Classificação de Contexto para Processamento da Linguagem Natural Baseado em Representação Vetorial de Palavras e no Agrupamento por K-Means

T.B.N. Silveira[1,2], H.S. Lopes[1], A.E. Lazzaretti[1], D.P. Araújo[2], C.F. Valério[2]

[1]Pós-Graduação em Engenharia Elétrica e Informática Industrial – CPGEI
Universidade Tecnológica Federal do Paraná – UTFPR
Curitiba, Brazil

[2]Divisão de Tecnologia – Telefônica Brasil
Curitiba, Brazil

tiago.silveira@telefonica.com, heitor.lopes@utfpr.edu.br

Abstract. *In this paper we propose the association of the skip-gram algorithm, which provides a vector representation of a word set, with the non-supervised clustering algorithm k-means. This methodology has presented an effectiveness of 75.3% on classifying context in natural language written documents. Besides detailing its architecture, in this paper we also seek to discuss the advantages and limitations of the proposed methodology in a long-term perspective, based both in the empirical methods for natural language processing as well as in the language descriptive models.*

Resumo. *Este trabalho traz como proposta a associação do algoritmo skip-gram, para representação vetorial de palavras, com o algoritmo de agrupamento não-supervisionado k-means. Ao longo do texto, além de detalhar esta metodologia e demonstrar sua eficácia de 75.3% na classificação de contexto em documentos escritos utilizando linguagem natural, buscamos discutir as vantagens e limitações do método proposto em uma perspectiva de longo prazo, inserida tanto na metodologia empírica de processamento da linguagem natural quanto na busca de modelos descritivos para a linguagem.*

1. Introdução

A popularização das redes sociais que teve início na última década tem sido acompanhada por uma crescente demanda de atendimento imediato por milhões de usuários de serviços digitais – sejam estes de consumo ou de comércio eletrônico – levando a um recente interesse em tecnologias para o processamento da linguagem natural (NLP – *Natural Language Processing*, em inglês), com destaque para análise de sentimentos e opiniões [Liu 2012] e classificação de informação não-estruturada [Kreimeyer et al. 2017], bem como aplicações decorrentes destas tais como inteligência artificial conversacional [Lee 2017] e resposta automática a questões em linguagem natural [GARTNER 2016].

Entretanto, embora inúmeras técnicas tenham sido desenvolvidas para tais aplicações – uma extensa revisão de métodos e algoritmos atualmente empregados pode ser encontrada em [Sun et al. 2017], [Kreimeyer et al. 2017], [Kumar e Ravi 2016] –, há

ainda limitações e desafios que impedem a aplicação efetiva de NLP na análise de sentimentos e na classificação de informação não-estruturada.

Em [Sun et al. 2017], estas limitações são endereçadas ao (i) número reduzido e limitado ao idioma inglês de corpus (i.e. bases de texto) com anotações para as classificações desejadas, sejam elas de sentimentos ou de categorias; (ii) aos erros cumulativos provenientes de tarefas de pré-processamento e (iii) à ausência de desenvolvimento significativo de *deep learning* para NLP.

Já de acordo com [Cambria e White 2014], estas limitações residem no fato de que a maioria dos métodos de NLP é baseada na representação sintática do texto, associada à frequência de ocorrência das palavras. Deste modo, tais algoritmos poderiam processar apenas a informação já conhecida, sem qualquer possibilidade de representação abstrata. Isto somado à incapacidade de perceber o sentido afetivo (*sentics*) e a ausência do conhecimento de senso comum, tornam os atuais algoritmos incapazes de processar a linguagem natural em domínios abertos (i.e. em aplicações sem delimitação de tema ou assunto), com percepção de contexto e de intenção.

Ainda em [Cambria e White 2014], uma perspectiva de evolução das técnicas de NLP é fundamentada nos paradigmas da sintática; semântica; e pragmática – esta última correspondente às capacidades simbólicas de alto nível [Dyer 1995] que possibilitariam, dentre outras tarefas, a percepção generalizada de contexto e a desambiguação de sentido. Neste cenário, métodos correspondentes à curva semântica já seriam capazes de trabalhar com contextos e significados, porém ainda limitados a ontologias (abordagem simbólica) ou a características intrínsecas ao documento (sem generalização). A transição entre os paradigmas, conforme apontada por [Cambria e White 2014] através de curvas de evolução e reproduzida na Figura 1, culminaria em modelos computacionais capazes de representar a informação tal qual humanos. Uma vez obtidos tais modelos, estes poderiam ser construídos associando-se redes neurais, conjuntos *fuzzy* e computação evolucionária para aprendizagem online.

