Etiquetação morfossintática de textos em português do Brasil no domínio do *e-commerce*

Márcio Lima Inácio[1,2], **Helena de Medeiros Caseli**[1]

[1]Departamento de Computação – Universidade Federal de São Carlos (UFSCar)
13.565-905 – São Carlos – SP – Brasil

[2]Boolabs – http://boolabs.com.br/

marcio.lima.inacio@gmail.com, helenacaseli@dc.ufscar.br

1. Introdução

Segundo [Loper e Bird 2002], a tarefa de etiquetação morfossintática (em inglês, *Part-of-Speech Tagging* ou *POS Tagging*) se preocupa em classificar os *tokens* de uma sentença de acordo com suas classes morfológicas (substantivo, verbo, adjetivo, entre outros). A automação desta tarefa, por meio de técnicas de aprendizado de máquina, faz com que o processo seja desenvolvido em um tempo reduzido e com menos esforço em comparação com antigos métodos utilizados para a realização da mesma tarefa, os quais envolviam o processo manual de criação de regras, por exemplo. Essas são características desejáveis para áreas como o mercado varejista *online* (*e-commerce*) em franca expansão no Brasil.[1]

A maior parte da informação veiculada em páginas de *e-commerce* é apresentada na forma de texto em língua natural ou imagens e vídeos descritivos dos produtos à venda. O processamento mais inteligente dessas informações, possível por meio de técnicas de aprendizado de máquina, pode garantir a precisão no reconhecimento e classificação de textos e imagens, bem como a vinculação destes ao produto pesquisado.

Com esse pensamento, este artigo traz o resultado do treinamento de uma ferramenta de etiquetação morfossintática aplicada a um corpus de *e-commerce*. Para tanto, na Seção 2 apresenta-se a tarefa de etiquetação morfossintática, com a descrição das principais abordagens adotadas para sua realização, bem como os resultados obtidos para o português do Brasil. Em seguida, na Seção 3 é descrito o corpus de treinamento usado neste trabalho e os resultados obtidos nos experimentos realizados são relatados na Seção 4. Por fim, a Seção 5 aponta direções para trabalhos futuros.

2. Etiquetação Morfossintática

Para ilustrar o processo de etiquetação morfossintática no domínio de interesse, considere o exemplo apresentado na Figura 1, o qual se refere à descrição de um produto (*pen drive*).

```
-_- Armazena_V dados_N de_PRP documentos_N ,_, fotos_N ,_, músicas_N ,_,
vídeos_N e_KC muito_ADV mais_ADV
```

Figura 1. Exemplo de sentença etiquetada morfossintaticamente

[1]Segundo dados do e-bit (www.ebitempresa.com.br/web-shoppers.asp), o mercado varejista *online* no Brasil alcançou um faturamento de R$16,06 bilhões no primeiro semestre de 2014, o que representa um aumento de 26% em relação ao mesmo período de 2013.

Na anotação da Figura 1, os *tokens* etiquetados estão acompanhados de suas etiquetas (após o caractere "_"): N (substantivo), V (verbo), PRP (preposição), ADV (advérbio), KC (conjunção coordenativa). Os sinais de pontuação recebem uma classificação com eles mesmos.

Apesar de a tarefa ser muito intuitiva para grande parte das pessoas, o processo de automatização desse trabalho não é muito trivial. Um dos maiores problemas presentes em todos os idiomas é a presença de ambiguidade entre palavras, como: morro (substantivo) e morro (verbo) em português, *object* (substantivo) e *object* (verbo) em inglês e *sein* (verbo) e *sein* (pronome) em alemão.

Como mostrado em [Branco e Silva 2004], as acurácias de *taggers* (etiquetadores morfossintáticos automáticos) adaptados para o português são: 97,09% para o TBL [Brill 1995], 97,08% para o MXPOST, 96,87% para o TnT [Brants 2000] e 89,97% para o QTag [Mason e Tufis 1997].

Existem várias abordagens para resolver o problema de classificação das palavras, como: método estocástico [Brants 2000], método baseado em regras [Voutilainen 1995] e o etiquetador baseado em transformação [Brill 1995]. Destes será aqui abordado apenas o método estocástico, selecionado para investigação neste trabalho por sua simplicidade, seu bom desempenho e por possuir uma implementação no *Natural Language Toolkit* (NLTK) com licença que permite reuso comercial, uma necessidade do projeto maior no qual este trabalho está inserido.

