

UNIVERSIDADE DE CAXIAS DO SUL
ÁREA DE CONHECIMENTO DE CIÊNCIAS EXATAS E ENGENHARIA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

GUILHERME HENRIQUE SANTOS ANDREATA

**O Uso de Processamento de
Linguagem Natural para a Análise de
Sentimentos na Rede Social Reddit.**

André Luis Martinotto
Orientador

Caxias do Sul, Maio de 2017

O Uso de Processamento de Linguagem Natural para a Análise de Sentimentos na Rede Social Reddit.

por

Guilherme Henrique Santos Andreato

Projeto de Diplomação submetido ao curso de Bacharelado em Sistemas de Informação da área de conhecimento de ciências exatas e engenharia, como requisito obrigatório para graduação.

Projeto de Diplomação

Orientador: André Luis Martinotto

Banca examinadora:

André Gustavo Adami

CCTI/UCS

Carlos Eduardo Nery

CCTI/UCS

Projeto de Diplomação apresentado em
5 de Dezembro de 2013

Daniel Luís Notari
Coordenador

SUMÁRIO

LISTA DE ACRÔNIMOS	3
LISTA DE FIGURAS	4
LISTA DE TABELAS	5
RESUMO	6
1 INTRODUÇÃO	7
1.1 Objetivos do Trabalho	8
1.2 Estrutura do Trabalho	8
2 PROCESSAMENTO DE LINGUAGEM NATURAL	9
2.1 Linguística	9
2.2 Métodos de Processamento de Linguagem Natural	10
2.2.1 Método Simbólico	10
2.2.2 Método Estatístico	11
2.3 Classificadores	14
2.3.1 Naive Bayes	14
2.3.2 <i>Maximum Entropy</i>	16
3 FRAMEWORKS	18
3.1 Natural Language Toolkit	18
3.1.1 Análise de Sentimentos	18
3.2 Stanford CoreNLP	19
3.2.1 Análise de Sentimentos	19
4 REDE SOCIAL REDDIT	21
4.1 API	22
REFERÊNCIAS	23

LISTA DE ACRÔNIMOS

NLP	<i>Natural Language Processing</i>
NLTK	<i>Natural Language Toolkit</i>
MaxEnt	<i>Maximum Entropy</i>
RNTN	<i>Recursive Neural Tensor Networks</i>

LISTA DE FIGURAS

Figura 2.1: Caminhos possíveis de classificação	13
Figura 2.2: Caminhos já decididos de classificação	14
Figura 3.1: Frase já classificada disponível no Sentiment Treebank	20
Figura 3.2: Exemplo de implementação	20

LISTA DE TABELAS

Tabela 2.1: Tabela de Probabilidade de Transição	12
Tabela 2.2: Tabela de Probabilidades de Associação	12
Tabela 2.3: Tabela de Carro e Categoria.	15
Tabela 2.4: Tabela de anos, carros, portas e categorias.	16
Tabela 2.5: Tabela de Probabilidades A	17
Tabela 2.6: Tabela de Probabilidades B	17

RESUMO

Palavras-chave: Kinect, Blender, Animação 3D.

1 INTRODUÇÃO

A linguagem é a forma com que nós nos comunicamos, seja ela escrita ou falada, através de símbolos ou sons. De fato, a linguagem é a forma como expressamos nossas idéias, sentimentos e experiências. O Processamento de Linguagem Natural, é o termo utilizado para descrever um software ou componente de hardware que tem como função analisar a linguagem escrita ou falada (JACKSON; MOULINIER, 2007).

Existem duas abordagens de Processamento de Linguagem Natural, sendo a primeira delas é chamada de simbólica ou racionalista e a outra de empírica. A primeira abordagem consiste em uma série de regras para a manipulação de símbolos, como as regras gramaticais que permitem identificar se uma frase está malformada ou não. A abordagem empírica está centrada na análise estatística da linguagem através de grandes quantidades de textos, como por exemplo, a utilização de modelos de Markov para reconhecer padrões na escrita (JACKSON; MOULINIER, 2007).

Existem diversos *frameworks open source* que facilitam o desenvolvimento de *software* para o Processamento de Linguagem Natural, sendo que entre esses destacam-se o *Stanford's Core NLP Suite* (Stanford CoreNLP, 2017), *Natural Language Toolkit* (Natural Language Toolkit, 2017), *Apache OpenNLP* (Apache OpenNLP, 2017) e *Spacy* (Spacy, 2017). Esses *frameworks* nos permitem, entre outras coisas, efetuar análise de sentimentos, identificar tópicos e conteúdos.

