARLE, J. R. A classification of illocutionary acts. *Language in society*, Cambridge University Press, v. 5, n. 1, p. 1–23, 1976.
- [7] LIU, K.; LI, W. *Organisational semiotics for business informatics*. [S.l.]: Routledge, 2014.
- [8] STAMPER, R. K. Organisational semiotics: Informatics without the computer? In: *Information, organisation and technology*. [S.l.]: Springer, 2001. p. 115–171.
- [9] LIU, K. *Semiotics in Information Systems Engineering*. [S.l.]: Cambridge University Press, 2000. ISBN 9781139425537.
- [10] BISHOP, C. M. *Pattern recognition and machine learning*. [S.l.]: springer, 2006.
- [11] BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001.
- [12] HARTMANN, N. et al. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. *arXiv preprint arXiv:1708.06025*, 2017.
- [13] MIKOLOV, T. et al. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [14] LING, W. et al. Two/too simple adaptations of word2vec for syntax problems. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. [S.l.: s.n.], 2015. p. 1299–1304.
- [15] BOJANOWSKI, P. et al. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, MIT Press, v. 5, p. 135–146, 2017.
- [16] PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. [S.l.: s.n.], 2014. p. 1532–1543.
- [17] PEIRCE, C. S. *Collected Papers*. Cambridge: Harvard University Press, 1931–1935.

- [18] AUSTIN, J. *How to Do Things with Words?* [S.l.]: Clarendon Press, London, 1962.
- [19] POESIO, M.; TRAUM, D. R. Conversational actions and discourse situations. *Computational Intelligence*, v. 13, n. 3, p. 309–347, 1997. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1111/0824-7935.00042>>.
- [20] SEARLE, J. R. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge, London: Cambridge University Press, 1969.
- [21] D'ANDRADE, R. G.; WISH, M. Speech act theory in quantitative research on interpersonal behavior. *Discourse Processes*, Routledge, v. 8, n. 2, p. 229–259, 1985. Disponível em: <<https://doi.org/10.1080/01638538509544615>>.
- [22] OLNEY, A. et al. Utterance classification in autotutor. In: *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing - Volume 2*. USA: Association for Computational Linguistics, 2003. (HLT-NAACL-EDUC '03), p. 1–8. Disponível em: <<https://doi.org/10.3115/1118894.1118895>>.
- [23] STILES, W. B. *Describing talk: A taxonomy of verbal response modes*. [S.l.]: Sage Publications Newbury Park, CA, 1992.
- [24] ALEXANDERSSON, J. et al. *Dialogue acts in VerbMobil 2*. [S.l.]: DFKI Saarbrücken, 1998.
- [25] HAWIZY, L.; PHILLIPS, I. W.; CONNOLLY, J. H. Intention modeling: a semiotic view. In: *Proceedings of International Conference Applied Computing*. [S.l.: s.n.], 2006. p. 478–482.
- [26] MOOR, A. D. Language/action meets organisational semiotics: Situating conversations with norms. *Information Systems Frontiers*, Springer, v. 4, n. 3, p. 257–272, 2002.
- [27] BONACIN., R. et al. Interacting with dynamic social knowledge - revealing challenges through an analysis of pragmatic aspects of problem solving. In: *INSTICC. Proceedings of the 14th International Conference on Enterprise Information Systems - Volume 3: ICEIS*. [S.l.]: SciTePress, 2012. p. 54–63. ISBN 978-989-8565-12-9. ISSN 2184-4992.
- [28] REIS, J. C. dos et al. Intenticons: Participatory selection of emoticons for communication of intentions. *Computers in Human Behavior*, v. 85, p. 146–162, 2018. ISSN 0747-5632. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0747563218301511>>.
- [29] REIS, J. C. dos et al. Design of Interactive Mechanisms to Support the Communication of Users' Intentions. *Interacting with Computers*, v. 30, n. 4, p. 315–335, 07 2018. ISSN 0953-5438. Disponível em: <<https://doi.org/10.1093/iwc/iwy013>>.
- [30] HWANG, M.; KIM, P.; CHOI, D. Information retrieval techniques to grasp user intention in pervasive computing environment. In: *2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. [S.l.: s.n.], 2011. p. 186–191.
- [31] GUPTA, V.; GARG, N.; GUPTA, T. Search bot: Search intention based filtering using decision tree based technique. In: *2012 Third International Conference on Intelligent Systems Modelling and Simulation*. [S.l.: s.n.], 2012. p. 49–54. ISSN 2166-0662.
- [32] MENDOZA, M.; BAEZA-YATES, R. A web search analysis considering the intention behind queries. In: *2008 Latin American Web Conference*. [S.l.: s.n.], 2008. p. 66–74.
- [33] NOOR, S.; MARTINEZ, K. Using social data as context for making recommendations: An ontology based approach. In: *Proceedings of the 1st Workshop on Context, Information and Ontologies*. New York, NY, USA: ACM, 2009. (CIAO '09), p. 7:1–7:8. ISBN 978-1-60558-528-4.
- [34] ZINGLÉ, H. Modelling knowledge with zdoc for the purposes of information retrieval. In: _____. *Proceedings of 19th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2006, Annecy, France, June 27-30, 2006. Proceedings*. [S.l.]: Springer Berlin Heidelberg, 2006. p. 1053–1058. ISBN 978-3-540-35454-3.
- [35] ASAI, H.; YAMANA, H. Intelligent ink annotation framework that uses user's intention in electronic document annotation. In: *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*. New York, NY, USA: ACM, 2014. (ITS '14), p. 333–338. ISBN 978-1-4503-2587-5.
- [36] GÓMEZ, M. Montes-y; GELBUKH, A. F.; LÓPEZ-LÓPEZ, A. Document title patterns in information retrieval. In: _____. *Text, Speech and Dialogue: Second International Workshop, TSD'99 Plzen, Czech Republic, September 13–17, 1999 Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999. p. 372–375. ISBN 978-3-540-48239-0.
- [37] MUSTAKIM, M.; WARDOYO, R.; MUSTOFA, K. Review of intent diversity in information retrieval: Approaches, models and trends. *Journal of Information Technology and Computer Science*, v. 3, n. 2, p. 132–145, 2018.
- [38] TEH, P. L.; CHENG, C.-B.; CHEE, W. M. Identifying and Categorising Profane Words in Hate Speech. In: *Proceedings of the 2nd International Conference on Compute and Data Analysis - ICCDA 2018*. New York: ACM Press, 2018. (ICCCA 2018), p. 65–69. ISBN 9781450363594.
- [39] AGARWAL, S.; SUREKA, A. But I did not Mean It!—Intent Classification of Racist Posts on Tumblr. In: *2016 European Intelligence and Security Informatics Conference (EISIC)*. [S.l.]: IEEE, 2016. p. 124–127. ISBN 978-1-5090-2857-3.
- [40] Figueroa, A.; Atkinson, J. Ensembling classifiers for detecting user intentions behind web queries. *IEEE Internet Computing*, v. 20, n. 2, p. 8–16, Mar 2016. ISSN 1941-0131.