O NLTK[2] [Loper e Bird 2002] possui quatro anotadores. O anotador mais básico é o *DefaultTagger*, que seleciona a melhor etiqueta para os *tokens* de acordo com o tipo de caractere utilizado. Por exemplo, se o *token* for um número, ele o classificará como numeral. Sua precisão é por volta de 20-30%, apresentando uma baixíssima performance se utilizado sozinho. Outro anotador disponível no NLTK é o *UnigramTagger*, que calcula a probabilidade de um *token* receber uma dada etiqueta com base nas frequências obtidas de um corpus de treinamento. De acordo com [Nau 2010], o cálculo é feito a partir de um modelo probabilístico simples onde:

$$P(t_i|w) = \frac{c(w,t_i)}{c(w,t_1) + ... + c(w,t_k)} \tag{1}$$

Sendo w a palavra a ser classificada e $t_1, ..., t_k$ uma lista das *tags* (etiquetas) possíveis. $c(w,t_i)$ indica quantas vezes a correspondência da palavra w com a etiqueta t_i apareceu no corpus de treinamento. Por exemplo, se a palavra "morro", em português, foi etiquetada 15 vezes como verbo e 65 como substantivo no corpus de treinamento, então $P(verbo|morro) = \frac{15}{80} = 0,19$ e $P(substantivo|morro) = \frac{65}{80} = 0,81$.

Nesta estratégia, a etiqueta a ser selecionada é aquela com maior probabilidade de ser a correta de acordo com o corpus de treinamento. Isto é, as entradas de w e t_i que maximizam a função $P(t_i|w)$. No caso do exemplo anterior, uma nova ocorrência de "morro" seria etiquetada como substantivo, pois $(substantivo, morro)$ é a opção que maximiza a equação de acordo com o corpus de treinamento.

Como todo modelo probabilístico, a performance do *UnigramTagger* depende

[2]Disponível em: `http://www.nltk.org/`. Acesso em: 28 jun. 2015.

muito da qualidade do corpus de treinamento fornecido, por esse motivo, optou-se pela criação de um corpus próprio para o caso de *e-commerce* (como descrito na Seção 3). Estendendo o contexto de análise, o modelo *BigramTagger* utiliza o mesmo método do *UnigramTagger* porém também levando em consideração a etiqueta da palavra anterior. Sua implementação é mais complexa, porém ela gera uma precisão maior por depender, também, do contexto onde as palavras se encontram e não apenas da palavra em si.

Seguindo o pensamento de [Carlberg e Kann 1999], seja $W = w_1, ..., w_n$ uma sequência de palavras (sentença) e $T = t_1, ..., t_n$ uma sequência de etiquetas, então uma representação para o modelo do etiquetador de bigrama seria:

$$\prod_{i=1}^{n} P(t_i|t_{i-1})P(w_i|t_i) \tag{2}$$

Onde, $P(w_i|t_i) = \frac{c(w_i,t_i)}{c(t_i)}$ e $P(t_i|t_{i-1}) = \frac{c(t_{i-1},t_i)}{c(t_{i-1})}$.

Sendo $c(t_i)$ a frequência da etiqueta t_i no corpus, $c(w_i, t_i)$ a frequência de $w_i|t_i$ (uma palavra w_i associada à etiqueta t_i) no corpus e $c(t_{i-1}, t_i)$ a frequência de $t_{i-1}t_i$ (a presença de duas classes gramaticais t_{i-1} e t_i em sequência) no corpus; a equação 2 mostra que a probabilidade da etiqueta ser a escolhida depende da etiqueta do *token* imediatamente anterior a ela na sentença, ou seja, do contexto em que aquela etiqueta se encontra. Por exemplo, realizando um aprendizado com base na sentença apresentada na Figura 2, a probabilidade de uma nova aparição da palavra "caminho" precedida de artigo (ART) ser classificada como substantivo (N) é maior do que a chance de ser classificada como verbo (V).

O_ART caminho_N ,_, por_PRP o_ART qual_PRO-KS-REL eu_PROPESS caminho_V todos_PROADJ os_ART dias_N ,_, é_V muito_ADV movimentado_ADJ ._.