A rede social Reddit é o vigésimo terceiro *website* mais acessado na internet e o sétimo mais acessado nos Estados Unidos da América (Alexa, 2017). Através deste *website*, seus usuários podem criar ou se inscrever em comunidades, também conhecidas como *subreddits*. Uma vez que as comunidades são criadas pelos próprios usuários, podemos encontrar comunidades sobre todos os assuntos, sejam notícias do mundo, comunidades partidárias, comunidades criadas para pessoas de uma mesma localidade ou comunidades de imagens engraçadas.

Nestas comunidades é possível visualizar e comentar *links* enviados por outros usuários. Além disso, o usuário pode efetuar um voto de forma positiva, caso acredite que aquele *link* é útil para a comunidade. Caso contrário, é possível efetuar um voto

negativo. Uma vez que os próprios usuários podem submeter *links*, os eventos e notícias de todo o mundo são reportados no *website*, como exemplo, pode-se citar as eleições ocorridas no ano de 2016 nos Estados Unidos e o tiroteio ocorrido em Paris.

Neste trabalho será desenvolvido um software que permita realizar a análise dos comentários do *website* Reddit. Mais especificamente esses comentários serão analisados com o objetivo de identificar padrões de sentimentos, ou seja, determinar se a opinião expressada com relação a um determinado tópico é neutra, positiva ou negativa.

1.1 Objetivos do Trabalho

Este trabalho tem como objetivo a análise dos comentários disponíveis no *website* Reddit, identificando padrões de sentimentos entre os usuários de suas comunidades. De forma a atingir o objetivo principal desse trabalho, os seguintes objetivos específicos devem ser realizados:

- Desenvolver uma ferramenta para o Processamento Natural de Linguagem através de *frameworks* já existentes.
- Construção de uma base de dados a partir do *website* Reddit.
- Efetuar o processamento da base de dados utilizando a ferramenta desenvolvida.

1.2 Estrutura do Trabalho

2 PROCESSAMENTO DE LINGUAGEM NATURAL

O objetivo da área de Processamento de Linguagem Natural é interpretar e analisar a linguagem natural, ou seja, a linguagem utilizada pelo seres humanos não importando se essa é escrita ou falada (MANNING; SCHÜTZE, 1999).

O Processamento de Linguagem Natural é uma área antiga, sendo anterior a invenção dos computadores modernos. De fato, sua primeira grande aplicação foi um dicionário desenvolvido no Birkbeck College em Londres no ano de 1948. Por ser uma área complexa, seus primeiros trabalhos foram notavelmente falhos o que causou uma hostilidade por parte das agências fundadoras de pesquisas.

Os primeiros pesquisadores eram muitas vezes bilíngues, como por exemplo, nativos alemães que imigraram para os Estados Unidos. Acreditava-se que pelo fato destes terem conhecimento de ambas as línguas, Inglês e Alemão, eles teriam capacidade de desenvolver programas de computadores que efetuariam a tradução das línguas de modo satisfatório. Uma vez que esses encontraram muitas dificuldades, ficou claro que o maior problema não era o conhecimento de ambas as línguas e sim como expressar esse conhecimento na forma de um programa de computador (HANCOX, 2017).

Para fazermos com que um computador seja capaz de interpretar uma língua, precisamos antes entender como nós efetuamos essa interpretação. Por isso, uma parte considerável do Processamento de Linguagem Natural está apoiado na área de Linguística.

2.1 Linguística

O objetivo da Linguística é compreender como os humanos adquirem, produzem e entendem as diversas línguas, ou seja, a forma com que conversamos, a nossa escrita e outras mídias de comunicação (MANNING; SCHÜTZE, 1999).

Na linguagem tanto escrita, como na falada, existem regras que são utilizadas para estruturar as expressões. Uma série de dificuldades no Processamento de Linguagem Natural são ocasionadas pelo fato de que as pessoas constantemente

mudam as regras para satisfazerem suas necessidades de comunicação (MANNING; SCHÜTZ, 1999). Uma vez que as regras são constantemente modificadas pelo locutor, se torna extremamente difícil a criação de um software ou hardware efetue a interpretação de uma língua.

2.2 Métodos de Processamento de Linguagem Natural

2.2.1 Método Simbólico

O método simbólico ou racionalista está baseado no campo da Linguística e faz o uso da manipulação dos símbolos, significados e das regras de um texto. Um exemplo de método simbólico é o método de Brill (BRILL, 1992). Por exemplo, no método de Brill a frase “João pintou a casa de branco”, será separada em palavras que serão classificadas através de um dicionário pré-definido.

