UNIVERSIDADE FEDERAL DE SANTA CATARINA
BIBLIOTECA UNIVERSITÁRIA

Suelen Cardoso Fenali

PRÉ-PROCESSAMENTO DE *TWEETS* VISANDO MELHORAR RESULTADOS DE *NERD*

Florianópolis
2016

Ficha de identificação da obra elaborada pelo autor
através do Programa de Geração Automática da Biblioteca Universitária
da UFSC.

A ficha de identificação é elaborada pelo próprio autor
Maiores informações em:
http://portalbu.ufsc.br/ficha

Suelen Cardoso Fenali

PRÉ-PROCESSAMENTO DE *TWEETS* VISANDO MELHORAR RESULTADOS DE *NERD*

Este trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de “Bacharel em Sistemas de Informação”, e aprovado em sua forma final pelo Departamento de Informática e Estatística da Universidade de Santa Catarina.

Florianópolis, 01 de dezembro de 2016.

_____
Prof. Dr. Frank Siqueira
Coordenador do Curso
Universidade Federal de Santa Catarina

Banca Examinadora:

_____
Prof. Dr. Renato Fileto
Orientador
Universidade Federal de Santa Catarina

_____
Prof.ª Dr.ª Carina Friedrich Dorneles
Universidade Federal de Santa Catarina

_____
Prof.ª Dr.ª Sílvia Modesto Nassar
Universidade Federal de Santa Catarina

Dedico aos meus pais, pois para mim eles são a razão de tudo...

AGRADECIMENTOS

Agradeço primeiramente aos meus pais e irmã, por todo apoio que sempre recebi. Pelos esforços empenhados para que eu pudesse sempre manter o foco nos estudos acima de qualquer outra coisa. Obrigada pai e mãe, por serem a base de quem eu sou e por acreditarem na minha capacidade quando muitas vezes eu mesma deixo de acreditar. Obrigada por sempre terem priorizado a educação e por terem trabalhado tanto para que eu pudesse ter um ensino de qualidade.

Agradeço a todos os professores que contribuíram para a minha formação. Agradeço de forma especial ao meu professor e orientador Renato Fileto, que me auxiliou e direcionou no desenvolvimento desse trabalho. Agradeço por ter sido não somente um professor, mas sim um mentor nessa minha jornada tanto acadêmica como profissional. Obrigada pelos puxões de orelha necessários, pelos elogios que sempre incentivam, por todo conhecimento compartilhado e por ser esse excelente professor, pesquisador e amigo.

Agradeço aos meus colegas de trabalho da HeyCheff, por toda compreensão sobre a minha ausência nessa reta final de curso. Obrigada por serem tão parceiros e por fazerem do nosso ambiente de trabalho um lugar sempre agradável.

Agradeço a todos os amigos que estiveram ao meu lado nessa trajetória. Aos que me ouviram reclamar, que me acalmaram, que viram meus *snaps* de noites viradas, aos que contribuíram para possíveis atrasos me convidando para festas e aos que me ajudaram não me convidando para sair. Agradeço de forma especial à Pâmella e aos Thiagos por dividirem comigo não apenas um apartamento, mas também muitos momentos divertidos. Agradeço ao Quarteto, por todos os anos de amizade e apoio. Agradeço ao Luan que está entre os amigos, mas é como se fosse

família. Agradeço a Naiane e Marina, que mesmo sendo amigas recentes sei que vieram para ficar e participaram de perto dessa desse momento importante.

Enfim, agradeço a todas as pessoas que passaram em minha vida e contribuíram para que eu fosse quem sou e para que conseguisse alcançar meus objetivos.

RESUMO

O enriquecimento semântico das postagens em mídias sociais pode trazer diversos benefícios em aplicações. Todavia, as técnicas e ferramentas de extração de informação atualmente presentes na literatura não trabalham adequadamente com dados provenientes dessas fontes, os quais estão sujeitos a ruídos diversos. Este trabalho propõe um método para filtragem de *tweets* baseado em normalização léxica visando diminuir ruídos e obter melhores resultados nas tarefas de reconhecimento e desambiguação de entidades nomeadas (*NERD*). Para realizar tal proposta, este trabalho apresenta uma revisão do estado-da-arte sobre o reconhecimento e desambiguação de entidades nomeadas com foco em mídias sociais, bem como revisa propostas para uma etapa preliminar de filtragem de *tweets.* De modo a verificar a qualidade do método proposto, foram realizados experimentos com a ferramenta FOX e observou-se um aumento de 5% no número de entidades nomeadas reconhecidas após a normalização léxica dos *tweets*.

Palavras-chave: Mídias Sociais. Reconhecimento de Entidades Nomeadas. Desambiguação de Entidades Nomeadas. FOX. Pré-processamento de dados de microblogs. Tweets.

ABSTRACT

The semantic enrichment of posts in social media can bring several benefits in applications. However, the information extraction techniques and tools currently available in the literature are not prepared to work with data from these sources, which are very affected by noises. This work proposes a method for filtering tweets based on lexical normalization to reduce noise and obtain better results in the recognition and naming entity disambiguation (NERD) tasks. To accomplish this, this paper presents a state-of-the-art review of the recognition and disambiguation of social media-focused entities, as well as reviews proposals for a preliminary tweeting filtering step. In order to verify the quality of the proposed method, experiments were performed with the FOX tool and a 5% increase in the number of named entities recognized after the lexical normalization of the tweets was observed.

Keywords: Social Medias. Named Entity Recognition. Named Entity Disambiguation. Microblogs data pre-processing. Tweets.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ACM	*Association for Computing Machinery*
AF	*Autômatos Finitos*
AM	*Aprendizado de Máquina*
API	*Application Programming Inteface*
BILOU	*Begin, In, Last, Out, Unit*
BIO	*Begin,In,Out*
CoNLL	*Conference on Computational Natural Language Learning*
CRF	*Conditional Random Fields*
ER	*Expressões Regulares*
HMM	*Hidden Markov Model*
IE	*Information Extraction*
IEEE	*Institute of Electrical and Electronics Engineers*
IV	*In vocabulary*
KB	*Knowledge Base*
LISA	*Laboratório de Integração de Sistemas e Aplicações*
MEM	*Maximum Entropy Models*
MEMM	*Maximum Entropy Markov Model*
ML	*Machine Learning*
MUC	*Message Understanding Conference*
MWE	*Multi-Word Expression*
NE	*Named Entity*
NER	*Named Entity Recognition*
NERD	*Named Entity Recognition and Disambiguation*
NED	*Named Entity Disambiguation*
NLTK	*Natural Language Toolkit*
OVV	*Out-of-Vocabulary*
PLN	*Processamento de Linguagem Natural*
POS	*Part-of-Speech Tagging*
SL	*Supervised Learning*
SSL	*Semi Supervised Learning*

SVM *Support Vector Machines*

SUMÁRIO

1 INTRODUÇÃO

O massivo uso das mídias sociais tem gerado grandes volumes de dados a cada dia. Postagens compartilhadas publicamente em tais mídias (e.g., *Twitter*, *Facebook*, *Flickr*) podem ser obtidas via APIs (*Application Programming Interface*) e apresentam conteúdos sobre os mais diversos assuntos, incluindo muitas vezes opiniões e sentimentos dos seus autores. O entendimento preciso do que é publicado em mídias sociais (BONTCHEVA; ROUT, 2014) pode contribuir na melhoria do desempenho de tarefas como mineração de opiniões (DAS; ACHARJYA; PATRA, 2014), recomendação de produtos e/ou serviços (PAZZANI; BILLSUS, 2007) e até mesmo detecção de catástrofes naturais (EARLE; BOWDEN; GUY, 2012).

Entretanto, postagens em mídias sociais têm conteúdos na forma de textos não estruturados em linguagem natural. A Extração de Informação (*Information Extraction* - IE) de textos (AGGARWAL; ZHAI, 2012; ZHAI; MASSUNG, 2016) visa detectar fragmentos relevantes e a eles associar semântica bem definida por meio de diversas tarefas de processamento. A tarefa de Reconhecimento de Entidades Nomeadas (*Named Entity Recognition – NER*) visa encontrar e classificar menções a entidades nomeadas (e.g., lugares, pessoas, instituições, valores, números de telefone, especificações de tempo), enquanto a tarefa subsequente de Desambiguação de Entidades Nomeadas (*Named Entity Disambiguation - NED*) (NADEAU; SEKINE, 2007), busca ligar precisamente tais menções a definições correspondentes em uma base de conhecimento (e.g., Avaí referindo-se à batalha da Guerra do Paraguai, município do estado de São Paulo, time de futebol de Florianópolis, Joinville, Laguna ou Guaramirim). Neste trabalho denominamos NERD

(*Named Entity Recognition and Disambiguation*) a composição das tarefas de NER e NED. Tais tarefas são realizadas com o uso de técnicas determinísticas e de aprendizado de máquina em diversas propostas presentes na literatura (KLEIN, 2015).

As abordagens do estado da arte em NER/NED podem produzir resultados de qualidade aceitável quando aplicadas a textos bem formados. Porém, isso não acontece com postagens em mídias sociais, que costumam conter muitos ruídos, tais como gírias, abreviações e erros de escrita que dificultam a extração de informações (RITTER, et al., 2011; IBRAHIM; YOSEF; WEIKUM, 2014; DERCZYNSKI, et al., 2015). Desta forma, várias abordagens têm sido propostas na literatura para efetuar NER/NED em conteúdos de postagens em mídias sociais. Entre tais abordagens algumas (HAN; BALDWIN, 2011; RITTER et al., 2011; AMITAVA et al., 2013; HAN; COOK; BALDWIN, 2013; OLIVEIRA et al., 2013; AHMED, 2015) baseiam-se em filtros, com o intuito de diminuir os ruídos presentes nos dados antes de realizar as tarefas de reconhecimento e desambiguação de entidades nomeadas. Além dessas abordagens, a técnica de aprendizado de máquina *ensemble learning* pode ser utilizada para aumentar precisão e cobertura de métodos de NER/NED (SPECK; NGONGA NGOMO, 2014).

Este trabalho propõe um método de pré-processamento de postagens do Twitter[1] para posterior reconhecimento e desambiguação de entidades nomeadas (NERD). O método proposto baseia-se em normalização léxica dos *tweets* e visa reduzir ruídos presentes no texto de modo a aumentar o número de entidades nomeadas reconhecidas pela ferramenta de NERD. O presente trabalho avalia o

1 https://twitter.com

desempenho do método proposto a partir de experimentos com a ferramenta de código aberto FOX[2], a qual faz uso da técnica de *ensemble learning* para combinar resultados de algumas das melhores ferramentas NERD abertas atuais para obter resultados melhores que os gerados por cada ferramenta da composição individualmente. Assim, os resultados da avaliação com o FOX, que também permite usar cada uma das ferramentas que ele engloba isoladamente, permite uma apreciação dos benefícios do método de filtragem proposto no estado-da-arte em NER.

1.1 OBJETIVOS

O objetivo geral deste trabalho é propor um método para filtrar *tweets* normalizando-os lexicalmente e avaliar o desempenho do método proposto com base no aumento do número de entidades nomeadas reconhecidas nos dados filtrados. Esta medida permite fazer uma apreciação preliminar dos resultados obtidos enquanto não há uma regra outro disponível para NERD em tweets.

1.1.1 Objetivos Específicos

Os objetivos específicos deste trabalho são:

1. Pesquisar, analisar e documentar o estado-da-arte em filtragem de dados de mídias sociais, principalmente *tweets*, com o intuito de melhorar os resultados da extração e desambiguação de entidades nomeadas (NERD).

2 http://aksw.org/Projects/FOX.html

2. Obter uma implementação de mecanismos para pré-processamento de *tweets* que permitam tratar ruídos (links, acrônimos, abreviações, erros de ortografia, etc.) e deste modo conseguir melhores resultados de NERD com os *tweets* filtrados do que com *tweets* brutos, usando ferramenta do estado-da-arte em NERD.
3. Descrever os mecanismos estudados, o método desenvolvido e os resultados obtidos em documentos a serem utilizados por membros do LISA e outros interessados em tal tema.

1.2 METODOLOGIA

O desenvolvimento deste trabalho foi dividido em 6 tarefas, as quais são descritas a seguir:

1. Produzir uma revisão bibliográfica atualizada sobre filtragem de dados de mídias sociais, com ênfase em *tweets*, visando melhorar os resultados da extração e desambiguação de entidades nomeadas (*NERD*).
2. Criar e implementar um método para tratamento de ruídos nos *tweets* mediante a conjugação de ideias propostas na literatura.
3. Aplicar o método de filtragem proposto a *tweets* selecionados da base de dados disponível no laboratório LISA.
4. Realizar experimentos de NERD usando a ferramenta FOX nos *tweets* brutos e pré-processados com o método proposto, visando analisar a contribuição de tal método na melhoria dos resultados de NERD.

5. Analisar os resultados obtidos, medindo o aumento da quantidade de entidades nomeadas encontradas nos tweets filtrados pelo método proposto em comparação com aquelas encontradas nos *tweets* não filtrados.
6. Documentar os resultados obtidos em uma monografia, uma apresentação, artigos e documentação de software.

Buscando entendimento do estado-da-arte, o estudo da literatura parte de trabalhos encontrados em bibliotecas digitais (e.g., IEEE Xplore, SpringerLink e ACM Digital Library). Foram incluídos na busca apenas trabalhos publicados após 2007, exceto aqueles que continham definições, técnicas e ferramentas relacionadas aos problemas estudados, e portanto, servem de base para a fundamentação deste trabalho.

A partir dos trabalhos estudados verificou-se a necessidade de filtragem das postagens de mídias sociais. Dessa forma, foi proposto um método de filtragem de *tweets* baseado em normalização léxica. Para fundamentar a proposta, foram escolhidos os trabalhos mais recentes e que tinham foco na filtragem de *tweets*. O método proposto foi implementado utilizando linguagem Python e bibliotecas disponíveis na *web*.

Para os experimentos foi utilizada uma amostra de dez mil *tweets* da base disponível no laboratório LISA que foram filtrados a partir do método proposto nesse trabalho. A ferramenta FOX que realiza as tarefas de NER/NED foi utilizada para experimento com os conjuntos de dados sem e com pré-processamento.

A avaliação dos experimentos focou em verificar o desempenho computacional (e.g., tempo de execução) da etapa de filtragem e das tarefas de NER/NED, assim como o aumento do número de anotações em *tweets* filtrados e não filtrados. A princípio, experimentos para medir ganhos de precisão e cobertura ficam para trabalhos futuros, devido à dificuldade de obter conjuntos de dados com regra ouro, dado o estágio atual das pesquisas sobre anotação semântica de postagens em mídias sociais.

O software desenvolvido está depositado no GitHub[3] para que possa ser estendido, aprimorado e utilizado por interessados dentro e fora do laboratório LISA. A sua documentação inclui este documento além documentos associados ao código fonte e ao executável, todos integrados em uma estrutura de diretórios contendo o *baseline* completo para compilação e execução.

1.4 ORGANIZAÇÃO DO DOCUMENTO

O presente trabalho divide-se nas seguintes seções:

- O Capítulo 2 apresenta a fundamentação teórica necessária para o entendimento do restante do trabalho, incluindo bases sobre processamento de linguagem natural e das tarefas de NER e NED.
- O Capítulo 3 apresenta uma revisão dos trabalhos encontrados na literatura com foco em pré-processamento de postagens de mídias sociais, principalmente tweets, viando melhorar os resultados de NER/NED.

[3] https://github.com/suelenfenali/TCC-Pre-processamento-tweets

- O capítulo 4 propõe e implementa um método de filtragem de *tweets* baseado em normalização léxica com o intuito de melhorar os resultados de NERD.
- O Capítulo 5 relata os experimentos realizados e discute os resultados obtidos.
- O Capítulo 6 conclui o trabalho e enumera trabalhos futuros.

2 FUNDAMENTAÇÃO

2.1 DEFINIÇÕES BÁSICAS

O termo **entidade nomeada** (*named entity* - NE) foi usado pela primeira vez por Grishman e Sundheim (1996) na 6ª Message Understanding Conference (MUC-6). Uma entidade nomeada é uma "unidade de informação" que pode referir-se a pessoas, lugares, organizações, tempo (horas e datas) e quantidades (valores monetários, percentuais) (NADEAU; SEKINE, 2007). As entidades nomeadas foram separadas na MUC-7 (CHINCHOR, 1998) em sete classes que compõem três grupos. Para cada grupo as entidades são anotadas com o uso de *tags* que descrevem a classe para a qual foram classificadas. O grupo ENAMEX inclui as classes de organização, pessoa e localização; o segundo grupo chamado de TIMEX representa as expressões de tempo para data e hora; por fim, o grupo NUMEX abrange as classes de valores monetários e percentagens.

Um termo escrito em linguagem natural de forma a designar uma entidade nomeada é chamado **nome de superfície** (e.g., "BSB" e "Brasília" são nomes de superfície para a capital do Brasil). Um **menção** é a ocorrência de um nome de superfície em um texto (KLEIN, 2015; RAO; MCNAMEE; DREDZE, 2013). O **contexto textual** de uma menção corresponde a sentença ou parágrafo onde ela ocorre, ou a uma janela de k palavras à esquerda e à direita da mesma.

Entidades nomeadas podem apresentar fenômenos de **polissemia** e **sinonímia** (KALLOUBI; NFAOUI; BEQQALI, 2014). A polissemia ocorre quando um nome de superfície se refere a diferentes entidades nomeadas (e.g., "São Paulo" pode referir-se a um estado brasileiro, à cidade paulista ou até mesmo ao time de

futebol). A sinonímia, por sua vez, ocorre quando há mais de um nome de superfície para descrever a mesma entidade nomeada (e.g., "Florianópolis", "Ilha da magia", "Floripa" são nomes de superfície usados para descrever a cidade que é a capital do estado de Santa Catarina) (KLEIN, 2015; KALLOUBI; NFAOUI; BEQQALI, 2014).

Para dar sentindo a menções de entidades pode-se ligá-las a definições precisas que podem ser encontradas em **bases de conhecimento** (Knowledge Bases - KB), tais como: Wikipedia[4], DBPedia[5], LinkedGeoData[6] (RAO; MCNAMEE; DREDZE, 2013). Tais ligações são chamadas anotações semânticas. Elas auxiliam no entendimento preciso das informações e no processo de expansão semântica, i.e., acesso ou derivação de novas informações a partir das definições semânticas usadas nas anotações. Por exemplo, a partir da ligação de uma menção à sua definição em base de conhecimento ou coleção de dados ligados, pode-se ter acesso a várias informações (e.g., a partir da ligação com a cidade correta mencionada em um texto pode-se ter acesso a muitos outros dados tais como população, clima, contexto geográfico e histórico).

