ARTIGO

Classificação de sentimentos em avaliações de livros na Amazon

Sentiment classification of Amazon book reviews

João Paulo Anibal Mondoni[1]

Submetido em: 23/04/2019 **Aceito em:** 17/06/2019 **Publicado em:** 29/08/2019

Resumo: *Ao longo dos anos, o volume de dados na web cresceu drasticamente, alavancando o uso de Inteligência Artificial (IA) para a extração de informações. O processamento de linguagem natural, via IA, classifica em arquivos de fala ou texto, sentimentos em relação a um artefato ou entidade. Neste trabalho foram utilizados dois classificadores da biblioteca NLTK e o Watson Tone Analyzer da IBM, em conjunto com um dataset composto por avaliações de oito livros na loja Amazon, determinando as opiniões positivas, neutras e negativas de cada produto sob a ótica de diferentes consumidores.*

Palavras-Chave: *análise de sentimentos, inteligência artificial, nltk.*

Abstract. *Over the years, the data volume on the web has grown drastically, leveraging the use of Artificial Intelligence (AI) for information extraction. The natural language processing, through IA, classifies in speech or text files, sentiment towards an artifact or entity. In this project, two classifiers of NLTK library and IBM's Watson Tone Analyzer were used, together with a dataset made of eight Amazon store book reviews, determining the positives, neutrals and negatives reviews of each product in different costumers' perspective.*

Keywords: *analysis, artificial intelligence, sentiment, nltk.*

1.Introdução

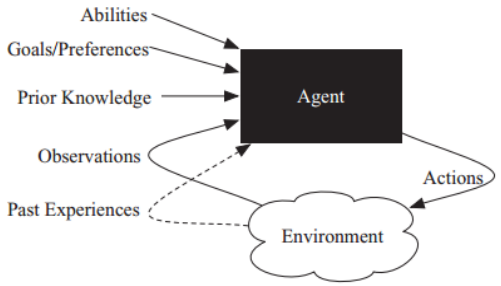
A IA (Inteligência Artificial) é uma forma de buscar, de forma simulada computacionalmente, a automação do comportamento inteligente (LUGER, 2013). Portanto, permite a busca pela resolução de problemas novos, através de dispositivos e métodos, com a automação do comportamento racional. Isso nos leva a um patamar onde podemos presumir que a IA é utilizada no cenário da tecnologia atual para resolver problemas e trazer soluções que demandam de muito intelecto humano de forma rápida, automática e confiável. Dois componentes muito importantes para a IA são os Agentes Inteligentes e o Conhecimento Prévio.

1.1. Agentes Inteligentes

[1] Escola de Engenharia de Piracicaba, Piracicaba (SP), Brasil. Email: jp.mondoni@gmail.com

As soluções de IA e suas automações ocorrem através de agentes inteligentes, atuadores e sensores. Segundo Poole e Mackworth (2010), agentes são entidades que agem em um ambiente com um propósito definido, sendo inteligente quando o que faz é correto de acordo com os objetivos, se adapta às mudanças no ambiente e nas metas, aprende por experiência e toma decisões baseadas em suas limitações de percepção e de recursos computacionais. Um ambiente é tudo aquilo que um agente, sendo inteligente ou não, pode agir através de seus atuadores. A Figura 1 descreve esses aspectos de forma simplificada.

Figura 1 - A interação de um agente com o ambiente



Fonte: Poole e Macworth (2010).

1.2. Conhecimento Prévio

O conhecimento prévio se refere à todas as informações que o agente inteligente possui antes de qualquer observação adicional que venha a fazer. Esse tipo de informação é de suma importância para que o agente produza comportamentos apropriados. Sistemas baseados em regras consistem em uma base de conhecimento como uma orientação para a maneira correta como se comportar, seguindo o que deve ser feito em determinadas situações até que sua inteligência se torne capaz de se adaptar às necessidades da situação. Um agente com conhecimento prévio robusto e com um conjunto de regras bem definido, será menos suscetível a decisões questionáveis como vimos anteriormente, pois será mais fechado de uma forma geral (COPPIN, 2004).

2. Processamento de Linguagem Natural

Para um computador, processar *strings* ou dados que contenham caracteres de um alfabeto conhecido não é uma tarefa complexa, pois o mesmo possui capacidade de computar este tipo de informação. Entretanto é necessária uma técnica que, na eventualidade desse tipo de dado ter um *reasoning* de alto nível, ou seja, que o mesmo seja qualitativo, a IA seja capaz de entender o significado

aplicado a um contexto. Humanos começam a associação entre entidades e palavras quando ainda são crianças e bebês, através da interação com o ambiente.