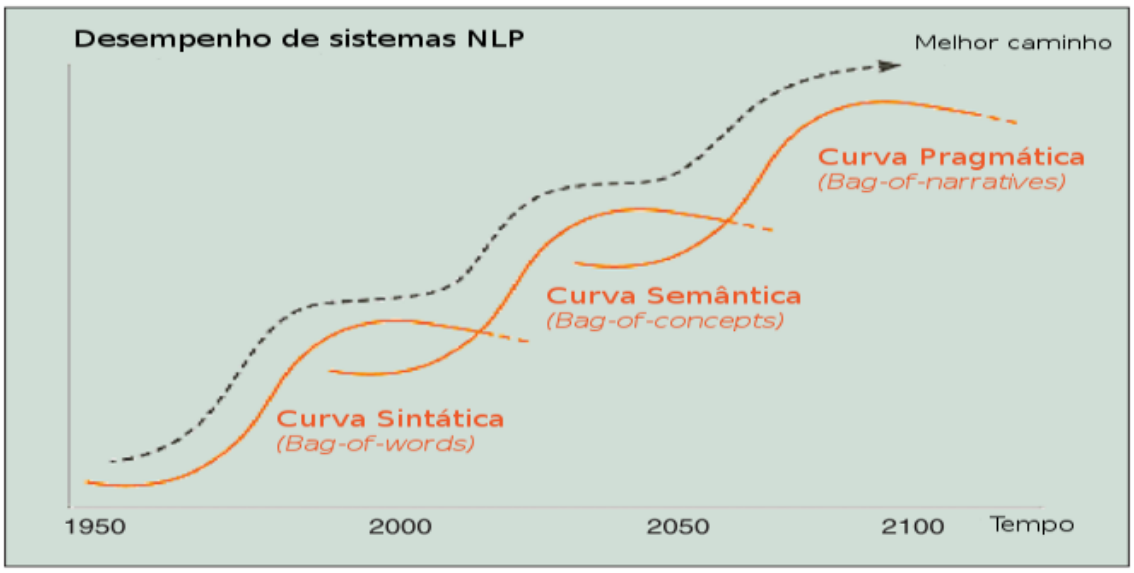

Figura 1. Evolução da pesquisa em NLP através de diferentes curvas, proposta por [Cambria e White 2014].

Tomando como objetivo as aplicações de interesse anteriormente citadas, este trabalho busca contribuir para o desenvolvimento das técnicas de NLP ao propor uma solução para classificação de contexto associando a abordagem empírica – empregada em [Palmer et al. 2017] e [Su et al. 2016], entre outros, e fundamentada na inferência indutiva através de exemplos e observações – porém contextualizando seu desenvolvimento nos paradigmas apresentados por [Cambria e White 2014], cuja expectativa é a obtenção de um modelo generativo da linguagem. Para tanto, a estratégia utilizada neste trabalho consiste em representar vetorialmente as palavras de um grande conjunto de texto em linguagem natural, decorrente de reclamações referentes a serviços de telecomunicações, e classificar o contexto ao qual estão inseridas (e.g. financeiro, produtos, qualidade ou defeito) a partir de um agrupamento não supervisionado.

Sendo assim, a Seção 2 traz a definição de contexto a qual será utilizada neste trabalho. A Seção 3 apresenta com detalhes a técnica de representação vetorial de palavras, com especial atenção ao modelo *skip-gram* desenvolvido em [Mikolov et al. 2013b], no qual a arquitetura do classificador, apresentado na Seção 4, é embasada. A discussão dos resultados obtidos é elaborada na Seção 5, seguida da conclusão e das perspectivas de trabalhos futuros na Seção 6.

2. Definição de contexto em NLP

Embora o significado de contexto possa ser tacitamente obtido, sua formalização teve início no surgimento da linguística, quando [De Saussure 1972] definiu a linguagem como um sistema cujos termos são interdependentes e cujo valor de um é atribuído em relação aos outros. Na computação, o termo passou a ser empregado com o surgimento da computação ubíqua e de suas aplicações sensíveis ao contexto [Weiser 1999], este então definido como qualquer informação utilizada para caracterizar a situação de um determinado agente [Dey 2001].

