PLATAFORMA DE COMPUTAÇÃO COGNITIVA PARA DESENVOLVIMENTO DE APLICAÇÕES COM PROCESSAMENTO DE LINGUAGEM NATURAL

Addam Cauê Peres RAFACHO, Prof. MSc. Guilherme de Cleva FARTO

addamcaue@hotmail.com, guilherme.farto@gmail.com

Instituto Municipal de Ensino Superior de Assis (IMESA)

Fundação Educacional do Município de Assis (FEMA) – Assis/SP (Brasil)

RESUMO: A Computação Cognitiva é uma área de pesquisa que vêm ganhando destaque nos últimos anos por fornecer um ambiente que compreende e aprende sobre a nossas ações e formas de nos comunicar por meio de tecnologias, tais como a de Processamento de Linguagens Naturais (PLN). O uso de PLN pode possibilitar outras subáreas de pesquisa como Deep Learning, Big Data, Redes Neurais, entre outros, de mudar completamente a forma em como interagimos com o ambiente e fornecer uma maior interatividade com o mesmo. Esta pesquisa tem como objetivo explorar os conceitos de Processamento de Linguagem Natural, implementada pela plataforma *Natural Language Toolkit* em Python. Como resultado, este trabalho apresentará um protótipo de plataforma que integra PLN com Python, possibilitando seu uso em diferentes contextos como, por exemplo, jogos e análise de sentimentos.

PALAVRAS-CHAVE: Computação Cognitiva; Processamento de Linguagem Natural; Natural Language Toolkit; Python; Análise de Sentimento

ABSTRACT: Cognitive Computing is a field of research that has been gaining a lot of popularity lately, by providing an environment that comprehends and learns about our actions and means of communication between us through technologies such as Natural Language Processing (NLP). The use of NLP might enable other subareas of research like Deep Learning, Big Data, Neural Networks, among others, to completely change the way we interact with the environment and provide a bigger interactivity with it. This research has as a goal to explore Natural Language Processing concepts, implemented by the functionalities of Natural Language Toolkit platform in Python. As a result, this project will present a platform prototype that integrates NLP with Python, enabling its use in differents contexts like, for example, games and sentiment analysis.

KEYWORDS: Cognitive Computing; Natural Language Processing; Natural Language Toolkit; Python; Sentiment Analysis

1. Introdução

A computação cognitiva é uma área da tecnologia que vem sendo explorada em diversos ramos do mercado. As técnicas responsáveis pela sua composição possibilitam que sistemas informatizados aprendam por meio de linguagem natural (MAROLDI, 2006; LEE et al., 2015).

Com isso, em contradição à computação tradicional, as abordagens de cognição computacional auxiliam na análise de situações dinâmicas, assim como dados variáveis, tornando possível que sistemas encontrem soluções para um problema proposto. Como resultado, a computação cognitiva pode e tem sido aplicada para apoiar a tomada de decisões com base na análise de grandes quantidades de dados, como no contexto de BigData (LEE et al., 2015).

A complexidade dos conceitos associados, bem como a insuficiência de abordagens específicas para Computação Cognitiva e Processamento de Linguagem Natural são problemas ou desafios que dificultam a experimentação prática de tais áreas em crescimento, contextualizando esta pesquisa de inicialização científica.

Um trabalho realizado com objetivos bem similar ao deste é o de Perna, Lopes e Rollsing (2017), pois fazem um estudo sobre técnicas de análise de *corpora* a fim de se aprofundarem no tema de Português para Fins Acadêmicos (PFA), utilizando de *Part of Speech Taggers* (PoSTagger) e da ferramenta de software ExATO (Extrator Automático de Ontologias) para extrair corpos de textos para análise.

Outra pesquisa que é conduzida de maneira similar ao discutido anteriormente e que parte da motivação desta pesquisa é o de Eckhard Bick (2000). A proposta de sua pesquisa é desenvolver um programa de análise de textos em português, sejam eles escritos a mão, digitais ou escaneados, como entrada, de forma a gerar uma saída que seja o menos ambíguo e o mais ausente de erros possível.