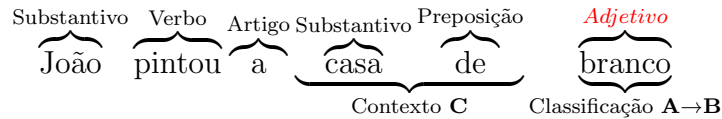
Palavra	João	pintou	a	casa	de	branco
Classificação:		Verbo	Artigo	Substantivo	Preposição	Adjetivo

Observa-se que algumas palavras não foram identificadas, como “João” ou classificadas de forma errada, como “branco”. Desta forma, o método utiliza-se de outras duas regras para uma classificação inicial. A primeira regra classifica todas as palavras desconhecidas que iniciam em maiúsculo como substantivos, por exemplo, a palavra “João”. Já a segunda regra, atribui para a palavra desconhecida a mesma classificação de outras palavras que terminam com as mesmas três letras. Por exemplo, supondo que o método não tivesse encontrado a palavra “pintou” no dicionário, ele irá associar essa palavra a outras palavras terminadas com o sufixo “tou”. Ou seja, ele iria classificar a palavra “pintou” como verbo.

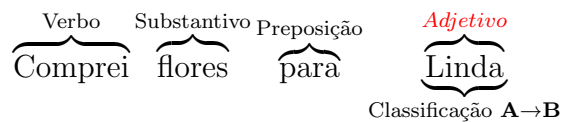
Palavra	João	pintou	a	casa	de	branco
Classificação:	Substantivo	Verbo	Artigo	Substantivo	Preposição	Adjetivo

Após essa classificação inicial, o método executa o seguinte conjunto de regras:

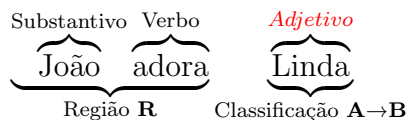
- Se uma palavra tem a classificação **A** e está no contexto **C** então a sua classificação deverá ser mudada para **B**. Por exemplo, se uma palavra **A** (branco no exemplo) é um adjetivo e uma das duas palavras anteriores é uma preposição (“casa de” no contexto **C**), mude para sua classificação para substantivo (classificação **B**).



- Se uma palavra tem a classificação **A** e tem uma propriedade **P** então a sua classificação deverá ser alterada para **B**. Por exemplo, se uma palavra **A** (“Linda”) foi classificada como um adjetivo e essa inicia com letra maiúscula (propriedade **P**), sua classificação deverá ser alterada para substantivo (classificação **B**).



- Se uma palavra tem a classificação **A** e uma palavra com a propriedade **P** está na região **R**, sua classificação deverá ser **B**. Por exemplo, se uma das duas palavras anteriores (“João adora” na região **R**) iniciam com letra maiúscula (propriedade **P**), sua classificação deverá ser alterada para substantivo (classificação **B**).



2.2.2 Método Estatístico

O método estatístico ou empírico utiliza-se de grandes quantidades de texto como páginas da internet, procurando por padrões e associações a modelos, sendo que esses podem ou não estarem relacionados com regras sintáticas ou semânticas. Como exemplo, podemos citar a utilização de Modelos de Markov com o algoritmo de Viterbi. Neste, a partir de um conjunto de dados já classificados, ou seja, um *training set* é verificada a possibilidade de transição entre as classes gramaticais. Como por exemplo, se no nosso *training set*, temos 10000 substantivos que em 7000 dos casos são seguidos por um verbo:

$$P(VB|SM) = \frac{C(SM, VB)}{C(SM)} = \frac{7000}{10000} = 0.7$$

Aonde que **VB** são verbos e **SM** substantivos. Portanto, a probabilidade de um

substantivo ser seguido de um verbo ($(\mathbf{VB}|\mathbf{SM})$) é igual a ocorrência de substantivos seguidos de verbos ($(\mathbf{C}(\mathbf{SM},\mathbf{VB}))$) dividida pela quantidade total de substantivos em nosso *training set*.

Após, é verificada a probabilidade de cada palavra ser associada com uma classe. Supondo que dos 10000 substantivos, 150 são a palavra “um” classificada dentre artigo, substantivo e pronome, como substantivo:

$$P(um|SM) = \frac{C(SM, um)}{C(SM)} = \frac{150}{10000} = 0.015$$

Aonde que **um** é a palavra que está sendo classificada e **SM** são os substantivos existentes no nosso *training set*. Portanto a probabilidade de um substantivo ser associada a palavra “um” ($(\mathbf{P}(\mathbf{um}|\mathbf{SM}))$) é igual a quantidade da palavra “um” classificada como substantivo ($(\mathbf{C}(\mathbf{SM},\mathbf{um}))$) dividida pela quantidade de substantivos em nosso *training set*. Como exemplo, vamos supor os seguintes dados compilados através dos dois métodos citados anteriormente, aonde a primeira tabela representa a probabilidade de transição e a segunda tabela representa a probabilidade de associação:

	Substantivo	Verbo	Artigo	Pronome
Início	0.30	0.25	0.35	0.4
Substantivo	0.4	0.2	0.015	0.85
Verbo	0.5	0.2	0.3	0.07
Artigo	0.1	0.001	0.030	0.005
Pronome	0.7	0.005	0.005	0.8

Tabela 2.1: Tabela de Probabilidade de Transição

	João	comprou	um	carro
Substantivo	1		0.015	1
Verbo		1		
Artigo			0.030	
Pronome			0.005	

Tabela 2.2: Tabela de Probabilidades de Associação

Portanto, para a classificação da frase anterior, as palavras João, Comprou e Carro, só podem ser classificadas como substantivo (SM), verbo (VB) e substantivo (SM) respectivamente. Porém, a palavra “um” pode ser classificada em substantivo (SM), artigo (ART) e pronome (PRO). A seguinte imagem representa as possibilidades de caminho que o classificador pode percorrer neste exemplo:

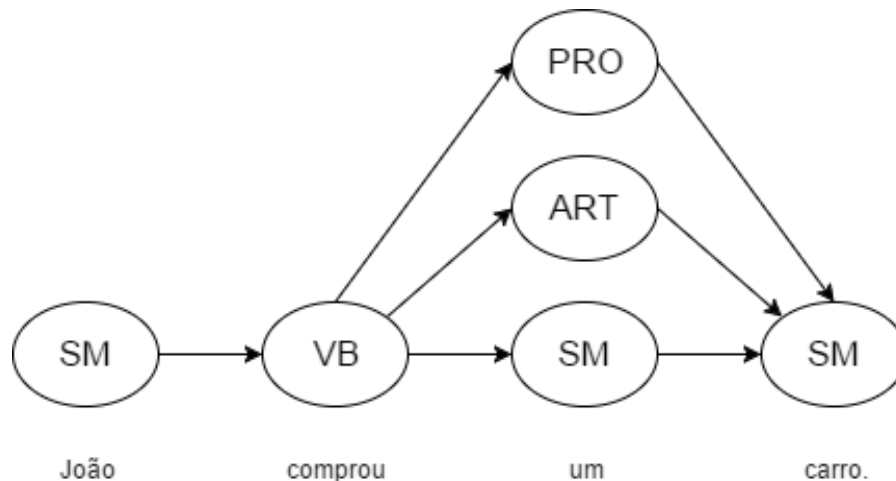


Figura 2.1: Caminhos possíveis de classificação

Para a identificação da classe da palavra “um” é realizada a seguinte equação seguindo o caminho representado pela figura anterior, começando pela palavra João:

$$v_t(j) = v_{t-1}a_{ij}b_j(o_t)$$

Onde que v_t é a probabilidade do caminho atual, v_{t-1} é a probabilidade do caminho anterior, a_{ij} é a probabilidade de transição e $b_j(o_t)$ é a probabilidade de associação.

Portanto a palavra “João”, v_{t-1} é representada pelo valor 1, visto que essa é a primeira palavra e não existe probabilidade de caminho anterior, a_{ij} é a probabilidade de transição entre “Início” e um substantivo, disponível na tabela 2.1 e $b_j(o_t)$ é a probabilidade de associação da palavra João com substantivo, disponível na tabela 2.2:

$$v_t(j) = 1 * 0.3 * 1 = 0.3$$

Para “comprou”, além dos valores retirados das tabelas 2.1 e 2.2, v_{t-1} é representado pelo cálculo anterior do caminho anterior, ou seja, 0.3:

$$v_t(j) = 0.3 * 0.2 * 1 = 0.06$$

Ao efetuar o cálculo de todos os caminhos, para determinar qual a classificação correta de uma palavra, é escolhido o caminho que tem maior probabilidade, no caso apresentado, a palavra “um” é classificada como artigo.