Segundo Vygotsky (2000), gestos, palavras, linguagem corporal e outros traços característicos de um indivíduo são absorvidos e ao longo do tempo transformados em conhecimento, um processo gradual. O entendimento da criança vem da repetição e reprodução daquilo que é apresentado no ambiente, através de imitações. O processo com um computador toma um rumo distinto a partir do momento que se torna possível importar um dicionário inteiro em um robô, a fim de que o mesmo conheça as palavras da língua e consiga criar conexões linguísticas baseado em um conjunto de sinônimos e antônimos.

Contudo existem situações subjetivas que necessitam de maior atenção quando se trata de processamento de linguagem natural. O processamento da linguagem segue uma cadeia de ações, conforme a Figura 2, que se iniciam na leitura de arquivos brutos de texto, selecionando dados de interesse a serem observados e minerando informações através da pré-seleção realizada. Então são utilizados padrões obtidos através da etapa anterior para obter conhecimento específico, que neste trabalho são os sentimentos contidos em textos.

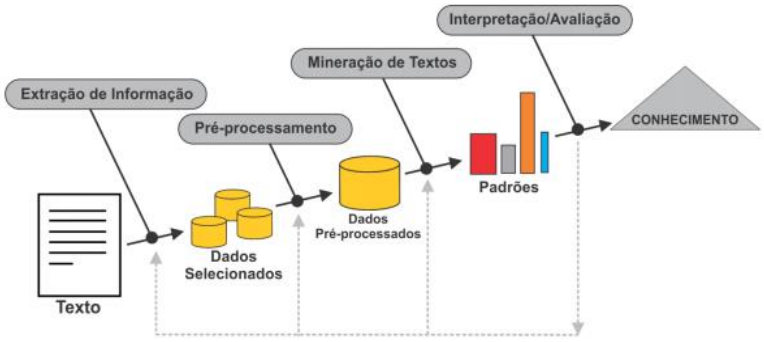


Figura 2 - Etapas de processamento de linguagem natural

Fonte: Gonçalves (2001).

O ser humano se comunica com o que conhecemos por linguagem natural, linguagem que pode conter frases extremamente objetivas como frases completamente subjetivas ou um misto de ambas. Ao definir para um robô uma estrutura básica de oração ou frase, que contenha um sujeito, um verbo, uma preposição, um substantivo e um adjetivo, será entendida por um Processador de Linguagem Natural (PLN) de forma relativamente simples, afinal sua estrutura linguística não representa nenhum desafio maior. Se as palavras tiverem significados objetivos, isentando-se de figuras de linguagem, o PLN conseguirá tomar conhecimento do que está sendo transmitido. Todavia a maioria das frases em linguagem natural possuem estruturas mais complexas e peculiares. Ainda em português, uma frase estrutura similar poderia conter sinestesia, ou ironia, o que teria complexidade maior no entendimento por parte da inteligência artificial.

2.1. Tokens e Tagging

O distrinchamento em *tokens* de um texto é o processo de separar de acordo com delimitadores a *string* de entrada, criando uma separação para o nível de análise de palavras. Sem esse processo, o texto é apenas um dado bruto e tem significado abstrato (BOWE, 2011). Este procedimento cria, no ambiente de processamento de linguagem natural, a separação da estrutura de um documento, abrindo o leque para possíveis apreciações. Uma delas é a que envolve o tema do trabalho que é análise de sentimentos. Ao submeter uma avaliação a este processo, cria-se uma estrutura de palavras conhecidas e utilizadas e faz-se uma referência cruzada ao dicionário que temos (campo léxico das palavras) e obtém-se uma frequência de repetição das mesmas, propiciando de acordo com a semântica uma maneira de lograr o sentimento de positividade, neutralidade ou negatividade, concebendo uma pontuação global consequentemente. Na NLKT, é utilizado junto com esse processo, uma expressão regular para que se filtre precisamente o tipo de conteúdo que deve ser identificado e extraído.

