Análise de Métodos e Ferramentas para Reconhecimento de Palavras Relevantes em Microblogs

Alternative Title: Analysis of Methods and Tools for Relevant Words Recognition in Microblogs

Danielly Sorato Fábio B. Goularte Silvia M. Nassar Renato Fileto
Dep. de Informática e Estatística, Universidade Federal de Santa Catarina (UFSC)
C.P. 476, 88040-900, Florianópolis-SC, Brasil.

danielly.sorato@grad.ufsc.br fabio.bif@posgrad.ufsc.br silvia.nassar, r.fileto@ufsc.br

RESUMO

Extrair informações acuradas dos enormes volumes de dados, muitos dos quais não estruturados, gerados em mídias sociais é um grande desafio atualmente, mas com diversas aplicações relevantes, muitas delas ainda latentes. Um dos primeiros e mais decisivos passos deste processo de extração de informação é o reconhecimento de palavras relevantes em textos. Este artigo apresenta um estudo comparativo de métodos e ferramentas para reconhecer palavras relevantes em postagens de microblogs. Dentre diversas ferramentas analisadas, cinco delas foram selecionadas para experimentos com 100 mil tweets. Tais experimentos mostraram alta variabilidade dos resultados de ferramentas distintas, o que sugere a necessidade de melhorias.

Palavras-Chave

Reconhecimento de palavras relevantes, Mídias sócias, Ferramentas de PLN.

ABSTRACT

Extracting accurate information from the huge volumes of data, much of them unstructured, generated in social media is currently a big challenge. However, it has several relevant applications, some of them latent yet. One of the first and most decisive steps in this information extraction process is the recognition of relevant words in texts. This article presents a comparative study of methods and tools for recognizing relevant words on microblog posts. Among several analyzed tools, five have been selected for experments with 100,000 tweets. These experiments showed high variability of the results generated by different tools, suggesting a need for improvements.

Categories and Subject Descriptors

I.2.7 [**Artificial Inteligence**]: Natural Language Processing – *language parsing and understanding*, *text analysis*.

General Terms

Algorithms, Measurement, Performance, Experimentation.

SBSI 2016, May 17–20, 2016, Florianópolis, Santa Catarina, Brazil.

Keywords

Relevant words recognition, social media data, natural language processing tools, text mining.

1. INTRODUÇÃO

A massiva utilização das Tecnologias da Informação (TICs) nas últimas décadas tem levado a um enorme crescimento na quantidade de informação disponível na World Wide Web [1] [53]. Por exemplo, milhões de *tweets* são gerados diariamente. A análise de palavras relevantes presentes nos textos de conjuntos de dados tão extensos como postagens em microblogs (e.g., Twitter) e redes sociais (e.g., Facebook) é inviável sem o uso de ferramentas apropriadas. Ferramentas utilizadas neste contexto necessitam fazer bom uso de recursos computacionais para processar em tempo hábil, além de gerarem resultados confiáveis para textos provenientes de mídias sociais, sujeitos a muitos ruídos e problemas de má formação linguística.

Mídias sociais apresentam um novo e desafiador estilo de texto para as tecnologias de processamento computacional. Por exemplo, textos de *tweets* têm natureza informal, tamanho limitado, semântica pouco definida e muito ruído (e.g., gírias, erros de gramática, abreviações, entre outros). Tais fatores tornam o processo de extração e classificação de informações do conteúdo de microblogs particularmente complexo, sendo difícil obter bons resultados por meio de métodos e ferramentas de Processamento de Linguagem Natural (PLN) [2] e alternativas clássicas. Além disso, diversas aplicações que usam dados de mídias sociais precisam analisar eficientemente a sintaxe e a semântica de grandes quantidades de dados gerados constantemente, cujos padrões linguísticos podem ser dinâmicos.

Contudo, apesar das dificuldades para extrair informações de postagens feitas em mídias sociais e, particularmente microblogs, pode-se encontrar nos *posts* componentes relevantes e semanticamente ricos, tais como entidades nomeadas (menções a pessoas, locais, organizações, datas, etc.) [3], além de palavras com classificação gramatical (substantivos, adjetivos, verbos, etc.) [2], que podem auxiliar no entendimento do que se passa com os usuários que fazem as postagens. Esses componentes linguísticos podem permitir, por exemplo, identificar eventos ocorridos ao longo do espaço e do tempo [4, 5, 6], locais frequentemente visitados [7] ou até alimentar com informação precisa métodos de análise de informação [8, 9], mineração de dados [10], análise de sentimento [11], recomendação [12], sistemas de TICs e parsers.

A análise do conteúdo em mídias sociais é uma área emergente no cenário de tecnologia e pesquisa. Atualmente, as ferramentas existentes para o reconhecimento de palavras relevantes apresentam-se implementadas em diferentes linguagens de programação e funcionam sob as plataformas mais comuns. Tais ferramentas podem gerar saídas em diversos formatos (e.g., XML, JSON, RDF, N-Triplas) funcionar como serviço remoto via Internet ou localmente instaladas, entre diversas outras possibilidades. O desempenho computacional e a qualidade dos resultados gerados pelas ferramentas dependem de fatores como a natureza do texto analisado, o domínio de aplicação, a técnica empregada no processo de reconhecimento, a arquitetura do sistema computacional e restrições de processamento [13].

Este artigo apresenta uma análise de métodos e ferramentas capazes de reconhecer palavras relevantes no conteúdo textual de *tweets* em Língua Portuguesa. Este estudo enfatiza métodos e ferramentas que utilizam PLN, pois a análise do texto traz a necessidade de pré-processamento do conteúdo, e técnicas como fonetização, normalização e *POS tagging* [2], podem ser aplicadas para melhorar os resultados. Os experimentos buscaram comprovar se existe diferença significativa entre os resultados gerados por ferramentas distintas, pois isso sugere a necessidade de estudos mais aprofundados e sistemáticos sobre o seu desempenho, particularmente com dados do Twitter, e também potencialmente de outras fontes análogas.

O restante deste artigo está organizado conforme segue. A Seção 2 descreve os problemas e abordagens. A Seção 3 apresenta uma análise comparativa das ferramentas encontradas na literatura relacionada a reconhecimento de palavras relevantes em textos. A Seção 4 relata e discute os resultados de experimentos e as ferramentas selecionadas. Finalmente, a Seção 5 tece as conclusões e enumera trabalhos futuros.