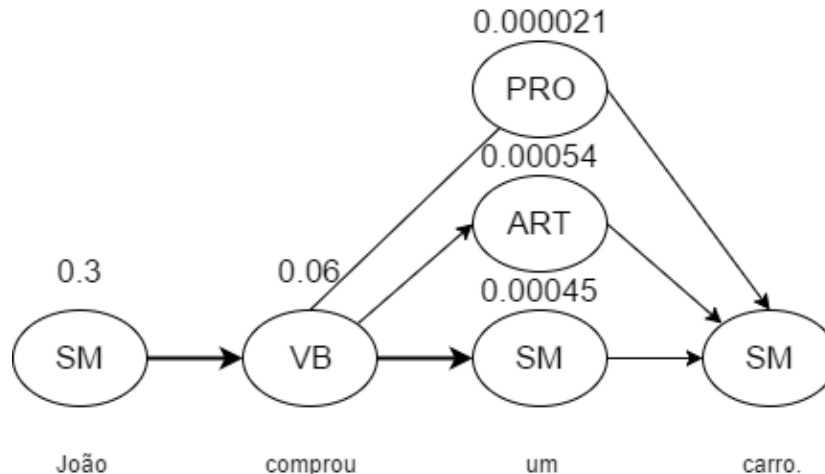


Figura 2.2: Caminhos já decididos de classificação

Como visto, o método simbólico para resolver problemas de Processamento de Linguagem Natural faz uso da criação de regras baseadas no conhecimento humano, enquanto o método estatístico, decide através de cálculos probabilísticos apoiados em estatísticas de um banco de dados para a resolução correta do problema.

2.3 Classificadores

Para o *Natural Language Processing* (NLP) e também para o campo de estatísticas, classificadores são algoritmos que identificam a qual categoria determinado item pertence. Essa classificação é feita a partir de dados já classificados corretamente, ou seja, um *training set*.

2.3.1 Naive Bayes

O classificador Naive Bayes é um classificador baseado no teorema de Bayes com independência entre seus atributos.

O teorema de Bayes é representado da seguinte forma:

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

Supondo que precisamos determinar se o carro que João comprou na frase “João comprou um Focus.” é o modelo sedan ou hatch.

- $P(c|d)$ é a probabilidade de **d** pertencer a classe **c**. Ou seja, a probabilidade do carro Focus ser um sedan.
- $P(d|c)$ é a probabilidade da classe **c** ser **d**. Ou seja, dentre todas as sedans, a probabilidade de um sedan ser um Focus.
- $P(c)$ é a probabilidade da classe **c**. Ou seja, a frequência que sedans aparecem no nosso banco de dados.

- $P(d)$ é a probabilidade de d . Ou seja, a frequência que Focus aparecem no nosso banco de dados.

Levando em consideração que temos o banco de dados representado pela tabela abaixo:

Carro	Categoria
Focus	Sedan
Gol	Hatch
Focus	Hatch
Focus	Sedan
Focus	Hatch
Fox	Hatch
Fiesta	Hatch
Cruze	Sedan
Focus	Hatch

Tabela 2.3: Tabela de Carro e Categoria.

Probabilidade do Focus ser sedan:

$$P(Sedan|Focus) = \frac{P(Focus|Sedan)P(Sedan)}{P(Focus)}$$

$$P(Sedan|Focus) = \frac{2/3 * 3/10}{5/10} = \frac{0,2}{0,5} = 0,4$$

Probabilidade do Focus ser um hatch:

$$P(Hatch|Focus) = \frac{P(Focus|Hatch)P(Hatch)}{P(Focus)}$$

$$P(Hatch|Focus) = \frac{3/6 * 7/10}{5/10} = \frac{0,35}{0,5} = 0,7$$

No caso utilizado como exemplo, o Focus(Atributo ou *Feature*) é um hatch (Rótulo ou *Label*). Porém, caso tenhamos mais um atributo para utilizar na classificação o classificador Naive Bayes não considera nenhuma dependência. Como por exemplo a frase “O carro era ano 2010 e tinha 2 portas” com os seguintes atributos:

O cálculo será feito da seguinte forma:

$$P(d|c) = P(d_1|c) * P(d_2|c) * \dots * P(d_n|c)$$

$$P(Sedan|Focus) = P(Portas = 2|Sedan) * P(Ano = 2010|Sedan) \dots *$$

Carro	Ano	Portas	Categoria
Focus	2010	2	Sedan
Gol	2010	4	Hatch
Focus	2011	4	Hatch
Focus	2011	2	Sedan
Focus	2011	2	Hatch
Fox	2012	4	Hatch
Fiesta	2012	2	Hatch
Cruze	2013	4	Sedan
Focus	2013	2	Hatch

Tabela 2.4: Tabela de anos, carros, portas e categorias.

$$P(\text{Sedan}|\text{Focus}) = 2/3 * 1/3 \dots *$$

Por assumir independência entre seus atributos, os valores obtidos nos cálculos podem ser armazenados no banco de dados e reaproveitados. Por isso, sua performance é considerada incrivelmente boa até mesmo para casos aonde temos forte dependência de atributos (DOMINGOS; PAZZANI, 1997).

