Dissertação de Mestrado

SSP: A Linguistic Pattern Mining Approach for Discourse Analysis and Information Extraction in Short Texts using Word Embeddings

Danielly Sorato

Universidade Federal de Santa Catarina
Programa de Pós-Graduação em
Ciência da Computação

Danielly Sorato

SSP: A Linguistic Pattern Mining Approach for Discourse Analysis and Information Extraction in Short Texts using Word Embeddings

Florianópolis
2019

Danielly Sorato

SSP:
A Linguistic Pattern Mining Approach for Discourse Analysis and Information Extraction in Short Texts using Word Embeddings

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina para a obtenção do Grau de Mestre em Ciência da Computação.
Orientador: Prof. Dr. Renato Fileto

Florianópolis
2019

Ficha gerada automaticamente pelo LaTeX.

Sorato, Danielly
SSP : A Linguistic Pattern Mining Approach for Discourse Analysis and Information Extraction in Short Texts using Word Embeddings / Danielly Sorato; orientador, Renato Fileto, 2019.
91 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Ciência da Computação, Florianópolis, 2019.

Inclui referências

1. Ciência da Computação. 2. Processamento de Linguagem Natural. 3. Extração de Informação. 4. Análise de Discurso. 5. *Word Embeddings*. 6. Mídias Sociais. I. Fileto, Renato. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Ciência da Computação. III. SSP A Linguistic Pattern Mining Approach for Discourse Analysis and Information Extraction in Short Texts using Word Embeddings.

Danielly Sorato

SSP: A Linguistic Pattern Mining Approach for Discourse Analysis and Information Extraction in Short Texts using Word Embeddings

Esta Dissertação foi julgada adequada para obtenção do Título de "Mestre em Ciência da Computação", e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Florianópolis, 24 de Abril de 2019.

José Luís A. Güntzel
Coordenador do Curso

Banca Examinadora:

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

Prof. Dr. Thiago Alexandre Salgueiro Pardo
Universidade de São Paulo
(participação por videoconferência)

Profª. Drª. Silvia Modesto Nassar
Universidade Federal de Santa Catarina

Prof. Dr. Mauro Roisenberg
Universidade Federal de Santa Catarina

All my affection and gratitude to the family, friends, colleagues and all others who accompanied me during this step of my journey, for even the very wise cannot see all ends. In particular, thank you Alexandre de Limas Santana, you are the star that guides my way in dark times.

ACKNOWLEDGEMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

“You shall know a word by the company it keeps.
(Firth, J. R. 1957) ”

RESUMO

Postagens em microblogs, tais como *tweets*, frequentemente contêm opiniões e pensamentos de usuários sobre eventos, produtos, pessoas, entre outras possibilidades. Contudo, o uso de mídias sociais para propagar discursos de ódio, promover desinformação e manipular opiniões não são ocorrências incomuns. A análise de postagens problemáticas é crucial para entender, combater e desencorajar tais ações. Repetições de expressão, i.e. padrões de discurso, ocorrem na linguagem natural. Extrair fragmentos de texto com semântica recorrente podem levar à descoberta de padrões linguísticos usados em certos tipos de discurso textualmente expressos em postagens de microblogs. Nessa dissertação, esses padrões são usados no contexto de extração de informação, análise de discurso e classificação de texto. Através da abordagem aqui desenvolvida, chamada mineração de Padrões Semânticos Curtos (em inglês *Short Semantic Patterns* - SSP), é possível descobrir dinamicamente, bem como extrair, sequências de palavras que compartilham significado similar em relação à sua representação vetorial. O uso de vetores de palavras (*word embeddings*) permite a extração eficiente de padrões flexíveis, que não estão restritos à similaridade e ordem lexical. Primeiramente, os SSP são formalmente descritos e sua incidência é mostrada em *tweets* reais. Depois, a abordagem de mineração é aplicada para executar tarefas de Extração de Informação e Análise de Discurso em dois estudos de caso distintos, especificamente *tweets* da campanha presidencial de Donald Trump e de discurso de ódio. Por fim, os SSP extraídos no caso de discurso de ódio são usados como *features* para construir classificadores para detectar se um *tweet* contém discurso de ódio (classificação binária) e também para distinguir entre *tweets* contendo racismo, sexismo, ou conteúdo normal (classificação ternária). A análise das instâncias de SSP em relação aos *tweets* de Donald Trump evidenciaram que sua estratégia de campanha consistia em sistematicamente difamar a mídia e seus oponentes. As instâncias de SSP encontradas nos *tweets* contendo sexismo mostraram que um grande número de *tweets* sexistas com a introdução *'I'm not sexist but'* e *' Call me sexist but'*. Enquanto isso, instâncias do SSP encontradas em *tweets* sobre racismo revelaram uma proeminência de discursos contra a religião islâmica, entidades e organizações associadas.

Palavras-chave: Processamento de Linguagem Natural. Extração de Informação. Análise de Discurso. *Word Embeddings*. Mídias Sociais.

RESUMO EXPANDIDO

Introdução

Nos últimos anos, as redes sociais tornaram-se um dos tipos mais proeminentes de *websites* na Internet. Consequentemente, a quantidade de dados gerados por usuários na Web teve um crescimento notável, caracterizando um exemplo importante de *big data* (WATANABE; BOUAZIZI; OHTSUKI, 2018). Nas postagens de mídias sociais, os usuários expressam suas opiniões e pensamentos sobre pessoas, produtos, eventos, etc. Portanto, essas postagens são fontes ricas de informações para realizar análise de dados e extração de informação. Anotação semântica e análise automática de texto possuem várias aplicações computacionais. A partir de análises textuais, pode-se atacar problemas importantes da era da informação, tais como detecção de discurso de ódio (DJURIC et al., 2015; DAVIDSON et al., 2017; WASEEM; HOVY, 2016), detecção de notícias falsas (CONROY; RUBIN; Y. CHEN, 2015; MONTEIRO et al., 2018), entre outros aspectos de análise do discurso.