Ambas as conceituações acima contribuem com o entendimento da definição apresentada em [Goldberg e Levy 2014], a qual é considerada neste estudo: dado um corpus T contendo uma sequência de N palavras, $T = \{w_i \mid 0 \leq i < N\}$, o *contexto* c_{w_t} para uma dada ocorrência da palavra w_t será definido pelas palavras vizinhas a w_t. Considerando-se uma janela de tamanho $d = 2j$, onde j é um parâmetro arbitrado indicando o número de palavras vizinhas consideradas, o contexto $c_{w_t}(w_t)$ é dado por:

$$c_{w_t}(w_t) = w_{t-j}, ..., w_{t-1}, w_{t+1}, ..., w_{t+j}. \quad (1)$$

Partindo desta representação – e considerando a hipótese distribucional [Harris 1981] – uma vez que duas palavras distintas w_n e w_m possuam os mesmos contextos ($c(w_n) \equiv c(w_m)$), é possível inferir que seus significados sejam também equivalentes.

3. *Word embeddings*: representação vetorial da linguagem

Inicialmente empregada por [Hinton et al. 1986], a representação de palavras em um espaço vetorial, denominado *word embedding*, tem sido aplicada de distintas maneiras tanto em modelos estatísticos baseados em frequência, tais como a análise semântica latente [Hofmann 1999] e *N-gram* [Brants et al. 2007], como também em modelos não

lineares baseados em representação distribuída [Bengio et al. 2003]. Entretanto, foi somente a partir dos modelos log-lineares propostos por [Mikolov et al. 2013a] que a representação vetorial de um grande volume de texto pôde ser efetivamente construída.

Em uma descrição simplificada, os modelos propostos em [Mikolov et al. 2013a] possibilitam o treinamento de um *word embedding*, a partir de um grande volume de texto, para prever uma palavra a partir de seu contexto (*continuous bag of words*, CBOW) ou, a partir de uma determinada palavra, prever o contexto no qual ela está inserida (*Skip-gram*), conforme arquiteturas dispostas na Figura 2. Além de possibilitar aplicações distintas, para grandes conjuntos de dados a arquitetura *skip-gram* se prova mais eficiente e com menor custo computacional, sendo portanto o modelo adotado para este estudo.

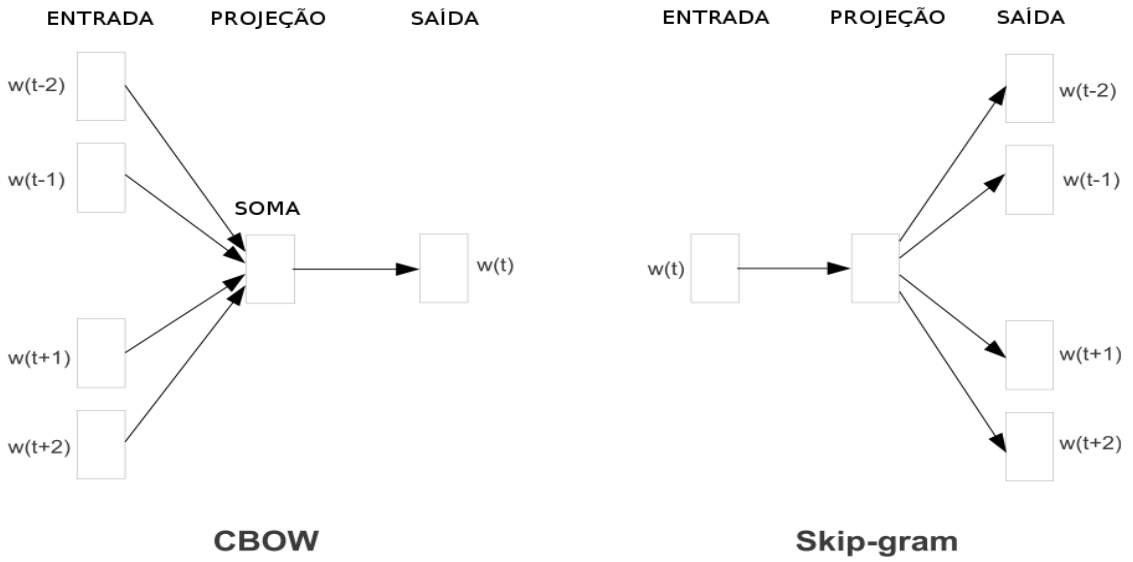

Figura 2. Arquiteturas dos modelos (i) *continuous bag of words*, CBOW, à esquerda; e (ii) *skip-gram*, à direita. Adaptado de [Mikolov et al. 2013a].