2.2 DEFINIÇÃO DO PROBLEMA

O estudo do Processamento de Linguagem Natural (PLN) iniciou-se por volta de 1950 com o uso de inteligência artificial e linguística (NADKARNI; OHNO-MACHADO; CHAPMAN, 2011). A realização de PLN inclui tarefas de **tokenização**, **reconhecimento** e **desambiguação de entidades nomeadas**. De acordo com

4 https://wikipedia.org
5 http://dbpedia.org
6 http://linkedgeodata.org/About

Kaplan (2005), no pré processamento de um texto é necessária uma etapa de **tokenização**, na qual uma sequência de caracteres deve ser quebrada em diferentes unidades com significado único, conhecidas como **tokens**. Entretanto, a ocorrência de algumas particularidades nessa etapa pode resultar na extração do token de forma incorreta. Tais particularidades incluem: a identificação de tokens que contém mais de uma palavra (*Multi-Word Expressions - MWE*) (e.g., a priori, Santa Catarina) e a extração de tokens com caracteres não alfanuméricos (e.g., d'água).

Após a tokenização ocorrem as etapas de NER/NED. Para definir formalmente os problemas de NER/NED, usamos neste trabalho as notações definidas em Klein (2015), onde:

- *D* é um conjunto de documentos, tal que um documento $d \in D$ é uma produção textual escrita em linguagem natural, da qual pretende-se extrair entidades nomeadas. As produções textuais podem ser semi-estruturadas ou não-estruturadas (e.g., postagem no Twitter, notícia de jornal, texto em um *blog*).
- *M* é um conjunto de menções a entidades.
- *E* é um conjunto de definições de entidades nomeadas armazenadas em uma base de conhecimento (KB).
- *T* é o conjunto de possíveis tipos de entidades ou menções.

2.2.1 Reconhecimento de Entidades Nomeadas

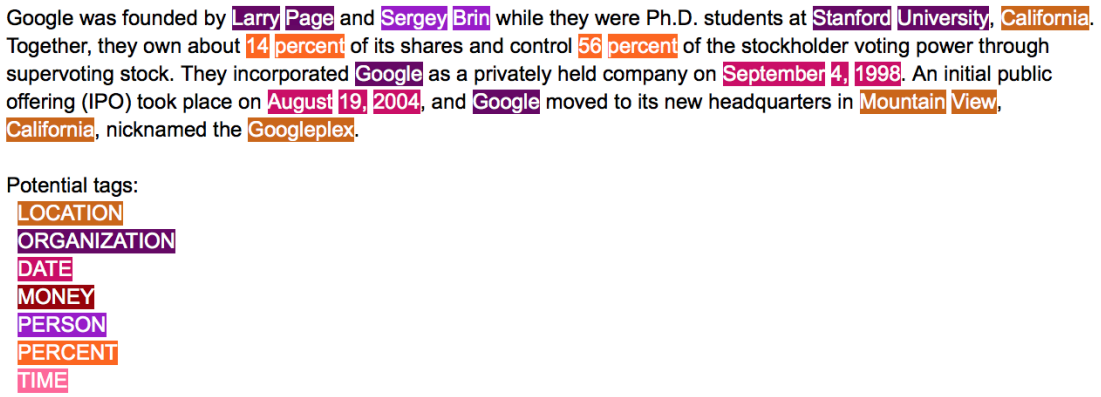
O Reconhecimento de Entidades Nomeadas (*Named Entity Recognition* - NER) é uma tarefa de extração de informação que consiste na identificação de

entidades nomeadas presentes em textos não estruturados (DERCZYNSKI, et al., 2015). NER serve de base para diversas áreas, tais como: realização de anotações semânticas, população de ontologias e mineração de opinião (MARRERO et al., 2013). Podemos definir NER a partir de duas subtarefas (SPECK; NGONGA NGOMO, 2014; KLEIN, 2015):

1. identificação: refere-se a identificação de menções $m \in M$ presentes em documentos $d \in D$;
2. classificação: consiste na geração de tuplas $\{ (m, t_m) \mid m \in M \wedge t_m \in T \wedge t_m$ é o tipo de m $\}$.

Na Figura 1 pode-se observar um exemplo do resultado da aplicação de NER. O reconhecimento das entidades foi realizado utilizando a ferramenta Stanford NER[7] configurada para encontrar as classes de entidades nomeadas de acordo com as definições da MUC-7 (CHINCHOR 1998).

Figura 1 – Exemplo de texto processado com a ferramenta Stanford Named Entity Tagger.



Fonte:http://nlp.stanford.edu:8080/ner/process.

7 http://nlp.stanford.edu/software/CRF-NER.shtml

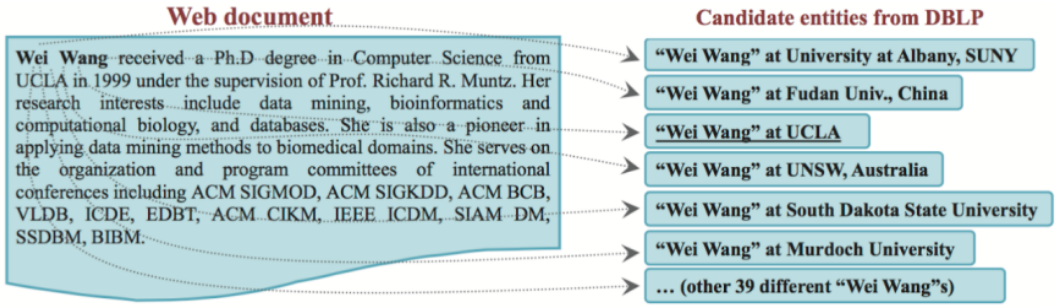
2.2.2 Desambiguação de Entidades Nomeadas

A Desambiguação de Entidades Nomeadas (*Named Entity Disambiguation - NED*) busca encontrar a definição mais apropriada para uma menção a entidade nomeada em uma base de conhecimento (DERCZYNSKI et al., 2014). Ao buscar por entidades em KBs, a tarefa de NED ajuda a resolver casos de polissemia e sinonímia. Todavia, há situações onde há uma ausência de definição para a menção na base de conhecimento (SHEN; WANG; HAN, 2014). Pode-se definir NED formalmente por (SHEN; WANG; HAN, 2014; KLEIN, 2015):

1) geração de entidades candidatas: consiste em identificar as definições de entidades $C_m \subseteq E$ candidatas a corresponder a cada menção $m \in M$;
2) *ranking* de candidatas: seleciona uma definição baseada na atribuição de um escore para cada $c_m \in C_m$, tal que o escore indica a probabilidade de c_m definir *m* corretamente.
3) detecção da ausência de definição: em casos onde não é encontrada uma definição para a menção, ou seja, $C_m = \varphi$ ou a candidata com maior escore não é considerada uma definição aceitável, então seleciona-se ε (épsilon – representando ausência de definição).

A Figura 2 mostra a seleção de entidades candidatas para a menção "Wei Wang". Um dos desafios de NED é a identificação da entidade correta em um determinado contexto, sendo necessário o uso de técnicas que auxiliem no tratamento desse problema. Nas seções seguintes deste trabalho as principais técnicas utilizadas serão discutidas.

Figura 2 - Exemplo de geração de entidades candidatas na tarefa de NED. A entidade correta está sublinhada.



Fonte: SHEN; WANG; HAN, 2014.

2.3 TÉCNICAS PARA NER E NED

Segundo Konkol (2012), as técnicas para efetuar NER e/ou NED podem ser categorizadas segundo diferentes aspectos. Toda técnica passa pela fase de sua criação e outra fase posterior de seu uso. A criação de uma técnica para realizar NER e/ou NED pode ser feita manualmente (*hand-crafted*), i.e. com parâmetros e regras criados por um ser humano, ou de maneira automática, i.e. por um programa de computador que a partir da observação de dados e uso de aprendizado de máquina, identifica automaticamente parâmetros, padrões e regras para solucionar o problema. Atualmente, técnicas baseadas em aprendizado de máquina e técnicas mistas têm suplantado as técnicas tradicionais baseadas em padrões linguísticos codificados manualmente em sistemas de NER/NED, principalmente no que se refere à qualidade dos resultados gerados. Técnicas de aprendizado de máquina podem ser classificadas em aprendizado supervisionado, semi supervisionado e não

supervisionado. Essa categorização é feita de acordo com o tipo de dado necessário para a máquina criar os parâmetros de entrada.

No que se refere à fase de uso, Konkol (2012) divide as técnicas de NER/NED em determinísticas e estocásticas. A principal diferença entre elas está na geração dos resultados. Técnicas determinísticas escolhem apenas uma classe para cada palavra, enquanto técnicas estocásticas atribuem um conjunto de classes e valores de probabilidade de cada classe para uma palavra. Para este trabalho será usada a classificação definida em Nadeau (2007) que trata de **técnicas determinísticas** e técnicas de **aprendizado de máquina**. Nesta divisão, técnicas baseadas em dicionário de nomes e sistemas baseados em regras são consideradas técnicas determinísticas. A seguir tem-se um aprofundamento sobre cada técnica.

2.3.1 Técnicas Determinísticas

A técnica mais simples para NER é a baseada em **dicionário de nomes**. Os dicionários de nomes utilizam uma lista de entidades nomeadas e tentam identificar cada palavra ou grupo de palavras no texto. Tais dicionários, entretanto, não apresentam uma alta performance devido a sua simplicidade (RATINOV; ROTH, 2009; KONKOL, 2012). Alguns dos problemas dessa técnica ocorrem quando existem entidades podem ser expressas por mais de um nome de superfície, sendo assim necessário enumerar todas as possíveis formas. Outra questão é que por possuírem uma quantidade restrita de nomes de superfície, os dicionários não funcionam bem para o reconhecimento de nomes próprios (SPECK, NGONGA NGOMO, 2014). Técnicas baseadas em regras surgiram com a intenção de melhorar

o desempenho dos dicionários de nomes, que às vezes podem ser inusitados. Dessa forma, dicionários são combinados com algoritmos baseados em regras para estender a lista de entidades reconhecidas (SPECK, NGONGA NGOMO, 2014).

Expressões Regulares (ER) e **Autômatos Finitos** (AF) são técnicas muito utilizadas no processamento de texto. Como todo AF pode ser representado de maneira determinística, o reconhecimento de entidades a partir dos mesmos pode ser executado de maneira rápida. ERs costumam ser usadas para a construção de sistemas baseados em regras (KONKOL, 2012).

A s **Gramáticas Livres de Contexto** representam uma classe mais abrangente de gramáticas de Chomsky que as expressões regulares e assim tornam possível a criação de regras mais complexas, funcionando melhor para os sistemas baseados em regras. Porém, é importante lembrar que quanto maior a complexidade de uma regra, maior a probabilidade de que a mesma seja usada em apenas um domínio do reconhecimento de entidades (e.g., usada para realizar apenas tokenização) (KONKOL, 2012).

Técnicas determinísticas são geralmente utilizadas quando não há um conjunto de dados de treino. Antigamente essas eram as principais técnicas a serem usadas e estudadas. Um exemplo disso é que na MUC-7 em 1996, cinco dos oito sistemas avaliados eram baseados em regras. Contudo, o foco da comunidade acadêmica vem mudando desde então, e ainda em 2003 na Conference on Computational Natural Language Learning (CONLL) dezesseis sistemas baseados em aprendizado de máquina foram apresentados (NADEAU, 2007).

2.3.2 Técnicas de Aprendizado de Máquina

Aprendizado de Máquina (AM ou em inglês ML - *Machine Learning*) consiste em aplicar algoritmos a grandes coleções de documentos anotados de modo a permitir que o computador aprenda a criar regras e explorar propriedades dos conjuntos de dados que auxiliem a solucionar um problema (NADKARNI; OHNO-MACHADO; CHAPMAN, 2011). Um dos problemas de AM é a dependência de um **corpus** (e.g. conjunto de instâncias) anotado, usado para realizar o treinamento. Um **corpus anotado** contém um atributo que indica a classificação correta do dado, enquanto um **corpus não anotado** não apresenta tal característica. A anotação em um corpus é custosa, pois geralmente necessita que especialistas anotem os documentos, isso consequentemente demanda recursos financeiros e de tempo (KLEIN, 2015). Devido à ausência de corpus anotados para alguns domínios, divide-se o aprendizado de máquina em diferentes tipos: aprendizado supervisionado, aprendizado semi supervisionado e aprendizado não supervisionado (RUSSELL; NORVIG, 2009).

O **aprendizado supervisionado** (*Supervised Learning - SL*) é a principal técnica utilizada para a resolução do problema de NER. Nesse caso, há um corpus anotado que é utilizado para treinar os parâmetros que irão classificar palavras do corpus de teste de acordo com as entidades anotadas no corpus de treinamento (NADEAU; SEKINE, 2007). Algumas técnicas de aprendizado supervisionado incluem: *Hidden Markov Models* (HMM), (BIKEL; SCHWARTZ; WEISCHEDEL, 1999), *Maximum Entropy Models* (ME), *Support Vector Machines* (SVM) (ASAHARA; MATSUMOTO, 2003) e *Conditional Random Fields* (CRF) (MCCALLUM; LI, 2003).

O **Aprendizado semi supervisionado** (*Semi Supervised Learning - SSL*) faz uso tanto de corpus anotados quanto não anotados. A técnica mais conhecida nesse tipo de aprendizado é chamada *bootstrapping*. O *bootstrapping* usa um pequeno corpus anotado para treinar o classificador e então usa os dados do treinamento para melhorar o próprio desempenho. Para facilitar o entendimento, Nadeau e Sekine (2007) nos dão um exemplo disso considerando um sistema que faz busca de nomes de doenças em textos. Nesse caso, seria necessário que uma pessoa anotasse um corpus com alguns exemplos, e a partir de então o *bootstrapping* treinaria seus classificadores buscando contextos onde as entidades anotadas são usadas. Assim, a técnica poderia buscar novas entidades que aparecem em contextos semelhantes e reaplicar os classificadores aos novos exemplos encontrados. Wu et al. (2009) propõem uma variação do *bootstrapping* chamada *bootstrapping* adaptável ao domínio, na qual um classificador treinado com corpus anotado para um domínio S (e.g., obras literárias) é utilizado para classificar entidades de um domínio T (e.g., postagens em tweets) que possui corpus não anotado.

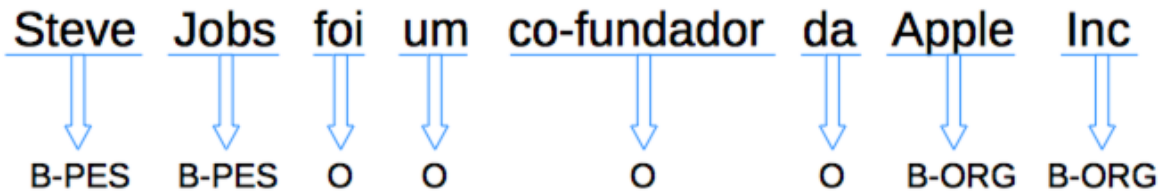
O **aprendizado não supervisionado** usa apenas corpus não anotados para o treinamento de parâmetros, sendo assim é necessário o uso de formas auxiliares (e.g., verificação manual) para avaliar o resultado do classificador. A técnica mais utilizada de aprendizado não supervisionado é *clustering* (e.g., pode-se gerar *clusters* baseados em similaridade de contexto e tentar inferir as entidades nomeadas a partir desses grupos). Há também outras técnicas, mas no geral elas dependem de recursos e padrões léxicos e de estatísticas retiradas de corpus não anotados (NADEAU; SEKINE, 2007).

Algumas técnicas de aprendizado de máquina fazem uso de classificação de sequências (*sequence labeling*), onde: para uma sequência de tokens s = $w_1w_2...w_n$, gera-se uma lista de marcadores $T = t_1t_2...t_n$ que indicam a presença de menções em *s* (KLEIN, 2015). A marcação pode ser realizada de diferentes formas (BIKEL; SCHWARTZ; WEISCHEDEL, 1999; NGUYEN; GUO, 2007; AMARAL; VIEIRA, 2014):

- **marcação de presença**: identifica a ocorrência de entidades aplicando múltiplos rótulos a uma mesma entidade, sendo necessário o uso de técnica complementar para identificar seu tipo. Essa marcação possui duas notações:
    a. **BIO**: um token inicia (*Begin*), está dentro (*In*) ou fora (*Out*) de uma entidade nomeada;
    b. **BILOU**: amplia a notação BIO acrescentando o marcador de final (*Last*) da entidade e um marcador de entidades compostas por apenas um token (*Unit*);
- **marcação de tipo**: cada token é associado a um tipo (classe) (e.g., pessoa, localização, data). Para tokens que não têm um tipo de entidade associado é atribuído o tipo adicional representado por épsilon - ε.

Há ferramentas que usam notação híbrida, utilizando a marcação de presença e de tipo simultaneamente. A Figura 3 ilustra um exemplo de classificação de entidades nomeadas utilizando notação híbrida na sentença "Steve Jobs foi um co-fundador da Apple Inc". As entidades encontradas são: Steve Jobs (pessoa) e Apple Inc (organização).

Figura 3 - Exemplo de classificação de entidades nomeadas utilizando a notação BIO.



Fonte: Adaptado de (AGGARWAL; ZHAI, 2012).

Alguns elementos são necessários para a técnica de classificação de sequências (SUTTON; MCCALLUM, 2011; KLEIN, 2015):

- classificadores: os mais apropriados para a técnica são *Hidden Markov Models* (HMM), *Maximum Entropy Markov Models* (MEMM) e *Conditional Random Fields* (CRF).
- um conjunto de treinamento: um corpus anotado de acordo com o aprendizado de máquina utilizado.
- características: são funções que indicam características dos dados de entrada. Exemplos de características são: capitalização do texto e sufixos (e.g., termina em "mente"). A função f_t: (x_t , y_t) → (0, 1), onde x_t é um token e y_t é o seu estado é usada para identificar características (e.g., definir se x_t é capitalizado, tal que x_t = "Google" e y_t = ORG) (SHA; PEREIRA, 2003; KLEIN, 2015).
- pesos: a cada característica do dado é atribuído um peso;

- algoritmo de treinamento: usado para adequar o peso dado a cada característica; o treinamento auxilia a aumentar a precisão e cobertura do classificador.