- [41] COHEN, W.; CARVALHO, V.; MITCHELL, T. Learning to classify email into “speech acts”. In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. [S.l.: s.n.], 2004. p. 309–316.
- [42] LAMPERT, A.; DALE, R.; PARIS, C. Classifying speech acts using verbal response modes. In: CAVEDON, L.; ZUKERMAN, I. (Ed.). *Proceedings of the Australasian Language Technology Workshop, ALTA 2006, Sydney, Australia, November 30-December 1, 2006*. Australasian Language Technology Association, 2006. p. 34–41. Disponível em: <<https://www.aclweb.org/anthology/U06-1007/>>.
- [43] O’SHEA, J.; BANDAR, Z.; CROCKETT, K. A machine learning approach to speech act classification using function words. In: JEDRZEJOWICZ, P. et al. (Ed.). *Agent and Multi-Agent Systems: Technologies and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 82–91. ISBN 978-3-642-13541-5.
- [44] KIM, H.-S.; SEON, C.-N.; SEO, J.-Y. Review of korean speech act classification: machine learning methods. *Journal of Computing Science and Engineering*, Korean Institute of Information Scientists and Engineers, v. 5, n. 4, p. 288–293, 2011.
- [45] MOLDOVAN, C.; RUS, V.; GRAESSER, A. C. Automated speech act classification for online chat. *MAICS*, v. 710, p. 23–29, 2011.
- [46] SAMEI, B. et al. Context-based speech act classification in intelligent tutoring systems. In: TRAUSAN-MATU, S. et al. (Ed.). *Intelligent Tutoring Systems*. Cham: Springer International Publishing, 2014. p. 236–241. ISBN 978-3-319-07221-0.
- [47] HU, Y.; WANG, S. Research on Crime Degree of Internet Speech Based on Machine Learning and Dictionary. In: *Proceedings - 2016 3rd International Conference on Information Science and Control Engineering, ICISCE 2016*. [S.l.: s.n.], 2016. p. 532–537. ISBN 9781509025350.
- [48] KOO, Y. et al. Automatic speech act classification of korean dialogue based on the hierarchical structure of speech act categories. In: *Pacific Asia Conference on Language, Information and Computation (PACLIC 33)*. [S.l.: s.n.], 2019. p. 432–441.
- [49] NIGAM, A.; SAHARE, P.; PANDYA, K. Intent detection and slots prompt in a closed-domain chatbot. *arXiv preprint arXiv:1812.10628*, 2018.
- [50] WANG, Y.; TANG, L.; HE, T. Attention-based cnn-blstm networks for joint intent detection and slot filling. In: *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*. [S.l.]: Springer, 2018. p. 250–261.
- [51] CHEN, Q.; ZHUO, Z.; WANG, W. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*, 2019.
- [52] DEVLIN, J. et al. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. Disponível em: <<http://arxiv.org/abs/1810.04805>>.
- [53] PANDEY, R. et al. Distributional Semantics Approach to Detect Intent in Twitter Conversations on Sexual Assaults. *Proceedings - 2018 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2018*, p. 270–277, 2019.
- [54] KIM, M.; KIM, H. Integrated neural network model for identifying speech acts, predicators, and sentiments of dialogue utterances. *Pattern Recognition Letters*, v. 101, p. 1–5, 2018. ISSN 0167-8655. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0167865517304129>>.
- [55] ALGOTIML, B.; ELMADANY, A.; MAGDY, W. Arabic tweet-act: Speech act recognition for Arabic asynchronous conversations. In: *Proceedings of the Fourth Arabic Natural Language Processing Workshop*. Florence, Italy: Association for Computational Linguistics, 2019. p. 183–191. Disponível em: <<https://www.aclweb.org/anthology/W19-4620>>.
- [56] LIU, C.; XU, X. Amff: A new attention-based multi-feature fusion method for intention recognition. *Knowledge-Based Systems*, v. 233, p. 107525, 2021. ISSN 0950-7051. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0950705121007875>>.
- [57] ARGYRIS, Y. A. et al. Using speech acts to elicit positive emotions for complainants on social media. *Journal of Interactive Marketing*, v. 55, p. 67–80, 2021. ISSN 1094-9968. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1094996821000153>>.
- [58] TANG, D.; QIN, B.; LIU, T. Deep learning for sentiment analysis: successful approaches and future challenges. *WIRES Data Mining and Knowledge Discovery*, v. 5, n. 6, p. 292–303, 2015. Disponível em: <<https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1171>>.
- [59] AGRONÔMICO, I. *O Agrônômico - Boletim Técnico-Informativo do Instituto Agrônômico - Volume 70 - 2018, Série Técnica APTA - ISSN 0365-2726*. Disponível em: <http://www.iac.sp.gov.br/publicacoes/agronomico/pdf/oagronomico_v70.pdf>.
- [60] PELEGRINI, D. F.; BEZERRA, L. M. C.; HASPARYK, R. G. Dinâmica da produção de feijão no brasil: progresso técnico e fragilidades. *Informe Agropecuário, Belo Horizonte*, v. 38, n. 298, p. 84–91, 2017.
- [61] KAWTRAKUL, A. et al. Development of collective intelligence platform for early warning and risk reduction in pest management. *CI@PracticeDay, Campinas, Brazil*, November 2015.
- [62] MIRANDA, E. *Tons de Verde - A sustentabilidade da agricultura brasileira*. [S.l.]: Metalivros, 2018. ISBN 9788582200186.
- [63] WHITMEE, S. et al. Safeguarding human health in the anthropocene epoch: report of the rockefeller