Certos padrões linguísticos ocorrem na linguagem humana (MONDAL; SILVA; BENEVENUTO, 2017; BÉCHET et al., 2012; SCHWARTZ; REICHART; RAPPOPORT, 2015). Este trabalho visa extrair dinamicamente, ou seja, sem estruturas sintáticas pré-definidas, padrões que podem retratar pensamentos frequentes expressos por usuários falando sobre um determinado assunto. Esses padrões podem ser usados, por exemplo, para minerar os discursos e opiniões das pessoas sobre produtos, eventos, organizações, entre outros. Portanto, propõe-se a exploração de tais padrões para detectar fragmentos de texto semanticamente semelhantes que expressam certos tipos de discurso, como linguagem abusiva e discurso de ódio. Nesse contexto, a repetição de certos discursos pode revelar pensamentos e conceitos comuns espalhados pelos usuários. Devido às comparações de similaridade flexíveis e eficientes fornecidas pelos vetores de palavras, pretendemos aplicar essa técnica para detectar sequências de palavras que possuem significados semelhantes. Como tal, o método proposto para minerar esses padrões, chamado Padrões Semânticos Curtos (em inglês, *Short Semantic Patterns* - SSP), é capaz de detectar fragmentos de texto que podem ter componentes lexicais distintos, mas exibem sentidos semelhantes. Nos experimentos, os SSPs são utilizados para investigar expressões e conceitos frequentemente associados a alvos de discurso de ódio, bem como discursos recorrentes de Donald Trump sobre a mídia e seus adversários. Posteriormente, classificadores foram implementados para examinar o impacto do uso de SSPs como *features*.

As principais contribuições deste trabalho são: (i) definição formal de SSPs; (ii) algoritmo para minerar SSPs em documentos de texto curto, como

postagens em microblogs; (iii) análises de discurso baseadas na mineração de instâncias SSP relacionadas a palavras-chaves (alvos) fornecidas e na inspeção de tais padrões para verificar os pensamentos e significados usualmente associados a tais alvos nas instâncias SSP mineradas; (iv) um classificador binário para detectar o discurso de ódio em tweets e; (v) um classificador ternário que distingue entre tweets sexistas, racistas e sem conteúdo problemático. Nos experimentos, foram investigadas instâncias de SSPs referentes ao sexismo e ao racismo encontradas nos *tweets* sobre discurso de ódio e das instâncias de SSPs relacionadas aos discursos de Donald Trump.

Objetivos

O objetivo principal deste trabalho é conceber e desenvolver uma abordagem semi-automática para extrair informações de textos curtos, através da mineração de Short Semantic Patterns (SSP). Dado o objetivo principal deste trabalho, emergem os seguintes objetivos específicos:

1. Compreender o estado da arte sobre o reconhecimento de padrões textuais através de uma revisão sistemática da literatura;

2. Definir SSP, instâncias SSP e investigar a incidência de tais padrões em conjuntos de dados reais;

3. Implementar um método para mineração e agrupamento de instância do SSP, portanto, caracterizando padrões;

4. Aplicar a mineração SSP no contexto de extração de informações, análise de discurso e classificação de texto.

Nossa abordagem oferece certas vantagens, especialmente ao lidar com textos de microblog. Devido à utilização da representação vetorial de palavras, os SSPs não são restringidos por similaridade ou ordem lexical. Além disso, é possível treinar vetores de palavras com conjuntos de dados de *tweets*, por exemplo, adquirindo assim um vocabulário mais adequado a algum domínio em particular.

Método

O método de extração de SSP baseia-se na representação de cada palavra em uma frase como um vetor de palavras produzido por algum modelo neural de *word embedding*. Estes vetores podem então ser iterativamente agregados por alguma função (por exemplo, soma, média) e filtrados para formar uma estrutura de tamanho arbitrário. Um SSP refere-se a sequências de palavras em textos que podem ser distintas, por exemplo em termos de componentes

lexicais e tamanho, mas semanticamente semelhantes entre si, correspondendo a um significado ou pensamento frequente. Cada uma dessas sequências de palavras deve ser encontrada em documentos distintos, por exemplo duas ou mais postagens de mídias sociais distintas, e é chamada de instância do respectivo SSP. Embora documentos distintos possam representar pensamentos e opiniões usando diversas combinações de palavras e expressões, essas são apenas maneiras variadas de fazer declarações semelhantes.

A mineração de SSP tem como objetivo localizar instâncias de SSP em uma coleção de textos curtos, como *tweets*. Para obter instâncias de SSP, é necessário localizar sequências de palavras semanticamente semelhantes em documentos distintos. Como tal, o processo de mineração depende da expansão de janelas de contexto que englobam sequências de palavras, partindo de uma palavra-chave como ponto central. A expansão de janelas de contexto é utilizada para encontrar a sequência mais longa no entorno de cada palavra-chave que aparece em um texto, preservando similaridade semântica em relação ao conteúdo de outra janela de contexto (referente a um documento distinto). Neste trabalho, propõe-se o uso de palavras de domínio como palavras-chave. Portanto, um dado documento que contiver ao menos uma palavras-chave do conjunto referente às de palavras de domínio, será marcado para mineração de padrão.

Cada janela de contexto é expandida gradualmente de seu centro (ocorrência de uma palavra-chave dada como entrada) para localizar a sequência de palavras em seu respectivo documento que ainda é semelhante a outra sequência de palavras. A janela de contexto é expandida em ambas as direções enquanto os limites do documento são testados. Se a janela de contexto atingir o início ou o final da frase, a janela de contexto não será mais expandida nessa direção. Durante o processo de mineração, os limites do entorno à direita e à esquerda de cada palavra-chave são dinamicamente aumentados, enquanto a semelhança semântica entre o conteúdo de duas janelas de contexto é mantida acima de um limiar. Quando as janelas de contexto não puderem ser mais expandidas, os *embeddings* das palavras contidas em seus limites são agregados e armazenados como duas instâncias SSP distintas. Posteriormente, outros documentos passam pelo mesmo processo, a fim de encontrar mais instâncias. Um SSP é caracterizado por instâncias correlatas, i.e. sequências de palavras expandidas a partir de certas palavras-chaves mantendo similaridade semântica dentro de certo limiar, nos documentos analisados.