2.2.3 Ensemble Learning

Buscando melhorar os resultados de um classificador tem-se feito uso da combinação de diferentes técnicas. Uma técnica de AM para confecção automática de combinações é ***ensemble learning***. Um algoritmo de ensemble learning S tem como objetivo gerar um classificador F com uma alta performance preditiva através da combinação de k classificadores básicos $C_1,...,C_m$ (SPECK; NGONGA NGOMO, 2014). Essa técnica é a utilizada pela ferramenta FOX, a ser avaliada neste trabalho. Existem diferentes maneiras de realizar ensemble learning, sendo algumas delas:

- **empilhamento**: a saída de um classificador é utilizada como entrada para o próximo, assim inúmeros classificadores podem ser combinados.
- **votação**: essa abordagem divide-se em **votação por maioria** e **votação por peso**. Na votação por maioria, é atribuída a um token a classe que foi predita pela maioria dos classificadores. A votação por peso é uma extensão da primeira, onde para cada classificador é atribuído um peso e o algoritmo retorna a classe que obteve o maior peso total.

Speck e Ngomo (2014) apontam que, devido a muitos anos de pesquisa, a precisão e a cobertura das técnicas utilizadas para NER são hoje aceitáveis para diversas coleções de dados. Entretanto, a técnica de *ensemble learning* ainda apresenta a desvantagem de depender de alguma abordagem de votação realizada

sobre a saída dos classificadores. Essa dependência do método de votação pode acarretar dois problemas: (a) se todos os classificadores utilizados obtiverem resultados errados, o ensemble learning provavelmente também apresentará erros; (b) votação não considera diferentes níveis de acurácia para diferentes classes de entidades. Ainda assim, a combinação de classificadores gera melhores resultados que o uso dos mesmos individualmente, principalmente quando é realizada uma combinação de classificadores que utilizem diferentes técnicas ou conjuntos de características (KONKOL, 2012).

3 *NER* E *NED* DE *TWEETS*

Recentemente as pesquisas sobre NER/NED aplicados a dados de mídias sociais têm ganhado atenção da comunidade acadêmica. Uma dessas mídias é o *Twitter*, onde usuários podem compartilhar textos (*tweets*) de até 140 caracteres. As técnicas atuais de extração de conteúdo relevante funcionam razoavelmente bem para textos escritos de maneira apropriada (e.g., textos jornalísticos, obras literárias e etc.). Entretanto, o reconhecimento e a desambiguação de entidades nomeadas em *tweets* é um grande desafio. Derczynski et al. (2015) apontam que técnicas de NER quando aplicadas a textos longos apresentam entre 85 e 90% de precisão, enquanto quando aplicados a *tweets* obtêm entre 30 e 50% de precisão, fora os problemas de cobertura. Alguns trabalhos (RITTER et. al., 2011; AMITAVA et al., 2013; HAN; COOK; BALDWIN, 2013; OLIVEIRA et al., 2013; AHMED, 2015; DERCZYNSKI et al., 2015) mostram os desafios encontrados ao processar postagens no *Twitter*:

a) tamanho reduzido do texto, tornando difícil a realização de anotações semânticas, particularmente a desambiguação, devido à carência de informação de contexto;

b) presença de ruídos provenientes do uso da linguagem natural sobre a Web (e.g., erros gramaticais, gírias, abreviações, *emoticons* e *hashtags*);

c) dependência de contexto (situação do usuário e seu ambiente, contexto da mensagem em meio a outras), o que impede em alguns casos a ligação às entidades nomeadas corretas. Em grandes textos as técnicas conseguem desambiguar entidades de acordo com o contexto onde elas aparecem. Porém com

tweets, usualmente pobres em informação de contexto, isso torna-se muito mais difícil.

Alguns trabalhos tem buscado solucionar tais problemas a partir do desenvolvimento de algoritmos de NER e NED específicos para microblogs, sendo que boa parte deles realiza uma etapa de filtragem dos *tweets* visando a melhorar os resultados das tarefas de NER e NED. Seguem descrições de trabalhos que tentam reconhecer e desambiguar entidades nomeadas em *tweets* e os filtros que cada um utiliza.

3.1 RITTER E A PROPOSTA DE RECONSTRUÇÃO DA *PIPELINE*

Ritter et al. (2011) reconstruíram uma *pipeline* de PLN buscando alcançar melhor desempenho em relação à mera utilização das atuais ferramentas de NER/NED que não tem foco em *tweets*. Para tal, antes de realizar a atividade de NER eles realizam uma filtragem nos tweets em três passos. O primeiro é *Part-of-Speech Tagging* (POS *Tagging*), no qual eles anotaram morfossintaticamente cerca de 16 mil *tokens* contidos em 800 *tweets* para serem usados como conjunto de dados de treinamento. Eles usaram *tags* adicionais, além das usualmente empregadas para anotar classes de palavras (substantivos, verbos, adjetivos, etc.) para indicar *retweets*, *@usernames*, *#hashtags* e urls. Para tratar casos de palavras consideradas "fora de vocabulário" (*Out-Of-Vocabulary - OOV*, em inglês) que apresentam variação léxica, os autores utilizaram *clusters* hierárquicos que agrupam palavras similares. Em sequência fizeram uso da abordagem proposta por Han e Baldwin (2011) que realiza normalização léxica com base em dicionários, similaridade de palavras e de contextos. Como resultado deste passo, eles

conseguiram uma redução de 41% na taxa de erro em comparação com a ferramenta *Stanford Tagger* e uma acurácia de 0.883. O segundo passo da filtragem foi *shallow parsing*. Essa sub tarefa do processamento de linguagem natural inclui identificar a categoria gramatical da palavra no texto (e.g. pronomes, verbos, substantivos, preposições) junto a uma análise sintática rasa (RITTER et al., 2011). Utilizando o mesmo conjunto de *tweets* anotados na tarefa de POS *Tagging* e aplicando um algoritmo de CRF, a ferramenta proposta obteve 22% de redução na taxa de erro quando comparada à ferramenta *OpenNLP* da *Apache*. Por fim, foi realizada uma tarefa de capitalização das palavras utilizando um classificador baseado em máquinas de vetores de suporte (SVM). Tal classificador buscava identificar se a capitalização em um *tweet* era informativa, ou apenas utilizada como uma figura de linguagem para, por exemplo, dar ênfase. O comparativo em relação às ferramentas que usam filtro de capitalização mostrou melhoria tanto na precisão quando cobertura.

3.2 DICIONÁRIO NORMALIZADO POR HAN, COOK E BALDWIN

Han, Cook e Baldwin (2013) propõem uma tarefa de normalização léxica das mensagens postadas. Este trabalho é uma continuidade dos trabalhos de Han e Baldwin (2011) e Han, Cook e Baldwin (2012). O processo de normalização utiliza um classificador que encontra palavras que possuem erro de escrita, abreviação ou substituição fonética (e.g. *2morrow*) e a partir de similaridades morfológica e fonêmica cria um dicionário de nomes com pares contendo a palavra encontrada no *tweet* e a sua forma padrão (e.g. *2morrow* → *tomorrow*). Tanto a similaridade de

contexto quanto a similaridade de *string* - o que engloba as similaridades morfológica e fonética - são analisadas para encontrar o melhor candidato para a palavra. Apesar do método ter a limitação de não desambiguar usos diferentes do mesmo *token*, os experimentos mostraram que a ferramenta proposta alcança o estado-da-arte no indicador de acurácia, apresenta maior cobertura que os dicionários existentes e tem um valor de precisão razoável. Além disso, a solução é leve e apropriada pra o processamento de grandes volumes de postagens de mídias sociais (HAN; COOK; BALDWIN, 2013).

3.3 FERRAMENTA DE FILTRAGEM *FS-NER*

Oliveira et al., (2013) propõem uma ferramenta chamada *FS-NER* (*Filter Stream Named Entity Recognition*) que utiliza cinco filtros, os quais além de rápidos independem de regras gramaticais, podendo ser combinados de maneira sequencial, aumentando a precisão e a cobertura. Os filtros usados incluem:

- Termos – estima a probabilidade de um termo ser uma entidade;
- Nomes – considera apenas termos com a primeira letra em maiúsculo para assim verificar se o termo é uma entidade;
- Afixos – (e.g. **re**fazer, legal**mente**) usa fragmentos do termo observado podendo reconhecer entidades que possuam afixos similares a afixos de entidades já reconhecidas;
- Contextos – busca identificar entidades desconhecidas analisando o contexto onde elas aparecem, considerando uma janela de *n* termos próximos ao termo que está sendo analisado;

- Dicionários – utiliza dicionário de nomes para determinar se um termo é uma entidade.

Para a avaliação desta ferramenta de filtragem os autores utilizaram três conjuntos de dados. Um com postagens em inglês, um com postagens em português e um contendo postagens de diversos idiomas. A *FS-NER* mostrou uma melhoria de 3% em precisão, cobertura e *F-score* quando comparada a outras ferramentas que utilizam CRF. Entretanto, em termos de desempenho computacional (tempo de execução) mostrou-se significantemente melhor que as outras ferramentas usadas (OLIVEIRA et al., 2013).

3.4 NORMALIZAÇÃO DE *TWEETS* POR AHMED

Ahmed (2015) propôs um método de normalização de *tweets* baseado em Ritter et al. (2011), que por sua vez usa alguns conceitos propostos por Han e Cook em 2011. Em seu trabalho, Ahmed propõe três passos para a normalização. O primeiro passo deles é a tokenização. O passo seguinte é chamado classificação e utiliza um dicionário de nomes para verificar se um *token* é classificado como:

- *in vocabulary* (IV) - quando o mesmo existe no dicionário
- *non-candidate* – quando o *token* não é uma palavra (e.g. #, !, @)
- *out of vocabulary* (OOV) - quando o *token* não é encontrado no dicionário

No passo três, os *tokens* considerados *OOV* são normalizados em quatro etapas:

1. Cálculo da distância de Levensthein, onde as entradas são a palavra *OOV* e um dicionário de nomes e a saída é um vetor contendo as palavras com distância de Levensthein menor ou igual a 2 a alguma palavra do

dicionário. De acordo com Han, Cook e Baldwin (2013) o cálculo de similaridade para distância > 2 é computacionalmente custoso, assim utiliza-se o *threshold* ≤ 2 sendo que segundo os autores, tal número apresenta um bom valor de *recall*.

2. Análise fonética utilizando o método *Refined Soundex*. Nessa etapa é usado como entrada o vetor resultante da etapa anterior e é gerado um novo vetor com as palavras filtradas foneticamente;
3. Aplicação do algoritmo de Peter Norvig. O algoritmo de Peter Norvig recebe como entrada um *token* e realiza operações de inserção, alteração, exclusão, separação e transposição criando uma lista com as palavras que foram geradas com um número de edições menor ou igual a dois. As palavras geradas são buscadas num arquivo de texto e para cada palavra encontrada é calculada a probabilidade de ser a palavra correta. A palavra que apresentar a maior probabilidade é a retornada como resultado (Ahmed, 2015);
4. Compara o vetor da etapa 2 com a sugestão da etapa 3 e toma a decisão da seguinte forma:

- se o vetor da etapa 2 for vazio, o resultado da normalização será a palavra sugerida na etapa 3;
- se a etapa 2 retornou apenas uma palavra, sendo essa a mesma sugerida na etapa 3, então essa palavra é resultado da normalização;
- se o vetor da etapa 2 possuir uma lista de palavras, para cada palavra é realizada uma análise de contexto considerando *n-grams*, onde para sentenças de tamanho = 5 são contadas quantas vezes a palavra

analisada aparece e é selecionada a candidata que aparece o maior número de vezes.

A avaliação do método proposto por Ahmed (2015) mostrou que o uso combinado das técnicas (distância de Levensthein, Refined Soundex, Peter Norvig e n-grams) resulta no aumento da acurácia na normalização léxica.

A Tabela 1 sumariza comparação dos métodos de pré-processamento descritos neste trabalho. Além dos trabalhos já descritos que utilizam filtragem no pré processamento dos dados, outros autores têm realizado pesquisas na área de NER/NED para *tweets*. Liu et al. (2011) combinam um classificador de K-Nearest Neighbors (KNN) com um classificador CRF em um *framework* baseado em aprendizado de máquina semi supervisionado. Os experimentos realizados em tal trabalho mostraram resultados efetivos e indicaram que a junção do aprendizado de máquina semi supervisionado com o uso de dicionários contribui na redução de problemas causados por falta de dados de treinamento. Mais recentemente, Derczynski et al. (2015) descrevem um novo conjunto de dados para realizar desambiguação de entidades nomeadas e fazem uma análise sobre NER e NED examinando a robustez dos sistemas presentes no estado-da-arte. Uma das conclusões apresentadas é sobre a necessidade da criação de um corpus anotado para melhorar os algoritmos utilizados. Entretanto, devido ao alto custo para que os corpus sejam anotados por especialistas, eles sugerem que a falta de dados anotados para treino seja resolvida a partir do uso de *crowd-sourcing*, ou seja, que a própria comunidade de pesquisadores em PLN crie através de ferramentas online as anotações nos corpus a serem utilizados (BONTCHEVA; DERCZYNSKI; ROBERTS, 2014).

Tabela 1 - Tabela comparativa entre as ferramentas que utilizam filtragem no pré processamento dos *tweets* para NER/NED.

Característica	***Ritter et al. (2011)***	***Han, Cook e Baldwin (2013)***	***Oliveira et al. (2013)***	***Ahmed (2015)***
Proposta	Filtra os tweets de acordo com uma nova pipeline de PLN buscando melhorias em relação as ferramentas do estado-da-arte	Realiza normalização léxica criando um dicionário baseado em similaridade de contexto, de morfologia e fonética das palavras	Aplica cinco filtros sobre os tweets de modo a aumentar precisão, cobertura e F-measure na tarefa de NER	Realiza normalização léxica de palavras que apresentam erro de escrita ou abreviações e gírias
Software Disponível	Github aritter/twitter_nlp	Não	Github dmoliveira/FSNER	Não
Idiomas	Inglês	Inglês	Inglês, Português e Outros	Inglês – adaptável para outros idiomas
Benefícios	Aumento da acurária nas sub tarefas (POS *Tagging, shallow parsing*) de PLN	Aumento da acurácia ao realizar a tarefa de normalização léxica	Aumento de 3% em relação a outras ferramentas de CRF e velocidade no tempo de execução	Aumento da acurácia ao realizar a tarefa de normalização léxica

4 PROPOSTA DE FILTRAGEM DOS DADOS

Entre as ferramentas para filtragem de dados de mídias sociais revisadas neste trabalho, duas delas (Ritter *et al.*, 2011 e Oliveira *et al.,* 2013) possuem código disponível em repositórios no *Github* e ambas obtiveram resultados satisfatórios na filtragem de *tweets*. Desta forma, a primeira abordagem realizada foi a tentativa de instalação local de tais ferramentas. Infelizmente, nenhum dos repositórios está atualizado corretamente para que a instalação pudesse ser completada sem erros. A *FS-NER*, proposta por Oliveira *et al.* (2013) não é atualizada desde sua publicação e seu código apresenta algumas configurações com código fixo, indicando caminho e arquivos da máquina local onde ela foi desenvolvida. A ferramenta de Ritter *et al.* (2011), por sua vez, realiza não apenas a filtragem, mas também o reconhecimento de entidades nomeadas, possuindo assim muitas dependências que dificultam a instalação e uso. Finalmente, dos trabalhos estudados, Ahmed (2015), além de ser o mais atual, mostrou-se o mais simples (mas ainda assim retornando bons resultados) e focado na filtragem dos *tweets*. Sendo assim, o autor de tal trabalho foi contatado para verificar a possibilidade de disponibilizar a ferramenta proposta. Todavia, não houve resposta do autor em tempo hábil para o desenvolvimento deste trabalho.

Após tais tentativas, devido à falta de sucesso e ao tempo gasto para tentar fazer ferramentas funcionarem, uma outra abordagem foi adotada. Dadas tais dificuldades e requisitos específicos de filtragem e análise de dados de mídias sociais, optou-se por codificar um método e um software próprios para a filtragem dos *tweets*, a partir de técnicas e ferramentas mais básicas disponíveis (e.g.,

técnicas e ideias descritas nos artigos estudados, bibliotecas de métricas de similaridade textual e fonética). Tal método é proposto e explicado nas seções a seguir.

4.1 O MÉTODO PROPOSTO

O método proposto para pré-processamento de *tweets* antes do seu enriquecimento semântico tem foco na normalização léxica do texto dos *tweets* e utiliza a *baseline* proposta por Ahmed (2015), com algumas modificações que visam melhorar o processo de normalização. A Figura 4 apresenta um fluxograma com as etapas do método proposto.

Um estudo (CARTER; WEERKAMP; TSAGKIAS, 2013) realizado em 1.1 milhão de *tweets* em inglês mostrou que 26% possuem URLs, 16.6% apresentam o uso de *hashtags* e 54.8% possuem menção a outro *username*. Sendo assim, antes de iniciar o método proposto por Ahmed (2015) é realizada uma limpeza dos três fenômenos citados acima, mediante o uso de autômatos baseados em expressões regulares. Em seguida, são realizadas três tarefas:

1. remoção de *stopwords*;
2. tokenização dos *tweets*;
3. classificação das palavras *in vocabulary* (*IV*) e *out-of-vocabulary* (*OOV*). Nessa tarefa é utilizado um dicionário do idioma dos *tweets* processados, onde palavras não encontradas no dicionário são classificadas como OOV e palavras encontradas são classificadas como IV. Ao realizar essa classificação são desconsideradas as palavras que forem capitalizadas.

> Derczynski et al. (2015) mostra que capitalização em textos comportados (e.g. textos jornalísticos, livros) geralmente indica uma entidade nomeada, mas que em postagens do Twitter isso nem sempre se reflete. Ainda assim, optou-se por desconsiderar as palavras capitalizadas para evitar que uma possível entidade nomeada fosse normalizada como sendo uma palavra pertencente ao dicionário utilizado.

Denominamos esses processamentos iniciais **etapa 0**.

Com os *tokens* devidamente separados é iniciada a normalização léxica. Para cada OOV encontrada são executadas as etapas de 1 a 4 listadas abaixo.

A **etapa 1** da normalização proposta por Ahmed (2015) realiza o cálculo da distância de Levensthein para avaliar a similaridade do *token OOV* encontrado no *tweet* com palavras presentes em dicionários. O resultado desta etapa é uma lista de palavras do dicionário cuja distância em relação ao *token* do *tweet* é menor ou igual a dois. A escolha por um valor de distância menor ou igual a 2 apoia-se no trabalho d e Han, Cook e Baldwin (2013) que mostra que o cálculo de similaridade para distância > 2 é computacionalmente custoso. Paralelo a isso, o *threshold* ≤ 2 gera um número significativo de palavras lexicalmente similares para a próxima etapa.