2. FUNDAMENTOS

Esta seção primeiramente define e classifica problemas relacionados ao reconhecimento de palavras relevantes em textos e, posteriormente, delineia as principais abordagens da literatura.

2.1 Definição do Problema

O Reconhecimento de Palavras Relevantes (*Relevant Words Recognition* - RWR) consiste em identificar e classificar entidades nomeadas e outros componentes com valor sintático e/ou semântico significativo em textos [14]. RWR é considerado em diversas áreas de pesquisa, sendo que cada uma dessas se utiliza de diferentes abordagens para resolver seus próprios subproblemas. RWR pode ser subdividido em extração de *tokens*, Reconhecimento de Entidades Nomeadas (*Named Entity Recognition* - NER), Desambiguação de Entidades Nomeadas (*Named Entity Desambiguation* – NED), *Shallow Parsing* e *POS Tagging*. A Figura 1 ilustra esses subproblemas de RWR, que podem ser vistos como um processo onde somente a tokenização é uma tarefa obrigatória, mas cujos resultados são potencialmente muito melhores com todas essas tarefas realizadas de maneira ordenada e, por vezes, cooperativa.

A extração de *tokens* pode ser feita com relativa facilidade, usando autômatos finitos, porém tem importância fundamental, pois a tokenização é geralmente o primeiro passo para NER, NED, *Shallow Parsing* e *POS Tagging*. Não obstante, muita pesquisa e desenvolvimento em NER e NED têm sido realizada, resultando em ferramentas que variam bastante com relação às técnicas utilizadas. NER [3] é uma tarefa de processamento de linguagem natural que consiste em extrair menções a entidades nomeadas (e.g., pessoas, lugares, instituições). NER frequentemente alimenta outras tarefas mais complexas, tais como *Relation Extraction* [15, 50], *Question Answering* [16] e NED [17]. NED visa ligar menções de entidades nomeadas detectadas à sua definição em uma base de conhecimento. Técnicas para tratar NER/NED fazem uso de regras sintáticas, dicionários de nomes e/ou *machine learning*. Todavia, tais tarefas só identificam e classificam um subconjunto de palavras relevantes. *Shallow Parsing* e *POS Tagging* [2], que por sua vez, são tarefas de PLN bastante similares, que podem ser usadas para complementar o processo de extração de palavras relevantes, identificando componentes morfossintáticos, tais como verbos (que podem ajudar no entendimento de ações, por exemplo), adjetivos (que podem ser úteis para detectar polaridade, entre outras possibilidades). *Shallow Parsing* (também conhecida por *chunking*) identifica os principais constituintes de uma sentença (e.g., frases, orações, grupos de verbos, grupos de substantivos), mas sem especificar sua estrutura interna e o papel de cada constituinte específico. *POS Tagging*, por outro lado, anota cada palavra em um texto com sua classe morfossintática, com base tanto em definições quanto nos contextos desses componentes no texto. *POS Tagging* é um problema central em PLN. A identificação precisa dos elementos morfossintáticos de uma sentença é de grande importância, pois ao classificar uma única palavra incorretamente, pode-se gerar erros de processamento subsequentes [54]. *POS Taggers* que usam *machine learning* podem explorar dados de treinamento rotulados para se adaptar a novos gêneros ou mesmo línguas, por meio de aprendizagem supervisionada [18].

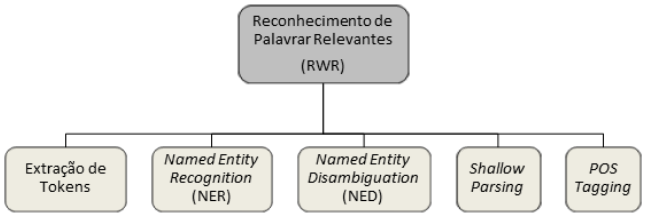


Figura 1. Subproblemas de RWR.

2.2 Abordagens

RWR pode ser realizado com diversas abordagens. Dicionário de nomes é uma das mais fáceis para NER, onde cada entrada corresponde a um par <*nome de superfície, entidade nomeada*>. A busca por um nome de superfície em um dicionário seleciona todas as entidades com tal nome [19]. Apesar da sua simplicidade, essa abordagem é restrita (só reconhece entidades previstas no dicionário) e não trata ambiguidade (um nome de superfície associado a várias entidades). Assim, geralmente se utilizam dicionários de nomes em conjunto com outras técnicas.

O Processamento de Linguagem Natural (PLN) é um campo de pesquisa interdisciplinar baseado em conhecimentos de várias áreas, principalmente da Linguística e da Inteligência Computacional (IC), que tem oferecido aplicações para diversos propósitos (e.g., tradutores automáticos, revisores gramaticais, sistemas de apoio à escrita, sumarizadores e simplificadores de texto, para citar alguns). Um dos objetivos deste novo campo é fazer com que os computadores possam executar tarefas úteis que envolvem a linguagem humana, como a comunicação homem-

máquina, ou mesmo a comunicação entre humanos por meio do processamento de texto ou discurso [2].

De maneira geral, a pesquisa em PLN volta-se, essencialmente, aos seguintes níveis de processamento da língua natural: fonologia, morfologia, sintaxe, semântica e pragmática [20]. O nível fonológico está relacionado à análise dos fonemas da língua; o morfológico relacionado aos morfemas utilizados na construção de palavras, subdividido em grupo dos afixos (prefixos, sufixos, radicais, entre outros), grupo da análise morfológica (gênero, número, grau, modos e tempos das conjugações verbais, entre outros), classes gramaticais (substantivo, verbo) e subnível léxico (formas canônicas); o nível sintático está relacionado ao arranjo dos lexemas em uma frase; o nível semântico está relacionado ao significado dos lexemas e das frases; e o nível pragmático está relacionado ao propósito do falante ao produzir frases.

A Mineração de texto (MT) é uma área de pesquisa da Ciência da Computação que tenta analisar uma grande quantidade de informação através da combinação de técnicas de Mineração de dados (*Data mining*) [21], Aprendizado de Máquina (*Machine Learning*) [22, 23], PLN, recuperação de informação [24] e gestão do conhecimento [25]. De forma análoga a mineração de dados, a mineração de texto procura extrair informações úteis a partir de fontes de dados através da identificação e exploração de padrões interessantes. No caso da mineração de texto, as fontes de dados são coleções de documentos, e os padrões interessantes são encontrados não entre os registros formalizados do banco de dados, mas nos dados textuais não estruturados dos documentos.