Conforme observado na Figura 2, o modelo *skip-gram* pode ser entendido como um *multi-layer perceptron* (MLP) onde a camada de entrada é um vetor representando a palavra w_t, na forma *one-for-K* [Pedregosa et al. 2011]. Se considerarmos o MLP como um classificador de regressão logística, e tomarmos a função *softmax* para a conversão de scores em probabilidades, a camada de saída conterá a probabilidade de que uma determinada palavra w pertença ao contexto de $c(w_t)$. Uma vez treinado o MLP para as N palavras selecionadas e seus respectivos contextos, a camada oculta da rede neural (*camada de projeção*) consistirá no *word embedding* W do respectivo corpus T. O número de neurônios da camada de projeção será determinado em função de j (definido na Seção 2), resultando em uma representação vetorial de dimensão $N \times d$.

Em relação ao treinamento do *word embeddings*, utilizando a definição adotada na Seção 2 e considerando D o conjunto de todos os pares $(w_t, c(w_t))$ extraídos de T, o objetivo do modelo *skip-gram* é o de determinar θ de modo a maximizar a seguinte função distribuição de probabilidade [Goldberg e Levy 2014]:

$$argmax_\theta \prod_{(w,c)\in D} p(c|w;\theta). \quad (2)$$

Em sua implementação básica, a parametrização da Equação 2 é dada através da função *softmax*, de modo que $p(c|w;\theta)$ seja definido como:

$$p(c|w;\theta) = \frac{\exp^{v_c \cdot v_w}}{\sum_{c' \in C} \exp^{v_{c'} \cdot v_w}}, \tag{3}$$

Onde C é o conjunto de todos os contextos possíveis, v_w é a representação vetorial das palavras e v_c é a representação vetorial dos contextos. Os parâmetros correspondentes a θ são v_{c_i}, v_{w_j}, para $w \in W$, $c \in C$, considerando $0 \leq i < d-1$, e $0 \leq j < N-1$.

A partir desta formulação, a computação da distribuição de probabilidade $p(c|w)$ tem um custo proporcional à dimensão do *word embedding*, o que a torna inviável para um grande conjunto de dados. Para tanto, são utilizadas estratégias como (i) *hierarchical softmax*; (ii) *noise contrastive estimation* (NCE); (iii) amostragem negativa (NEG); e (iv) subamostragem de palavras frequentes. A implementação utilizada neste estudo utiliza amostragem negativa, cuja formalização é encontrada em [Mikolov et al. 2013b].

4. Arquitetura proposta

O classificador de contexto proposto neste trabalho utiliza como base uma implementação em código aberto do modelo *skip-gram*, denominado *word2vec*[1], e construído sobre o *framework* TensorFlow™ [2] – uma plataforma *open source* para computação numérica baseado em fluxo de dados entre grafos, utilizada em aplicações de *machine learning* e *deep learning*. A Figura 3 apresenta a arquitetura completa do classificador proposto, segmentada em etapas sequenciais cujo detalhamento é dado nas subseções a seguir.

4.1. Aquisição dos dados

Conforme discutido na Seção 1, as técnicas de NLP têm se mostrado mais efetivas quando aplicadas a bases de domínio fechado, isto é, de temas mais específicos e menos generalistas. Desta forma, optamos por construir uma base de dados em linguagem natural através das informações registradas publicamente no portal ReclameAqui[3] relacionadas a produtos e serviços de telecomunicações, com o objetivo de classificar o contexto de cada reclamação nas categorias: (i) *financeiro*; (ii) *produtos* ou *serviços*; (iii) *qualidade*; ou (iv) *defeito técnico*. A busca dos registros na Internet possibilitou a construção de um banco de dados em MongoDB com mais de 70.000 registros referentes ao ano de 2016, o equivalente a mais de 11 milhões de palavras.