Experimentos, Resultados e Discussão

Para os experimentos em relação ao discurso de Trump, utilizou-se um conjunto

de dados contendo 3,219 *tweets*[1] postados ou retweetados por Donald Trump durante a campanha para a eleição presidencial de 2016 nos Estados Unidos. Três conjuntos de dados distintos foram utilizados para os experimentos de detecção de discurso de ódio, nomeadamente Waseem et al. (WASEEM; HOVY, 2016)[2], SemEval 2019 Task 5 (hatEval) (BASILE et al., 2019) e Davidson et al. (DAVIDSON et al., 2017)[3].

Para os experimentos de detecção de discurso de ódio, em adição de um modelo de *embeddings* pré-treinado, treinou-se um modelo FastText customizado usando os conjuntos de dados de Davidson et al. e hatEval. A tarefa de mineração e classificação de SSPs foi realizada usando apenas o conjunto de dados de Waseem et al. Para ambos os estudos de caso, primeiramente os *tweets* foram pré-processados, com o intuito de remover *stop words*, pontuação, emojis, url e menções de nome de usuário. Em seguida, os textos dos *tweets* foram tokenizados e lematizados.

A análise das instâncias de SSP em relação aos *tweets* de Donald Trump mostraram que sua estratégia de campanha consistia em sistematicamente difamar a mídia e seus oponentes. Nos discursos propagados via *tweets*, ele tenta retratar a mídia e seus adversários como incompetentes e não confiáveis. Assim, ele surge como a única solução viável e fonte de informação correta. A análise das instâncias de SSP encontradas nos tweets referentes ao sexismo revelou que um grande número de tweets sexistas começava com a introdução *'I'm not sexist but'* e *' Call me sexist but'*. Enquanto isso, instâncias do SSP encontradas em *tweets* sobre racismo revelaram uma proeminência de discursos contra a religião islâmica, entidades e organizações associadas. A técnica de mineração de instâncias de SSP mostrou-se útil para extrair discursos recorrentes (em termos de similaridade semântica) que apareciam no conjunto de dados sem ter que revisá-lo manualmente.

A precisão e a cobertura do melhor desempenho na classificação binária foram 79,1% e 80,2%, respectivamente. O modelo implementado superou a abordagem de Waseem et al. (WASEEM; HOVY, 2016) na sua melhor configuração (*n*-gramas de caracteres + gênero: precisão 72,9%, cobertura 77,7%). A precisão e a cobertura do melhor desempenho na classificação ternária foram 76,7% e 77,8%, respectivamente. A maior parte das classificações incorretas correspondem a instâncias *tweets* contendo discurso de ódio que foram categorizadas como conteúdo não problemático. Em especial, instâncias de *tweets* contendo formas veladas de sexismo e sarcasmo foram muitas vezes erroneamente previstas como não problemáticas. *Tweets* contendo xingamentos explícitos e fortes tendem a ser precisamente

[1] https://www.kaggle.com/benhamner/clinton-trump-tweets/home
[2] https://github.com/ZeerakW/hatespeech
[3] https://github.com/t-davidson/hate-speech-and-offensive-language

classificados como discurso de ódio.

Considerações Finais

Nesta dissertação o conceito de Padrões Semânticos Curtos (SSP) foi apresentado, bem como uma abordagem para a mineração de SSP e a análise da aplicação de tais padrões foi realizada em dois estudos de caso distintos, com *tweets* reais. Características (*features*) extraídas de SSP minerados também foram utilizadas para classificação binária e ternária de documentos de texto, usando classificadores distintos. Apesar dos ganhos marginais de desempenho obtidos usando os SSP em classificadores, foram notadas melhorias consistentes. Portanto, pretende-se investigar se a expansão semântica visando aumentar o número e a cobertura das palavras-chave levaria à descoberta de mais instâncias de SSP, para aumentar a relevância estatística dos padrões. Um aumento no número de SSP poderia causar mais impacto nos resultados do classificador implementado.

A abordagem aqui desenvolvida oferece certas vantagens, especialmente ao lidar com textos de microblog. Ao utilizar a representação vetorial de palavras (*word embeddings*), tais padrões não estão ligados à semelhança lexical. Assim, os SSP não precisam conter a mesma ordenação de palavras, nem as mesmas palavras, mas focam na similaridade semântica. Além disso, existe a possibilidade de treinar vetores de palavras com conjuntos de dados de *tweets*, assim adquirindo um vocabulário mais apropriado ao domínio desejado. Ao fazer isso, também pode-se capturar o contexto (palavras adjacentes) das palavras fora do vocabulário, como neologismos e gírias, em um determinado conjunto de dados. Esses fatores tornam os SSP menos restritivos e mais adequados para aplicação em textos de microblog.

O método proposto ainda sofre certas limitações. Existe uma restrição de granularidade, uma vez que os documentos são marcados para mineração através de palavras-chave predefinidas. O Além disso, ainda não há um método para validar automaticamente os padrões. No entanto, um especialista humano pode facilmente validar os padrões minerados como padrões linguísticos relevantes. Também se faz necessário automatizar a atribuição de valores para os parâmetros de limiar de semelhança semântica e suporte mínimo. Ademais, é necessário avaliar se o tratamento de palavras compostas (por exemplo, São Paulo) causaria um impacto benéfico na mineração de instâncias

Palavras-chave: Processamento de Linguagem Natural. Extração de Informação. Análise de Discurso. *Word Embeddings*. Mídias Sociais.

ABSTRACT

Microblog posts such as tweets frequently contain users opinions and thoughts about events, products, people, among other possibilities. However, the usage of social media to propagate hate speech, promote online disinformation and manipulation is not an uncommon occurrence. Analyzing such problematic social media posts is essential for understanding, fighting, and discouraging such actions. Repetition of discourses, i.e. speech patterns, occur in natural language. Extracting recurrent fragments of text which are semantically similar can lead to the discovery of linguistic patterns used in certain kinds of discourse. Therefore, we aim to use these patterns to encapsulate frequent discourses textually expressed in microblog posts. In this dissertation, we propose to exploit such linguistic patterns in the context of Information Extraction and Discourse Analysis. Though the technique developed in this work, called SSP (Short Semantic Pattern) mining, we are able to dynamically discover and extract sequences of words that share a similar thought in their word embedding representation. The use of word embeddings allows the efficient extraction of flexible patterns, which are not restricted to lexical and syntactic similarity. First, we formally describe our SSPs and show its incidence in real tweets. Then, we apply our technique to perform Information Extraction and Discourse Analysis in two case studies, namely Donald Trump's presidential campaign and hate speech tweets. Afterwards, we experiment using SSPs as features to build classifiers to detect if a tweet contains hate speech (binary classification) and to distinguish between sexism, racism and clean tweets (ternary classification). The analysis of SSP instances regarding Donald Trump's tweets showed that his campaign strategy consisted in systematically defaming the media and his opponents. The SSP instances encountered in tweets containing sexism have shown that a large number of sexist tweets with the introduction *'I'm not sexist but'* and *'Call me sexist but'*. Meanwhile, SSP instances found in tweets depicting racism revealed a prominence of discourses against the Islamic religion, associated entities and organizations.