A **etapa 2** da normalização verifica a similaridade fonética do *token* OOV com as palavras morfologicamente similares resultantes da etapa 1. Nessa etapa, de forma análoga a proposta de Ahmed (2015), são utilizadas apenas as palavras que apresentaram similaridade morfológica de modo a criar uma lista de palavras que possui simultaneamente duas características de similaridade.

Em sequência, na **etapa 3** é utilizado como entrada o *token* OOV e aplicado o algoritmo de Peter Norvig que retornará a palavra que apresentou a maior probabilidade de ser a forma lexicalmente correta do *token* de entrada.

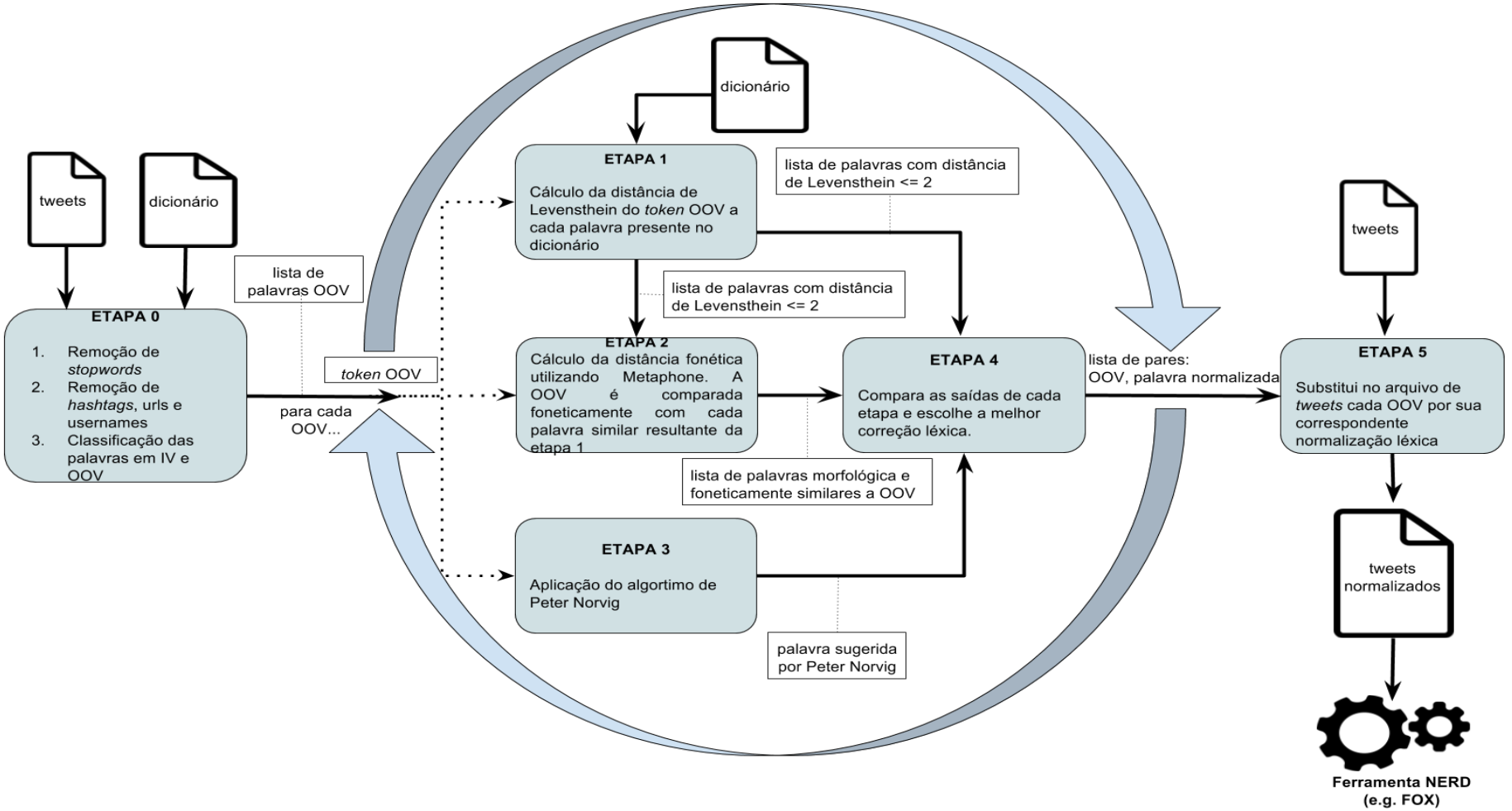
Na última etapa do processo de normalização léxica, representada pela **etapa 4**, é realizada a comparação das listas de palavras resultantes das etapas anteriores e é selecionada a melhor sugestão de normalização seguindo os seguintes critérios:

- se a lista de palavras morfológica e foneticamente similares resultante da etapa 2 for vazia, o resultado da normalização será a palavra sugerida na etapa 3;
- se a etapa 2 retorna apenas uma palavra, sendo essa a mesma sugerida na etapa 3, então essa palavra é resultado da normalização;
- se a etapa 2 resultou em mais de uma palavra, as possíveis decisões são:
    - se uma das palavras da lista for a mesma sugerida na etapa 3, então esta é a selecionada;
    - se nenhuma das palavras da lista for a mesma que a resultante da etapa 3, então é selecionada a que tiver menor distância de Levensthein;
    - se todas as palavras tiverem o mesmo valor de distância é selecionada uma delas randomicamente.

Ao final da execução das etapas de 1 a 4 para todas as OOVs tem-se então uma lista de pares contendo o *token* OOV encontrado nos *tweets* e sua forma normalizada.

Por fim, na **etapa 5** recebe-se como entrada o arquivo contendo os *tweets* não normalizados e a lista resultante da etapa 4 e realiza-se a substituição dos *tokens* OOV pelas palavras lexicalmente corretas.

Figura 4 – Fluxograma do método de filtragem dos *tweets*



4.2 MPLEMENTAÇÃO DO MÉTODO

O método proposto foi codificado utilizando a linguagem Python, a plataforma *Natural Language Toolkit*[8] (NLTK) e bibliotecas adicionais para a realização dos cálculos dos algoritmos utilizados. Tais bibliotecas são listadas no decorrer dessa seção. Na **etapa 0**, a limpeza de URLs, *hashtags* e *usernames* dos *tweets* foi feita com autômatos baseados em expressões regulares. Para a remoção de *stopwords* e tokenização utilizou-se, respectivamente, o corpus de *stopwords* e a função *RegexpTokenizer* disponíveis na NLTK. Na identificação de palavras *IV* e *OOV* foi utilizado o vocabulário do WordNet[9], que pode ser instalado juntamente com a ferramenta NLTK e acessado de maneira simples a partir da função nltk.corpus.wordnet.synsets(x), onde x representa a palavra a ser buscada no corpus.

Para as etapas 1, 2 e 3 representadas na Figura 4 foram utilizadas as seguintes configurações:

- **etapa 1**: Utiliza o arquivo word.txt contendo 235886 palavras em inglês, esse arquivo faz parte de sistemas Linux e é utilizado como dicionário pelo sistema operacional para sugerir correções quando o usuário digita uma palavra de maneira incorreta. Esse arquivo foi utilizado nessa etapa por ser o mesmo utilizado por Ahmed (2015). Para a função de Levensthein é utilizada a biblioteca *editdistance*[10] que possui a função *eval(x, y)*, onde *x* é a palavra existente no

8 http://www.nltk.org
9 https://wordnet.princeton.edu/wordnet/
10 https://pypi.python.org/pypi/editdistance

dicionário e *y* o *token OOV*; o retorno da função é a distância entre as palavras *x* e *y*.

- **etapa 2:** Ahmed (2015) utilizou o método *Refined Soundex* (disponível apenas em linguagem Java). Entretanto, para este trabalho foi escolhido o método *Metaphone* que é equivalente ao *Refined Soundex*, porém está disponível como uma biblioteca em Python. O cálculo de distância fonética é feito com a biblioteca *Metaphone*[11] a partir da chamada de função *doublemetaphone(z)* que retorna um código *hash* correspondente a pronuncia de *z*. Palavras que são pronunciadas da mesma forma tem o mesmo código *hash.* Assim, para cada palavra presente no array resultante da etapa 1, a função *doublemetaphone* é chamada duas vezes, uma onde *z = token do tweet* e outra onde *z = palavra presente no array*. Se o *hash* resultante de ambas for igual, então elas são similares foneticamente e são guardados em um segundo *array*.
- **etapa 3**: o algoritmo de Peter Norvig é realizado a partir do uso de uma classe chamada *spell*.py[12]. Tal classe precisa ser importada no *script* Python, que implementa o método proposto e, com o uso da função *spell.correction(oov)*, é obtida a melhor sugestão de correção léxica pelo algoritmo de Peter Norvig. A classe *spell*.py faz uso do corpus big.txt[13] e portanto, o mesmo também precisa estar no mesmo diretório da classe.

11 https://pypi.python.org/pypi/Metaphone/0.4
12 http://norvig.com/spell.py
13 http://norvig.com/big.txt é composto por 1 milhão de palavras e apresenta a junção de livros do projeto Gutenberg, bem como as palavras mais comuns do *Wiktionary* e do *British National Corpus*

A **etapa 4** realiza a comparação das listas de palavras resultantes das etapas anteriores e gera uma lista contendo o *token* OOV e a palavra mais provável de ser a correção léxica, conforme mostrado na proposição do método. Por fim, na **etapa 5** são substituídos nos *tweets* os *tokens* OOV pelas palavras corrigidas lexicalmente e então gera-se um novo arquivo com os *tweets* lexicalmente normalizados. A Tabela 2 mostra exemplos dos tweets antes e depois do processo de normalização léxica.

Tabela 2 - Exemplo de *tweets* antes e depois da normalização léxica.

Tweet não normalizado	*Tweet* normalizado
RT **@Harry_Styles**: Chocolate coin problems.	RT Chocolate coin problems.
I'm at Via Sul **@Shopping_viasu**l in Fortaleza, CE **https://t.co/lHlpnqQhFE**	I'm at Via Sul in Fortaleza, CE
I wanna hear **u callin** my name like HEY MAMA	I wanna hear u **calling** my name like HEY MAMA
Why can't we give love one more chance? **#queen**	Why can't we give love one more chance?

Em negrito aparecem *usernames*, *hashtags*, urls e palavras lexicalmente incorretas que foram tratadas na normalização.

5 EXPERIMENTOS E RESULTADOS

Os experimentos deste trabalho tem como objetivo avaliar o desempenho do método proposto na melhoria dos resultados de NERD. Para tal, foram realizadas as seguintes tarefas:

1. Coleta de amostra tweets da base de dados do laboratório LISA;
2. Aplicação do método proposto aos tweets selecionados;
3. Execução da ferramenta de NERD utilizando os tweets brutos;
4. Execução da ferramenta de NERD utilizando os tweets pré-processados na tarefa 2

Para verificar a contribuição do método proposto, analisou-se o aumento no número de entidades nomeadas reconhecidas nos tweets que passaram pelo pré-processamento.

Os experimentos deste trabalho foram realizados com um conjunto de 10 mil *tweets* em inglês. A escolha pelo uso de *tweets* no idioma inglês deve-se ao fato de a ferramenta de NERD utilizada para os experimentos não trabalhar com o idioma português, e portanto, descreve-se experimentos na língua portuguesa como trabalhos futuros. Os tweets foram aleatoriamente selecionados de milhões de *tweets* em inglês enviados de um *bounding box* em torno do território brasileiro e coletados em tempo real da API do Twitter entre janeiro de 2015 e outubro de 2016. Estes *tweets* representam uma amostra da base de dados disponível no laboratório LISA. A ferramenta de NERD escolhida para os experimentos é o FOX, o qual é oferecido pela Universidade de Leipzig –

Alemanha, via *Web service* baseado em arquitetura REST-full e possui razoável documentação (SPECK; NGONGA NGOMO, 2014). Para acessar o serviço, foi implementado um código simples em JAVA que envia cada *tweet* e recebe como resposta um objeto JSON contendo as entidades nomeadas encontradas no *tweet*. A seguir são mostrados os resultados obtidos com os experimentos.

5.1 FILTRAGEM DOS DADOS

A etapa de filtragem dos *tweets* levou 58 minutos e foi executada de forma independente antes das tarefas de *NER* e *NED*. A Tabela 2 mostra as estatísticas obtidas durante a execução. Os *tokens* analisados referem-se ao número de tokens encontrados já excluindo *hashtags*, urls e menções a usuários. Como já mostrado na **etapa 0**, antes da tokenização realiza-se uma limpeza dessas ocorrências. O número de diferentes *stopwords* encontradas é 153, sendo que as mesmas se repetem nos *tweets* totalizando 24031 ocorrências. As palavras relevantes, referem-se aos tokens analisados excluindo a s *stopwords*. As palavras não capitalizadas incluem os *tokens* que foram classificados como *in vocabulary* (IV) ou *out-of-vocabulary* (OOV). Lembrando que ao classificar os *tokens* são excluídos os que tiverem capitalização, pois existe a possibilidade de os mesmos serem entidades nomeadas (DERCZYNSKI et al., 2015). Os *tokens* capitalizados são 30,43% do total de *tokens* encontrados

nos *tweets* deste trabalho, valor que está de acordo com o que é explorado no trabalho de Derczynski et al. (2015), no qual três diferentes *corpus* de *tweets* apresentam entre 23% e 30% de *tokens* capitalizados. Por fim, as palavras *in vocabulary* (IV) referem-se às palavras (não capitalizadas) encontradas no dicionário, e as palavras *out-of-vocabulary (*OOV*)* indica quantos *tokens* não se encontram no dicionário utilizado, sendo assim candidatos à normalização.

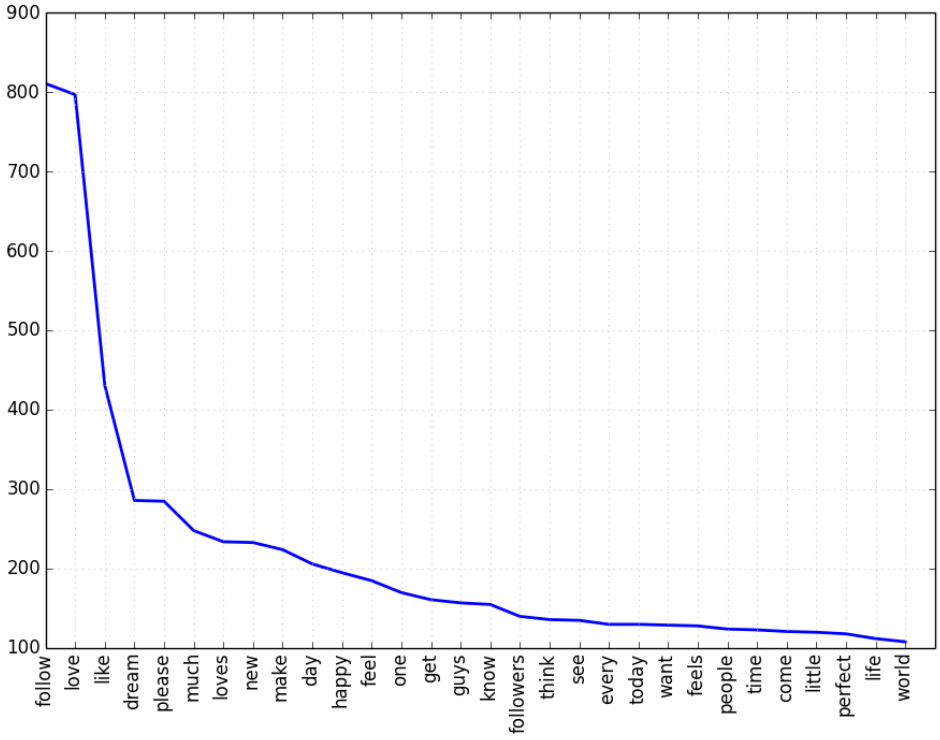
Tabela 3 - Número de palavras encontradas e como se classificam.

Tipo de Ocorrência	**Quantidade encontrada**
Tokens analisados	75293
Stopwords	24031
Palavras relevantes	51262
Palavras não capitalizadas	28348
In vocabulary (IV)	24060
Out-of-vocabulary (OOV)	4288

A Figuras 5 e 6 mostram gráficos gerados ao final da execução da filtragem. Elas mostram as 30 palavras *IV* e *OOV*, respectivamente, mais frequentes nos *tweets analisados*. Observa-se que entre essas palavras não há nenhuma que seja uma entidade nomeada, porém elas são importantes para outros tipos de anotações semânticas que não são cobertas nesse trabalho. Um exemplo são as palavras “*love*” e “*like*” que podem ser utilizadas para anotações referentes à análises de sentimento. Muitas dessas palavras são verbos que indicam ações

(e.g. seguir - *“follow”*, fazer - *“make”*, pensar - *“think”*) reforçando a necessidade de ligar palavras com seus significados.

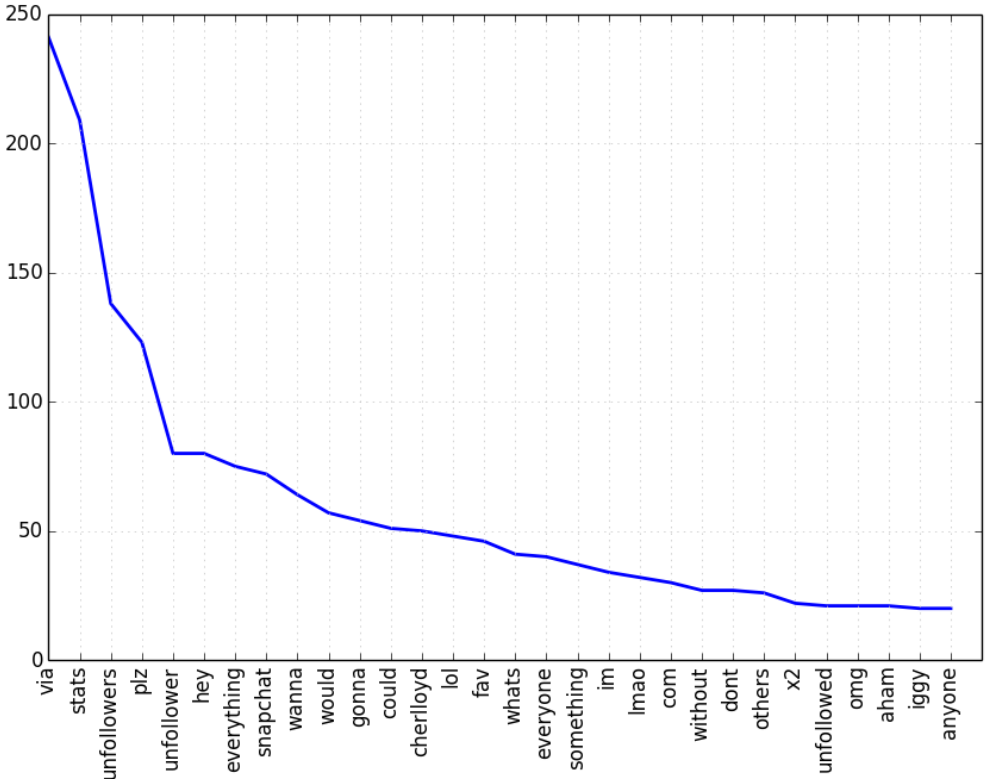
Figura 5 - As 30 palavras *IV* mais frequentes nos *tweets* processados.