As abordagens mais citadas para RWR até o momento são baseadas em padrões linguísticos e programação convencional. Todavia, ferramentas que usam técnicas baseadas em aprendizado de máquina tais como: classificação de sequências, *ensemble learning* e *bootstrapping* têm gerado melhores resultados atualmente. Classificação de sequências é uma técnica robusta que classifica uma sentença por vez. Dada uma sequência de *tokens*, gera-se uma lista de marcadores que indicam a presença de menções. Uma vez que o classificador de sequências é treinado, utiliza-se o algoritmo de Viterbi para calcular uma sucessão de estados que mais provavelmente gera uma coleção de observações, através de programação dinâmica. A classificação de sequências pode se beneficiar do uso de *bootstrapping* quando não existe grande quantidade de dados anotados [19]. *Ensemble learning* [26] combina resultados de várias ferramentas usando algum esquema de votação. Dessa forma, teoricamente os resultados finais são potencialmente melhores, porém obtidos com maiores custos (e.g., tempo de processamento, uso de memória).

3. ANÁLISE DAS FERRAMENTAS

Esta seção descreve as principais ferramentas avaliadas no âmbito desta pesquisa e compara suas características. Os critérios de seleção dessas ferramentas a partir da literatura técnico-científica pesquisada foram: prominência, grau de recomendação, qualidade da documentação e aderência a padrões da Web, principalmente os relacionados a enriquecimento semântico e sistemas baseados em conhecimento. Essas ferramentas ainda não foram classificadas em categorias, pois algumas têm soluções para diversos problemas e utilizam técnicas combinadas.

FOX (*Federated knOwledge eXtraction framework*) [27]: é um *framework* de código aberto que implementa serviços Web RESTful para NER e NED. FOX usa AGDISTIS, Weka e um Perceptron multicamadas [28]. AGDISTIS[1] [29] é um *framework* de código aberto para desambiguação de palavras relevantes que consegue relacionar entidades de bases de conhecimento na forma de dados ligados e fazer a desambiguação usando a DBpedia [30]. Weka [31] é um framework para mineração de dados. Os algoritmos nele disponíveis baseiam-se principalmente em aprendizado de máquina e podem ser aplicados diretamente a um conjunto de dados usando sua interface homem-máquina ou a partir de código Java. Weka contém ferramentas para pré-processamento de dados, classificação, regressão, *clustering*, regras de associação e visualização. FOX usa *ensemble learning* para combinar os resultados de algumas de das melhores ferramentas atuais para NER e NED. Os autores afirmam que a ferramenta consegue atingir um percentual de medida-F de aproximadamente 95.23%.

O *workflow* realizado pela ferramenta consiste em quatro passos. No primeiro passo, de pré-processamento, o usuário fornece como entrada uma URL, texto com *tags* HTML ou texto simples. Se a entrada for uma URL, FOX envia uma requisição a URL fornecida para receber os dados de entrada. Para todos os formatos de entrada, FOX remove as *tags* HTML e detecta frases e *tokens*. No segundo passo é realizado o NER, via combinação dos resultados de quatro outras ferramentas: *Stanford NER*, *Illinois Named Entity Tagger*, *Ottawa Baseline Information Extraction* e *Apache OpenNLP Name Finder*. Na terceira etapa é feita a ligação de entidades usando AGDISTIS. Na última etapa é realizada a serialização dos resultados. O usuário pode escolher dentre os seguintes formato de saída: JSON-LD, N-Triplas, RDF/JSON, RDF/XML, Turtle, TriG e N-Quádruplas. É possível usar a ferramenta por meio de serviço Web[2], bem como uma API (a qual possui *bindings* para Java e Python). FOX é de código aberto, o que possibilita fazer um *fork*, disponível no Github[3].

T-NER [32]: é uma ferramenta capaz de alcançar medida-F de 59% ao classificar entidades nomeadas em três classes (pessoa, local e organização), enquanto o *Stanford NER* alcançou apenas 29% para tal tarefa sobre os mesmos dados. O *workflow* do T-NER consiste nas fases de segmentação e de classificação. Na fase de segmentação, cada sentença do texto é rotulada segundo a notação BIO (*Beginning, the Inside and Outside of text segments*) [33]. Um *Condition Random Field* (CRF) é responsável pelo aprendizado e inferência a partir de características como a ortografia, o contexto textual e um dicionário de nomes e classes extraído do *Freebase* [34]. Na fase de classificação, a ferramenta LabeledLDA é utilizada para modelar tópicos de acordo com 10 classes de entidades extraídas da *Freebase* [35]. O LabeledLDA é treinado com um *dataset* de 60 milhões de *tweets*. O T-NER introduz novas ferramentas de *POS Tagging* [36], *Shallow Parsing* [37], classificação de capitalização e reconhecimento de palavras relevantes exclusivas para *tweets* [32, 19].

Stanford NER (*Stanford Named Entity Recognizer*) [38]: é um *framework* de código aberto para NER e NED. A ferramenta baseia-se em classificação de sequências e utiliza dez características locais para treinar um CRF. O *framework* foi implementado em Java pelo grupo de PLN de Stanford, e rotula nomes (de pessoas, empresas, genes e proteínas, entre outros) no texto. Pode-se usar o código da StanfordNER para construir

1. Disponível em <https://github.com/AKSW/AGDISTIS>
2. Disponível em <http://139.18.2.164:4444/demo/index.html#!/demo>
3. Código fonte disponível em <https://github.com/AKSW/FOX>

modelos de sequência para reconhecimento de palavras relevantes ou qualquer outra tarefa. Também é possível usá-lo via GUI, linha de comando, usando a API ou serviço e *standalone*. A distribuição atual permite o acesso a todos os recursos do *pipeline* do Stanford CoreNLP, acessados através da classe *NERClassifierCombiner*. Experimentos com o StanfordNER resultaram em 85,51% na medida-F ao utilizar o *dataset* da CoNLL[4] e 92,29% ao utilizar o *dataset* da CMU[5].

Illinois Named Entity Tagger [33]: é um *tagger* que marca texto simples com palavras relevantes. Atualmente as palavras podem se encaixar nas categorias pessoa, organização, localização e variados, ou ainda dezoito outros tipos, se a classificação for realizada com base no corpus OntoNotes. Ela usa *gazetteers* extraídos da Wikipedia, modelos de classe palavra derivados de texto não rotulado e recursos não-locais expressivos.