4.2. Pré-processamento da base de dados

Uma vez construída a base de dados, algumas manipulações foram realizadas por questões legais, bem como para garantir um melhor desempenho: (i) normalização do corpus na codificação ISO-8859-1; (ii) substituição de quaisquer referências a pessoas jurídicas e/ou marcas comerciais pelo termo ”operadora”; (iii) substituição de quaisquer referências a pessoas físicas pelo termo ”cliente”; (iv) transformação de todas as letras do texto em minúsculas; (v) utilizando expressões regulares (*RegEx*), remoção dos endereços *web*, endereços de e-mail, *hashtags* e pontuações.

[1]Disponível em *https://code.google.com/p/word2vec/*.
[2]Disponível em *https://www.tensorflow.org/*
[3]Disponível em *http://reclameaqui.com.br*.

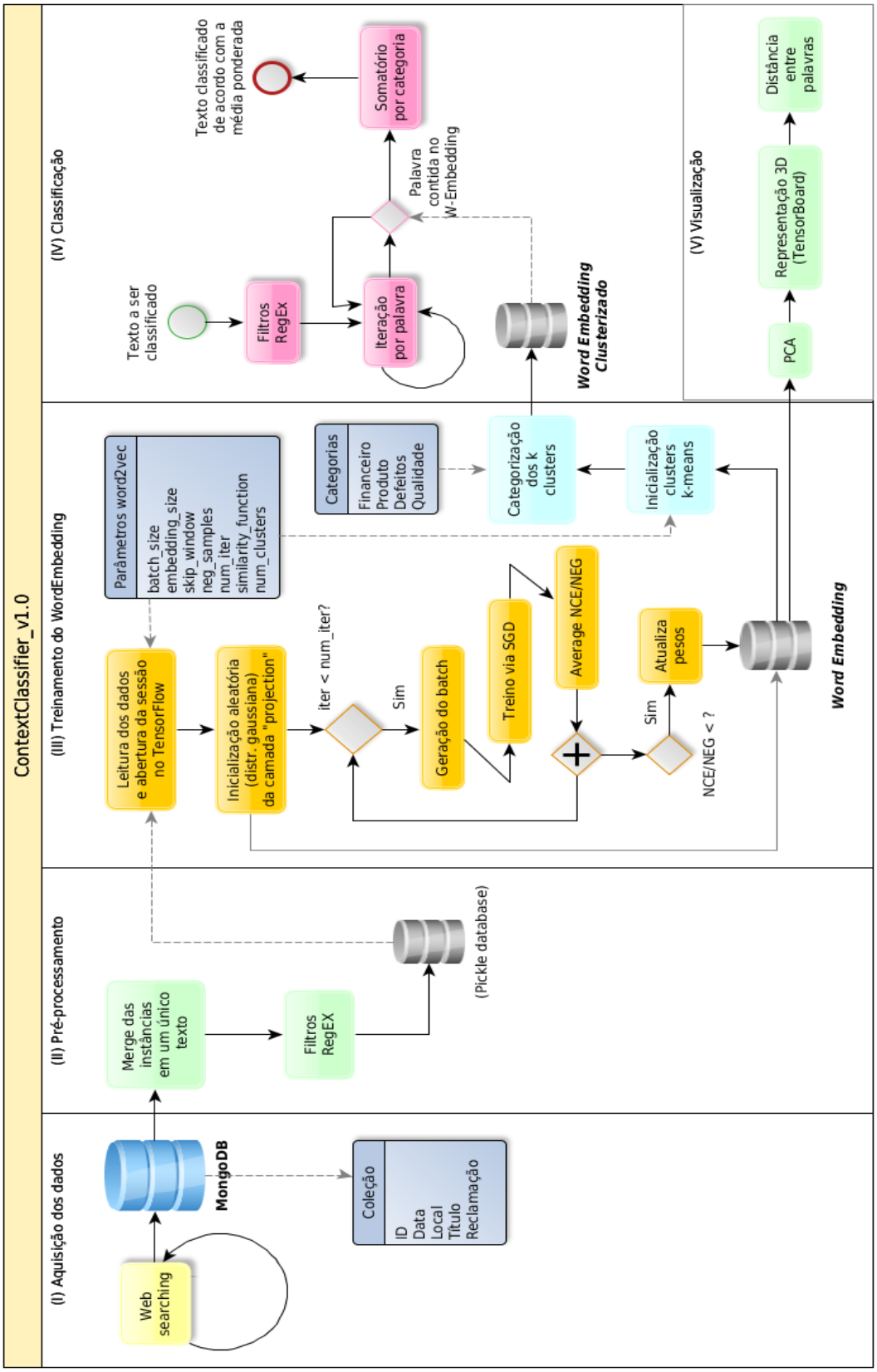

Figura 3. Visão geral da arquitetura do classificador de contexto proposto. Após o treinamento, novas entradas de texto são classificadas pelo processo detalhado na Etapa (IV).