- foundation–lancet commission on planetary health. *The Lancet*, Elsevier, v. 386, n. 10007, p. 1973–2028, 2015.
- [64] MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2013. p. 3111–3119.
- [65] CHAWLA, N. V. et al. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, v. 16, p. 321–357, 2002.
- [66] LEMAÎTRE, G.; NOGUEIRA, F.; ARIDAS, C. K. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, v. 18, n. 17, p. 1–5, 2017. Disponível em: <<http://jmlr.org/papers/v18/16-365.html>>.
- [67] LORENA, A. C.; CARVALHO, A. C. de. Uma introdução às support vector machines. *Revista de Informática Teórica e Aplicada*, v. 14, n. 2, p. 43–67, 2007.
- [68] OSHIRO, T. M.; PEREZ, P. S.; BARANAUSKAS, J. A. How many trees in a random forest? In: PERNER, P. (Ed.). *Machine Learning and Data Mining in Pattern Recognition*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 154–168. ISBN 978-3-642-31537-4.
- [69] SHIN, H.-C. et al. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, IEEE, v. 35, n. 5, p. 1285–1298, 2016.
- [70] REZENDE, E. et al. Malicious software classification using transfer learning of resnet-50 deep neural network. In: *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. [S.l.: s.n.], 2017. p. 1011–1014.
- [71] REZENDE, E. et al. Malicious software classification using vgg16 deep neural network’s bottleneck features. In: LATIFI, S. (Ed.). *Information Technology - New Generations*. Cham: Springer International Publishing, 2018. p. 51–59. ISBN 978-3-319-77028-4.
- [72] WEISS, K.; KHOSHGOFTAAR, T. M.; WANG, D. A survey of transfer learning. *Journal of Big data*, SpringerOpen, v. 3, n. 1, p. 9, 2016.
- [73] HUH, M.; AGRAWAL, P.; EFROS, A. A. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016.
- [74] REZENDE, E. R. D. et al. Exposing computer generated images by using deep convolutional neural networks. *Signal Processing: Image Communication*, Elsevier, v. 66, p. 113–126, 2018.
- [75] DERCZYNSKI, L. Complementarity, f-score, and nlp evaluation. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*. [S.l.: s.n.], 2016. p. 261–266.
- [76] FONSECA, E. R.; ROSA, J. L. G.; ALUÍSIO, S. M. Evaluating word embeddings and a revised corpus for part-of-speech tagging in portuguese. *Journal of the Brazilian Computer Society*, Springer, v. 21, n. 1, p. 2, 2015.
- [77] CER, D. et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.
- [78] REIS, J. C. dos; BONACIN, R.; BARANAUSKAS, M. C. C. Recognizing intentions in free text messages: Studies with portuguese language. In: *IEEE. 2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. [S.l.], 2017. p. 302–307.
- [79] MECHULAM, N. et al. Building dynamic lexicons for sentiment analysis. *Inteligencia Artificial*, v. 22, n. 64, p. 1–13, 2019.