Figura 2. Sentença "O caminho, pelo qual eu caminho todos os dias, é muito movimentado." etiquetada

Seguindo a mesma lógica, como mostra [Hess et al. 2000], a implementação pode ser estendida para um *TrigramTagger* como:

$$\prod_{i=1}^{n} P(t_i|t_{i-2}, t_{i-1})P(w_i|t_i) \tag{3}$$

Vale salientar que quanto maior o número de *tokens* anteriores verificados pelo anotador automático, maior o tempo de execução da ferramenta e maior a necessidade de uma ampla variedade de sequências de etiquetas possíveis no corpus de treinamento (abrangência) pois os contextos presentes no texto a ser classificado podem não se apresentar no treino gerando, assim, erros de etiquetação.

Neste trabalho, utilizou-se o *TrigramTagger* com um *backoff* para o *BigramTagger*, isto é, caso a ferramenta não encontre a melhor etiqueta utilizando o *Trigram*, ela tentará classificar a palavra com o uso do *Bigram*. De modo semelhante, o *BigramTagger* pode chamar o *Unigram*, que por sua vez, pode chamar o *DefaultTagger* que atribuirá a etiqueta padrão (no caso deste trabalho, a etiqueta de substantivo N).

3. Corpus de treinamento

O corpus de treinamento está composto por textos retirados de páginas de *e-commerce*. Os textos do corpus de treinamento foram previamente etiquetados com a ferramenta Aelius[3] [Alencar 2010]. O Aelius, segundo o próprio desenvolvedor, é uma biblioteca para a linguagem Python que utiliza o pacote NLTK para realizar o pré-processamento de textos em português do Brasil, sendo a etiquetação morfossintática uma função inclusa. Esse pacote possui diversos etiquetadores implementados, dos quais o MXPOST[4], treinado com o corpus Mac-Morpho, foi o escolhido para gerar a etiquetação inicial.

A partir dessa etiquetação inicial, os textos foram revisados por dois linguistas utilizando a ferramenta BRAT[5] [Stenetorp et al. 2012] e com uma concordância de $\kappa =$ 0,97 [Carletta 1996]. O corpus de treinamento resultante é composto por 316 arquivos, num total de 48.510 *tokens* com as quantidades de etiquetas especificadas na Tabela 1.

Tabela 1. Quantidades de etiquetas presentes no corpus de treinamento

Etiqueta	Quantidade
Substantivo	13.083
X (pontuação, por exemplo)	8.801
Preposição	6.590
Adjetivo	4.203
Artigo	3.670
Verbo	2.903
Numeral	1.864
Conjunção coordenativa	1.660
Nome próprio	1.563
Pronome adjetivo	1.074
Advérbio	979
Pronome pessoal	439
Verbo auxiliar	400
Verbo no particípio	368
Pronome relativo	317
Conjunção subordinativa	228
Palavra denotativa	181
Pronome substantivo	137
Pronome subordinado	24
Advérbio relativo	11
Interjeição	8
Advérbio subordinado	7
TOTAL	48.510

4. Resultados

A avaliação da ferramenta treinada como descrito anteriormente foi realizada com base na porcentagem de etiquetas corretas (acurácia), calculada como:

[3]Disponível em: `http://aelius.sourceforge.net/`. Acesso em: 28 jun. 2015.

[4]Disponível em: `http://www.inf.ed.ac.uk/resources/nlp/local_doc/MXPOST.html`. Acesso em: 28 jun. 2015.

[5]Disponível em: `http://brat.nlplab.org/index.html`. Acesso em: 28 jun. 2015.

$$A = \frac{|W_t \cap W_c|}{|W_t|} \tag{4}$$

Onde, W_t é o conjunto de etiquetas atribuídas pela ferramenta e W_c é o conjunto de etiquetadas anotadas pela equipe de linguistas (corpus de referência/treinamento).

Primeiramente, utilizou-se o *k-fold cross-validation* [Clark et al. 2010][6] para estimar uma média de valores em que as anotações geradas foram corretas. A acurácia em cada *fold* é medida pela fórmula 4. Ao final, a média das acurácias de todas as 10 iterações dá uma estimativa do desempenho da ferramenta. Para a ferramenta treinada como descrito anteriormente, a estimativa obtida com 10-*fold cross-validation* foi de 83,45% de acurácia.