2. Objetivos

Este artigo tem como objetivos gerais (i) propor e implementar uma plataforma para Computação Cognitiva, com foco no tópico de Processamento de Linguagem Natural, bem como (ii) avaliar, por meio de experimentos e projetos práticos, a abordagem e a arquitetura tecnológica que compõem a plataforma proposta.

A realização desses objetivos depende dos seguintes objetivos específicos:

- Explorar e adotar NLTK com Python para manipular sentenças a partir dos recursos de Processamento de Linguagem Natural;
- Adotar padrões como JSON para a troca de dados entre tecnologias juntamente com NLTK e Python;
- Implementar um protótipo de jogo com Processing, integrando-o com Processamento de Linguagem Natural.

3. Metodologia

As propostas e objetivos definidos nesta pesquisa foram alcançados por meio de uma metodologia inicialmente amparada pela revisão bilbiográfica em fontes confiáveis como artigos técnico-científicos, monografias e livros.

Após os estudos de trabalhos semelhantes relacionados aos tópicos desta pesquisa, foi proposto uma implementação que utilize uma biblioteca de Processamento de Linguagem Natural disponível para a plataforma Python. Por meio dela, foram aplicados conceitos conhecidos na própria área de PLN e foram explorados minuciosamente alguns conceitos de *sentiment analysis*.

Também foi desenvolvido tendo em mente as diferentes abordagens na análise de linguagem.

4. Revisão da literatura

4.1. Processamento de Linguagem Natural

De acordo com Oliveira (2009), é uma área de pesquisa que possui o objetivo de obter uma comunicação que seja a mais "natural" possível, por meio da linguagem que os seres humanos estão acostumados. Assim, elimina-se a necessidade de adaptação a formas e variações de interação.

Para Chowdhury (2003), é uma área de pesquisa e aplicação que explora como computadores podem ser usados para entender e manipular textos ou discursos de linguagem natural para realizar tarefas desejadas.

O Processamento de Linguagem Natural é utilizado em conjunto com conceitos de linguística para a análise da linguagem, sendo definido assim algumas formações que são usadas para o entendimento da linguagem natural (GASPERIN; LIMA, 2001; MARTINS et al., 2010).

Segundo Martins et al. (2010), O compreendimento da linguagem natural toma como base as seguintes definições:

- **Fonético**: Relacionamento das palavras com os sons tanto para a fala quanto para a escuta.;
- **Morfológico**: Construção das palavras a partir de unidades de significado primitivas e de como classifica-las em categorias morfológicas;
- **Sintático**: Relacionamento das palavras entre si, cada uma com seu papel estrutural nas sentenças;
- **Semântico**: Relacionamento das palavras com seus significados e de como eles são compostos para que a sentença se torne compreensível
- **Pragmático**: Uso de sentenças em diferentes contextos, afetando o significado.

4.2. Python

A linguagem Python foi criada por Guido Van Rossum no final dos anos 80. Tornou-se famosa por sua simplicidade de uso para quem não tem fundamentos em programação.

Isso facilitou o contato de pesquisadores de diversas outras áreas com o mundo de programação, fazendo com que a plataforma se tornasse destaque para desenvolvimento de pesquisas e análises de dados.

Python é uma linguagem que utiliza tipagem dinâmica, além de também apresentar um coletor de lixo para manuseamento de memória (VAN ROSSUM et al, 2007).

Possui diversas bibliotecas especializadas em análise de dados, como Matplotlib para visualização de gráficos; Numpy, uma biblioteca que fornece uma estrutura de arranjo robusto para realizar cálculos matemáticos complexos com uma grande quantidade de dados e com maior eficiência de tempo em relação à estrutura que a própria linguagem fornece (OLIPHANT, 2006; WALT et al., 2011); entre várias outras.