Quanto às palavras OOV, podemos confirmar que o uso de gírias e abreviações contam como boa parte dos *tokens* encontrados em postagens de mídias sociais. Por exemplo, a quarta palavra mais frequente é *“*plz*”* que é utilizada como abreviação de “*please*”. Enquanto entre as 30 palavras IV mais frequentes não observou-se nenhuma ocorrência de possível entidade nomeada, nas palavras OOV aparecem duas que são nomes de superfície de entidades, sendo elas: “cherlloyd” e “iggy”, ambas referentes a cantoras

americanas. Isso mostra a importância de tratar capitalização para o reconhecimento de entidades nomeadas. A falta de capitalização dessas palavras fez com que as mesmas passassem pelo processo de filtragem, o que não deveria ser necessário já que se tratam de nomes.

Figura 6 - As 30 palavras *out-of-vocabulary OOV* mais frequentes nos *tweets* processados.



5.2 RESULTADOS DE NERD COM FILTROS E SEM FILTROS

A primeira execução realizada foi nos *tweets* sem a etapa de filtragem e a segunda foi nos *tweets* já pré-processados. Como pode-se observar pela Tabela

4, o tempo de execução de NERD foi praticamente o mesmo para os *tweets* sem e com filtragem. A seguir é apresentada uma tabela comparativa dos resultados obtidos em cada execução.

Tabela 4 – Tabela comparativa do desempenho do FOX no processamento de *tweets* com e sem filtragem

	Sem filtros	**Com filtros**
Tempo de execução do FOX	2 horas e 9 min.	2 horas e 12 min.
Número de anotações por classe:		
Localização	669	649
Organização	226	241
Pessoa	770	865
Total de *tweets* processados	10 mil	10 mil
Total de *tweets* anotados	1440	1505
Percentual de *tweets* anotados	14.4%	15.05%
Total de anotações realizadas	1665	1755
Média de anotações por *tweet* processado	0.166	0.175
Média de anotações por *tweet* anotado	1.156	1.166

Sobre o número de entidades anotadas em cada experimento foi observado um aumento de 100 entidades, ou seja, 5% a mais em relação às anotações realizadas no experimento sem utilização de filtros. Apesar de 5% não ser um valor consideravelmente alto, dentre as publicações revisadas neste trabalho

encontram-se métodos de filtragem (OLIVEIRA et al., 2013) que apresentaram uma melhoria de apenas 3%. Considerando que os dados utilizados são provenientes da *web* e não apresentam nenhuma estrutura a melhoria na tarefa de NERD obtida com o método proposto mostra-se interessante. Porém, ainda é necessário muita pesquisa para que se façam variações do método proposto (e.g. adicionar tratamentos de capitalização das palavras) para que se alcance resultados com boa relevância estatística. A seguir, as Figuras 7 e 8 mostram as 30 instâncias mais frequentes encontradas nos *tweets* sem e com filtragem, respectivamente.

Figura 7 - As 30 entidades mais frequentemente encontradas nos *tweets* sem filtragem.

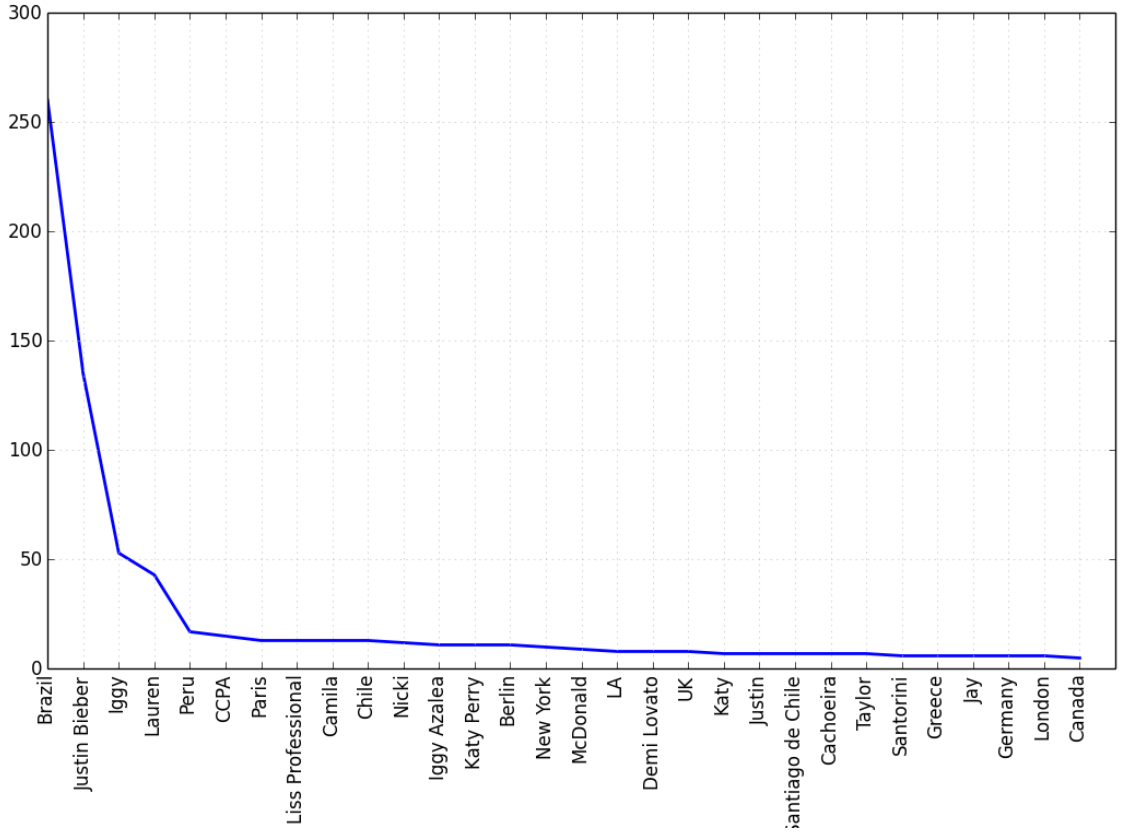
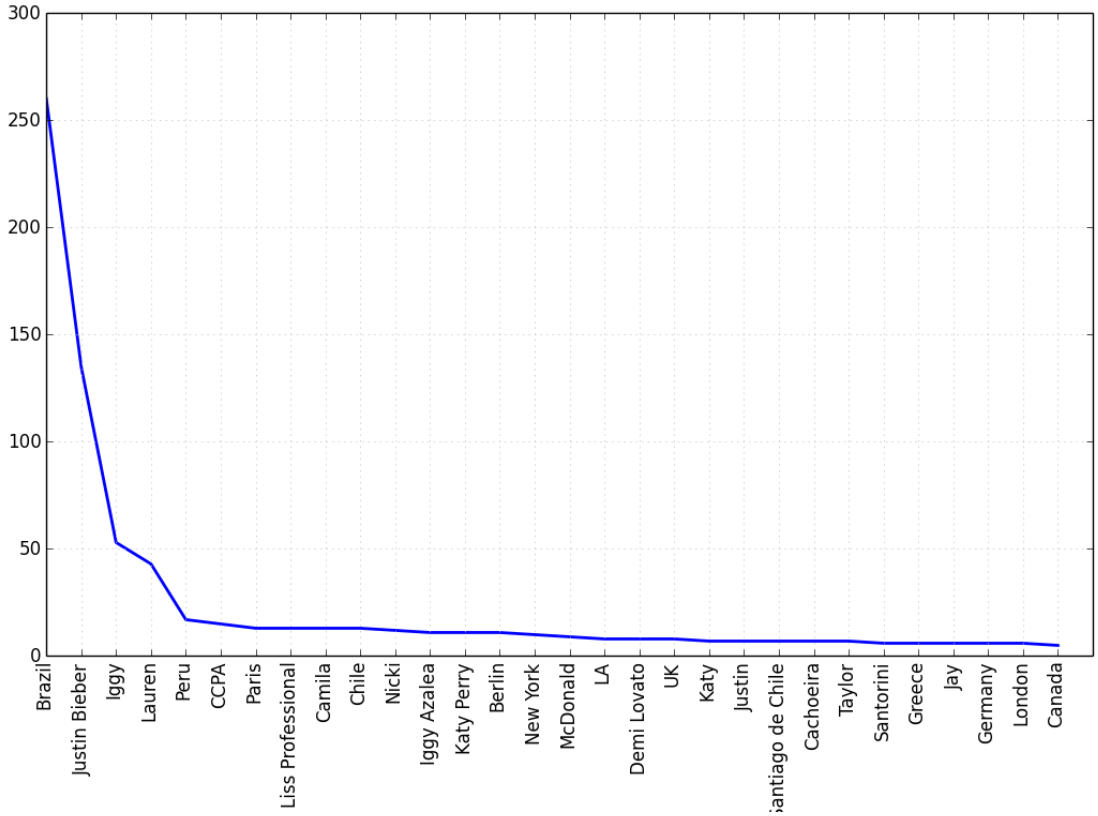

300
250
200
150
100
50
0
Brazil
Justin Bieber
Iggy
Lauren
Peru
CCPA
Paris
Liss Professional
Camila
Chile
Nicki
Iggy Azalea
Katy Perry
Berlin
New York
McDonald
LA
Demi Lovato
UK
Katy
Justin
Cachoeira
Taylor
Santorini
Greece
Jay
Germany
London
Canada

Figura 8 - As 30 entidades mais frequentemente encontradas nos *tweets* com filtragem.



Analisando as entidades nomeadas mais frequentemente encontradas em ambas as execuções da ferramenta de NERD, pode-se observar diferenças nas entidades encontradas, bem como alguns problemas a serem tratados. Apesar da maioria das entidades serem as mesmas em ambos os gráficos, observa-se que algumas tornaram-se menos frequentes, como no caso da entidade "Walmart" da classe Organização, a qual não está mais presente entre as 30 mais frequentes. Outro caso que mostra a necessidade de melhorias no método é o aparecimento de "RT" como uma entidade frequente após a filtragem dos *tweets*, já que a sigla RT significa *retweet* e é muito utilizada nos *tweets* analisados, porém foi identificada pela ferramenta de NERD como sendo uma Organização, o canal de televisão "*Russia Today"*. Além disso outro caso interessante são as entidades "Justin Bieber" e "Justin" que aparecem em ambos os gráficos. Ao realizar uma análise manual dos *tweets* foi possível observar que a maioria das menções a Justin referem-se a entidade Justin Bieber da classe Pessoa. Entretanto o FOX reconheceu as menções a Justin com sendo a uma entidade diferente de Justin Bieber. Isso mostra a dificuldade da extração e desambiguação de entidades nomeadas em *tweets*, bem como reforça a necessidade de uma regra ouro para NERD em que mídias sociais.

6 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho apresentou uma fundamentação teórica trazendo os principais conceitos necessários para o entendimento das tarefas de reconhecimento (NER) e de desambiguação (NED) de entidades nomeadas. Em seguida, as técnicas utilizadas para NER e NED (NERD) foram explicadas dividindo-as em: técnicas determinísticas e de aprendizado de máquina, enfatizando nas últimas *ensemble learning*.

Foi realizada também uma revisão bibliográfica sobre NERD em *tweets*, dando foco a trabalhos que fizeram uso de filtragem dos dados antes da realização das tarefas de NER e NED. Com base nos trabalhos estudados, este trabalho propõe e implementa um processo para filtragem dos *tweets* antes da realização da tarefa de NERD. A proposta apresentada é baseada no trabalho de Ahmed (2015) e consiste de 5 etapas, sendo elas: 0 - limpeza de ruídos (e.g. *hashtags, usernames* e urls) e separação em IV e OOV; 1 – cálculo de distância morfológica; 2 – cálculo de distância fonética; 3 - aplicação do algoritmo de Peter Norvig; 4 – comparação dos resultados de cada etapa e escolha da melhor candidata como resultado da normalização; 5 – substituição no arquivo de *tweets* das OOVs pelas suas respectivas correções.

Os experimentos foram realizados com o objetivo de verificar o aumento no número de entidades nomeadas reconhecidas após a filtragem dos dados. Dessa forma, foi feita a filtragem dos *tweets* pelo método proposto e então

realizou-se duas execuções da ferramenta FOX para medir a melhoria obtida na tarefa de NERD. Como resultados dos experimentos

observou-se que o tempo de execução foi praticamente o mesmo nos dois conjuntos de *tweets.* Entretanto, quanto ao aumento de entidades nomeadas reconhecidas foi possível observar uma melhoria de 5% no experimento com *tweets* pré-processados utilizando o método proposto. Por ser tratar de análise de conteúdo não estruturado proveniente da *web* e com base em trabalhos atuais presentes na literatura (OLIVEIRA et al., 2013) que mostram melhorias de apenas 3%, o resultado obtido neste trabalho mostra que o método proposto pode auxiliar na tarefa de NERD. Porém, o método ainda precisa ser melhorado não só para aumentar esse percentual de entidades reconhecidas, como também para evitar que palavras que não são menções a entidades nomeadas sejam identificas como tal.

Este trabalho apresenta um primeiro esforço do nosso grupo de pesquisa no uso de filtros para pré-processar dados de mídias sociais. Todavia, muito ainda há para se desenvolver em tal área de pesquisa. Entre os possíveis trabalhos futuros podemos citar:

- Implementar o método de filtragem proposto utilizando dicionários da língua portuguesa;
- Adicionar na última etapa da filtragem dos dados o uso de contexto como proposto por Ahmed (2015) para assim melhorar a escolha da palavra normalizada;

- Adicionar tratamentos para capitalização de palavras;
- Testar outras abordagens para filtragem de dados visando melhorias dos resultados de *NER* e *NED*;

- Incluir no FOX o idioma Português para realizar *NER* e *NED* nos *tweets* da base de dados do laboratório LISA, cuja maioria está em língua portuguesa;
- Instalar localmente a ferramenta FOX e incorporar a etapa de filtragem de dados no FOX
- , visando melhoria em termos de tempo de execução
- Obter conjuntos de dados com regra ouro e realizar experimentos para medir ganhos de precisão e cobertura

REFERÊNCIAS

AGGARWAL, C.ZHAI, C. **Mining text data**. New York: Springer, 2012.

AHMED, BILAL. **Lexical Normalisation of Twitter Data**. CoRR, v. abs/1409.4614, 2015. Disponível em: <http://arxiv.org/abs/1409.4614>.

AMARAL, D. O. F. do; VIEIRA, R. **NERP-CRF: uma ferramenta para o reconhecimento de entidades nomeadas por meio de conditional random fields**. v. 6, n. 1, p. 41–49, Julho 2014. ISSN 1647–0818. Disponível em: <http://linguamatica.com/index.php/linguamatica/article/view/v6n1-03>.

AMITAVA, Das et al. NER from Tweets: SRI-JU System @MSM 2013. In: MAKING SENSE OF MICROPOSTS #MSM2013, 3., 2013, Rio de Janeiro. **Proceedings of the Concept Extraction Challenge**. Rio de Janeiro: Ceur Workshop Proceedings, 2013. v. 1019, p. 62 - 66. Disponível em: <http://ceur-ws.org/Vol-1019/paper_33.pdf>. Acesso em: 28 nov. 2016.

ASAHARA, Masayuki; MATSUMOTO, Yuji. Japanese named entity extraction with redundant morphological analysis. In: NAACL '03, 04., 2003, Edmonton. **Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology**. Stroudsburg: Association For Computational Linguistics, 2003. v. 1, p. 8 - 15. Disponível em: <http://dx.doi.org/10.3115/1073445.107344>. Acesso em: 28 nov. 2016.

BIKEL, Daniel M.; SCHWARTZ, Richard; WEISCHEDEL, Ralph M.. An Algorithm that Learns What's in a Name. **Machine Learning**. [s. L], p. 211-231. fev. 1999. Disponível em: <http://dx.doi.org/10.1023/A:1007558221122>. Acesso em: 30 nov. 2016.

BONTCHEVA, Kalina.; ROUT, Dominic. **Making sense of social media streams through semantics: A survey**. Semantic Web, v. 5, n. 5, p. 373-403, 2014.

BONTCHEVA, Kalina; DERCZYNSKI, Leon; ROBERTS, Ian. **Crowdsourcing Named Entity Recognition and Entity Linking Corpora**. 2014. In Nancy Ide and James Pustejovsky, eds.: The Handbook of Linguistic Annotation.

CARTER, Simon; WEERKAMP, Wouter; TSAGKIAS, Manos. Microblog language identification: overcoming the limitations of short, unedited and idiomatic text. **Language Resources And Evaluation**. [s. L], p. 195-215. mar. 2013. Disponível em: <http://dx.doi.org/10.1007/s10579-012-9195-y>. Acesso em: 28 nov. 2016.

CHINCHOR, N. Overview of muc-7. In: Seventh Message Understanding Conference (MUC-7): **Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998.** [s.n.], 1998. Disponível em: <http://www.aclweb.org/anthology/M98-1001>.

DAS, T.; ACHARJYA, D.; PATRA, M. Opinion mining about a product by analyzing public tweets in Twitter. **2014 International Conference on Computer Communication and Informatics**, p. 1-4, 2014.

DERCZYNSKI, L. et al. Analysis of named entity recognition and linking for tweets. **Information Processing & Management**, Elsevier, v. 51, n. 2, p. 32–49, 2015.