2.3.2 *Maximum Entropy*

O classificador *Maximum Entropy* (MaxEnt) tem como característica principal a preferência por modelos de dados uniformes sem efetuar nenhuma suposição injustificada.

Podemos utilizar um exemplo similar ao anterior para demonstrar a lógica do classificador MaxEnt.

“João comprou um carro.”.

Supondo que temos que classificar o tipo de carro comprado em três categorias:

- Hatch.
- Sedan.
- Cupê.

Podemos afirmar que:

$$P(\text{Hatch}) + P(\text{Sedan}) + P(\text{Cupe}) = 1$$

Como só temos essas três possibilidades de classificação no nosso exemplo, o carro só pode ser classificado em uma dessas três possibilidades, ou seja, a soma das três probabilidades deve ser 100% ou 1 essa é a primeira restrição ou *constraint*. Abaixo duas tabelas que satisfazem essa restrição:

Sem nenhum conhecimento prévio da distribuição desses carros, ou seja, a quanti-

Tipo	%
Sedan	33%
Hatch	33%
Cupê	33%

Tabela 2.5: Tabela de Probabilidades A

Tipo	%
Sedan	50%
Hatch	50%
Cupê	0%

Tabela 2.6: Tabela de Probabilidades B

dade de carros comprados por tipo, o classificador assume uma distribuição uniforme das probabilidades, portanto, com maior entropia.

- Hatch - 33%.
- Sedan - 33%.
- Cupê - 33%.

Agora, supondo que a partir do nosso banco de dados conseguimos verificar que em 80% dos casos o veículo comprado era um sedan ou hatch, temos uma nova restrição:

$$P(Hatch) + P(Sedan) = 0.8$$

Podemos novamente ter n distribuições diferentes, porém a distribuição mais uniforme que satisfaz as nossas duas restrições são:

- Hatch - 40%.
- Sedan - 40%.
- Cupê - 30%.

Esse é o princípio da Máxima Entropia utilizado nessa forma de classificação. Primeiro é descoberta a frequência de cada atributo, depois é procurada a distribuição que maximiza a entropia, ou seja, a mais uniforme.

3 FRAMEWORKS

3.1 Natural Language Toolkit

O *Natural Language Toolkit* (NLTK) é um *Framework* para Python criado em 2001 na Universidade de Pensilvânia. Ele contém mais de 50 dicionários e modelos já treinados incluindo:

- *Sentiment Polarity Dataset Version 2.0* - Conjunto de dados já classificados que contém mais de 1000 filmes avaliados de forma positiva e 1000 filmes avaliados de forma negativa.
- *SentiWordNet* - Provém um dicionário com as palavras extraídas do WordNet já classificadas em positividade, negatividade e objetividade.
- *VADER Sentiment Lexicon* - Dicionário especificamente ajustado para análise de sentimentos expressos em mídias sociais.

3.1.1 Análise de Sentimentos

Para a análise de sentimentos, o NLTK já possui implementado as dois classificadores citados anteriormente, *Naive Bayes* e também *MaxEnt*.

Podemos utilizar o classificador Naive Bayes a partir da classe

`nltk.classify.naivebayes.NaiveBayesClassifier` através dos seguintes métodos:

- *classify(featureset)* - Classifica a partir de um conjunto de atributos.
- *most_informative_features(n=100)* - A partir de um classificador treinado, retorna os atributos mais relevantes.
- *train(trainingset)* - Treina um classificador a partir de um *training set*.

Podemos utilizar o classificador MaxEnt a partir do módulo **`nltk.classify.maxent`** através dos seguintes métodos:

- *train(train_toks, algorithm=None, trace=3, encoding=None, labels=None, gaussian_prior_sigma=0, **cutoffs)* - Treina um classificador MaxEnt a partir de um *training set*.

- *train_toks* - *Training set*.
 - *algorithm* - Algoritmo a ser usado para treinar o classificador.
 - *trace* - Nível de detalhe utilizado no log.
 - *encoding*
 - *labels* - Uma lista de possíveis rótulos, se nenhuma for especificada, todos os labels do *training set* serão utilizados.
 - *gaussian_prior_sigma=0* - Somente utilizado no LM-BFGS.
 - *cutoffs* - Argumentos que especificam condições em que o processo será terminado.
- *classify(featureset)* - Classifica a partir de um conjunto de atributos.
 - *explain(featureset, columns=4)* - Mostra uma tabela demonstrando os efeitos de cada atributo e como eles combinam para determinar a probabilidade de cada rótulo.
 - *show_most_informative_features(n=10, show='all')* - A partir de um classificador treinado, retorna os atributos mais relevantes.