4.3. NLTK

A *Natural Language Toolkit* (NLTK) é uma plataforma líder para contrução de programas em Python para trabalhar com dados da linguagem humana. Esta ferramenta fornece interfaces fáceis de uso para mais de 50 *corpora* e recursos léxicos, além de também possuir um conjunto vasto de bibliotecas de processamento de textos para classificação, tokenização, *stemming*, marcação, análise e raciocínio semântico (NLTK, 2017).

Uma das funções desta ferramenta que se destaca nesta pesquisa é a de tokenização de palavras, do módulo *tokenize*, a qual é responsável por separar as palavras de uma sentença em uma lista para ser iterada posteriormente na implementação. Esse módulo também traz uma outra função similar, a tokenização de sentenças, podendo quebrar um texto completo em uma lista de sentenças, sendo possível definir o fator de quebra como, por exemplo, uma vírgula ou um ponto final.

5. Desenvolvimento

Para a plataforma de processamento de linguagem natural, foi definido desenvolver a funcionalidade de classificar uma sentença definida em uma linguagem pelo usuário. Com isso, a ideia é tornar possível a indentificação de qual abordagem realizar para cada linguagem e também indentificar os substantivos e adjetivos de uma sentença.

Utilizando desta funcionalidade, é planejado um jogo, com o qual o usuário identifique objetos contidos na fase, descrevendo suas características visuais, assim como informando o que é o objeto.

5.1. NLTK

Nesta abordagem, o NLTK é usado para treinar o programa sobre a linguagem a ser processada, para isso, contou-se com a utilização de corpos léxicos denominados *corpus* ou *corpora*, isto é, grandes corpos de texto que possuem informações e conteúdos diversos sobre uma linguagem em questão (BIRD et al., 2009). A ferramenta posssui diversos corpos que possuem funcionalidades únicas para cada. No português, destacam-se os corpos léxicos mac-morpho e machado, já o inglês, possui o *brown*.

O mac-morpho é composto por 109 arquivos com textos de jornais da Folha de São Paulo, além de ser dividido em 10 seções, cada um possuindo um tema específico, como agricultura, política, esporte, entre outros (FONSECA; ROSA, 2013). O *corpus* machado possui as obras completas de Machado de Assis. O *brown* foi o primeiro *corpus* eletrônico do inglês com mais de mil palavras, além de conter textos de mais de 500 fontes categorizadas por gênero, como, por exemplo, notícias e editoriais (NLTK, 2017).

A ferramenta também dispõe de diversos algoritmos de análise de sentenças, alguns como os de análise de sentimento, os de polaridade de sentenças, treinamento de classificação, *stemming*, descrita na Seção 5.3.2, tokenização de palavras ou sentenças, análise de frequência de conteúdo, entre outros.

5.2. Visão geral

A plataforma desenvolvida baseia-se em receber uma frase escolhida pelo usuário, bem como a linguagem a qual ele deseja que seja processada. Cada linguagem resulta numa abordagem diferente para a plataforma.

Com a linguagem definida, o programa escolhe qual *corpus* usar para uma certa linguagem, treinando o programa por meio de textos préviamente classificados do próprio *corpus*. Essa classificação prévia pode ser feita por profissionais ou pela saída de um outro programa classificador de palavras.

O treinamento será usado para classificar a sentença de entrada, isto é, com os dados obtidos e analisando palavra por palavra, dessa forma, necessitando de um tokenizador de textos. Por fim, cada classificador morfológico é atribuido às palavras, retornando uma classificação completa. Isto é obtido por meio da funcionalidade *PoSTagger*, um pedaço de *software* que processa uma sequência de palavras numa dada linguagem e atribui classificações como substantivo, verbo, entre outros, para cada componente. (BIRD et al., 2009).

A qualidade da classificação, neste caso, depende completamente do processo de treinamento, pois se a mesma não ser realizada com qualidade, o processamento irá produzir *outputs* indesejados para seu propósito.

O método de tokenização utilizado separa um conjunto de dados em pedaços menores significativos para serem usados para um determinado objetivo. No caso desta pesquisa, o conjunto é qualquer sentença informada, e os *tokens* são as palavras separadas da sentença. Estes *tokens*, na pesquisa, foram utilizados para serem analisados e classificados.