O processo seguinte é o de marcação de um texto, em inglês conhecido como *tagging*, faz a análise e definição *tokens* de acordo com a gramática, determinando se uma palavra é um verbo, um adjetivo ou um pronome, por exemplo. A marcação é um processo realizado através de um treinamento disponibilizado de forma automática pelo NLTK, garantindo a precisão das informações, é vital em classificações supervisionadas.

2.2. Classificação de textos

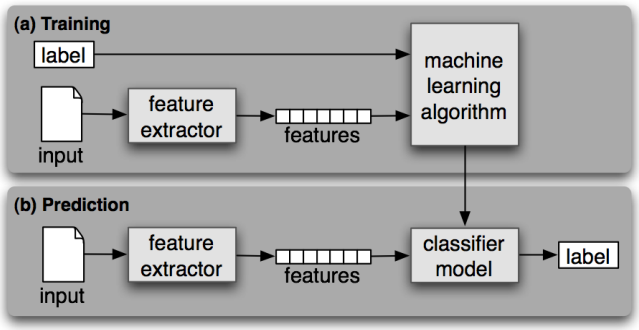
Conforme o livro da NLTK (BIRD, LOPER E KLEIN, 2009), detectar padrões é uma parte central do PLN. Alguns padrões conforme os autores na língua inglesa são palavras que terminam em –ed que tendem a ser verbos no passado e o uso frequente de *will* aumenta a probabilidade de o texto ser uma notícia. Para que

a detecção de padrões seja feita da melhor forma possível, existem tipos distintos de classificação, como a classificação supervisionada.

A classificação supervisionada (CS) é "dado uma entrada qualquer, escolher a classe de rótulo correta" (BIRD, LOPER E KLEIN, 2009), com intenção de encontrar a melhor classificação possível. A CS tem como objetivo resolver algumas questões como determinar o tópico de uma notícia a partir de uma lista fixada de possibilidades, indicar se um e-mail é ou não spam ou definir se a palavra banco se refere à instituição financeira ou o objeto de sentar. Essas distinções são imprescindíveis na análise de sentimentos, porque interfere no resultado positivo, neutro ou negativo na mineração de opiniões.

Um treinador para CS consiste em criar um modelo de classificação (Figura 3 (a)) com o objetivo de que a predição (Figura 3 (b)) utilize as características abduzidas em um modelo classificador e então gerar um rótulo final.

Figura 3 - Fluxo da classificação supervisionada



Fonte: Bird, Loper e Klein (2009).

Citando casos análogos, a detecção de gênero utiliza CS (Figura 4), através de um modelo definido de padrões para os gêneros masculino e feminino da língua inglesa, por intermédio da leitura do último caractere de um nome. Com o intuito de que seja a catalogação mais correta o possível, é preciso preparar uma lista de amostras de palavras com a classificação, fazer o treinamento do conjunto e então a determinação do gênero.

Figura 4 - Códigos da NLTK em Python para determinar gênero (binário) de um nome

```
>>> from nltk.corpus import names
>>> labeled_names = ([(name, 'male') for name in names.words('male.txt')] +
... [(name, 'female') for name in names.words('female.txt')])
>>> import random
>>> random.shuffle(labeled_names)
```

```
>>> featuresets = [(gender_features(n), gender) for (n, gender) in labeled_names]
>>> train_set, test_set = featuresets[500:], featuresets[:500]
>>> classifier = nltk.NaiveBayesClassifier.train(train_set)
```

```
>>> classifier.classify(gender_features('Neo'))
'male'
>>> classifier.classify(gender_features('Trinity'))
'female'
```

Fonte: Bird, Loper e Klein (2009).

Utilizando o teorema de Bayes é possível determinar a probabilidade de um determinado conjunto de características (*features*) possuir um rótulo (*label*). Neste projeto, este rótulo seria o sentimento daquela característica, sendo positiva ou negativa. A equação é dada da seguinte maneira:

P(label | features) = P(label) * P(features | label) / P(features)

Onde calcula-se o resultado da probabilidade de forma binária, ou seja, determina quais as chances de aquela característica possuir aquele rótulo. Como os *labels* são mutuamente exclusivos, apenas um dos dois é escolhido. Esta escolha binária pode diminuir a acurácia da análise, principalmente em casos onde não há uma probabilidade muito alta para nenhum dos dois lados. No caso de um *feature* ser classificado como 55% positivo e 45% negativo, ele vai ser totalmente considerado como positivo, por mais que o seu valor contrário demonstre outra informação.