4.3. Treinamento do *word embedding*

O modelo apresentado na Seção 3 foi implementado através do *framework* TensorFlow™ na linguagem Python, cujos módulos estão detalhados na Etapa (III) da Figura 3. Os seguintes parâmetros foram utilizados:

- *batch_size = 128*; número de instâncias em cada lote de treino.
- *embedding_size = 128*; equivalente ao parâmetro $d = 2j$.
- *vocabulary_size = 50.000*; equivalente ao parâmetro N.
- *num_iter = 5.000.000*; número de iterações durante o treinamento.
- *num_clusters = 1.000*; número de *clusters* criados pelo *k-means*.

O algoritmo foi executado utilizando processadores CPU Intel® Core™ i7 e GPU Nvidia® GeForce 940MX e equivalentes. Em tais equipamentos, e com os parâmetros definidos acima, o treinamento dos *word embeddings* teve duração média de 6 horas.

Uma vez finalizado o treinamento do *word embedding*, tem-se uma representação das 50.000 palavras mais frequentes em um espaço vetorial de 128 dimensões. De acordo com o exposto na Seção 3, palavras inseridas no mesmo contexto têm a maior probabilidade de possuírem o mesmo significado e, portanto, terão as menores distâncias entre si no hiperespaço definido pelo *word embedding*. O algoritmo *k-means* [Hotho et al. 2005], portanto, torna-se a opção adequada para o agrupamento não-supervisionado dos pontos próximos e, consequentemente, o agrupamento das palavras que compartilham do mesmo contexto.

4.4. Classificador ponderado

Uma vez agrupados, os *k-clusters* são categorizados pelos rótulos definidos na Seção 4.1 através de inspeção manual, resultando em 44 *clusters* anotados. Dada uma nova entrada em linguagem natural para a qual deseja-se classificar seu contexto, (i) aplica-se a ela o mesmo pré-processamento descrito na Seção 4.2; (ii) em seguida o texto de entrada pré-processado é iterado em cada palavra, verificando se ela está contida no *clusterized word embedding* e, em caso positivo, contabilizando a ocorrência de sua categoria. Ao final, (iii) o contexto atribuído à entrada será aquele(s) de maior ocorrência ponderada em relação às demais classes.

5. Discussão dos resultados

Uma vez adquirida a base de dados, aplicou-se a ela os métodos dispostos nas etapas (II) e (III) da Figura 3, obtendo assim um *word embedding* de 50.000 palavras. A partir deste ponto, aplicando-se os métodos da etapa (V), é possível visualizar a representação vetorial de palavras espacialmente, inferindo intuitivamente a distância e correspondência entre elas.

A Figura 4 mostra a visualização do *word embedding* criado através do *TensorBoard*, módulo do TensorFlow™, com a redução da dimensionalidade de 128 para 3 dimensões através de *PCA – principal component analysis*. Na mesma figura, à direita, são exibidas as ocorrências mais próximas do termo buscado – no caso, ”atendente”. É interessante observar, nesta representação, a capacidade semântica do *word2vec*, uma vez que os erros ortográficos (e.g. ”atentende”, ”atendende”e ”tendente”) possuem distâncias pequenas em relação ao termo correto.