NERD [39]: é um *framework* que propõe unificar as saídas de dez diferentes extratores PLN publicamente disponíveis na Web. A abordagem baseia-se no desenvolvimento da ontologia NERD[6], que provem uma interface para anotação de elementos e uma API REST usada para unificar as saídas. NERD é uma aplicação web que funciona sobre várias ferramentas de PLN. Sua arquitetura segue os princípios *REpresentational State Transfer* (REST) e fornece um acesso HTML web para os usuários e uma API para que computadores possam fazer intercâmbio de conteúdo em formato JSON ou XML. Ambas as interfaces são movidas pela *engine* NERD REST. A *engine* NERD REST e as ferramentas de PLN usados pelo NERD fazem o reconhecimento e desambiguação de palavras relevantes apontando URIs para objetos do mundo real, como eles poderiam ser definidos na Web de dados.

LX-Tagger [40]: desenvolvido pelo Natural Language and Speech Group da Universidade de Lisboa, a ferramenta é focada no processamento da Língua Portuguesa, e composta pelos módulos de *Sentence chunker, POS-tagger, Tokenizer, Nominal featurizer e Nominal lemmatizer*. O *Sentence chunker* é um autômato de estado finito, onde as transições são ativadas por sequências de caracteres especificadas na entrada e os símbolos emitidos correspondem aos limites da sentença e do parágrafo. O *POS-tagger* utilizado é o Brant's TnT tagger, treinado com 90% de um corpus de 280,000 *tokens* [41]. O *Tokenizer* identifica *tokens* a partir dos espaços em branco. O *Nominal featurizer* é responsável por atribuir *tags* para inflexão (gênero e número) e grau (diminutivo, superlativo e comparativo) para palavras de categorias morfossintática nominais. O *Nominal lemmatizer* é responsável pela tarefa de atribuir adjetivos e nomes comuns a uma forma normalizada (masculino singular). A ferramenta pode ser baixada livremente, porém não tem código aberto.

GATE: é uma arquitetura, ambiente de desenvolvimento e estrutura para construção de sistemas que processam a linguagem humana [42]. Ele foi desenvolvido na Universidade de Sheffield e é usado em projetos para extração de informações em vários idiomas. Sua arquitetura é baseada em componentes, sendo os três principais: *Language Resources*, *Processing Resources* e *Visual Resources*. O componente *Language Resources* armazena tipos de dados linguísticos, como documentos e ontologias e provê serviços para acessá-los. *Processing Resources* trata de recursos cuja função é principalmente programática ou algorítmica, como um *POS-tagger* ou *parser*. O componente *Visual Resources* trata somente de elementos gráficos de interface. O GATE provê um conjunto de recursos de processamento reutilizáveis para tarefas comuns de processamento de linguagem natural. Juntos, esses recursos formam o ANNIE (um sistema para extração de informação), mas também podem ser usados individualmente ou em conjunto acoplado com novos módulos, a fim de criar novas aplicações. Os diferentes componentes de pré-processamento do *framework* (e.g., *tokenizer*, *gazetteer* e *POS-tagger)* foram projetados para ser facilmente adaptáveis a novas línguas.

GATE Twitter POS tagger [18]**:** é um *tagger* desenvolvido por especificamente para o processamento de *tweets*. Para reduzir o impacto da dispersão de dados a ferramenta usa *vote-constrained bootstrapping* (uma variação de bootstrapping, detalhes sobre o funcionamento da técnica estão no artigo que descreve o *tagger*). Também são utilizados métodos para o tratamento de erros característicos do gênero e gírias. A ferramenta é destinada à *tweets* na língua inglesa e contém uma instância do Stanford Tagger[7]. Ela tem várias opções de utilização (*plugin* do GATE, *Standalone*, etc) e também está incluso em um *pipeline* para processamento de linguagem natural de código aberto customizado para textos de microblogs chamado TwitIE [43].

OpenNLP [44]**:** é um conjunto de ferramentas baseadas em aprendizagem de máquina visando o processamento de linguagem natural. Possui suporte para tarefas como tokenização, segmentação de frases, *POS tagging*, extração de entidades nomeadas, *chunking*, *parsing* e resolução de correferência. O *POS-tagger* marca os *tokens* com o seu tipo de palavra correspondente com base no próprio *token* bem como no seu contexto. A ferramenta usa um modelo de probabilidade para prever a *tag* correta para o *token* a partir do conjunto de *tags*. No site do OpenNLP, existem modelos pré-treinados para dinamarquês, alemão, inglês, holandês, sueco e português. Porém, a ferramenta também permite o treinamento de modelos em outras linguagens. Isso pode ser feito por meio de um conjunto de dados que consiste em coleções de textos anotados com *tags*.

FreeLing [45]**:** é uma biblioteca que realiza tarefas como identificação da linguagem, tokenização divisão de sentenças, análise morfológica, detecção e classificação de entidades nomeadas, *POS tagging*, S*hallow parsing* e resolução de correferência. Inclui dois módulos diferentes capazes de realizar *POS tagging*. O primeiro é a classe *tagger_hmm*, baseada no método de [41]. O segundo módulo é o *relax_tagger*, um sistema híbrido capaz de integrar conhecimentos sobre estatística e códigos feitos a mão, seguindo [46]. A aplicação pode instanciar qualquer um dos dois módulos.

TreeTagger [47]**:** desenvolvida por Helmut Schmid, no Instituto de Linguística Computacional da Universidade de Stuttgart, usa *decision trees* para lidar com dados esparsos e tem sido aplicada com sucesso em vários idiomas. A *decision tree* é construida recursivamente a partir de um conjunto de treinamento de trigramas, usando uma versão modificada do algoritmo ID3. Além de *tagger*, a ferramenta também pode ser usada como *chunker* para as línguas inglesa, espanhola, francesa e alemã.

O Quadro 1 sintetiza as principais características das ferramentas estudadas. Nota-se que Java e Python são as linguagens predominantes para acoplamento dessas ferramentas, linha de comando é a forma de uso mais comum e algumas ferramentas

4. Disponível em: <http://cnts.uia.ac.be/conll2003/ner/>
5. Disponível em: <http://nlp.shef.ac.uk/dot.kom/resources.html>
6. Disponível em <http://nerd.eurecom.fr/>
7. Disponível em <http://nlp.stanford.edu/software/tagger.shtml>

estão disponíveis como *open source*. Abordagens baseadas em PLN ainda são preponderantes, embora alguns dos melhores resultados venham de ferramentas que usam aprendizagem de máquina (FOX, T-NER, StanfordNER, GATE Twitter POS tagger TreeTagger), muitas vezes em combinação com outras técnicas. Formato texto tem sido o mais comum para entrada e saída, embora algumas ferramentas processem entradas em HTML e ofereçam saídas em formatos padrão da Web, sendo somente o FOX aderente a padrões da Web semântica.