Keywords: Natural Language Processing. Information Extraction. Discourse Analysis. Word Embeddings. Social Media.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

LIST OF ALGORITHMS

CONTENTS

1 INTRODUCTION

During the last years, social networks became one of the most prominent types of websites on the Internet. Consequently, the amount of user-generated data on the Web had an outstanding growth, comprising an appealing example of big data (WATANABE; BOUAZIZI; OHTSUKI, 2018). In social media posts, users express their opinions and thoughts about people, products, events, etc. Therefore, such postings are rich sources of information to perform Data Analysis (DA) and Information Extraction (IE). Such possibility awakened the interest of both industry and academia. Consequently a myriad of techniques and tasks for computational linguistics was conceived and developed.

Semantic annotation and automatic text analysis have several applications. Through textual analyzes one can attack important problems of the information age, such as hate speech recognition (DJURIC et al., 2015; DAVIDSON et al., 2017; WASEEM; HOVY, 2016), fake news detection (CONROY; RUBIN; Y. CHEN, 2015; MONTEIRO et al., 2018) and other aspects of discourse analysis. Semantically annotated texts can be used to feed large volumes of data with well-defined semantics to improve the results of recommendation systems (MEEHAN et al., 2013; LU; LAM; ZHANG, 2012), question answering (Y. LIU; ALEXANDROVA; NAKAJIMA, 2013), sentiment analysis (KHAN; ATIQUE; THAKARE, 2015; H. WANG et al., 2012), opinion mining (SIDOROV et al., 2012; O'CONNOR et al., 2010), etc.

Recently, the usage of neural models that generate dense word vectors has become popular, providing efficient learning of suitable numerical vectors to represent and handle words based on their contexts (neighboring words). Such word embedding models are especially attractive due to the ability of capturing relevant syntactic and semantic information from extensive volumes of data (IACOBACCI; PILEHVAR; NAVIGLI, 2016) and even dealing with out-of-vocabulary words (JOULIN et al., 2016). The usage of word embeddings contributed positively to the performance of many Natural Language Processing (NLP) tasks, such as Word Sense Disambiguation (ROTHE; SCHÜTZE, 2015; IACOBACCI; PILEHVAR; NAVIGLI, 2015), Sentiment Analysis (TRASK; MICHALAK; J. LIU, 2015; L.-C. YU et al., 2017; DRAGONI; PETRUCCI, 2017; TANG et al., 2016), Question Answering (BORDES; CHOPRA; WESTON, 2014; XU; SAENKO, 2016; ZHOU et al., 2015), Discourse Analysis (LEI et al., 2017; BRAUD; DENIS, 2015; J. CHEN et al., 2016; C. WU et al., 2017) and Machine Translation (ZOU et al., 2013; BAHDANAU; CHO; BENGIO, 2014; LAMPLE et al., 2018), among others.

Certain linguistic patterns occur in the human language. There are several types of patterns that can be extracted from texts, varying greatly in purpose (MONDAL; SILVA; BENEVENUTO, 2017; BÉCHET et al.,

2012; SCHWARTZ; REICHART; RAPPOPORT, 2015). In this work we are concerned in extracting dynamically, i.e. without predefined syntactical structures, patterns that might depict frequent thoughts expressed by users talking about a given subject. These patterns could be used, for instance, to mine people's discourses and opinions about products, events, organizations, among others. We propose to exploit such patterns to detect semantically similar text fragments that express certain kinds of discourse, such as abusive language and hate speech. In this context, the repetition of certain discourses may reveal common ideas and concepts spread by the users. Due to the flexible and efficient similarity comparisons provided by word embeddings, we intend to apply this technique to detect sequences of words that carry similar meanings. As such, our proposed method to mine these patterns, called Short Semantic Patterns (SSP), is capable of detecting fragments that may have distinct lexical components but display similar senses. In the experiments, one case study analyzes recurrent discourses of Donald Trump about the media and his adversaries. Other case study analyzes the discourses found on tweets towards certain minorities. We use our SSPs to investigate expressions aimed at hate speech targets and concepts frequently associated with these targets in the view of hate speakers. Later, we build classifiers and examine the impact of using SSPs as features. By doing so, we intend to discover if such patterns may be helpful to detect whether a tweet contains hateful content or not (binary classification) and to distinguish between sexism, racism and clean contents (ternary classification).

The main contributions of this work are: (i) an algorithm to mine SSPs in short text documents such as posts on microblogs; (ii) a comprehensive analysis of SSP instances related to sexism and racism found in tweets and of SSP instances related to Donald Trump's discourses; (iii) a binary classifier to detect hate speech on tweets and; (iv) a ternary classifier that distinguishes between sexist, racist and clean tweets. The extracted patterns can be explored and used in a myriad of NLP tasks, such as discourse analysis, word sense induction and disambiguation, text classification, etc. In our experiments, we utilize SSP in the context of IE, discourse analysis and text classification, namely hate speech recognition.

1.1 PROBLEM DEFINITION

Microblogs posts are rich sources of information. Users often write in their posts their opinions, visions and emotions about several concepts and entities, which may be people, music, products, etc. Therefore, the exploration of such posts is very valuable to several NLP tasks, such opinion mining, question answering, and text classification, among others.

Although the state of the art advanced greatly with remarkably powerful language models and optimization of tasks, results of automatic text analysis are quite behind human performance. Linguistic phenomena such as polysemy, figures of speech, and implicit semantics hinder the transformation of language constructs into a formal structure that enables straightforward linguistic analysis through NLP methods. In fact, natural languages can be complex and dynamic. New slangs, regionalisms, and neologisms can be incorporated into colloquial vocabularies very fast, specially in social media, where the linguistic idiosyncrasy plays an important role. The informal style of writing predominant in social media inflicts additional difficulties for NLP methods, such as slangs, misspellings, and disregard to formal grammar, which combined with the scarce context information of microtexts challenges the limits of NLP.