Apêndice A

Tabela 12. Número de ocorrências por pessoa e total ocorrências em que uma combinação *Word Embedding*/Dimensão/Estrutura/Classificador obteve macro F1-Score maiores do que a média somada com o desvio-padrão, calculadas com base nos 5% dos maiores macro F1-Score da validação cruzada.

Word Embedding / Dimensão / Estrutura / Classificador	Pessoa			Total
	A	B	C	
FastText / 300 / Skip-Gram / ANN	13	1	3	17
FastText / 300 / Skip-Gram / RF	1	0	1	2
FastText / 50 / Skip-Gram / RF	0	0	1	1
FastText / 600 / Skip-Gram / ANN	12	3	0	15
FastText / 600 / Skip-Gram / RF	0	0	1	1
Glove / 100 / RF	0	0	2	2
Glove / 300 / ANN	8	0	1	9
Glove / 300 / RF	0	0	4	4
Glove / 50 / RF	0	0	2	2
Glove / 600 / ANN	6	0	0	6
Glove / 600 / RF	0	0	4	4
Wang2Vec / 300 / Skip-Gram / ANN	1	0	1	2
Wang2Vec / 600 / CBOW / ANN	0	10	0	10
Wang2Vec / 600 / Skip-Gram-Gram / ANN	0	0	4	4
Wang2Vec / 600 / Skip-Gram-Gram / RF	0	0	1	1
Word2Vec / 300 / Skip-Gram / ANN	0	0	1	1
Word2Vec / 300 / Skip-Gram / RF	0	0	4	4
Word2Vec / 600 / Skip-Gram / ANN	0	1	0	1
Word2Vec / 600 / Skip-Gram / RF	0	0	1	1

Tabela 13. Número de ocorrências por pessoa e total ocorrências em que uma combinação *Word Embedding*/Dimensão/Estrutura/Classificador obteve macro F1-Score maiores do que a média somada com o desvio-padrão, calculadas com base nos 5% dos maiores macro F1-Score da avaliação de 80% para treino e 20% para teste.