Quadro 1. Principais características das ferramentas. PT = Português, EN = Inglês, ES = Espanhol, IT = Italiano, DE = Alemão, FR = Francês, NL = Holandês, RU = Russo, SV = Sueco, ZH = Chinês, AR = Árabe, CEB = Cebuano, HI = Hindi, RO = Romeno, DA = Dinamarquês, CA = Catalão, GL = Galego, CY = Galês, BG = Búlgaro, SW = Swahili, SK = Eslovaco, SL = Esloveno, PL = Polonês, LA = Latim, ET = Estoniano.

Ferramenta e Ling.	Forma / Condição de Uso	Abordagens	Entrada	Saída	Idioma
FOX (Java/Python)	API / *open source*	*Ensemble learning*	txt, HTML, URL	RDF(XML,JSON), JSON-LD, TriG, N-Triple, N-Quads, Turtle	EN, FR, NL, DE
T-NER (Python)	linha de comando / *open source*	Dicionário de nomes, *bootstrapping*	txt, HTML, URI	txt	EN
StanfordNER (Java/ Javascript/Pearl/PHP/ Ruby/C#)	*standalone*, API ou linha de comando / *open source*	Classificação de sequências	txt, HTML	XML, txt	EN, ZH, DE
Illinois Named Entity Tagger (Java)	*standalone*	PLN	txt	txt	EN
NERD (Java/Python/ Nodejs/Ruby)	*open source*	PLN	txt, HTML, URI	JSON	EN, FR, DE, IT, PT, ES, RU, SV
LX-Tagger (Java)	linha de comando	PLN	txt, HTML	txt	PT
Gate (Java)	localmente instalável, API / *open source*	PLN	txt, HTML, XML, email, SGML, RTF	XML, XHTML, RTF, txt, email	EN, FR, DE, IT, RU, ZH, AR, CEB, HI, RO
GATE Twitter POS tagger (Java)	*plugin*, *standalone*, linha de comando / *open source*,	PLN, *vote-constrained bootstrapping*	txt, HTML	txt	EM
OpenNLP (Java)	linha de comando, API	PLN	txt	txt	DA, DE, EN, ES, NL, PT, SV
Freeling (Java, Phyton)	linha de comando, API	PLN	txt	tag	PT, RU, EN, ES, IT, FR, CA, GL, CY
TreeTagger (Perl)	linha de comando, Interface Gráfica (Windows)	PLN, *decision tree*	txt	txt	DE, EN, FR, IT, NL, ES, BG, RU, PT, GL, ZH, SW, SK, SL, LA, ET, PL PT, EN

4. EXPERIMENTOS

Esta seção descreve os experimentos realizados para a avaliação de ferramentas selecionadas sobre uma coleção de *tweets*.

4.1 Seleção das ferramentas

As ferramentas selecionadas precisam processar uma grande quantidade de texto, principalmente, em Língua Portuguesa, classificar as palavras, ter código fonte aberto ou permitir a utilização em pesquisa, além da possibilidade de integração a outras aplicações. Usando esses critérios foram selecionadas para os experimentos: LX-Tagger, TreeTagger, TwitIE, FreeLing e OpenNLP. Todas elas realizam a atribuição de uma classe gramatical, na forma de *tag*, a cada palavra do texto de entrada (*part-of-speech tagging* - *POS Tagging*). Por exemplo, a frase "*Cheguei ontem e apaguei*", após passar pelo *POS Tagging* pode ser representada como "*Cheguei/V ontem/ADV e/CJ apaguei/V*", com a *tag V* indicando verbo, *ADV* – advérbio e *CJ* – conjunção. Priorizou-se a análise dos resultados de *POS Tagging* por este ser um componente fundamental na análise linguística, contribuindo para a recuperação de informação, a desambiguação de palavras e a redução de *parsers* (análise sintática automática de frases em termos de suas funções gramaticais). Ademais, *POS Tagging* não se restringe a análise de um grupo específico de palavras, pois anota todos *tokens*. Isso o torna uma tarefa com forte potencial de aplicação em pesquisas sobre RWR e enriquecimento semântico de dados desenvolvidas no nosso laboratório. Existem outras ferramentas de PLN para o português (e.g., corretor gramatical CoGrOO) que implementam *POS Tagging*, porém para incluí-las neste estudo seria preciso alterar a forma de execução.

4.2 Dataset

Os experimentos utilizaram *tweets* postados entre 17/12/2015 e 04/01/2016 no território brasileiro, o que resultou em cerca de 67 milhões de *tweets*, ocupando um arquivo texto de 7,3 GB. Em virtude do volume de informação, foram selecionados 100 mil *tweets* referentes aos primeiros dias de dezembro de 2015. A Tabela 1 apresenta estatísticas dos dados e o Quadro 2 alguns exemplos de *tweets*.

Tabela 1. Estatística do dataset

Descrição	Quantificação
Tweets	100000
Tokens	1122561
Caracteres por palavra:	
Média	4,49
Desvio padrão	3,18
Tamanho arquivo (MB)	~6

Quadro 2. Exemplos de *tweets*

1. *A folga do cara https://t.co/lxvzZfkmKK;* ***2.*** *Daqui a pouco vou assistir Campeonato Indiano no EI Plus;* ***3.*** *RT @_: odeio gente q coloca só k nas frases*

Os valores na Tabela 1 foram gerados pelas ferramentas Word Smith tool 5.0 e AntConc 3.2.1. Ambas fazem análise linguística e calculam estatísticas relativas a um texto. A primeira oferece versão demo enquanto a segunda é gratuita. Os valores na Tabela 1 podem mudar caso o conjunto de dados seja processado por outras ferramentas ou com configurações distintas de parâmetros.