O classificador Naive Bayes é um componente presente no módulo NLTK. Sua principal vantagem é aplicar o teorema de Bayes de forma simples e com processamento rápido. Grandes conjuntos de texto podem ser rapidamente utilizados como treino para o classificador, como por exemplo o *dataset* nativo da NLTK *movie_reviews*, que possui 5331 avaliações positivas e 5331 avaliações negativas já corretamente classificadas.

3. Análise de Sentimentos

Feldman (2013) apresenta a análise de sentimento como “a tarefa de encontrar opiniões de autores sobre entidades específicas”. Esta incumbência se encarrega de, dado um documento com algum conteúdo que possa ser analisado, averiguar e classificar os sentimentos contidos no mesmo, superando desafios propostos pelo processamento de linguagem natural, em busca de informações que agreguem valor ao objeto de estudo.

3.1. Aplicações para Análise de Sentimentos

Com a definição apresentada, torna-se possível desenvolver algumas das aplicações possíveis para esse tema. Pang e Lee (2008) citam alguns desses usos possíveis para a análise de sentimentos: avaliações da web, inteligências de negócios, agências de inteligência governamental e como um subcomponente tecnológico.

A área de inteligência de um governo é beneficiária dos avanços na área de análise de sentimentos, afinal este tipo de conhecimento viabiliza o monitoramento em massa da população, buscando "fontes de aumento de comunicações hostis e negativas" (PANG E LEE, 2008). O uso ético correto destas informações seria capaz de auxiliar na prevenção de situações que coloquem em risco a vida de mais pessoas.

Como um subcomponente de tecnologia, esta metodologia de análise tem um uso levemente distinto, entretanto que contribui com grande importância para outras metodologias. Entender o sentimento de um usuário em relação à uma entidade habilita o aumento de uso de sistemas baseados em recomendação, seguindo a premissa de que um usuário possui um perfil na web e que uma recomendação baseada no conhecimento prévio que se tem do retrato daquele indivíduo, poderia trazer informações relevantes para o mesmo ou para o detentor de uma solução.

4. Experimentos, Implementações e Resultados

Através de uma metodologia indutiva, a inteligência artificial da aplicação será capaz de identificar o teor sentimental de cada *review* do conjunto de dados, através de um outro *dataset* padrão contido no toolkit utilizado para o desenvolvimento do projeto. Este corpus linguístico tem uma série de opiniões sobre filmes e foi coletado na referência da plataforma e foi introduzido em alguns artigos dos autores Bo Pang e Lillian Lee, os mesmos de "*Opinion mining and sentiment analysis*" (2008).

4.1. Normalização do conjunto de dados

Para que os dados pudessem ser analisados, foi necessária a execução de uma etapa de normalização, onde foram retiradas informações impertinentes e que possam prejudicar a análise posterior. Nesta etapa de preparação, foi seguido um roteiro a fim de remover marcações desnecessárias dos textos, atributos HTML, tabulações e possíveis link. Além disso, foi realizada uma padronização para cada avaliação do conjunto, adicionando um caractere de fim de linha ao fim de cada um, dado pela sequência de escape (como por exemplo, \r\n), conhecido também como CRLF (*carriage return*, *line feed*). Todo o *dataset* foi duplicado em novos arquivos normalizados, evitando o comprometimento das informações originais.

4.2. Experimentos e Implementações

Para obter os resultados, foram realizados diversos experimentos buscando obter melhor acurácia e testar os modelos classificadores ou a metodologia de classificação. Para as implementações utilizando as bibliotecas NLTK e Vader Sentiment, foi utilizado o corpus *movie_reviews*.

Com o classificador de Bayes, foi utilizado um filtro de palavras de parada (*stopwords*), que pertencem a um conjunto de palavras que não possuem valor para o contexto, afinal seu significado não agrega a classificação, uma vez que a probabilidade de uma *stopword* ser positiva ou negativa é de 50%, sendo perfeitamente neutra. Essas palavras são artigos, pronomes e auxiliares verbais. A seguir são utilizadas funções de Jacob Perkins para obter características dos rótulos do corpus de treino, para remover quaisquer possíveis vieses do classificador e também para separar características dos *labels* (PERKINS, 2010). Com esta etapa concluída seguindo os ensinamentos do autor, passamos à etapa de criar um classificador utilizando o teorema de Bayes, para as avaliações pré-definidas no corpus opiniões sobre filmes. Utilizando a biblioteca *pickle*, é possível criar um arquivo de despejo do classificador para ser utilizado posteriormente. Por fim, para testar a acurácia, usa-se o método *accuracy*, passando como parâmetros a variável do classificador e um conjunto de características de testes com o corpus de texto comparativo *movie_reviews*.