EARLE, Paul S.; BOWDEN, Daniel C.; GUY, Michelle. Twitter earthquake detection: earthquake monitoring in a social world. **Annals of Geophysics**, [S.l.], v. 54, n. 6, jan. 2012. ISSN 2037-416X. Disponível em: <http://www.annalsofgeophysics.eu/index.php/annals/article/view/5364>

GRISHMAN, R.; SUNDHEIM, B. Message understanding conference-6: A brief history. In: **COLING**. [S.l.: s.n.], 1996. v. 96, p. 466–471.

HAN, Bo; BALDWIN, Timothy. Lexical Normalisation of Short Text Messages: Makn Sens a \#Twitter. In: HLT '11, 49., 2011, Portland. **Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies**. Stroudsburg: Association For Computational Linguistics, 2011. v. 1, p. 368 - 378. Disponível em: <http://dl.acm.org/citation.cfm?id=2002472.2002520>. Acesso em: 28 nov. 2016.

HAN, Bo; COOK, Paul; BALDWIN, Timothy. Automatically Constructing a Normalisation Dictionary for Microblogs. In: EMNLP-CONLL '12, 09., 2012, Jeju Island. **Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning**. Stroudsburg: Association For Computational Linguistics, 2012. p. 421 - 432. Disponível em: <http://dl.acm.org/citation.cfm?id=2390948.2391000>. Acesso em: 28 nov. 2016.

HAN, Bo; COOK, Paul; BALDWIN, Timothy. Lexical Normalization for Social Media Text. **ACM Transactions On Intelligent Systems And Technology**. New York, p. 1-27. jan. 2013. Disponível em: <http://doi.acm.org/10.1145/2414425.2414430>. Acesso em: 28 nov. 2016.

IBRAHIM, Y.; YOSEF, M.; WEIKUM, G. AIDA-Social : Entity Linking on the Social Stream. **Proceedings of the 7th International Workshop on Exploiting Semantic Annotations in Information Retrieval** - ESAIR '14, p. 17-19, 2014.

KALLOUBI, F.; NFAOUI, E. H.; BEQQALI, O. E. Graph based tweet entity linking using dbpedia. In: **11th IEEE/ACS International Conference on Computer Systems and Applications, AICCSA 2014**, Doha, Qatar, November 10-13, 2014. [s.n.], 2014. p. 501–506. Disponível em: <http://dx.doi.org/10.1109/AICCSA.2014.7073240>.

KAPLAN, R. M. A method for tokenizing text. I**nquiries into words, constraints and contexts**, p. 55, 2005.

KLEIN, Douglas. **Reconhecimento e Desambiguação de Entidades nomeadas com foco em Mídias Sociais**. 2015. 72 f. TCC (Graduação) - Curso de Sistemas de Informação, Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis, 2015.

KONKOL, M. **Named entity recognition**. 2012. Disponível em: <http://www.kiv.zcu.cz/site/documents/verejne/vyzkum/publikace/technicke-zpravy/2012/tr-2012-04.pdf>.

LIU, X. et al. Recognizing named entities in tweets. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1**. [S.l.], 2011. p. 359–367.

MARRERO, M. et al. Named Entity Recognition: Fallacies, challenges and opportunities. **Computer Standards & Interfaces**, v. 35, n. 5, p. 482-489, 2013.

MCCALLUM, Andrew; LI, Wei. Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-enhanced Lexicons. In: CONFERENCE ON NATURAL LANGUAGE LEARNING, 17., 2003, Edmonton. **Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003**. Stroudsburg: Association For Computational Linguistics,

2003. v. 4, p. 188 - 191. Disponível em: <http://dx.doi.org/10.3115/1119176.1119206>. Acesso em: 28 nov. 2016.

NADEAU, David. **Semi-Supervised Named Entity Recognition:** Learning to Recognize 100 Entity Types with Little Supervision. 2007. 142 f. Tese (Doutorado) - Curso de Ciência da Computação, School Of Information Technology And Engineering, University Of Ottawa, Ottawa, 2007.

NADEAU, D.; SEKINE, S. A survey of named entity recognition and classification. **Lingvisticae Investigationes**, John Benjamins publishing company, v. 30, n. 1, p. 3–26, 2007.

NADKARNI, P.; OHNO-MACHADO, L.; CHAPMAN, W. Natural language processing: an introduction. **Journal of the American Medical Informatics Association**, v. 18, n. 5, p. 544-551, 2011.

NGUYEN, N.; GUO, Y. Comparisons of sequence labeling algorithms and extensions. In: **ACM. Proceedings of the 24th international conference on Machine learning**. [S.l.], 2007. p. 681–688.

OLIVEIRA, D. M. de et al. Fs-ner: A lightweight filter-stream approach to named entity recognition on twitter data. In: **Proceedings of the 22Nd International Conference on World Wide Web**. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2013. (WWW '13 Companion), p. 597–604. ISBN 978-1-4503-2038-2. Disponível em: <http://dl.acm.org/citation.cfm?id=2487788.2488003>.

PAZZANI, M.BILLSUS, D. **Content-Based Recommendation Systems**. The Adaptive Web, p. 325-341, 2007.

RAO, Delip; MCNAMEE, Paul; DREDZE, Mark. Entity Linking: Finding Extracted Entities in a Knowledge Base. In: POIBEAU, Thierry et al (Ed.). **Multi-source, Multilingual Information Extraction and Summarization**. Heidelberg: Springer Berlin Heidelberg, 2013. Cap. 2. p. 93-115. (Theory and Applications of Natural Language Processing). Disponível em: <http://dx.doi.org/10.1007/978-3-642-28569-1_5>. Acesso em: 30 nov. 2016.

RATINOV, Lev; ROTH, Dan. Design Challenges and Misconceptions in Named Entity Recognition. In: CONLL '09, 13., 2009, Boulder. **Proceedings of the Thirteenth Conference on Computational Natural Language Learning**.

Stroudsburg: Association For Computational Linguistics, 2009. p. 147 - 155. Disponível em: <http://dl.acm.org/citation.cfm?id=1596374.1596399>. Acesso em: 30 nov. 2016.

RITTER, A. et al. Named entity recognition in tweets: An experimental study. In: **Proceedings of the Conference on Empirical Methods in Natural Language Processing**. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011. (EMNLP '11), p. 1524–1534. ISBN 978-1-937284-11-4. Disponível em: <http://dl.acm.org/citation.cfm?id=2145432.2145595>.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 3rd. ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2009. ISBN 0136042597, 9780136042594.

SHA, F.; PEREIRA, F. Shallow parsing with conditional random fields. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1**. [S.l.], 2003. p. 134–141.

SHEN, W.; WANG, J.; HAN, J. Entity linking with a knowledge base: Issues, techniques, and solutions. **Knowledge and Data Engineering**, IEEE, v. 27, p. 443–460, 2014.

SPECK, R.; NGONGA NGOMO, A.-C. Ensemble learning for named entity recognition. In: **The Semantic Web – ISWC 2014**. Springer International Publishing, 2014, (Lecture Notes in Computer Science, v. 8796). p. 519–534. Disponível em: <http://svn.aksw.org/papers/2014/ISWCEL4NER/public.pdf>.

SUTTON, C.; MCCALLUM, A. An introduction to conditional random fields. **Machine Learning**, v. 4, n. 4, p. 267–373, 2011.

WU, D. et al. Domain adaptive bootstrapping for named entity recognition. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3**. [S.l.], 2009. p. 1523–1532.

ZHAI, C.MASSUNG, S. **Text data management and analysis: A Practical Introduction to Information Retrieval and Text Mining**. Association for Computing Machinery and Morgan, 2016.

APÊNDICE A – Artigo Produzido com Base no TCC

PRÉ-PROCESSAMENTO DE *TWEETS* VISANDO MELHORAR RESULTADOS DE *NERD*

Suelen C. Fenali[1], Renato Fileto[1]

[1]Departamento de Informática e Estatística
Universidade Federal de Santa Catarina (UFSC) – Florianópolis, SC – Brazil

suelen.cfenali@gmail.com, r.fileto@ufsc.br

Abstract. *The semantic enrichment of posts in social media can bring several benefits in applications. However, the information extraction techniques and tools currently available in the literature are not prepared to work with data from these sources, which are very affected by noises. This work proposes a method for filtering tweets based on lexical normalization to reduce noise and obtain better results in the recognition and naming entity disambiguation (NERD) tasks. In order to verify the quality of the proposed method, experiments were performed with the FOX tool and a 5% increase in the number of named entities recognized after the filtering.*

Resumo. *O enriquecimento semântico de postagens em mídias sociais pode trazer diversos benefícios em aplicações. Todavia, as técnicas e ferramentas de extração de informação presentes na literatura não trabalham adequadamente com dados provenientes dessas fontes, os quais estão sujeitos a ruídos diversos. Este trabalho propõe um método para filtragem de tweets baseado em normalização léxica visando diminuir ruídos e obter melhores resultados nas tarefas de reconhecimento e desambiguação de entidades nomeadas (NERD). Para verificar a qualidade do método proposto, foram realizados experimentos com a ferramenta FOX e observou-se um aumento de 5% no número de entidades nomeadas reconhecidas após a filtragem.*

1. Introdução

O massivo uso das mídias sociais tem gerado grandes volumes de dados a cada dia. Postagens compartilhadas publicamente em tais mídias (e.g., *Twitter*, *Facebook*, *Flickr*) podem ser obtidas via APIs (*Application Programming Interface*) e apresentam conteúdos sobre os mais diversos assuntos, incluindo muitas vezes opiniões e sentimentos dos seus autores. O entendimento preciso do que é publicado em mídias sociais [4] pode contribuir na melhoria do desempenho de tarefas como mineração de opiniões [7], recomendação de produtos e/ou serviços [19] e até mesmo detecção de catástrofes naturais [9].

Entretanto, postagens em mídias sociais têm conteúdos na forma de textos não estruturados em linguagem natural. A Extração de Informação (*Information Extraction* - IE) de textos [1][24] visa detectar fragmentos relevantes e a eles associar semântica bem definida por meio de diversas tarefas de processamento. A tarefa de Reconhecimento de Entidades Nomeadas (*Named Entity Recognition – NER*) visa encontrar e classificar menções a entidades nomeadas (e.g., lugares, pessoas, instituições, valores, números de telefone, especificações de tempo), enquanto a tarefa subsequente de Desambiguação de

Entidades Nomeadas (*Named Entity Disambiguation - NED*) [17], busca ligar precisamente tais menções a definições correspondentes em uma base de conhecimento (e.g., Avaí referindo-se à batalha da Guerra do Paraguai, município do estado de São Paulo, time de futebol de Florianópolis, Joinville, Laguna ou Guaramirim). Neste trabalho denominamos NERD (*Named Entity Recognition and Disambiguation*) a composição das tarefas de NER e NED. Tais tarefas são realizadas com o uso de técnicas determinísticas e de aprendizado de máquina em diversas propostas presentes na literatura [14].

As abordagens do estado da arte em NER/NED podem produzir resultados de qualidade aceitável quando aplicadas a textos bem formados. Porém, isso não acontece com postagens em mídias sociais, que costumam conter muitos ruídos, tais como gírias, abreviações e erros de escrita que dificultam a extração de informações [21][13][8]. Desta forma, várias abordagens têm sido propostas na literatura para efetuar NER/NED em conteúdos de postagens em mídias sociais. Entre tais abordagens algumas [11][21][3][12] [18][2] baseiam-se em filtros, com o intuito de diminuir os ruídos presentes nos dados antes de realizar as tarefas de reconhecimento e desambiguação de entidades nomeadas. Além dessas abordagens, a técnica de aprendizado de máquina *ensemble learning* pode ser utilizada para aumentar precisão e cobertura de métodos de NER/NED [23].

Este trabalho propõe um método de pré-processamento de postagens do Twitter[14] para posterior reconhecimento e desambiguação de entidades nomeadas (NERD). O método proposto baseia-se em normalização léxica dos *tweets* e visa reduzir ruídos presentes no texto de modo a aumentar o número de entidades nomeadas reconhecidas pela ferramenta de NERD. O presente trabalho avalia o desempenho do método proposto a partir de experimentos com a ferramenta de código aberto FOX[15], a qual faz uso da técnica de *ensemble learning* para combinar resultados de algumas das melhores ferramentas NERD abertas atuais para obter resultados melhores que os gerados por cada ferramenta da composição individualmente. Assim, os resultados da avaliação com o FOX, que também permite usar cada uma das ferramentas que ele engloba isoladamente, permite uma apreciação dos benefícios do método de filtragem proposto no estado-da-arte em NER.

Este artigo divide-se em 4 seções. A introdução que foi apresentada é a primeira. A segunda apresenta fundamentos e definições para o entendimento do trabalho realizado. A seção 3 apresenta trabalhos relacionados. A seção 4 apresenta o método proposto. A seção 5 apresenta os experimentos e resultados obtidos. Por fim, a seção 6 concluí o artigo e propõe trabalhos futuros.

2. Fundamentação

O termo **entidade nomeada** (*named entity* - NE) foi usado pela primeira vez por Grishman e Sundheim (1996) [10] na 6ª Message Understanding Conference (MUC-6). Uma entidade nomeada é uma "unidade de informação" que pode referir-se a pessoas, lugares, organizações, tempo (horas e datas) e quantidades (valores monetários, percentuais) [17].

Um termo escrito em linguagem natural de forma a designar uma entidade nomeada é chamado **nome de superfície** (e.g., "BSB" e "Brasília" são nomes de superfície para a capital do Brasil). Um **menção** é a ocorrência de um nome de superfície em um texto [14][20]. O **contexto textual** de uma menção corresponde a sentença ou parágrafo onde ela ocorre, ou a uma janela de k palavras à esquerda e à direita da mesma.

Para dar sentindo a menções de entidades pode-se ligá-las a definições precisas que podem ser encontradas em **bases de conhecimento** (Knowledge Bases - KB), tais

14 https://twitter.com

15 http://aksw.org/Projects/FOX.html

como: Wikipedia[16], DBPedia[17], LinkedGeoData[18] [20]. Tais ligações são chamadas anotações semânticas. Elas auxiliam no entendimento preciso das informações e no processo de expansão semântica, i.e., acesso ou derivação de novas informações a partir das definições semânticas usadas nas anotações.

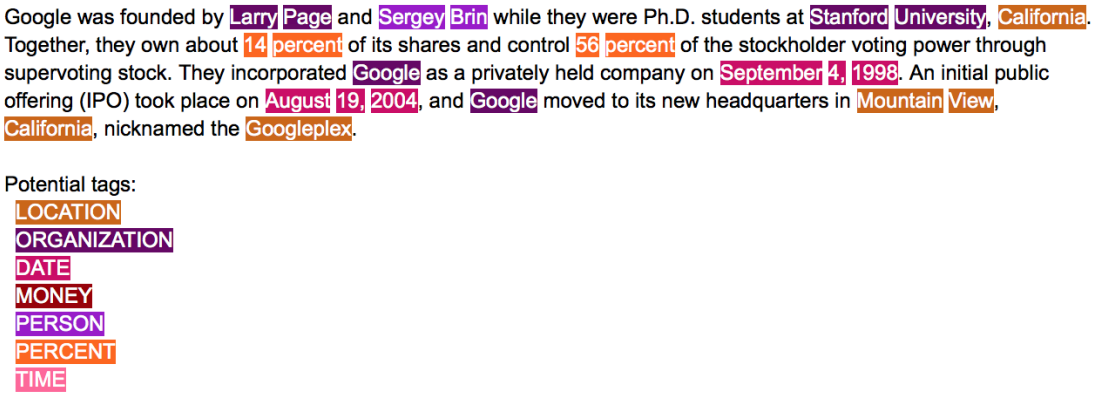
2.1. Reconhecimento de Entidades Nomeadas

O Reconhecimento de Entidades Nomeadas (*Named Entity Recognition* - NER) é uma tarefa de extração de informação que consiste na identificação de entidades nomeadas presentes em textos não estruturados [8]. NER serve de base para diversas áreas, tais como: realização de anotações semânticas, população de ontologias e mineração de opinião [16]. Podemos definir NER a partir de duas subtarefas [23][14]:

1. identificação: refere-se a identificação de menções $m \in M$ presentes em documentos $d \in D$;
2. classificação: consiste na geração de tuplas $\{ (m, t_m) \mid m \in M \wedge t_m \in T \wedge t_m$ é o tipo de m $\}$.

Pode-se observar (Figura 1) um exemplo do resultado da aplicação de NER. O reconhecimento das entidades foi realizado utilizando a ferramenta Stanford NER[19] configurada para encontrar as classes de entidades nomeadas de acordo com as definições da MUC-7 [6].

Figura 9 – Exemplo de texto processado com a ferramenta Stanford Named Entity Tagger.



2.2. Desambiguação de Entidades Nomeadas

A Desambiguação de Entidades Nomeadas (*Named Entity Disambiguation - NED*) busca encontrar a definição mais apropriada para uma menção a entidade nomeada em uma base de conhecimento [8]. Ao buscar por entidades em KBs, a tarefa de NED ajuda a resolver casos de polissemia e sinonímia. Todavia, há situações onde há uma ausência de definição para a menção na base de conhecimento [22].

16 https://wikipedia.org

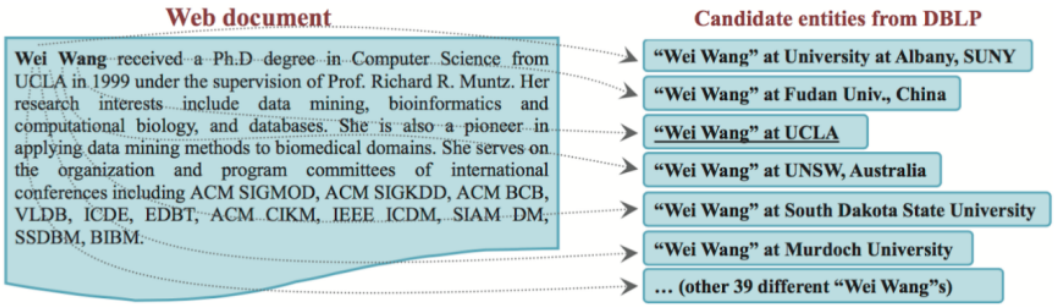
17 http://dbpedia.org

18 http://linkedgeodata.org/About

19 http://nlp.stanford.edu/software/CRF-NER.shtml

A Figura 2 mostra a seleção de entidades candidatas para a menção "Wei Wang". Um dos desafios de NED é a identificação da entidade correta em um determinado contexto, sendo necessário o uso de técnicas que auxiliem no tratamento desse problema.

Figura 10 - Exemplo de geração de entidades candidatas na tarefa de NED. A entidade correta está sublinhada [22].