4.3 Resultados e discussão

As ferramentas receberam como entrada os textos dos *tweets* e geraram como saída um arquivo texto com cada palavra relevante neles encontrada, anotada com a sua categoria morfossintática. A anotação variou conforme a especificidade da classificação morfossintática da ferramenta. Por exemplo, a TreeTagger usa as seguintes notações para adjetivo: AO – adjetivo ordinal, AOS – adjetivo ordinal superlativo, AQ – adjetivo qualificativo, AQA – adjetivo qualificativo aumentativo, AQC – adjetivo qualificativo diminutivo, AQS – adjetivo qualificativo superlativo; enquanto a TwitIE usa: JJ – adjetivo, JJR – adjetivo comparativo, JJS – adjetivo superativo. Em razão de tais diferenças, subcategorias foram agrupadas em grandes categorias.

Os resultados obtidos estão descritos na tabela e figuras a seguir. A Tabela 2 mostra a quantidade de *tokens* encontrados pelas ferramentas, conforme a distribuição das categorias morfossintáticas. Note que mesmo o número total de *tokens* detectado foi diferente para cada ferramenta. Isso ocorre em função dos diferentes algoritmos adotados nas ferramentas para realizar os processos de tokenização e *POS Tagging*. A FreeLing encontrou mais *tokens* que as demais porque possui funcionalidades para lidar com texto coloquial, e também por implementar vários módulos que identificam uma grande variedade de *tokens*, conseguindo enriquecer os *tweets* com novas informações linguísticas. Assim como a FreeLing, a TreeTagger faz lematização, gerando anotações mais detalhada e saídas de ~26MB e ~19MB, respectivamente. A arquivo gerado pela FreeLing foi o maior porque a ferramenta aplica a lematização a todos os *tokens*, inclusive símbolos, pontuação e números. Já a TreeTagger aplica a lematização apenas aos *tokens* identificados como palavras. LX-Tagger, TwitIE e OpenNLP geraram arquivos de ~10MB.

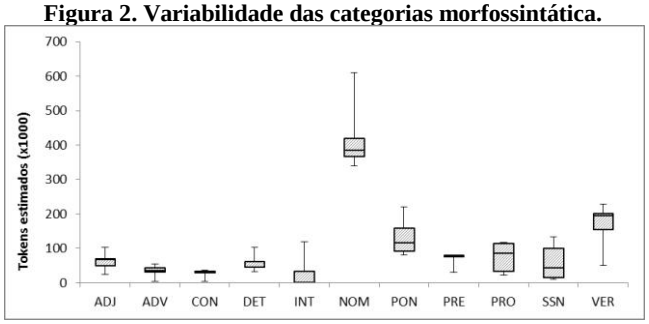
Tabela 2. Estimativa de tokens pelas ferramentas para cada categoria morfossintática. ADJ = Adjetivo, ADV = Advérbio, CON = Conjunção, DET = Determinante, INT = Interjeição, NOM = Nome, PON = Pontuação, PRE = Preposição, PRO = Pronome, SSN = Sigla/Símbolo/Número/Outros, VER = Verbo.

Categoria	Ferramentas				
	Lx-Tagger	**TreeTagger**	**TwitIE**	**FreeLing**	**OpenNLP**
ADJ	66816	102524	23609	49424	69228
ADV	31021	34231	3268	42566	53838
CON	36387	29937	2806	29378	31135
DET	102556	44620	44873	60522	31477
INT	1811	3590	118761	3053	-
NOM	367322	384437	610171	418822	338965
PON	95574	137535	-	220414	80778
PRE	81101	78293	29942	79572	75972
PRO	31893	85949	21937	117280	112746
SSN	132282	10312	99637	43522	13700
VER	194926	228687	50544	154172	200662
Total	**1141689**	**1140115**	**1005548**	**1218725**	**1008501**

As Figuras 2 e 3 mostram as diferenças entre as médias de ocorrências das categorias morfossintáticas nos resultados das ferramentas. O gráfico do tipo *boxplot* apresentado na Figura 2 permite analisar a variabilidade dos resultados (diferença entre o terceiro e o primeiro quartil), a mediana (linha central) e os máximos e mínimos (calda inferior e superior). A ordem decrescente de frequência média das categorias (NOM, VER, PON, PRO, PRE, ADJ, SSN, DET, ADV, INT e COM) condiz com a literatura, onde se demonstra que substantivo e verbo são as categorias mais frequentes e importantes para análise linguística. A mediana das categorias em muitos casos está deslocada do centro, e a cauda muito longa ou curta, denotando falta de simetria. Isso está relacionado a grande diferença dos resultados das ferramentas para cada categoria. Observa-se que para as categorias NOM, PONT, PRO, SSN e VER, a variabilidade é alta, em ordem, 51500, 66380, 80853, 85937 e 46490 tokens.

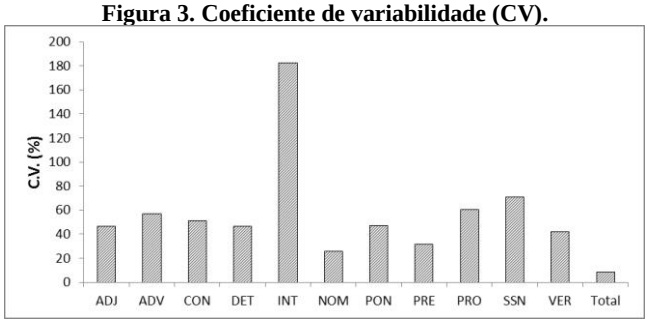
Incluir a lematização das palavras juntamente com a anotação morfossintática ajuda no enriquecimento semântico, mas deve-se ter o cuidado de verificar o contexto da aplicação. Se a quantidade de dados for grande, a inclusão de mais componentes linguísticos no processo de *POS Tagging* pode acarretar em maior tempo de processamento e tamanho do arquivo de saída.

Figura 2. Variabilidade das categorias morfossintática.



Outra forma de verificar a variabilidade dos resultados é usada na Figura 3, que ilustra o coeficiente de variabilidade (desvio-padrão / média) das categorias. Quanto menor tal coeficiente, mais homogêneo é o conjunto de dados. Ao observar-se a variabilidade do total de *tokens* (8%), pode-se afirmar que as ferramentas apresentam um desempenho de processamento semelhante. Porém, quando o foco volta-se para as categorias, todos os coeficientes foram maiores que 25%, valor esse considerado limite para afirmar que não existe homogeneidade.

Figura 3. Coeficiente de variabilidade (CV).