Word Embedding / Dimensão / Estrutura / Classificador	Pessoa			Total
	A	B	C	
FastText / 100 / CBOW / ANN	0	2	0	2
FastText / 100 / CBOW / RF	0	1	0	1
FastText / 100 / Skip-Gram / ANN	0	5	1	6
FastText / 100 / Skip-Gram / RF	0	0	1	1
FastText / 100 / Skip-Gram / SVM	0	1	0	1
FastText / 300 / CBOW / ANN	0	1	1	2
FastText / 300 / CBOW / RF	0	0	1	1
FastText / 300 / Skip-Gram / ANN	13	17	4	34
FastText / 300 / Skip-Gram / RF	2	0	3	5
FastText / 300 / Skip-Gram / SVM	1	0	0	1
FastText / 50 / CBOW / RF	0	1	1	2
FastText / 50 / Skip-Gram / RF	1	0	6	7
FastText / 50 / Skip-Gram / SVM	0	0	1	1
FastText / 600 / CBOW / RF	0	1	0	1
FastText / 600 / Skip-Gram / ANN	33	4	2	39
FastText / 600 / Skip-Gram / RF	0	2	6	8
Glove / 100 / ANN	0	1	2	3
Glove / 100 / RF	1	0	3	4
Glove / 300 / ANN	1	1	3	5
Glove / 300 / RF	1	2	5	8
Glove / 50 / RF	0	2	2	4
Glove / 600 / ANN	7	1	5	13
Glove / 600 / RF	0	0	2	2
Wang2Vec / 100 / CBOW / ANN	0	2	1	3
Wang2Vec / 100 / CBOW / RF	0	0	1	1
Wang2Vec / 100 / CBOW / SVM	0	1	0	1
Wang2Vec / 100 / Skip-Gram / RF	0	0	1	1
Wang2Vec / 300 / CBOW / ANN	1	5	17	23
Wang2Vec / 300 / CBOW / RF	3	2	2	7
Wang2Vec / 300 / Skip-Gram / ANN	3	1	7	11
Wang2Vec / 300 / Skip-Gram / RF	1	0	3	4
Wang2Vec / 300 / Skip-Gram / SVM	1	0	0	1
Wang2Vec / 50 / CBOW / RF	0	2	0	2
Wang2Vec / 50 / Skip-Gram / RF	1	1	2	4
Wang2Vec / 600 / CBOW / ANN	14	31	1	46
Wang2Vec / 600 / CBOW / RF	2	2	1	5
Wang2Vec / 600 / Skip-Gram / ANN	1	2	11	14
Wang2Vec / 600 / Skip-Gram / RF	0	0	1	1
Word2Vec / 100 / CBOW / RF	0	1	1	2
Word2Vec / 100 / Skip-Gram / ANN	0	24	1	25
Word2Vec / 100 / Skip-Gram / RF	0	0	9	9
Word2Vec / 100 / Skip-Gram / SVM	0	2	0	2
Word2Vec / 300 / CBOW / ANN	4	0	2	6
Word2Vec / 300 / CBOW / RF	1	0	0	1
Word2Vec / 300 / Skip-Gram / ANN	0	0	7	7
Word2Vec / 300 / Skip-Gram / RF	1	2	6	9
Word2Vec / 300 / Skip-Gram / SVM	0	0	2	2
Word2Vec / 50 / CBOW / ANN	0	0	9	9

Table 13 continued from previous page

Word Embedding / Dimensão / Estrutura / Classificador	Pessoa			Total
	A	B	C	
Word2Vec / 50 / CBOW / SVM	0	0	1	1
Word2Vec / 50 / Skip-Gram / ANN	0	0	5	5
Word2Vec / 50 / Skip-Gram / RF	0	1	5	6
Word2Vec / 50 / Skip-Gram / SVM	0	1	0	1
Word2Vec / 600 / CBOW / ANN	4	1	3	8
Word2Vec / 600 / CBOW / RF	0	0	8	8
Word2Vec / 600 / Skip-Gram / ANN	3	1	4	8
Word2Vec / 600 / Skip-Gram / RF	0	1	3	4
Word2Vec / 600 / Skip-Gram / SVM	0	0	1	1