A plataforma é utilizada pelo jogo no sentido de que, conforme o jogador vai informando as características dos objetos que ele foi avistando, essas descrições são enviadas à plataforma, quebrando essas descrições e analisando se elas coincidem com uma descrição concisa. Assim, valida-se a qualidade da informação informada, além de servir como um meio de pontuação do jogo.

5.3. Principais funcionalidades

5.3.1. Análise de sentimento

Subárea de PLN, análise de sentimento envolve habilitar computadores a reconhecerem e expressarem emoções (BO; LEE, 2008). É uma área de pesquisa importante, a qual implementa a capacidade de verificar o quanto uma pessoa possa estar sendo crítica ou plenamente rude. Um dos grandes desafios, entretanto, reside em fazer o computador compreender contextos subjetivos, ambiguos e com expressões ou gírias da linguagem.

NLTK possibilita implementar técnicas de análise de sentimento com diversas abordagens. Neste projeto, a análise foi usada de forma a auxiliar o processo de encontrar palavras positivas e negativas, além de fornecer a facilidade de procurar palavras através dos processos de *stemming* e filtragem de *stopwords*, o que diminuiu a complexidade deste processo.

5.3.2. Stemming

LOVINS (1969) determina algoritmos de *stemming* como procedimentos computacionais que reduzem as variantes de uma palavra. *Stemming* é uma técnica que permite uma abordagem para verificação de palávras mais abrangente, ou seja, possibilita analisar uma palavra pela sua raiz e não pela sua derivação.

Nesta pesquisa, a grande eficiência desse processo foi na hora de procurar palavras positivas ou negativas, pois mesmo que sejam encontradas palavras diferentes, se forem derivadas de uma mesma palavra de origem que está contida na lista de palavras negativas, por exemplo, então todas serão contadas como palavras negativas pelo programa.

A biblioteca disponibiliza diversos algoritmos diferentes pra *stemming*, dependendo da linguagem utilizada, assim, é necessário validar o procedimento correto para tratar cada linguagem especificamente.

5.3.3. Estrutura JSON

Um arquivo JSON é criado ao final de todo o processo da plataforma. Nele, é informado o conteúdo completo do processamento, contendo a sentença e linguagem escolhida para a realização do processamento; a classificação morfológica realizada por um *PoSTagger*, mostrando palavra por palavra seus classificadores; as palavras negativas e positivas encontradas; a quantidade de palavras positivas e negativas; a porcentagem dessas palavras em conjunto com todas as palavras da sentença; e por fim, a polaridade de sentença, a qual faz uma relação de todo o conteudo positivo e negativo, além de trazer o quão neutro a sentença também acaba por ser. O arquivo JSON, bem como a análise gerado pelo processamento são representados na Figura 1.


```
{
   "sentence polarity": 0.0,
   "sentence": "The main meal was very bad with horrible taste, but the dessert was delicious and tasty",
   "negative analysis": {
      "negative words": [
         "bad",
         "horribl"
      ],
      "negative percentage": 11.76,
      "negative counter": 2
   },
   "positive analysis": {
      "positive percentage": 11.76,
      "positive words": [
         "delici",
         "tasti"
      ],
      "positive counter": 2
   },
   "tagged sentence": {
      "CC": "and",
      "RB": "very",
      "DT": "the",
      "VBD": "was",
      "IN": "with",
      ",": ",",
      "NN": "dessert",
      "JJ": "tasty"
   },
   "lang": "EN"
}
```


Figura 1. Exemplo de JSON de saída

6. Jogo

O jogo é concentrado em trazer para o ambiente uma quantidade de objetos aleatórios, definidos num arquivo JSON como um banco de dados básico. Com ele, são armazenados os objetos que serão escolhidos pelo jogo, alem de obter as informações essenciais sobre os objetos que poderão ser trazidos pelo jogo e pela fase.