Estes resultados mostram que apesar das ferramentas utilizadas terem apresentado alguns bons resultados na literatura, neste estudo, há evidências de que o problema de *POS Tagging* precisa ser melhor investigado para textos de microblogs. Além disso, em uma breve análise verificou-se palavras anotadas com categoria incorreta. Uma solução para os problemas constatadas nos

resultados das ferramentas seria considerar tarefas que melhoram a qualidade dos dados na etapa de pré-processamento, como a normalização, ou ainda, subdividir o conjunto de dados em um corpus de teste e treinamento e repetir os experimentos. A normalização desta linguagem também pode ser crucial para facilitar o processamento de texto e melhorar o desempenho de ferramentas de PLN aplicadas a mídias sociais. Contudo, dependendo do propósito da aplicação, a normalização torna-se difícil, por exemplo, uma URI pode ser considerada como um *token* ou mesmo como vários (quanto utiliza-se palavras com sentido próprio na composição).

5. CONCLUSÃO

Atualmente, apesar da maior disponibilidade de ferramentas para processamento de texto, muitos problemas ainda persistem: baixa interoperabilidade (principalmente entre ferramentas comerciais); limitações no compartilhamento de recursos (em função da dependência de plataformas específicas para execução); falta de padrões; disponibilização apenas de verões *online* para teste e restrições de capacidade de processamento ao lidar com grandes volumes de dados. Outro ponto crítico é a escassez de recursos apropriados para processar textos naturais de mídias sociais e em certos idiomas, como é o caso da Língua Portuguesa.

A análise comparativa aqui apresentada de ferramentas para o RWR em textos distingue-se de trabalhos afins [48, 49, 51, 52] pelas seguintes contribuições: (i) cobertura de uma variedade de subproblemas, métodos e ferramentas de áreas de pesquisa tradicionalmente separadas; (i) foco na análise de postagens em mídias sociais usando a língua portuguesa e (iii) resultados experimentais mais abrangentes, analisando a alta disparidade dos resultados gerados por ferramentas distintas. O estudo do estado da arte mostra que propostas recentes tendem ao uso de técnicas baseadas em aprendizado de máquina, pois estas começam a gerar melhores resultados. Porém, há muita variação, tanto no que se refere a resultados retornados quanto ao tempo de processamento. Apesar deste trabalho não apresentar medidas de tempo de processamento, por limitações de espaço e por considerar ferramentas com formas de uso distintas, notou-se nos experimentos que algumas ferramentas geraram resultados em apenas alguns minutos, enquanto outras, depois de algumas horas. Possivelmente, um dos fatores que influencia na diferença do tempo de processamento é a complexidade do método utilizado, porém, comprovar tal hipótese requer acesso ao código das ferramentas ou ao menos documentação mais precisa e detalhada.

Os resultados obtidos sugerem a necessidade de aprofundar pesquisas visando melhorar o desempenho de ferramentas para RWR em postagens de mídias sociais. Apenas características técnicas e resultados publicados em outros estudos não são suficientes para afirmar quais ferramentas melhor resolvem, por exemplo, o problema de *POS Tagging* de *tweets.* Para avançar tais investigações, um corpus que servirá como padrão-ouro está sendo aperfeiçoado e os resultados das ferramentas serão aferidos segundo medidas como precisão, cobertura, medida-f e intervalo de confiança, em trabalhos futuros.

6. AGRADECIMENTOS

Este trabalho foi suportado pelo GBD - Grupo de Banco de Dados do INE/UFSC. Danielly Sorato é bolsista do CNPq – Brasil na modalidade de iniciação científica e Fábio B. Goularte é bolsista CAPES na modalidade de doutorado.

7. REFERÊNCIAS

[1] Habib, M. B., and Keulen, V. M. (2014). Information extraction for social media. ACL.

[2] Jurafsky, D. and Martin, J. H. (2008). Speech and Language processing: An introduction to natural language processing. 2nd Ed., Pearson Prentice Hall.

[3] Tjong Kim Sang, E. F., and De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In Proc. CNLL at HLT-NAACL (vol. 4, pp. 142-147). ACL.

[4] Wang, W., and Stewart, K. (2015). Spatiotemporal and semantic information extraction from Web news reports about natural hazards. Comp., Env. Urb. Syst., 50, 30-40.

[5] Anantharam, P., Barnaghi, P., Thirunarayan, K., & Sheth, A. (2015). Extracting city traffic events from social streams. ACM TIST (vol. 6, n. 4, pp. 43).

[6] Xia, C., Hu, J., Zhu, Y., Naaman, M. (2015). What Is New in Our City? A Framework for Event Extraction Using Social Media Posts. In: Advances in KDD Mining, Springer, 16-32.

[7] Fileto, R., May, C., Renso, C., Pelekis, N., Theodoridis, Y. (2015) The Baquara2 knowledge-based framework for semantic enrichment and analysis of movement data. Data & Knowledge Eng (vol. 98, pp 104-122.

[8] Lev, B., and Thiagarajan, S. R. (1993). Fundamental information analysis. JA research, 190-215.

[9] Sacenti, J.A.P., Salvini, F., Fileto, R., Raffaetà, A., Roncato, A. (2015) Automatically Tailoring Semantics-Enabled Dimensions for Movement DW. In: DaWaK 2015: 205-216

[10] Berry, M. J., and Linoff, G. (1997). Data mining techniques: for marketing, sales, and customer support. John Wiley & Sons, Inc.

[11] Das, T. K., Acharjya, D. P., and Patra, M. R. (2014). Opinion mining about a product by analyzing public tweets in Twitter. In ICCCI (pp. 1-4). IEEE

[12] Pazzani, M. J., and Billsus, D. (2007). Content-based recommendation systems. In The adaptive web (pp. 325-341). Springer Berlin Heidelberg.

[13] Jabeen, S., Shah, S., and Latif, A. (2013). Named entity recognition and normalization in tweets towards text summarization. In ICDIM (pp. 223-227). IEEE.

[14] Downey, D., Broadhead, M., and Etzioni, O. (2007). Locating Complex Named Entities in Web Text. In IJCAI (vol. 7, pp. 2733-2739).

[15] Doddington, G. R., Mitchell, A., Przybocki, M. A., Ramshaw, L. A., Strassel, S., and Weischedel, R. M. (2004). The Automatic Content Extraction Program-Tasks, Data, and Evaluation. In LREC (vol. 2, pp. 1).

[16] Voorhees, E. M. (1999). The TREC-8 Question Answering Track Report. In Trec (vol. 99, pp. 77-82).

[17] Derczynski, L., Maynard, D., Rizzo, G., van Erp, M., Gorrell, G., Troncy, R., Petrak, J., Bontcheva, K. (2015). Analysis of named entity recognition and linking for tweets. Inf. Process. Manage (vol. 51, n. 2, pp. 32-49).