Ele também contém um pacote contendo classes úteis para a análise de sentimentos chamado de *nlk.sentiment*. Nesse pacote temos os seguintes módulos:

- Classe *nlk.sentiment.sentiment_analyzer.SentimentAnalyzer* - Ferramentas para facilitar e implementar análise de sentimentos, especialmente para demonstrações e ensino.
- Módulo *nlk.sentiment.util* - Contém diversas classes de demonstrações e utilitários como conversão de *json* para *csv*.
- Módulo *nlk.sentiment.vader* - Ferramenta de Análise VADER.

3.2 Stanford CoreNLP

O Stanford CoreNLP é um conjunto de ferramentas escrito em Java para processamento de linguagem natural. Dentre essas ferramentas, estão incluídos: *Part-of-Speech Tagging* ou classificação gramatical, reconhecimento de entidade e análise de sentimentos. Também possui suporte a diversas linguas além do inglês, como: árabe, chinês, francês, alemão e espanhol.

3.2.1 Análise de Sentimentos

A análise de sentimentos do Stanford CoreNLP é realizada através de um novo modelo de rede neural construído em cima de estruturas gramaticais chamado de *Recursive Neural Tensor Networks* (RNTN). Seu modelo é treinado a partir do *Sentiment Treebank*, um banco de dados que possui 215.154 orações distribuídas em

11.855 árvores de frases com sentimentos já classificados.

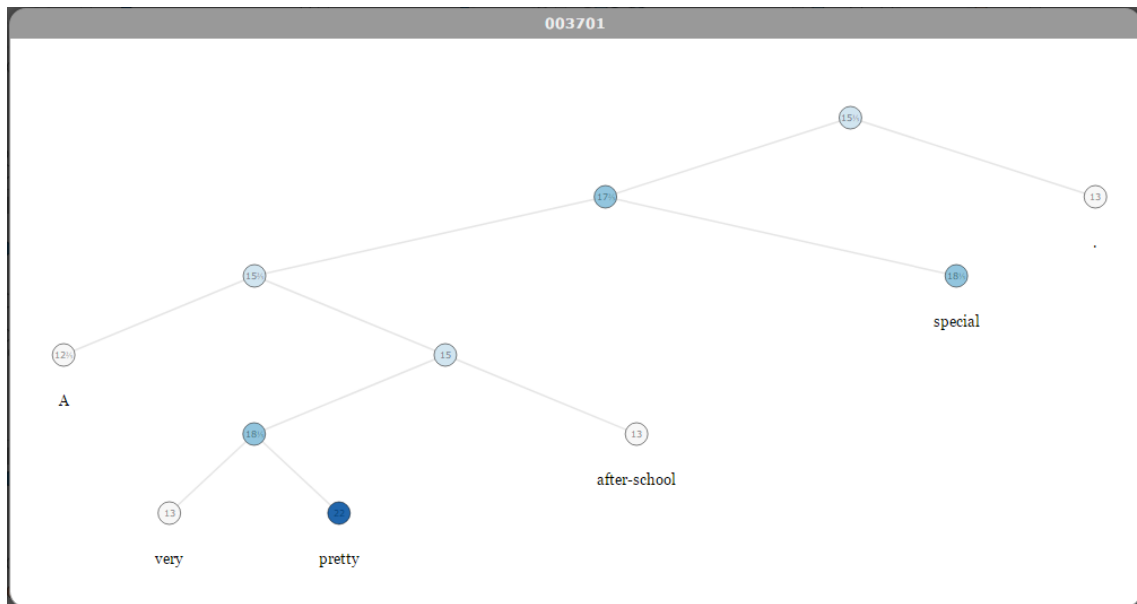


Figura 3.1: Frase já classificada disponível no Sentiment Treebank

A sua utilização pode ser feita de diversas formas, como linha de comando, através de um servidor *web* e através de sua API java:

```

1 public static void main(String[] args) throws IOException {
2     String text = "This World is an amazing place";
3     Properties props = new Properties();
4     props.setProperty("annotators", "sentiment");
5     StanfordCoreNLP pipeline = new StanfordCoreNLP(props);
6
7     Annotation annotation = pipeline.process(text);
8     List<CoreMap> sentences = annotation.get(CoreAnnotations.SentencesAnnotation.class);
9     for (CoreMap sentence : sentences) {
10         String sentiment = sentence.get(SentimentCoreAnnotations.SentimentClass.class);
11         System.out.println(sentiment + "\t" + sentence);
12     }
13 }

```

Figura 3.2: Exemplo de implementação

Como resultado, o console java irá imprimir que a frase é muito positiva ou *Very positive*.