In this dissertation, we explore patterns that we call Short Semantic Patterns (SSP), which are characterized by sequences of words with a intended meaning that appear repeatedly in the textual contents of such postings. These patterns depict a vision, opinion or though of a user or a group a user's talking about a given subject. Although distinct documents can pose thoughts by using diverse words and expressions, an example of SSP is displayed in Figure 1. The fragments featured by grey boxes in Figure 1, extracted from Donald Trump tweets, have similar meanings.

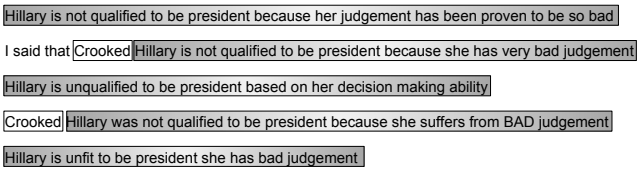

Figure 1: Examples of SSP instances in Donald Trump's tweets concerning Hillary Clinton.

However, detecting an SSP is not an easy task. Even amongst humans, detecting an SSP instance can generate disagreements. For instance, the word *'crooked'*, which appeared in the second and the fourth instances, should be incorporated in the pattern too, since it appeared in more than one instance? Would a human classify the tweets from Waseen et al. (WASEEM; HOVY, 2016) **"I'm not sexist but women do not belong espn analyst"** and **"I'm not sexist but hate female sport analysts"** as instances of the same SSP? The meaning similarity can be complicated, for instance, it is necessary world knowledge to know that *"espn"* is a sports channel.

By using word embeddings and other NLP techniques we try to develop

an approach to the dynamically discover such patterns, i.e., recognize SSP without the help of predesigned structures. The use of semantic similarity via word embeddings to identify the contents of SSP instances is an relaxation for the problems mentioned above. That means that this approach does not try to solve such problems, which remains open due to the need further studies and elaboration to tackle them.

1.2 RESEARCH HYPOTHESIS

Methods for Information Extraction and Discourse Analysis in formal texts (e.g, news, literature) have been extensively studied and refined. However, texts that present more complicated semantics, such as those in social media posts, are very challenging. Given the shortcomings in the state-of-the-art methods to capture difficult semantics and the difficulty of dealing with extensive volumes of unstructured data, the following research hypotheses arise:

1. There is an incidence of SSP in microblog posts;

2. SSP instances occurring in microblogs can help the semantic analysis of the discourse that they carry and can be used to aid classification tasks, through textual style analysis.

1.3 OBJECTIVES

The main objective of this work is to conceive and develop a semi-automatic approach to extract information from short texts, through Short Semantic Pattern (SSP) mining.

1.3.1 Specific Objectives

Given the main objective of this work, the following specific objectives emerge:

1. Understand the state of the art about textual pattern recognition through a systematic literature review;

2. Define SSP, SSP instances and investigate the incidence of such patterns in real datasets;

3. Implement a method for SSP instance mining and grouping, therefore characterizing patterns;

4. Apply SSP mining in the context of information extraction, discourse analysis and text classification.

Our approach offers certain advantages, specially when dealing with microblog texts. Since we use the word vector representation, our patterns are not tied to lexical similarity. Thus, SSPs don't have to contain the same word order, nor the same words. Beyond that, we have the possibility of training word vectors with tweet datasets, for instance, thus acquiring a vocabulary more appropriate some particular domain. By doing so, we also may capture the context (surrounding words) of Out of Vocabulary (OOV) words, such as neologisms and slangs, if they appear frequently in a given dataset. These factors make the SSPs less restrictive and more suitable for application in microblog texts.

1.4 DISSERTATION ORGANIZATION

The remainder of this dissertation is organized as follows: First, a brief theoretical review of the concepts that are used in this paper are presented in Chapter 2. Subsequently, the Chapter 3 discuss the related work. In Chapter 4, our Short Semantic Pattern (SSP) is formally defined and the algorithms for mining and grouping pattern instances are presented as well. Afterwards, Chapter 5 shows the experiments performed in the context of Information Extraction and Discourse Analysis. Finally, in Chapter 6 we present the scientific contributions of this research, conclusions and future works.

2 FUNDAMENTALS

In order to discuss our approach, it is important to clearly define the main concepts and techniques associated to the problem that we tackle and the solution that we propose in this dissertation. As such, this section aims to brief a theoretical background regarding semantic representation and annotation of texts, hate speech and Twitter discourse characteristics.

2.1 SEMANTIC REPRESENTATION

The Language allows humans to communicate, express thoughts, concepts, desires, among many others things. Hence, the words meanings must represent our mental conceptions of objects, actions, etc. There are two fields of linguistics that study the meaning of words: semantics and pragmatics. The study of semantics focuses in how grammatical processes build complex meanings out of simpler ones and the literal meaning of sentences. Therefore, semantics focuses on the link between the lexicon (vocabulary of a language), the grammar and the semantic (literal) meaning. Meanwhile, pragmatics aims to explain how factors outside of language contribute to both literal and non literal meaning that speakers communicate though language. Thus, pragmatics focuses on the connection between context of use, semantic and speakers intended meaning. Despite this theoretical division, linguist and computational linguistics researches usually investigate both semantics and pragmatics (FASOLD; CONNOR-LINTON, 2014).

Meaning representation has long fascinated philosophers, linguists, psychologists, and interdisciplinary fields, such as computational linguistics. Therefore a broad number of researches were developed in such fields (LEVELT; CARAMAZZA, 2007). The semantic representation is a formal and abstract way of representing senses of sentences. Following the principle of semantic compositionality (PELLETIER, 1994), the semantic meaning of any unit of a language is determined by the semantic meanings of its parts along with the way they are put together. Therefore, when dealing with semantic representation, three main aspects should be considered:

- The morphological and syntactic structure of the sentence;
- The representation of word meaning;
- The relation between words meanings;

It is a difficult task to provide an adequate structured lexical semantic representation that satisfies the needs of the rules that map meaning to syntax. Words can assume different roles (e.g., *address* to something/someone, home

address) and assume distinct meanings depending on the context (e.g., *bank* were people sit, *bank*, the financial institution). Consequently, we may represent semantics or aspects of semantics in several ways, but such representations can't fulfill the requirements of all domains. However, obvious advantages arise from publishing and consuming structured machine readable data. It is substantially easier to process data that follows formal structures, which allows querying, for instance[1].