Em seguida, a ferramenta também foi avaliada em um corpus de teste composto por 10 arquivos e 1.096 *tokens* que não estavam presentes no corpus de treinamento. Nesse caso, a acurácia obtida com a ferramenta treinada no domínio do *e-commerce* foi de 90,69% contra 32,8% obtida pelo Aelius (modelo disponível para o português do Brasil treinado com o Mac-Morpho). Esse ganho fica evidente ao analisar a saída produzida pelo Aelius (a) e pelo etiquetador gerado neste trabalho após treinamento no corpus de *e-commerce* (b), com referência à anotação produzida pelos linguistas (c), como mostra a Figura 3.[7]

a	Informações_N :_: Jarra_NPROP de_NPROP 1,25_NUM L_NPROP Acionamento_N automático_ADJ Motor_NPROP reversível_ADJ
b	Informações_N :_X Jarra_N de_PREP 1,25_NUM L_X Acionamento_N automático_N Motor_N reversível_N
c	Informações_N :_X Jarra_N de_PREP 1,25_NUM L_X Acionamento_N automático_ADJ Motor_N reversível_ADJ

Figura 3. Sentença anotada com as ferramentas Aelius (a) e a produzida neste trabalho (b), comparadas com a anotação de referência produzida pela equipe de linguistas (c)

5. Trabalhos futuros

Como trabalho futuro prevê-se a possibilidade de realizar um re-treinamento da ferramenta utilizando conjuntamente o corpus Mac-Morpho e o corpus de *e-commerce* gerado neste trabalho com o intuito de aumentar a abrangência do corpus de treinamento. Além disso, a ferramenta de etiquetação produzida neste trabalho será utilizada nas pesquisas que seguem dentro do projeto de extensão do qual este trabalho faz parte.

[6]Esta abordagem consiste em dividir o corpus anotado em k partes $\{f_1, ..., f_k\}$, no caso deste artigo $k = 10$. Então, cada *fold* f_i é anotado com o etiquetador treinado com os demais *folds*, resultando num total de k iterações.

[7]Vale mencionar que no caso dos caracteres de pontuação, etiquetados como X no corpus de referência e com o próprio caractere pelo Aelius, considerou-se correta a etiquetação do Aelius quando no corpus de referência a etiqueta era X.

Agradecimentos

Esse trabalho foi realizado em parceria e com suporte da Boolabs (`http://boolabs.com.br/`) e faz parte do projeto de extensão "Processamento de texto e de imagem na descrição *online* de produtos" (UFSCar/FAI #23112.003944/2014-81).

Referências

Alencar, L. F. (2010). Aelius: uma ferramenta para anotação automática de corpora usando o nltk. In *IX Encontro de Linguística de Corpus*.

Branco, A. e Silva, J. (2004). Evaluating solutions for the rapid development of state-of-the-art pos taggers for portuguese. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*, pages 507–510.

Brants, T. (2000). Tnt - a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000*, pages 224–231.

Brill, E. (1995). A simple rule-based part of speech tagger. In *Proceedings of the third conference on Applied natural language processing (ANLC'92)*, pages 152–155.

Carlberg, J. e Kann, V. (1999). Implementing an efficient part-of-speech tagger. In *Nada, Numerical Analysis and Computing Science Royal Institute of Technology*, pages 815–832.

Carletta, J. C. (1996). Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics*, 22(2):249–254.

Clark, A., Fox, C., e Lappin, S. (2010). *The Handbook of Computational Linguistics and Natural Language Processing*. Wiley-Blackwell.

Hess, M., Clematide, S., e Schneider, G. (2000). *Trigrams'n'Tags: Methoden der Korpusanalyse und des statistischen Parsings in NEGRA*.

Loper, E. e Bird, S. (2002). Nltk: the natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics*, volume 1, pages 63–70.

Mason, O. e Tufis, D. (1997). Probabilistic tagging in a multi-lingual environment: Making an english tagger understand romanian. In *Proceedings of the Third TELRI European Conference*, pages 165–168.

Nau, D. (2010). Part-of-speech tagging. http://www.cs.umd.edu/ñau/cmsc421/part-of-speech-tagging.pdf.

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., e Tsujii, J. (2012). Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107.

Voutilainen, A. (1995). A syntax-based part of speech analyser. In *Proceedings of the Seventh Conference of the European Chapter of the Association for Computational Linguistics*, pages 157–164.