Um objeto neste jogo possui seu nome, sua descrição e seu local de origem o qual pertence, como, por exemplo, o garfo, um objeto pertencente à cozinha. A descrição do objeto servirá para atribuir conteúdo para este ítem, fazendo com que a descrição informada pelo jogador tenha que, de alguma forma, ser próxima com a que o objeto possui. Quanto mais as informações do jogador coincidir com a descrição do objeto, melhor será a sua pontuação. Entretanto, quanto mais palavras tiver a descrição do jogador que não coincide com a do objeto, menos pontos valerá essa informação.

O processo de análise da informação descrita pelo jogador deve depender de uma API REST, que tem como principal propósito integrar o jogo com a plataforma, possibilitando o uso de suas funcionalidades.

7. Desafios e Limitações da Pesquisa

Os desafios encontrados estão concentrados na capacidade de fornecer uma boa integração entre às diversas funções disponíveis pela biblioteca. Sua complexidade vária de linguagem para linguagem, tornando mais prático de implementar os conceitos de processamento de linguagem natural na língua inglesa.

Outro desafio encontrado no desenvolvimento desta pesquisa foi o de compreender a utilização da biblioteca NLTK e como a mesma se encaixa nos objetivos, além de visualizar o conceito e abordagem da proposta principal.

7.1. Bigram e Trigrams

Na implementação da plataforma, a abordagem de análise utilizada foi a unigrama, algoritmo estatístico simples que define uma classificação ou etiqueta para cada elemento de um conjunto analisado (BIRD; KLEIN; LOPER, 2009). O problema desta abordagem é que as palavras são classificadas sem levar em consideração o contexto no qual estão inseridas, resultando em um conteúdo de saída superfícial e com pouca eficiência.

Para realizar implementações de análises mais precisas e completas, surgiu a necessidade de aumentar a alcançabilidade dos algoritmos *n-gram* e extender para *Bigrams* e

Trigrams, no mínimo. Ao definir essa abordagem, destacou-se uma necessidade de um ambiente com maior poder computacional para oferecer um tempo de resposta aceitável para treinamento e classificação, pois as análises mais completas levam mais tempo para serem finalizadas.

7.2. Stopwords

Com a intenção de obter uma melhor abordagem, o método de eliminação ou filtro de palavras *stopwords* foi usado. Desta forma, palavras de menor valor são retiradas para a intenção da análise da plataforma e fornecendo, assim, uma eficiência maior para o resultado.

A filtragem de palavras consideradas *stopwords* possui algumas abordagens possíveis, destacando-se as duas mais práticas estão (i) retirar palavras respeitando um limite de caracteres aceitáveis definidos para o programa, pois em grande maioria, as *stopwords* são palavras curtas, e (ii) possuir uma lista de palavras consideradas *stopwords* fazendo com que o programa percorra essa lista e procure por essas palavras na sentença para retirá-las.

Em ambos os casos gerou-se uma preocupação de que o programa possa retirar palavras que agregam valor para a análise. No caso (i), uma das palavras que poderia ser retirada e que possui valor para a plataforma seria a palavra “Sol”, já no caso (ii), qualquer palavra importante pode estar nesta lista.

Uma proposta de solução para esta situação seria de criar listas customizadas com palavras específicas que o programa deve ignorar para situações específicas de uma análise. A questão a ser levantada, entretanto, é se esta é uma abordagem prática e se resulta em análises mais ágeis e precisas.

8. Resultados

Ressalta-se como resultados desta pesquisa de iniciação científica, o estudo da subárea de Computação Cognitiva, mais precisamente dos fundamentos de Processamento de Linguagem Natural, por meio de uma ferramenta para a linguagem de programação Python, para a modelagem e desenvolvimento de sistemas capazes de aprender.

A plataforma desenvolvida pela pesquisa implementa diversas das funcionalidades dispostas pela ferramenta NLTK, obtendo a possibilidade de processar sentenças, classificá-las e analisá-las por meio de conceitos de análise de sentimentos. Além disso, destaca a possibilidade de verificar as informações geradas pelo resultado do processamento, geradas num arquivo de saída JSON.

Por fim, destaca-se que os resultados foram expostos durante o X Fórum Científico na FEMA, em outubro/2017.