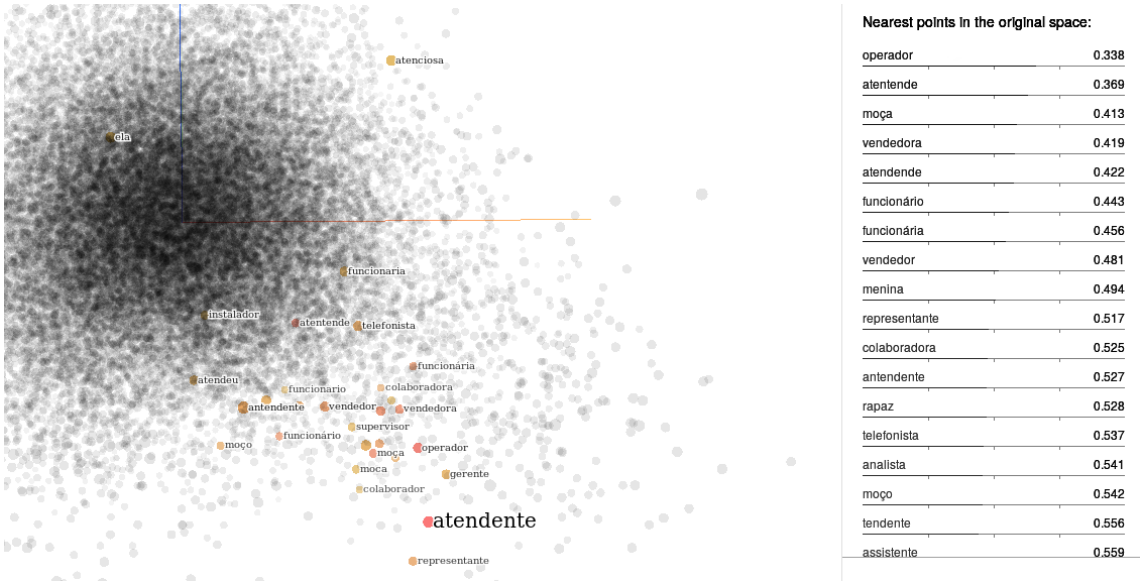

Figura 4. Representação tridimensional do *word embedding*, através da ferramenta *Tensorboard*, com destaque à palavra "atendente"e aos termos mais próximos a ela.

Já a Tabela 1 apresenta a matriz confusão para o classificador implementado na etapa (IV) da Figura 3, aplicado a 150 amostras aleatórias da base de dados, não utilizadas no treinamento do *word embedding* e contendo um número médio de 90 palavras. As linhas representam a classificação dada por humanos, enquanto as colunas indicam a classificação dada pelo algoritmo. Deste modo, a diagonal da tabela representa as categorias classificadas em consonância com a classificação humana, em uma acurácia equivalente a 75.3%. Vale observar que algumas amostras foram classificadas por humanos com um segundo tema predominante, muito embora optou-se por utilizar apenas a primeira classificação dada pelo humano e a de maior peso pelo algoritmo. Tal fato justifica o número de erros na categoria *Qualidade*, cujo texto está relacionado, na maioria das vezes, à descrição do atendimento a determinado produto.

Tabela 1. Matriz confusão para o classificador de contexto aplicado a 150 amostras aleatórias da base de dados.

	Financeiro	**Produtos**	**Qualidade**	**Defeitos**
Financeiro	53	-	-	-
Produtos	4	44	1	7
Qualidade	3	16	2	-
Defeitos	-	5	1	14

6. Conclusão

Pode-se dizer que a principal contribuição deste trabalho é apontar a ingenuidade filosófica, tal como definida por [Teixeira 2011], ao se abordar o problema da linguagem empregando métodos computacionais estritamente empíricos, justificando tal viés a partir das curvas propostas por [Cambria e White 2014]. Conforme este último, nossa ciência e

técnica estão limitadas ao processamento da semântica e atingirão um desempenho satisfatório somente quando possibilitarem a compreensão e o processamento da pragmática.

Deste modo, sustentamos a ideia de que tanto a busca por métodos empíricos para NLP quanto a pesquisa em modelos da linguagem devam ser estabelecidos em um paradigma de transdisciplinaridade, afinal, nas palavras de [Cambria e White 2014], a origem da linguagem humana (e talvez o futuro dela) tem sido taxada como o problema mais difícil da ciência. Ficou evidente, em nossa pesquisa, que as limitações encontradas nos algoritmos aqui discutidos são similares às limitações dos modelos formais debatidos na linguística, tal como em [Thá 2007] e [Santos e Santos 2016], ainda insuficientes para generalizar o processo da linguagem humana.

Por outro lado, abordagens empíricas tal como a arquitetura trazida neste trabalho têm demonstrado resultados satisfatórios, ainda que em aplicações de domínio fechado e escopo limitado. Acreditamos, portanto, que o pavimento para o caminho proposto por [Cambria e White 2014] – para alcançarmos a capacidade, dentre outras, de coompreender através de NLP o contexto em textos de domínio aberto – seja construído via uma abordagem transdisciplinar, onde desenvolvedores abandonem o puro empirismo em seus algoritmos e linguistas passem a igualmente considerar os volumes de dados e recursos técnicos na formulação de seus modelos.