Através da implementação com uso da API Vader Sentiment, a classificação independe do treino local, pois está embutido um dicionário léxico com pontuações de polarização de sentimentos. É recomendado que seja feita o uso de uma pontuação agregada, chamada de *compound*, para verificação da polaridade sentimental de uma opinião (HUTTO E GILBERT, 2014). Como forma de aumentar a acurácia, as opiniões do *dataset* foram divididas em frases, obtendo um *compound* para cada, efetuando posteriormente uma média aritmética para determinar a pontuação do texto completo. Foi percebido que este método resultava em um número muito grande de avaliações neutras, pois pela documentação da API, a faixa de valores neutro acaba sendo branda. Para dar mais sentido aos resultados, foram criadas duas novas classes: levemente positiva e levemente negativa.

4.3. Resultados – Vader Sentiment

Os resultados finais, utilizando como testes o corpus de texto *movie_reviews*. É utilizado todo o *dataset*, uma vez que o treino não utiliza do mesmo. O método Vader teve uma acurácia muito boa, conseguindo atingir 93% para os textos positivos e 87% para os negativos, resultando numa média de 90% de acurácia.

Tabela 1 - Matriz de Confusão – Vader Sentiment API

	Positivo	Negativo
Positivo	4958 (93%)	373 (7%)
Negativo	693 (87%)	4638 (13%)

Fonte: Elaboração Própria.

4.4. Resultados – Classificador de Bayes (NLTK)

Para aferir os resultados, foi utilizada uma porção de 20% do corpus *movie_reviews*, deixando de lado os 80% utilizados para o treino. Portanto, foram analisados 1066 textos positivos e 1066 textos negativos dos 5331 originais para ambas as classes. Com a *cross-validation* (CV), uma técnica para avaliar a capacidade de generalização de um modelo, o qual dividiu o *dataset* de treino em 10 partes (*folds*) para testar a acurácia generalizada em cada um, foi obtida acurácia média 84,1% para as classes positivas e média de 82,3% para as classes negativas, obtendo acurácia média total de 83,2%.

Tabela 2 - Matriz de Confusão – Naive Bayes

	Positivo	Negativo
Positivo (CV)	895 (84,1%)	171 (15,9%)
Negativo (CV)	192 (17,7%)	874 (82,3%)

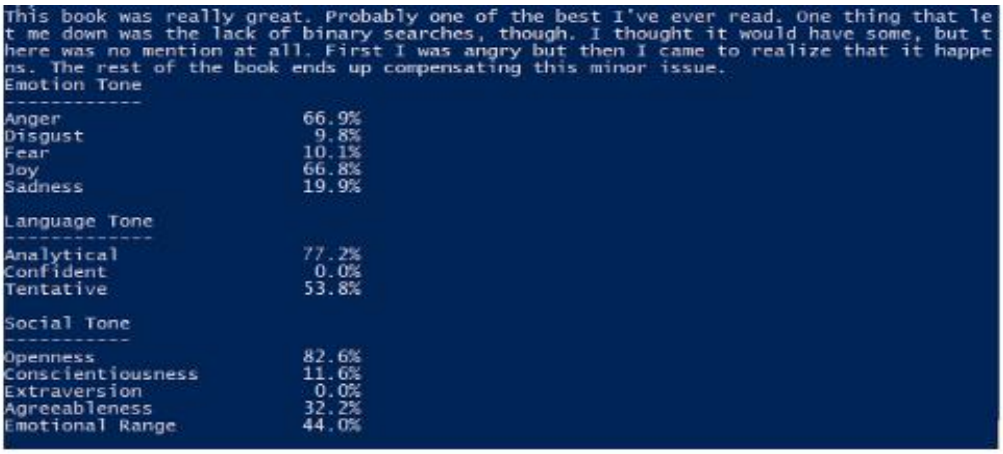
Fonte: Elaboração Própria.