2.1.1 Knowledge Graphs

Natural ambiguities (e.g., same name to describe different concepts) can be eliminated by using formal structures. Before accessing knowledge graphs, it is necessary to conceptualize ontologies. Ontology is a term originated from the philosophy, which means the study of concepts that are related to what is known as well as the basic categories of being and their relations [2]. In the Computer Science field, an ontology is a formal representation of knowledge though abstract categories (classes), objects (instances), properties, and relationships between them. Guarino (GUARINO, 1998) defines an ontology as a conceptualization of an universe of discourse. An ontology is used to represent the entities of a given domain as well as to describe the domain itself and reason about the properties of such domain. It is a formal and powerful method to describe taxonomies and classification networks, by characterizing the structure of knowledge for various domains. Usually, a ontology is created by using computational languages based on description logics, such as Web Ontology Language (OWL)[3].

A knowledge graph is described as a collection of relational facts that are generally represented in the form of a triplet *head entity, relation, tail entity* (Z. WANG et al., 2014). These entities may represent people, objects, events, situations, abstract concepts, etc. By describing the relationship between such entities in a formal structure, entity descriptions contribute to one another. Thus, each entity represents part of the description of the entities that are related to it, forming a network. Knowledge graphs can also be seen as tuple knowledge bases, such as Freebase (BOLLACKER et al., 2008), due to the fact that the data in it supports formal semantics expressed via tuples. Therefore, the data in knowledge graphs can be queryed and used to interpret the data and infer new facts. Also, the term knowledge graph is often used as a synonym of ontology. The Figure 2 shows and example of knowledge graph that can be constructed based on real tweets (documents d_i and d_j).

[1] https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html
[2] https://plato.stanford.edu/entries/logic-ontology/#Ont
[3] https://www.w3.org/OWL/

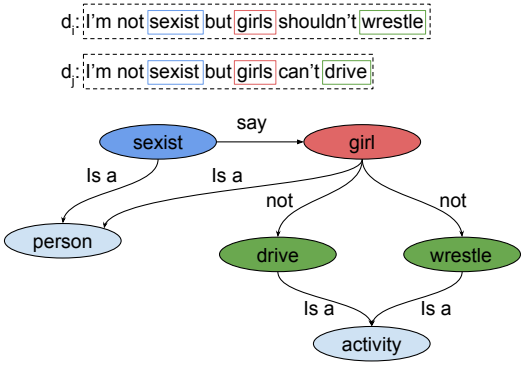

Figure 2: Examples of knowledge graph that can be build from real tweet texts.

The Linked Data (LD) can be seen as a type of knowledge graph with further restrictions. LD refers to machine readable data published on the Web, i.e., the meanings are explicitly defined, linked to other external datasets and can be linked from external datasets as well (L. YU, 2011). Conceptually, it also refers to a set of best practices for publishing and connecting structured data on the Web. Thus, the key idea behind LD follows two principles: (i) usage of the Resource Description Framework (RDF) data model to publish structured data on the Web and; (ii) usage of RDF links to interlink data from different data sources. By obeying these principles we should accomplish the creation of a Web of Data, which machines can read and understand. To ensure that publishers will correctly follow such principles, Tim Berners-Lee proposes the following rules in his 2006 Web architecture note [4]:

- Use URI as names for things;
- Use HTTP URI so that a client (machine or human reader) can look up these names;
- When someone looks up a URI, useful information should be provided;
- Include links to other URI, so that a client can discover more things.

By using the unique and universal name to identify (URI) a certain resource or concept, we can solve the ambiguity of using the same word in different documents to refer to distinct resources or concepts. The second rule above is a further restriction of the first, to guarantee that data publishers create globally unique identifiers without involving any centralized management.

[4] http://www.w3.org/DesignIssues/LinkedData.html

2.1.2 Lexical Databases

Traditional dictionaries are arranged alphabetically, whereas lexical databases, such as WordNet are arranged semantically, creating electronic lexical databases of nouns, verbs, adjectives, etc (BANERJEE; PEDERSEN, 2002). These databases contain synsets. A synset is a group of synonymous words (according with one of their denotations). If a word can be found in two or more distinct synsets, it is considerate polysemous. For instance, the word 'bank' in one sentence can mean a financial institution, a slope beside a body of water, or even the funds held by a gambling house or the dealer in some gambling games. Thus, the word 'bank' will occur in many different synsets. The synsets also have a brief explanation of the meaning of the concept being represented, called glosses.

There is a wide variety of semantic relations that connects the synsets in lexical databases. Exploring these relations may be very valuable for NLP methods. For nouns, the most relevant relations are hyponymy, hypernymy, holonymy and meronymy. In the hyponymy and hypernymy relantionship, the hypernymous terms are more generic and a the hyponymous are more specific. For example, *{foundation, base, fundament, foot, groundwork, substructure, understructure}* is a hypernymous synset of *{house}*, which by its turn is a hyponymous synset to the previous. If the synset refers to a part of something denoted by another synset, then there is a relation of holonymy and meronymy between them. For instance, the synset *{hand, manus, mitt, paw}* is a holonymy of the synset *{finger}*, which is a meronymy of the previous. On the other hand, the most important relations between verbs are hypernymy and troponymy. If a synset expresses a way of achieving some objective represented in another synset, then its a troponymy. For instance, the synset *{march, marching}* is troponymous to *{walk, walking}*.