Neste cenário, os próximos passos relativos a este estudo consistem em (i) expandir a base de dados, ampliando o domínio de classificação; (ii) automatizar o processo de categorização dos *clusters*, tornando o método totalmente não-supervisionado; (iii) construir bases de sentenças com contextos rotulados, viabilizando o treinamento de classificadores; e (iv) comparar, através de métricas de desempenho, a técnica proposta com as demais metodologias citadas para classificação de contexto.

Referências

Bengio, Y., Ducharme, R., Vincent, P., e Janvin, C. (2003). A Neural Probabilistic Language Model. *The Journal of Machine Learning Research*, 3:1137–1155.

Brants, T., Popat, A. C., Xu, P., Och, F. J., e Dean, J. (2007). Large Language Models in Machine Translation. *Google, Inc.*, pages 858–867.

Cambria, E., e White, B. (2014). Jumping NLP curves: A review of natural language processing research. *IEEE Computational Intelligence Magazine*, 9(2):48–57.

De Saussure, F. (1972). Cours de linguistique générale. Em *Collection Payothèque*, page 509.

Dey, A. (2001). Understanding and using context. *Personal and ubiquitous computing*, 5(1):4–7.

Dyer, M. G. (1995). Connectionist Natural Language Processing: A Status Report. Em *Computational Architectures Integrating Neural And Symbolic Processes*, pages 389–429. Springer US, Boston, MA.

GARTNER, I. (2016). Gartner's 2016 Hype Cycle for Emerging Technologies Identifies Three Key Trends That Organizations Must Track to Gain Competitive Advantage. Em *GARTNER*.

Goldberg, Y., e Levy, O. (2014). word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. (2):1–5.

Harris, Z. S. (1981). Distributional Structure. Em *Papers on Syntax*, pages 3–22. Springer Netherlands, Dordrecht.

Hinton, G. E., McClelland, J. L., e Rumelhart, D. E. (1986). Distributed representations. Em *Parallel distributed processing: Explorations in the microstructure of cognition*, pages 77—-109.

Hofmann, T. (1999). Probabilistic Latent Semantic Analysis. Em *Proc. UAI*.

Hotho, A., Nürnberger, A., e Paaß, G. (2005). A Brief Survey of Text Mining. *LDV Forum - GLDV Journal for Computational Linguistics and Language Technology*, 20:19–62.

Kreimeyer, K. et al. (2017). Natural language processing systems for capturing and standardizing unstructured clinical information: A systematic review. *Journal of Biomedical Informatics*, 73:14–29.

Kumar, B. S., e Ravi, V. (2016). A survey of the applications of text mining in financial domain. *Knowledge-Based Systems*, 114:128–147.

Lee, A. (2017). Conversational Artificial Intelligence. We Need To Talk About It. - Adrian Lee. Em *GARTNER*.

Liu, B. (2012). Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

Mikolov, T., Chen, K., Corrado, G., e Dean, J. (2013a). Efficient Estimation of Word Representations in Vector Space. pages 1–12.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., e Dean, J. (2013b). Distributed Representations of Words and Phrases and their Compositionality. pages 1–9.

Palmer, M., Hwa, R., e Riedel, S., editors (2017). *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, {EMNLP} 2017, Copenhagen, Denmark, September 9-11, 2017*. Association for Computational Linguistics.

Pedregosa, F. et al. (2011). Scikit-learn: Machine Learning in {P}ython. *Journal of Machine Learning Research*, 12:2825–2830.

Santos, F. S., e Santos, T. S. (2016). Linguística formal e linguística do discurso: continuidades e rupturas teóricas. *Revista Linguística Rio*, 2:31–49.

Su, J., Carreras, X., e Duh, K., editors (2016). *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, {EMNLP} 2016, Austin, Texas, USA, November 1-4, 2016*. The Association for Computational Linguistics.

Sun, S., Luo, C., e Chen, J. (2017). A review of natural language processing techniques for opinion mining systems. *Information Fusion*, 36:10–25.

Teixeira, J. d. F. (2011). *Mente cérebro e cognição*. Vozes.

Thá, F. (2007). *Categorias conceituais da subjetividade*. Annablume.

Weiser, M. (1999). The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):3–11.