9. Considerações Finais

Este projeto de pesquisa, ainda que desenvolvido e aplicado em um contexto experimental, apresentou uma abordagem composta por uma ferramenta de Processamento de Linguagem Natural para auxiliar no contato com temas de Computação Cognitiva e Processamento de Linguagem Natural, assim como temas derivados e relacionados.

9.1. Trabalhos futuros

Como trabalhos futuros, distintos tópicos podem ser explorados. Os autores desta pesquisa sugerem, entre outras evoluções:

- Melhorias e refatorações da plataforma para implementar novas funcionalidades como, por exemplo, identificação da linguagem da sentença informada, possibilidade de informar mais de uma sentença e abordagens melhores de análise de sentimentos.

- Implementação de uma *Application Programming Interface* (API) REST, com o propósito de integrar diferentes plataformas à esta, possibilitando o reaproveitamento de suas funcionalidades atuais.
- Aplicação e avaliação da abordagem quando integrada ao contexto de visão computacional para escaneamento de textos, *BigData*, assim como demais áreas diversas da ciência cognitiva.

Referências

BICK, Eckhard. **The Parsing System "PALAVRAS": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework**. 2000. 503p. Tese. Departamento de linguística – Universidade de Aarhus, Aarhus, Dinamarca, 2000.

BIRD, Steven; KLEIN, Ewan; LOPER, Edward. **Natural Language Processing with Python**, 1. ed. EUA: O'Reilly, 2009.

CHOWDHURY, Gobinda G. **Natural language processing**. Annual review of information science and technology, v. 37, n. 1, p. 51-89, 2003.

FONSECA, Erick Rocha; ROSA, João Luís G. **Mac-morpho revisited: Towards robust part-of-speech tagging**. In: Proceedings of the 9th Brazilian symposium in information and human language technology. 2013.

GASPERIN, C. V.; LIMA, V. L. S. **Fundamentos de processamento estatístico de linguagem natural**. n. 21, 2001. Relatório técnico. Disponível em <http://www.pucrs.br/facin-prov/wp-content/uploads/sites/19/2016/03/tr021.pdf>. Acesso em 21/10/2017.

LOVINS, Julie Beth. **Development of a stemming algorithm**. Mech. Translat. & Comp. Linguistics, v. 11, n. 1-2, p. 22-31, 1968.

MARTINS, D.; KATAOKA, K.; TRINDADE, L. **Processamento de Linguagem Natural**. 2010. Universidade Federal da Bahia, Bahia. Disponível em <http://homes.dcc.ufba.br/~leotavo/index.html/artigo2.pdf>. Acesso em 21/10/2017.

NLTK. **Natural Language Toolkit**. 2017. Disponível em <www.nltk.org>. Acesso em 20/10/2017.

OLIVEIRA, F. A. D. **Processamento de Linguagem Natural: Princípios básicos e a implementação de um analisador sintático de sentenças da língua portuguesa**. Universidade Federal do Rio Grande do Sul, Porto Alegre, 2009.

OLIPHANT, Travis E. **A guide to NumPy**. USA: Trelgol Publishing, 2006.

PANG, Bo; LILLIAN, Lee. **Opinion mining and sentiment analysis**. Foundations and Trends® in Information Retrieval, v. 2, n. 1–2, p. 1-135, 2008.

PERNA, Cristina; LOPES, Lucelene; ROLLSING, Lucas. **Português para Fins Acadêmicos sob o aporte da Linguística de Corpus e do Processamento de Linguagem Natural**. In: Domínios de Linguagem, 2., 2017, Uberlândia, Brasil. vol. 11, abril/junho, 2017, 379p., 393p.

VAN ROSSUM, Guido et al. **Python Programming Language**. In: USENIX Annual Technical Conference. 2007. p. 36.

WALT, Stéfan van der; COLBERT, S. Chris; VAROQUAUX, Gael. The NumPy array: a structure for efficient numerical computation. **Computing in Science & Engineering**, v. 13, n. 2, p. 22-30, 2011.