2.1.3 Word Embeddings

Word embeddings are vectors that represent words in a reduced vectorial space, which are produced by language modeling and feature learning techniques. They represent usual meanings and syntax of words according with the contexts in which they usually appear. The key idea behind such vectors is that words that occur in similar contexts tend to have similar meanings. This link between similarity in how words are distributed and sense similarity is called the distributional hypothesis, and is very antique, being formulated in the 1950s by linguists such as Joos, Harris, and Firth (JURAFSKY; MARTIN, 2014). Therefore, word embedding models aim to transform words in compact numeric vectors with properties that allow capturing the meaning and syntax of words in some corpus. This process of 'vectorization' of words transforms natural

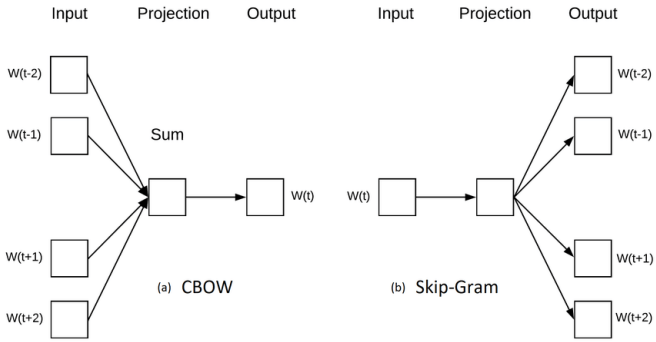

Figure 3: Illustration of CBOW and Skip-gram model architectures extracted from (MIKOLOV; K. CHEN, et al., 2013)

language to a machine readable representation.

There are sparse vectors and dense vectors. Sparse vectors such as those produced by Term-document and term-term matrix (TF-IDF) and Pointwise Mutual Information (PMI) have been broadly used in fields such as Information Retrieval. However, due to the very high dimensionality and sparsity of the vectors produced by such techniques, they are unable to capture semantic similarities. Due to this, dense vectors are more appropriate for approaches like the one developed in this work. Although there are several ways of generating dense word vectors, the most widespread techniques in the state of the art rely on neural networks, as seen in the Word2vec (MIKOLOV; K. CHEN, et al., 2013) model, which we use in this work. The type of information that can be embedded is not restricted to words. For instance, it is also possible to embed knowledge (Z. WANG et al., 2014), lexemes and synsets (ROTHE; SCHÜTZE, 2015). Beyond that, there are embeddings for closed domains, such as BioVectors, for biological sequences.

Word2Vec (MIKOLOV; K. CHEN, et al., 2013) and GloVe (PENNINGTON; SOCHER; MANNING, 2014) are the current most prominent models of word embeddings. Both methods are unsupervised and take a corpus or a dataset as input and output word vectors. Word2vec is a two-layer neural network model composed of two main learning algorithms: Continuous Bag of Words (CBOW) and Skip-gram. CBOW uses the context to predict a target word, whereas Skip-gram uses a word to predict a target context. The Figure 3 shows an illustration of such model architectures. Generally the Skip-gram model produces more accurate results on large datasets.

Global Vectors (GloVe) is a log-bilinear regression model that combines the advantages global matrix factorization (process of using matrix factorization

methods to perform rank reduction on a large term-frequency matrix) and local context window methods, such as the skip-gram model. The GloVe model leverages statistical information by training only nonzero elements in a word-word co-occurrence matrix, instead of the entire sparse matrix or individual context windows. Furthermore, instead of learning the raw co-occurrence probabilities, the GloVe learning algorithm learns the ratios of such co-occurrence probabilities. Compared to the raw probabilities, the ratio is more suitable to distinguish relevant words from irrelevant words and to recognize distinctions between relevant words (PENNINGTON; SOCHER; MANNING, 2014).

In contrast to classical distributional methods such as Latent Semantic Analysis (LANDAUER; DUMAIS, 1997), these models focus in trying to predict a target word given its neighboring words (IACOBACCI; PILEHVAR; NAVIGLI, 2015). In these models, relationships between words are present as vector offsets. As such, operations between vectors can be applied, resulting in interesting cases like *vector('king') − vector('man') + vector('woman')* being close to *vector('queen')*. Ultimately, it can be observed that word vectors conserve interesting properties that reveal implicit contextual information, which are very suitable for NLP tasks (MIKOLOV; YIH; ZWEIG, 2013). Beyond that, word vectors can be used for deriving word classes from large datasets by performing K-means clustering on top of the word vectors[5]. To illustrate the aforementioned concepts, Figure 4 presents word vectors similar to the *sexist* vector projected in a 2D space. The word similarities are calculated trough the word vectors cosine distance.

Aiming to tackle the high incidence of out-of-vocabulary words (e.g., slangs, neologisms) in social media text, one of the embedding models that we tested in this work is FastText (BOJANOWSKI et al., 2016). In FastText each word is represented as a sum of character n-gram representations. Therefore, this model takes into account the internal structure (morphology) of the words, being suitable to represent word forms that rarely occur in the training set.

2.2 SEMANTIC ANNOTATION OF TEXTS

The semantic annotation (or semantic tagging, semantic enrichment) of texts is the process of attributing additional information, such as concepts (e.g., people, organization) or senses, to words or sentence fragments. Thus, the semantic annotation process aims to enrich content by linking it to other resources. Semantically annotated texts facilitates and/or enables the extraction knowledge from sources, automatically interconnect content, disambiguate resource discoveries, information analytics, among many other possibilities.

[5] https://code.google.com/archive/p/word2vec/

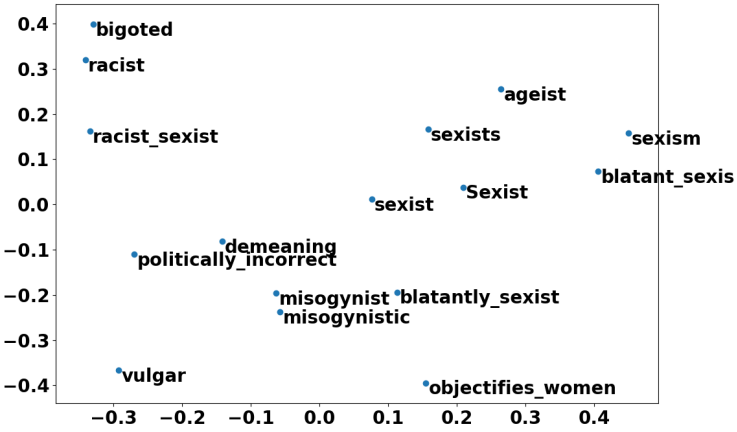

Figure 4: Word vectors related to the word *sexist*.