4.5. Demonstração – IBM Watson Tone Analyzer

O foco desta API é distinto do que pontuar somente a polarização sentimental, afinal analisa tons emocionais (raiva, repulsa, felicidade, medo, tristeza), tons de linguagem (analítico, confiante e tentativo) e tons sociais (abertura, conscienciosidade, extraversão, agradabilidade e range emocional). Pela limitação de chamadas na API gratuita, não foi possível realizar testes de acurácia semelhantes às implementações anteriores. O objetivo do uso desta API foi trazer ao estudo a demonstração de diferentes possibilidades de análise de sentimentos, não somente na polarização ternária, mas sim mais afundo nas acentuações sentimentais existentes. É permitido ao usuário sortear uma avaliação para classificar pela NLTK, Vader Sentiment e IBM Tone Analyzer simultaneamente, para comparar os resultados.

Para essa implementação, foi criado um projeto gratuito no painel IBM Cloud pessoal do autor e utilizada a documentação da API para chamadas de análise, recebendo como resultado instantaneamente a pontuação (Figura 5).

Figura 5 - Demonstração de resultados através da plataforma IBM Watson.



Fonte: Elaboração Própria.

5. Conclusão e trabalhos futuros

Os resultados obtidos no trabalho, após o algoritmo fazer análises de quase 200 mil textos de avaliações diferentes, cada um com sua peculiaridade de forma de escrita, desde os mais irônicos aos mais objetivos com apenas 3 palavras, são suficientes para demonstrar que o objetivo inicial de se classificar os textos foi atingido com sucesso, seguindo também os requisitos do nível de confiança e velocidade em processamento.

Comparando os métodos desenvolvidos, seguindo a limitação de chamadas da API da IBM, podemos verificar como os dois métodos de código aberto – Vader e NLTK – se saem. Por fim, para o modelo treinado e usado para classificação de Bayes, foi obtida uma acurácia média de 83,2%, contra 90% do modelo de classificação por dicionário léxico da Vader Sentiment. O Watson Tone Analyzer não teve a acurácia medida por limitações de taxa da API, entretanto, segundo a IBM, melhorias realizadas permitiram que o classificador fosse melhor que os modelos "estado da arte" de classificação de emoções (GUNDECHA, 2016).

Para uma melhoria de acurácia com o classificador de Bayes, poderiam ter sido melhor classificadas figuras de linguagem no texto. Além disso, textos muito longos na Vader Sentiment parecem tender a ser mais neutros do que na NLTK, esse aspecto poderia ter sido mais explorado para descobrir se há uma correlação entre concentração sentimental em textos longos, ou se é apenas uma característica da API.

Considerando os resultados obtidos, uma nova etapa deste trabalho será iniciada com a criação de um motor capaz de pesquisar postagens em redes sociais sobre determinado assunto e monitorar as emoções e sentimentos em relação a um tópico em tempo real.

Referências Bibliográficas

BIRD, S., LOPER, E., KLEIN, E. (2009) "Natural Language Processing with Python". O'Reilly Media Inc., 1st edition.

BOWE, A. (2011) "Au Naturale: An Introduction to NLTK". http://alexbowe.com/au-naturale/>.

COPPIN, B. (2004) "Artificial intelligence illuminated", Jones and Bartlett Publishers, 1st edition.

FELDMAN, R. (2013) "Techniques and Applications for Sentiment Analysis". In Communications of the ACM 56, pages 82-89.

GONÇALVES, L. (2001) "Avaliação de ferramentas de mineração de dados como fonte de dados relevantes para a tomada de decisão" Universidade Federal do Rio Grande do Sul.

GUNDECHA, P. (2016) "IBM Watson just got more accurate at detecting emotions". Disponível em https://www.ibm.com/blogs/bluemix/2016/10/watson-has-more-accurate-emotion-detection/. Último acesso: 19/04/2019.

HUTTO, C. J., GILBERT, E. E. (2014) "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text". Eight International Conference on Weblogs and Social Media (ICWSM-14).

LUGER, G. F. (2013) "Inteligência Artificial", Person Education do Brasil, 6th edition.

PERKINS, J. (2010) "Python 3 Text Processing with NLTK 3 Cookbook", Packt Publishing, 2nd edition.

VYGOTSKY, L. S. (2000) "A construção do pensamento e da linguagem", Martins Fontes, 2000.

POOLE, L. D., MACWORTH, K. A. (2010) "Artificial Intelligence: Foundantions of Computational Agents", Cambridge University Press, 1st edition.

PANG, B., LEE, L. (2008) "Opinion mining and sentiment analysis" Foundations and Tends in Information Retrieval, pages 1-135.