[18] Derczynski, L., Ritter, A., Clark, S., and Bontcheva, K. (2013). Twitter Part-of-Speech Tagging for All: Overcoming Sparse and Noisy Data. In RANLP (pp. 198-206).

[19] Klein, D. (2015). Estudo de técnicas e ferramentas aplicáveis a mídias sociais para reconhecimento e desambiguação de entidades nomeadas (TCC). Univ. Federal de Santa Catarina.

[20] Willingham, D. T. (2007). Cognition: The thinking animal. Englewood Cliffs, NJ: Pearson/Prentice Hall.

[21] Han, J., Kamber, M., Pei, J. (2006) Data mining: concepts and techniques. Morgan kaufmann.

[22] Goldberg, D. E., and Holland, J. H. (1988). Genetic algorithms and machine learning. Machine learning (vol. 3, n. 2, pp. 95-99).

[23] Witten, I.H., and Frank, E. (2011). Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 664pp.

[24] Manning, C. D., Raghavan, P., and Schütze, H. (2008). Introduction to information retrieval (vol. 1, n. 1, p. 496). Cambridge: Cambridge University Press.

[25] Feldman, R., and Sanger, J. (2007). The text mining handbook: advanced approaches in analyzing unstructured data. Cambridge University Press.

[26] Dietterich, T. G. (2002). Ensemble Learning. The Handbook of Brain Theory and Neural Networks, MA Arbib.

[27] Speck, R., and Ngomo, A. C. N. (2014). Named entity recognition using FOX. In Proc. of the 2014 Int. Conf. (vol. 1272, pp. 85-88). CEUR-WS. org.

[28] Ruck, D. W., Rogers, S. K., Kabrisky, M., Oxley, M. E., and Suter, B. W. (1990). The multilayer perceptron as an approximation to a Bayes optimal discriminant function. Neural Networks, IEEE Trans. on (vol. 1, n. 4, pp. 296-298).

[29] Usbeck, R., Ngomo, A. C. N., Röder, M., Gerber, D., Coelho, S. A., Auer, S., and Both, A. (2014). AGDISTIS-graph-based disambiguation of named entities using linked data. In The Semantic Web – ISWC 2014 (pp. 457-471). Springer International Publishing.

[30] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). Dbpedia: A nucleus for a web of open data (pp. 722-735). Springer Berlin Heidelberg.

[31] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. ACM SIGKDD expl. newsl., (vol. 11, n. 1, pp. 10-18).

[32] Ritter, A., Clark, S., and Etzioni, O. (2011). Named entity recognition in tweets: an experimental study. In Proc. of the Conf. on Empirical Methods in NLP (pp. 1524-1534). ACL.

[33] Ratinov, L., and Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In Proceedings CCNLL (pp. 147-155). ACL.

[34] Bollacker, K., Cook, R., and Tufts, P. (2007). Freebase: A shared database of structured general human knowledge. In AAAI (vol. 7, pp. 1962-1963).

[35] Ramage, D., Hall, D., Nallapati, R., and Manning, C. D. (2009). Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In Proc. CEMNLP (vol. 1, pp. 248-256). ACL.

[36] Voutilainen, A. (2003). Part-of-speech tagging. The Oxford handbook of computational linguistics (pp. 219-232).

[37] Sha, F., and Pereira, F. (2003). Shallow parsing with conditional random fields. In Proc. NC of the ACL: HLT (vol. 1, pp. 134-141). ACL

[38] Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into IE systems by gibbs sampling. In Proc. Annual Meeting on ACL (pp. 363-370).

[39] Rizzo, G., Troncy, R., Hellmann, S., and Bruemmer, M. (2012). NERD meets NIF: Lifting NLP Extraction Results to the Linked Data Cloud. LDOW, 937.

[40] Branco, A., and Silva, J. R. (2006). A suite of shallow processing tools for portuguese: Lx-suite. In Proc. Eleventh Conf. of the EU Chapter of the ACL (pp. 179-182).

[41] Brants, T. (2000). A statistical Part-of-Speech tagger. In Proceedings of the Sixth ANLP (pp. 224-231).

[42] Bontcheva, K., Maynard, D., Tablan, V., and Cunningham, H. (2003). Gate: A unicode-based infrastructure supporting multilingual information extraction. Proc. IE for Slavonic and other Central and Eastern EU Languages, Borovets, Bulgaria.

[43] Bontcheva, K., Derczynski, L., Funk, A., Greenwood, M. A., Maynard, D., and Aswani, N. (2013). TwitIE: An Open-Source IE Pipeline for Microblog Text. In RANLP (pp. 83-90).

[44] Baldridge, J. (2005). The opennlp project. URL: http://opennlp.apache.org. (acessado em janeiro de 2016).

[45] Padró, L., and Stanilovsky, E. (2012). Freeling 3.0: Towards wider multilinguality. In LREC2012.

[46] Padró, L., (1998). A Hybrid Environment for Syntax-Semantic Tagging. PhD thesis. Dept. LSI. Universitat Politècnica de Catalunya.

[47] Quilan, J.R., 1983. Learning efficient classification procedures and their application to chess end games. Machine Learning: An AI Approach, 1.

[48] Amaral, D.O.F., Fonseca, E.B., Lopes, L., Vieira, L. (2014) Comparative Analysis of Portuguese Named Entities Recognition Tools. In Int. CRE (vol. 1, pp. 2554-2558).

[49] Rizzo, G., Troncy, R. (2011). NERD: Evaluating Named Entity Recognition Tools in the Web of Data. In Workshop on WEKEX'11, Bonn, Germany, 1-16.

[50] Bowman, M., Debray, S. K., and Peterson, L. L. (1993). Reasoning about naming systems. ACM TOPLAS (vol. 15, n. 5, pp. 795-825).

[51] Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. JMLR, 3, 1289-1305.

[52] Liu, X., Zhang, S., Wei, F., and Zhou, M. (2011). Recognizing named entities in tweets. In Proc. of the 49th Annual Meeting of the ACL: HLT. (vol. 1, pp. 359-367).

[53] Diniz, H. B., Silva, E. C., and Gama, K. S. (2015). Uma Arquitetura de Referência para Plataforma de Crowdsensing em Smart Cities. In Proc. of the BSIS (pp. 87-97).

[54] Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In Proc. CNAC ACL: HLT (vol. 1, pp. 173-180)