2.3. Técnicas de NERD

Segundo Konkol (2012), as técnicas para efetuar NER e/ou NED podem ser categorizadas segundo diferentes aspectos. Atualmente, técnicas baseadas em Aprendizado de Máquina (AM) e técnicas mistas têm suplantado as técnicas tradicionais baseadas em padrões linguísticos codificados manualmente em sistemas de NERD, principalmente no que se refere à qualidade dos resultados gerados. Técnicas de aprendizado de máquina podem ser classificadas em aprendizado supervisionado, semi supervisionado e não supervisionado. Essa categorização é feita de acordo com o tipo de dado necessário para a máquina criar os parâmetros de entrada.

2.3.1. Ensemble Learning

Buscando melhorar os resultados de um classificador tem-se feito uso da combinação de diferentes técnicas. Uma técnica de AM para confecção automática de combinações é ***ensemble learning***. Um algoritmo de ensemble learning S tem como objetivo gerar um classificador F com uma alta performance preditiva através da combinação de k classificadores básicos $C_1,...,C_m$ [23]. Essa técnica é a utilizada pela ferramenta FOX, a ser avaliada neste trabalho.

Speck e Ngomo (2014) apontam que, devido a muitos anos de pesquisa, a precisão e a cobertura das técnicas utilizadas para NER são hoje aceitáveis para diversas coleções de dados. A combinação de classificadores gera melhores resultados que o uso dos mesmos individualmente, principalmente quando é realizada uma combinação de classificadores que utilizem diferentes técnicas ou conjuntos de características [15].

3. Trabalhos Relacionados

Recentemente as pesquisas sobre NER/NED aplicados a dados de mídias sociais têm ganhado atenção da comunidade acadêmica. As técnicas atuais de extração de conteúdo relevante funcionam razoavelmente bem para textos escritos de maneira apropriada (e.g., textos jornalísticos, obras literárias e etc.). Entretanto, o reconhecimento e a desambiguação de entidades nomeadas em *tweets* é um grande desafio. Derczynski et al. (2015) apontam que técnicas de NER quando aplicadas a textos longos apresentam entre 85 e 90% de precisão, enquanto quando aplicados a *tweets* obtêm entre 30 e 50% de

precisão, fora os problemas de cobertura. Alguns trabalhos [21][3][12][18][2][8] mostram os desafios encontrados ao processar postagens no *Twitter*:

a) tamanho reduzido do texto, tornando difícil a realização de anotações semânticas, particularmente a desambiguação, devido à carência de informação de contexto;

b) presença de ruídos provenientes do uso da linguagem natural sobre a Web (e.g., erros gramaticais, gírias, abreviações, *emoticons* e *hashtags*);

c) dependência de contexto (situação do usuário e seu ambiente, contexto da mensagem em meio a outras), o que impede em alguns casos a ligação às entidades nomeadas corretas.

Alguns trabalhos (Tabela 1) tem buscado solucionar tais problemas a partir do desenvolvimento de algoritmos de NERD específicos para microblogs, sendo que boa parte deles realiza uma etapa de filtragem dos *tweets* visando a melhorar os resultados das tarefas de NER e NED.

Tabela 1 – Tabela comparativa entre as ferramentas que utilizam filtragem no pré processamento dos *tweets* para NER/NED.

Característica	**Ritter et al. (2011)**	**Han, Cook e Baldwin (2013)**
Proposta	Filtra os tweets de acordo com uma nova pipeline de PLN buscando melhorias em relação as ferramentas do estado-da-arte	Realiza normalização léxica criando um dicionário baseado em similaridade de contexto, de morfologia e fonética das palavras
Software Disponível	Github aritter/twitter_nlp	Não
Idiomas	Inglês	Inglês
Benefícios	Aumento da acurária nas sub tarefas (POS *Tagging, shallow parsing*) de PLN	Aumento da acurácia ao realizar a tarefa de normalização léxica
Característica	**Oliveira et al. (2013)**	**Ahmed (2015)**
Proposta	Aplica cinco filtros sobre os tweets de modo a aumentar precisão, cobertura e F-measure na tarefa de NER	Realiza normalização léxica de palavras que apresentam erro de escrita ou abreviações e gíriasRealiza normalização léxica criando um dicionário baseado em similaridade de contexto, de morfologia e fonética das palavras
Software Disponível	Github dmoliveira/FSNER	Não
Idiomas	Inglês, Português e Outros	Inglês – adaptável para outros idiomas

Benefícios	Aumento de 3% em relação a outras ferramentas de CRF e velocidade no tempo de execução	Aumento da acurácia ao realizar a tarefa de normalização léxica

4. Método Proposto

O método proposto para pré-processamento de *tweets* antes do seu enriquecimento semântico tem foco na normalização léxica do texto dos *tweets* e utiliza a *baseline* proposta por Ahmed (2015), com algumas modificações que visam melhorar o processo de normalização. A Figura 3 apresenta um fluxograma com as etapas do método proposto.

Um estudo [5] realizado em 1.1 milhão de *tweets* em inglês mostrou que 26% possuem URLs, 16.6% apresentam o uso de *hashtags* e 54.8% possuem menção a outro *username*. Sendo assim, antes de iniciar o método proposto por Ahmed (2015) é realizada uma limpeza dos três fenômenos citados acima, mediante o uso de autômatos baseados em expressões regulares. Em seguida, são realizadas três tarefas:

1. remoção de *stopwords*;
2. tokenização dos *tweets*;
3. classificação das palavras *in vocabulary* (*IV*) e *out-of-vocabulary* (*OOV*). Nessa tarefa é utilizado um dicionário do idioma dos *tweets* processados, onde palavras não encontradas no dicionário são classificadas como OOV e palavras encontradas são classificadas como IV. Ao realizar essa classificação são desconsideradas as palavras que forem capitalizadas. Derczynski et al. (2015) mostra que capitalização em textos comportados (e.g. textos jornalísticos, livros) geralmente indica uma entidade nomeada, mas que em postagens do Twitter isso nem sempre se reflete. Ainda assim, optou-se por desconsiderar as palavras capitalizadas para evitar que uma possível entidade nomeada fosse normalizada como sendo uma palavra pertencente ao dicionário utilizado.

Denominamos esses processamentos iniciais **etapa 0**.

Com os *tokens* devidamente separados é iniciada a normalização léxica. Para cada OOV encontrada são executadas as etapas de 1 a 4 listadas abaixo.

A **etapa 1** da normalização proposta por Ahmed (2015) realiza o cálculo da distância de Levensthein para avaliar a similaridade do *token OOV* encontrado no *tweet* com palavras presentes em dicionários. O resultado desta etapa é uma lista de palavras do dicionário cuja distância em relação ao *token* do *tweet* é menor ou igual a dois. A escolha por um valor de distância menor ou igual a 2 apoia-se no trabalho [12] que mostra que o cálculo de similaridade para distância > 2 é computacionalmente custoso. Paralelo a isso, o *threshold* ≤ 2 gera um número significativo de palavras lexicalmente similares para a próxima etapa.

A **etapa 2** da normalização verifica a similaridade fonética do *token* OOV com as palavras morfologicamente similares resultantes da etapa 1. Nessa etapa, de forma análoga a proposta de Ahmed (2015), são utilizadas apenas as palavras que apresentaram similaridade morfológica de modo a criar uma lista de palavras que possui simultaneamente duas características de similaridade.

Em sequência, na **etapa 3** é utilizado como entrada o *token* OOV e aplicado o algoritmo de Peter Norvig que retornará a palavra que apresentou a maior probabilidade de ser a forma lexicalmente correta do *token* de entrada.

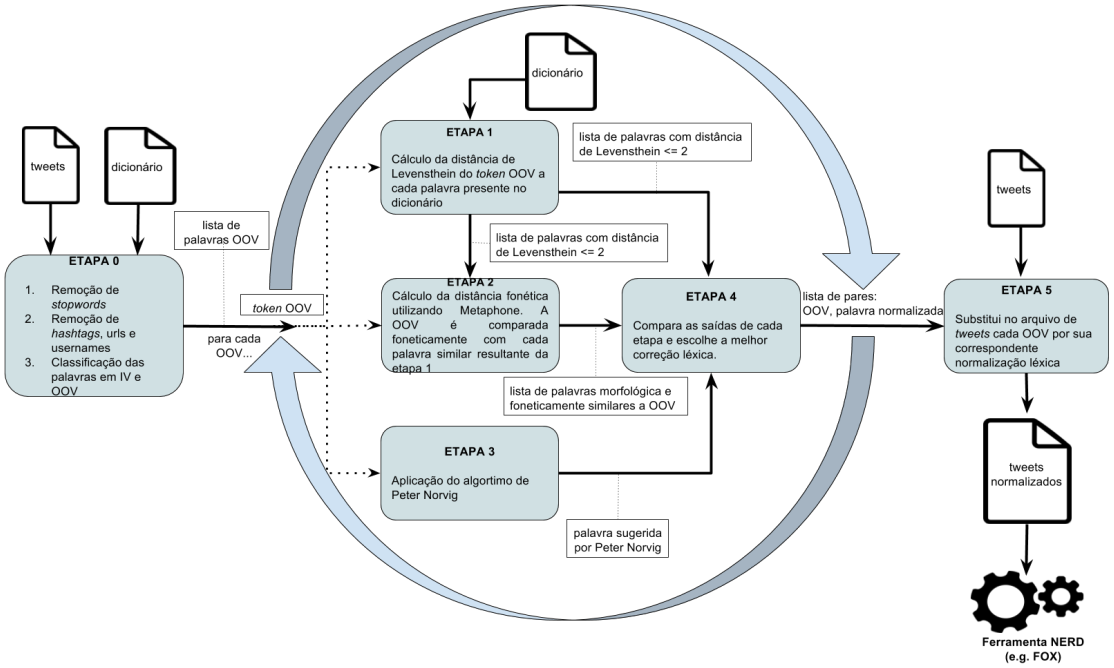
Na última etapa do processo de normalização léxica, representada pela **etapa 4**, é realizada a comparação das listas de palavras resultantes das etapas anteriores e é selecionada a melhor sugestão de normalização seguindo os seguintes critérios:

- se a lista de palavras morfológica e foneticamente similares resultante da etapa 2 for vazia, o resultado da normalização será a palavra sugerida na etapa 3;
- se a etapa 2 retorna apenas uma palavra, sendo essa a mesma sugerida na etapa 3, então essa palavra é resultado da normalização;
- se a etapa 2 resultou em mais de uma palavra, as possíveis decisões são:
  - se uma das palavras da lista for a mesma sugerida na etapa 3, então esta é a selecionada;
  - se nenhuma das palavras da lista for a mesma que a resultante da etapa 3, então é selecionada a que tiver menor distância de Levensthein;
  - se todas as palavras tiverem o mesmo valor de distância é selecionada uma delas randomicamente.

Ao final da execução das etapas de 1 a 4 para todas as OOVs tem-se então uma lista de pares contendo o *token* OOV encontrado nos *tweets* e sua forma normalizada.

Por fim, na **etapa 5** recebe-se como entrada o arquivo contendo os *tweets* não normalizados e a lista resultante da etapa 4 e realiza-se a substituição dos *tokens* OOV pelas palavras lexicalmente corretas.

Figura 3 – Fluxograma do método proposto



4.1. Implementação do Método

O método proposto foi codificado utilizando a linguagem Python, a plataforma *Natural Language Toolkit*[20] (NLTK) e bibliotecas adicionais para a realização dos cálculos dos algoritmos utilizados. Tais bibliotecas são listadas no decorrer dessa seção. N a **etapa 0**, a limpeza de URLs, *hashtags* e *usernames* dos *tweets* foi feita com autômatos baseados em expressões regulares. Para a remoção de *stopwords* e tokenização utilizou-se, respectivamente, o corpus de *stopwords* e a função *RegexpTokenizer*

20

http://www.nltk.org

disponíveis na NLTK. Na identificação de palavras *IV* e *OOV* foi utilizado o vocabulário do WordNet[21], que pode ser instalado juntamente com a ferramenta NLTK e acessado de maneira simples a partir da função nltk.corpus.wordnet.synsets(x), onde x representa a palavra a ser buscada no corpus.

Para as etapas 1, 2 e 3 representadas na Figura 4 foram utilizadas as seguintes configurações:

- **etapa 1**: Utiliza o arquivo word.txt contendo 235886 palavras em inglês, esse arquivo faz parte de sistemas Linux e é utilizado como dicionário pelo sistema operacional para sugerir correções quando o usuário digita uma palavra de maneira incorreta. Esse arquivo foi utilizado nessa etapa por ser o mesmo utilizado por Ahmed (2015). Para a função de Levensthein é utilizada a biblioteca *editdistance*[22] que possui a função *eval(x, y)*, onde *x* é a palavra existente no dicionário e *y* o *token OOV*; o retorno da função é a distância entre as palavras *x* e *y*.
- **etapa 2:** Ahmed (2015) utilizou o método *Refined Soundex* (disponível apenas em linguagem Java). Entretanto, para este trabalho foi escolhido o método *Metaphone* que é equivalente ao *Refined Soundex*, porém está disponível como uma biblioteca em Python. O cálculo de distância fonética é feito com a biblioteca *Metaphone*[23] a partir da chamada de função *doublemetaphone(z)* que retorna um código *hash* correspondente a pronuncia de *z*. Palavras que são pronunciadas da mesma forma tem o mesmo código *hash*. Assim, para cada palavra presente no array resultante da etapa 1, a função *doublemetaphone* é chamada duas vezes, uma onde *z = token do tweet* e outra onde *z = palavra presente no array*. Se o *hash* resultante de ambas for igual, então elas são similares foneticamente e são guardados em um segundo *array*.
- **etapa 3**: o algoritmo de Peter Norvig é realizado a partir do uso de uma classe chamada *spell*.py[24]. Tal classe precisa ser importada no *script* Python, que implementa o método proposto e, com o uso da função *spell.correction(oov)*, é obtida a melhor sugestão de correção léxica pelo algoritmo de Peter Norvig. A classe *spell*.py faz uso do corpus big.txt[25] e portanto, o mesmo também precisa estar no mesmo diretório da classe.

A **etapa 4** realiza a comparação das listas de palavras resultantes das etapas anteriores e gera uma lista contendo o *token* OOV e a palavra mais provável de ser a correção léxica, conforme mostrado na proposição do método. Por fim, na **etapa 5** são substituídos nos *tweets* os *tokens* OOV pelas palavras corrigidas lexicalmente e então gera-se um novo arquivo com os *tweets* lexicalmente normalizados.

5. Experimentos e Resultados

Os experimentos deste trabalho tem como objetivo avaliar o desempenho do método proposto na melhoria dos resultados de NERD. Para tal, foram realizadas as seguintes tarefas:

21 https://wordnet.princeton.edu/wordnet/

22 https://pypi.python.org/pypi/editdistance

23 https://pypi.python.org/pypi/Metaphone/0.4

24 http://norvig.com/spell.py

25 http://norvig.com/big.txt é composto por 1 milhão de palavras e apresenta a junção de livros do projeto Gutenberg, bem como as palavras mais comuns do *Wiktionary* e do *British National Corpus*

1. Coleta de amostra tweets da base de dados do laboratório LISA;
2. Aplicação do método proposto aos tweets selecionados;
3. Execução da ferramenta de NERD utilizando os tweets brutos;
4. Execução da ferramenta de NERD utilizando os tweets pré-processados na tarefa 2.

Para verificar a contribuição do método proposto, analisou-se o aumento no número de entidades nomeadas reconhecidas nos tweets que passaram pelo pré-processamento.

Os experimentos deste trabalho foram realizados com um conjunto de 10 mil *tweets* em inglês. A escolha pelo uso de *tweets* no idioma inglês deve-se ao fato de a ferramenta de NERD utilizada para os experimentos não trabalhar com o idioma português, e portanto, descreve-se experimentos na língua portuguesa como trabalhos futuros. Os tweets foram aleatoriamente selecionados de milhões de *tweets* em inglês enviados de um *bounding box* em torno do território brasileiro e coletados em tempo real da API do Twitter entre janeiro de 2015 e outubro de 2016. Estes *tweets* representam uma amostra da base de dados disponível no laboratório LISA. A ferramenta de NERD escolhida para os experimentos é o FOX, o qual é oferecido pela Universidade de Leipzig – Alemanha, via *Web service* baseado em arquitetura REST-full e possui razoável documentação [23]. Para acessar o serviço, foi implementado um código simples em JAVA que envia cada *tweet* e recebe como resposta um objeto JSON contendo as entidades nomeadas encontradas no *tweet*. A seguir são mostrados os resultados obtidos com os experimentos.

5.1 Filtragem dos Dados

A etapa de filtragem dos *tweets* levou 58 minutos e foi executada de forma independente antes das tarefas de *NER* e *NED*. A Tabela 2 mostra as estatísticas obtidas durante a execução. Os *tokens* analisados referem-se ao número de tokens encontrados já excluindo *hashtags*, urls e menções a usuários. Como já mostrado na **etapa 0**, antes da tokenização realiza-se uma limpeza dessas ocorrências. O número de diferentes *stopwords* encontradas é 153, sendo que as mesmas se repetem nos *tweets* totalizando 24031 ocorrências. As palavras relevantes, referem-se aos tokens analisados excluindo as *stopwords*. As palavras não capitalizadas incluem os *tokens* que foram classificados como *in vocabulary* (IV) ou *out-of-vocabulary* (OOV). Lembrando que ao classificar os *tokens* são excluídos os que tiverem capitalização, pois existe a possibilidade de os mesmos serem entidades nomeadas [8] Os *tokens* capitalizados são 30,43% do total de *tokens* encontrados nos *tweets* deste trabalho, valor que está de acordo com o que é explorado no trabalho [8] no qual três diferentes *corpus* de *tweets* apresentam entre 23% e 30% de *tokens* capitalizados. Por fim, as palavras *in vocabulary* (IV) referem-se às palavras (não capitalizadas) encontradas no dicionário, e as palavras *out-of-vocabulary (*OOV*)* indica quantos *tokens* não se encontram no dicionário utilizado, sendo assim candidatos à normalização.