4 REDE SOCIAL REDDIT

O *website* Reddit teve seu início em 2005 como um agregador de conteúdo e atualmente é o vigésimo terceiro *website* mais acessado na internet e o sétimo mais acessado nos Estados Unidos da América (Alexa, 2017). Seus usuários podem enviar links para conteúdos externos ao Reddit ou também mensagens de texto. A partir desse conteúdo enviado, seja ele uma mensagem de texto no próprio Reddit quanto um link a um *website* externo, seus usuários podem votar para cima (*upvote*) ou para baixo *downvote*, influenciando a sua posição no *website*. Esse algoritmo de ordenação de conteúdo é fechado portanto não está disponível para consulta. Além de votar no conteúdo, seus usuários podem enviar comentários como forma de expressar sua opinião.

Esse conteúdo é distribuído em *subreddits* que funcionam como comunidades que abordam certos assuntos. Os usuários podem se inscrever nesses *subreddits* para que seu conteúdo apareça na página inicial. Dentre os *subreddits* mais notáveis se encontram:

- */r/AskReddit* - Local para fazer perguntas gerais para outros usuários. Atualmente com 16.941.544 de inscritos.
- */r/worldnews* - Notícias do mundo. Atualmente com 16.570.606 de inscritos.
- */r/IAmA* - IAmA é um estilização de 'I am a' ou 'Eu sou um'. Local aonde os usuários podem fazer perguntas e respostas ao criador do tópico que se identifica por algo notável, como uma profissão ou algum feito. Atualmente com 16.941.544 de inscritos.

Dentre esses *subreddits* podemos destacar alguns dos tópicos mais acessados no ano de 2016:

- */r/IAmA - We're NASA scientists & exoplanet experts. Ask us anything about today's announcement of seven Earth-size planets orbiting TRAPPIST-1!* - Tópico de perguntas e respostas com cientistas da NASA após a descoberta dos planetas que orbitavam a estrela TRAPPIST-1.

- */r/IAmA - I'm Bill Gates, co-chair of the Bill & Melinda Gates Foundation. Ask Me Anything.* - Tópico de perguntas e respostas com Bill Gates.
- */r/worldnews - Fidel Castro is dead at 90.* - Link para anúncio da morte de Fidel Castro.
- */r/AskReddit - [Serious]South Koreans of Reddit, how did they teach you about the existence of North Korea in School when you were young?serious replies only* - Tópico perguntando para os usuários sul coreanos como que foi ensinado para eles sobre a existência da Coreia do Norte.

4.1 API

O *website* possui uma API *open source* localizada em <https://github.com/reddit/>. Sua documentação é gerada de forma automática a partir do código fonte e podemos encontrar ela em: <https://www.reddit.com/dev/api/>.

REFERÊNCIAS

Alexa. **Alexa Top 500 Global Sites**. <Disponível em: <http://www.alexa.com/topsites/>>. Acesso em: 27 de Fevereiro de 2017.

Apache OpenNLP. **Apache OpenNLP**. <Disponível em: <https://opennlp.apache.org/>>. Acesso em: 27 de Fevereiro de 2017.

BRILL, E. A Simple Rule-based Part of Speech Tagger. In: THIRD CONFERENCE ON APPLIED NATURAL LANGUAGE PROCESSING, Stroudsburg, PA, USA. **Proceedings...** Association for Computational Linguistics, 1992. p.152–155. (ANLC '92).

DOMINGOS, P.; PAZZANI, M. J. On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. **Machine Learning**, [S.l.], v.29, n.2-3, p.103–130, 1997.

HANCOX, P. J. **A brief history of Natural Language Processing**. <Disponível em: http://www.cs.bham.ac.uk/~pjh/sem1a5/pt1/pt1_history.html/>. Acesso em: 02 de Abril de 2017.

JACKSON, P.; MOULINIER, I. **Natural Language Processing for Online Applications**: text retrieval, extraction and categorization. [S.l.]: John Benjamins Publishing Company, 2007.

MANNING, C. D.; SCHÜTZE, H. **Foundations of Statistical Natural Language Processing**. [S.l.]: MIT Press, 1999.

Natural Language Toolkit. **Natural Language Toolkit**. <Disponível em: <http://www.nltk.org/>>. Acesso em: 27 de Fevereiro de 2017.

Spacy. **Spacy**. <Disponível em: <https://spacy.io/>>. Acesso em: 27 de Fevereiro de 2017.

Stanford CoreNLP. **Stanford CoreNLP**. <Disponível em: <http://stanfordnlp.github.io/CoreNLP/>>. Acesso em: 27 de Fevereiro de 2017.