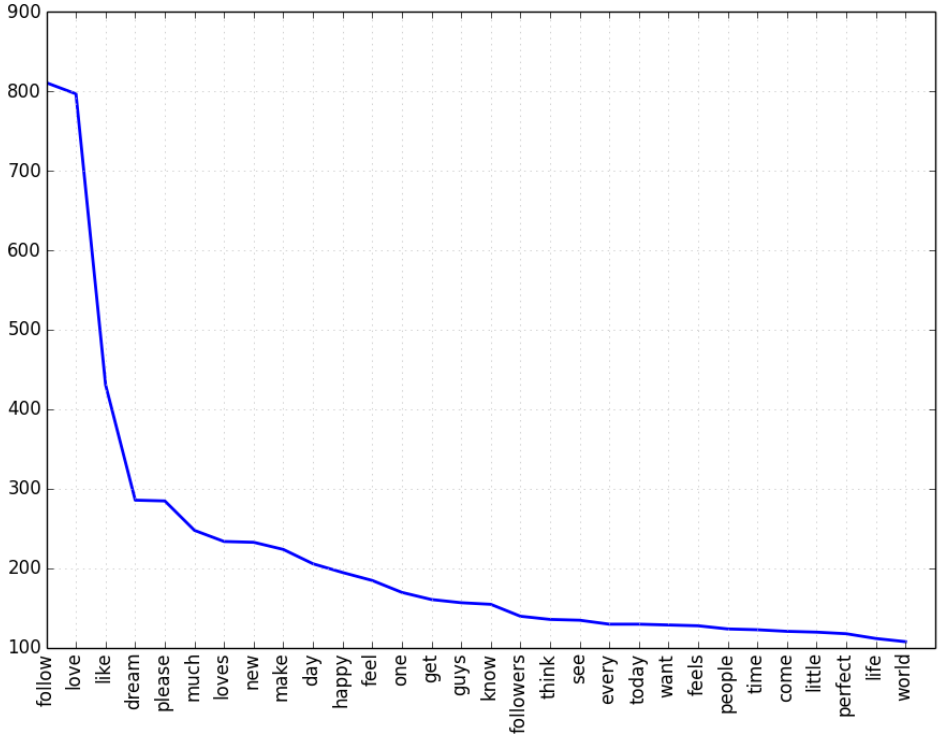
Tabela 2 - Número de palavras encontradas e como se classificam.

Tipo de Ocorrência	**Quantidade encontrada**
Tokens analisados	75293
Stopwords	24031
Palavras relevantes	51262
Palavras não capitalizadas	28348

In vocabulary (IV)	24060
Out-of-vocabulary (OOV)	4288

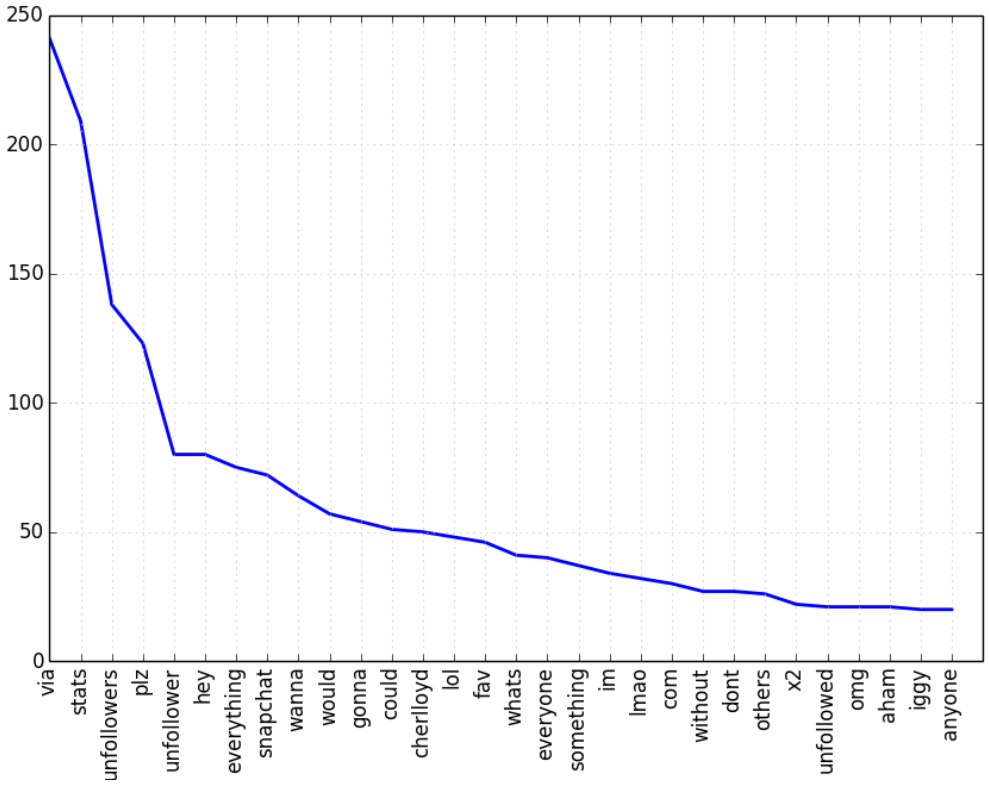
A Figuras 4 e 5 mostram gráficos gerados ao final da execução da filtragem. Elas mostram as 30 palavras *IV* e *OOV*, respectivamente, mais frequentes nos *tweets analisados*. Observa-se que entre essas palavras não há nenhuma que seja uma entidade nomeada, porém elas são importantes para outros tipos de anotações semânticas que não são cobertas nesse trabalho. Um exemplo são as palavras "*love*" e "*like*" que podem ser utilizadas para anotações referentes à análises de sentimento. Muitas dessas palavras são verbos que indicam ações (e.g. seguir - *"follow"*, fazer - *"make"*, pensar - *"think"*) reforçando a necessidade de ligar palavras com seus significados.

Figura 4 - As 30 palavras *IV* mais frequentes nos *tweets* processados.



Quanto às palavras OOV, podemos confirmar que o uso de gírias e abreviações contam como boa parte dos *tokens* encontrados em postagens de mídias sociais. Por exemplo, a quarta palavra mais frequente é "plz" que é utilizada como abreviação de "*please*". Enquanto entre as 30 palavras IV mais frequentes não observou-se nenhuma ocorrência de possível entidade nomeada, nas palavras OOV aparecem duas que são nomes de superfície de entidades, sendo elas: "cherlloyd" e "iggy", ambas referentes a cantoras americanas. Isso mostra a importância de tratar capitalização para o reconhecimento de entidades nomeadas. A falta de capitalização dessas palavras fez com que as mesmas passassem pelo processo de filtragem, o que não deveria ser necessário já que se tratam de nomes.

Figura 5 - As 30 palavras *out-of-vocabulary OOV* mais frequentes nos *tweets* processados.



5.2

5.2. Resultados de Nerd com Filtros e sem Filtros

A primeira execução realizada foi nos *tweets* sem a etapa de filtragem e a segunda foi nos *tweets* já pré-processados. Como pode-se observar pela Tabela 4, o tempo de execução de NERD foi praticamente o mesmo para os *tweets* sem e com filtragem. A seguir é apresentada uma tabela comparativa dos resultados obtidos em cada execução.

Tabela 3 – Tabela comparativa do desempenho do FOX no processamento de *tweets* com e sem filtragem

	Sem filtros	Com filtros
Tempo de execução do FOX	2 horas e 9 min.	2 horas e 12 min.
Número de anotações por classe:		
Localização	669	649
Organização	226	241
Pessoa	770	865
Total de *tweets* processados	10 mil	10 mil

Total de *tweets* anotados	1440	1505
Percentual de *tweets* anotados	14.4%	15.05%
Total de anotações realizadas	1665	1755
Média de anotações por *tweet* processado	0.166	0.175
Média de anotações por *tweet* anotado	1.156	1.166

Sobre o número de entidades anotadas em cada experimento foi observado um aumento de 100 entidades, ou seja, 5% a mais em relação às anotações realizadas no experimento sem utilização de filtros. Apesar de 5% não ser um valor consideravelmente alto, dentre as publicações revisadas neste trabalho encontram-se métodos de filtragem [18] que apresentaram uma melhoria de apenas 3%. Considerando que os dados utilizados são provenientes da *web* e não apresentam nenhuma estrutura a melhoria na tarefa de NERD obtida com o método proposto mostra-se interessante. Porém, ainda é necessário muita pesquisa para que se façam variações do método proposto (e.g. adicionar tratamentos de capitalização das palavras) para que se alcance resultados com boa relevância estatística.

Analisando as entidades nomeadas mais frequentemente encontradas em ambas as execuções da ferramenta de NERD, pode-se observar diferenças nas entidades encontradas, bem como alguns problemas a serem tratados. Um dos caso que mostra a necessidade de melhorias no método é o aparecimento de "RT" como uma entidade frequente após a filtragem dos *tweets*, já que a sigla RT significa *retweet* e é muito utilizada nos *tweets* analisados, porém foi identificada pela ferramenta de NERD como sendo uma Organização, o canal de televisão "*Russia Today"*. Além disso outro caso interessante são as entidades "Justin Bieber" e "Justin" que aparecem entre as 30 entidades encontradas mais frequentes. Ao realizar uma análise manual dos *tweets* foi possível observar que a maioria das menções a Justin referem-se a entidade Justin Bieber da classe Pessoa. Entretanto o FOX reconheceu as menções a Justin com sendo a uma entidade diferente de Justin Bieber. Isso mostra a dificuldade da extração e desambiguação de entidades nomeadas em *tweets*, bem como reforça a necessidade de uma regra ouro para NERD em que mídias sociais.

6. Conclusão e Trabalhos Futuros

Este trabalho apresentou uma fundamentação teórica trazendo os principais conceitos necessários para o entendimento das tarefas de reconhecimento (NER) e de desambiguação (NED) de entidades nomeadas. Em seguida, foram brevemente apresentados trabalhos relacionados sobre NERD em *tweets*, dando foco a trabalhos que fizeram uso de filtragem dos dados antes da realização das tarefas de NER e NED. Com base nos trabalhos estudados, este artigo propôs e implementou um processo para filtragem dos *tweets* antes da realização da tarefa de NERD. A proposta apresentada é baseada no trabalho [2] e consiste de 5 etapas, sendo elas: 0 - limpeza de ruídos (e.g. *hashtags, usernames* e urls) e separação em IV e OOV; 1 – cálculo de distância morfológica; 2 – cálculo de distância fonética; 3 - aplicação do algoritmo de Peter Norvig; 4 – comparação dos resultados de cada etapa e escolha da melhor candidata como resultado da normalização; 5 – substituição no arquivo de *tweets* das OOVs pelas suas respectivas correções.

Os experimentos foram realizados com o objetivo de verificar o aumento no número de entidades nomeadas reconhecidas após a filtragem dos dados. Dessa forma, foi feita a filtragem dos *tweets* pelo método proposto e então realizou-se duas execuções da ferramenta FOX para medir a melhoria obtida na tarefa de NERD. Como resultados dos experimentos observou-se que o tempo de execução foi praticamente o mesmo nos dois conjuntos de *tweets*. Entretanto, quanto ao aumento de entidades nomeadas reconhecidas foi possível observar uma melhoria de 5% no experimento com *tweets* pré-processados utilizando o método proposto. Por ser tratar de análise de conteúdo não estruturado proveniente da *web* e com base em trabalhos atuais presentes na literatura [18] que mostram melhorias de apenas 3%, o resultado obtido neste trabalho mostra que o método proposto pode auxiliar na tarefa de NERD. Porém, o método ainda precisa ser melhorado não só para aumentar esse percentual de entidades reconhecidas, como também para evitar que palavras que não são menções a entidades nomeadas sejam identificas como tal.

Este trabalho apresentou um primeiro esforço do grupo de pesquisa do laboratório LISA no uso de filtros para pré-processar dados de mídias sociais. Todavia, muito ainda há para se desenvolver em tal área de pesquisa. Entre os possíveis trabalhos futuros pode-se citar:

- Implementar o método de filtragem proposto utilizando dicionários da língua portuguesa;
- Adicionar na última etapa da filtragem dos dados o uso de contexto como proposto por Ahmed (2015) para assim melhorar a escolha da palavra normalizada;
- Adicionar tratamentos para capitalização de palavras;
- Testar outras abordagens para filtragem de dados visando melhorias dos resultados de *NER* e *NED*;
- Incluir no FOX o idioma Português para realizar *NER* e *NED* nos *tweets* da base de dados do laboratório LISA, cuja maioria está em língua portuguesa;
- Instalar localmente a ferramenta FOX e incorporar a etapa de filtragem de dados no FOX visando melhoria em termos de tempo de execução;
- Obter conjuntos de dados com regra ouro e realizar experimentos para medir ganhos de precisão e cobertura.

References

[1] AGGARWAL, C.ZHAI, C. Mining text data. New York: Springer, 2012.

[2] AHMED, BILAL. Lexical Normalisation of Twitter Data. CoRR, v. abs/1409.4614, 2015. Disponível em: <http://arxiv.org/abs/1409.4614>.

[3] AMITAVA, Das et al. NER from Tweets: SRI-JU System @MSM 2013. In: MAKING SENSE OF MICROPOSTS #MSM2013, 3., 2013, Rio de Janeiro. Proceedings of the Concept Extraction Challenge. Rio de Janeiro: Ceur Workshop Proceedings, 2013. v. 1019, p. 62 - 66. Disponível em: <http://ceur- ws.org/Vol- 1019/paper_33.pdf>. Acesso em: 28 nov. 2016.

[4] BONTCHEVA, Kalina.; ROUT, Dominic. Making sense of social media streams through semantics: A survey. Semantic Web, v. 5, n. 5, p. 373-403, 2014.

[5] CARTER, Simon; WEERKAMP, Wouter; TSAGKIAS, Manos. Microblog language identification: overcoming the limitations of short, unedited and idiomatic text.

Language Resources And Evaluation. [s. L], p. 195-215. mar. 2013. Disponível em: <http://dx.doi.org/10.1007/s10579-012-9195-y>. Acesso em: 28 nov. 2016.

[6] CHINCHOR, N. Overview of muc-7. In: Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998. [s.n.], 1998. Disponível em: <http://www.aclweb.org/anthology/M98-1001>.

[7] DAS, T.; ACHARJYA, D.; PATRA, M. Opinion mining about a product by analyzing public tweets in Twitter. 2014 International Conference on Computer Communication and Informatics, p. 1-4, 2014.

[8] DERCZYNSKI, L. et al. Analysis of named entity recognition and linking for tweets. Information Processing & Management, Elsevier, v. 51, n. 2, p. 32–49, 2015.

[9] EARLE, Paul S.; BOWDEN, Daniel C.; GUY, Michelle. Twitter earthquake detection: earthquake monitoring in a social world. Annals of Geophysics, [S.l.], v. 54, n. 6, jan. 2012. ISSN 2037-416X. Disponível em: <http://www.annalsofgeophysics.eu/index.php/annals/article/view/5364>

[10] GRISHMAN, R.; SUNDHEIM, B. Message understanding conference-6: A brief history. In: COLING. [S.l.: s.n.], 1996. v. 96, p. 466–471.

[11] HAN, Bo; BALDWIN, Timothy. Lexical Normalisation of Short Text Messages: Makn Sens a \#Twitter. In: HLT '11, 49., 2011, Portland. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg: Association For Computational Linguistics, 2011. v. 1, p. 368 - 378. Disponível em: <http://dl.acm.org/citation.cfm?id=2002472.2002520>. Acesso em: 28 nov. 2016.

[12] HAN, Bo; COOK, Paul; BALDWIN, Timothy. Lexical Normalization for Social Media Text. ACM Transactions On Intelligent Systems And Technology. New York, p. 1- 27. jan. 2013. Disponível em: <http://doi.acm.org/10.1145/2414425.2414430>. Acesso em: 28 nov. 2016.

[13] IBRAHIM, Y.; YOSEF, M.; WEIKUM, G. AIDA-Social : Entity Linking on the Social Stream. Proceedings of the 7th International Workshop on Exploiting Semantic Annotations in Information Retrieval - ESAIR '14, p. 17-19, 2014.

[14] KLEIN, Douglas. Reconhecimento e Desambiguação de Entidades nomeadas com foco em Mídias Sociais. 2015. 72 f. TCC (Graduação) - Curso de Sistemas de Informação, Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis, 2015.

[15] KONKOL, M. Named entity recognition. 2012. Disponível em: <http://www.kiv.zcu.cz/site/documents/verejne/vyzkum/publikace/technicke-zpravy/2012/tr-2012-04.pdf>.

[16] MARRERO, M. et al. Named Entity Recognition: Fallacies, challenges and opportunities. Computer Standards & Interfaces, v. 35, n. 5, p. 482-489, 2013.

[17] NADEAU, D.; SEKINE, S. A survey of named entity recognition and classification. Lingvisticae Investigationes, John Benjamins publishing company, v. 30, n. 1, p. 3– 26, 2007.

[18] OLIVEIRA, D. M. de et al. Fs-ner: A lightweight filter-stream approach to named entity recognition on twitter data. In: Proceedings of the 22Nd International Conference on World Wide Web. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2013. (WWW '13 Companion), p. 597–604. ISBN 978-1-4503-2038-2. Disponível em: <http://dl.acm.org/citation.cfm?id=2487788.2488003>.

[19] PAZZANI, M.BILLSUS, D. Content-Based Recommendation Systems. The Adaptive Web, p. 325-341, 2007.

[20] RAO, Delip; MCNAMEE, Paul; DREDZE, Mark. Entity Linking: Finding Extracted Entities in a Knowledge Base. In: POIBEAU, Thierry et al (Ed.). Multi-source, Multilingual Information Extraction and Summarization. Heidelberg: Springer Berlin Heidelberg, 2013. Cap. 2. p. 93-115. (Theory and Applications of Natural Language Processing). Disponível em: <http://dx.doi.org/10.1007/978-3-642-28569-1_5>. Acesso em: 30 nov. 2016.

[21] RITTER, A. et al. Named entity recognition in tweets: An experimental study. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011. (EMNLP '11), p. 1524–1534. ISBN 978-1-937284-11-4. Disponível em: <http://dl.acm.org/citation.cfm?id=2145432.2145595>.

[22] SHEN, W.; WANG, J.; HAN, J. Entity linking with a knowledge base: Issues, techniques, and solutions. Knowledge and Data Engineering, IEEE, v. 27, p. 443–460, 2014.

[23] SPECK, R.; NGONGA NGOMO, A.-C. Ensemble learning for named entity recognition. In: The Semantic Web – ISWC 2014. Springer International Publishing, 2014, (Lecture Notes in Computer Science, v. 8796). p. 519–534. Disponível em: <http://svn.aksw.org/papers/2014/ISWCEL4NER/public.pdf>.

[24] ZHAI, C.MASSUNG, S. Text data management and analysis: A Practical Introduction to Information Retrieval and Text Mining. Association for Computing Machinery and Morgan, 2016.