2.2.1 Named Entity Recognition

Essentially, named entities are components in a sentence that represents concepts such as persons, organizations, etc (TJONG KIM SANG; DE MEULDER, 2003). The classical category labels of named entities consists in four groups: person, organization, location, and miscellaneous. However, nowadays the majority of tools that process named entities covers more specific and meaningful labels or allows the inclusion of new ones. Some examples of non classical labels that are commonly used are product, monetary values, periods of date and time, events and numerals.

Named Entity Recognition (NER) refers to task of recognizing mentions of named entities in a text, followed by the annotation of the named entity with its respective category label. Those entities can be further disambiguated by linking them to resources, such as ontology concepts, in a process called Named Entity Disambiguation. Figure 5 shows an example of these processes. The components '#TheForceAwakens' and 'ONE WEEK' are recognized as two named entities, the fist as a product (movie) and the second as a time period and '#TheForceAwakens' can be linked to the concept on the Wikipedia ontology that describes the movie. There are several approaches for NER. The first works in the area were based in heuristic and handcrafted rules, but currently NER methods rely on supervised (RITTER; CLARK; ETZIONI, et al., 2011; MCCALLUM; LI, 2003), unsupervised (ELSNER; CHARNIAK; JOHNSON, 2009; ETZIONI et al., 2005) or semi-supervised (also called weakly supervised) learning (PAŞCA, 2007; KLEMENTIEV; ROTH, 2006;

Figure 5: Example of Named Entity Recognition

CARLSON et al., 2010).

2.2.2 Word Sense Induction and Disambiguation

The natural language is inherently ambiguous, thus several words have more than one sense. Word Sense Induction (WSI) and Word Sense Disambiguation (WSD) are two complementary linguistic tasks, that have the objective of attributing senses to words. While WSI aims to induce the possible senses for a word, WSD has the objective of defining which sense is the correct for a certain word, given a context. Word senses are commonly represented as a list of definitions in lexical databases, such as WordNet. This representation results in several problems, such as the lack of explicit semantic or contextual links between concepts (NAVIGLI et al., 2011) and more importantly, the definitions of such lexical databases often do not reflect the exact meaning of a word in a given context (VÉRONIS, 2004; MANANDHAR et al., 2010). Notice that due to word ambiguities, not every synset wherein a given word occurs has the desired meaning, therefore attributing the correct sense is not an easy task.

Most of the approaches to perform WSD rely on manually sense-tagged text or lexical databases (LESK, 1986; MIHALCEA; MOLDOVAN, 1999; BANERJEE; PEDERSEN, 2002; VASILESCU; LANGLAIS; LAPALME, 2004; SINHA; MIHALCEA, 2007). The biggest drawbacks of relying on such resources are the high cost of the corpora creation and the limitation of fixed senses. However, with the popularization of Word embeddings, many works concerning WSD now apply embeddings and obtained state-of-the-art results (ROTHE; SCHÜTZE, 2015; TAGHIPOUR; NG, 2015; TRASK; MICHALAK; J. LIU, 2015; IACOBACCI; PILEHVAR; NAVIGLI, 2015).

The Word Sense Induction (WSI) task aims to automatically discover the senses of a word, without relying on any hand-crafted resources such as manually labeled corpora. Some recent works in machine translation (VICKREY et al., 2005) and information retrieval (VÉRONIS, 2004) suggests

that sense induction has succeeded where methods based on a fixed sense inventory have previously failed (CARPUAT; D. WU, 2005; VOORHEES, 1993). Generally, sense induction is treated as an unsupervised clustering problem. The input to the clustering algorithm are instances of the ambiguous word with their accompanying contexts, represented by co-occurrence vectors, and the output is a grouping of these instances into classes corresponding to the induced senses. Thus, contexts that are classified in the same group represent a specific word sense (BRODY; LAPATA, 2009).

2.2.3 Part-of-Speech Tagging

The task of Part-of-Speech (PoS) Tagging (VOUTILAINEN, 2003) annotates each word in a text with its syntactic class (e.g., noun, adjective, verb), based on both the definitions and contexts (surrouding words) of these components in the text. The precise identification of the morphosyntactic elements of a sentence is of great importance, because when classifying a single word incorrectly, it can generate subsequent processing errors (TOUTANOVA et al., 2003). There are several techniques for the implementation of PoS Taggers, ranging from probabilistic ones to machine learning techniques and hybrids.

Ancient PoS-Taggers were mostly rule-based, as they try to assign tags to words using a set of hand-written rules. An obvious difficulty of this method is the dependence of specialists for the creation of rules. Due to the superior performance of current techniques, pure rule-based PoS-Taggers are no longer used. However, certain hybrid approaches, combine rules and probability, thus attributing the most likely tag based on a training corpus, and then apply a given set of rules to see if the tag should be changed. The use of machine learning in PoS-Taggers can explore labeled training data to adapt to new genres or even languages, through supervised learning.

2.2.4 Classification

In machine learning and statistics, classification is defined as the task that aims to predict the class, also called target, label or category, of data points in a given dataset. In other words, classification can be seen as the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y). Example of classification problems are spam detection, authorship identification, sentiment analysis, hate speech recognition and age/gender identification. Therefore, given a document d and a set of fixed classed $C = \{c_1, c_2, ..., c_n\}$, the classification of the document d is a predicted class $c \in C$. An algorithm that performs classification is called a classifier, which aim to assign some sort of output value to a given input value. The classification problem itself, is an example of a more generic problem, which is

pattern recognition. Usually pattern recognition systems use feature descriptors and a particular classification procedure to determine the true class of a given pattern (HO; HULL; SRIHARI, 1994).

There are several classifiers, such as Naïve Bayes, decision trees, Support Vector Machines (SVM), logistic regression, among many others. There are also very distinct classification algorithms, ranging from rules based combination of words or other features (e.g. black list addresses for spam detection) to machine learning ones. The most suitable architecture of classifier depends on various factors besides the target problem itself, such as the size of the training dataset.

2.2.5 Discourse Analysis

On a broad perspective, discourse analysis is the study of the ways in which language is used between people (written or spoken). Instead of investigating individual parts of language (e.g., words and phrases, grammar), discourse analysis takes in consideration other aspects, such as the social and cultural context of texts. Therefore, discourse analysis focuses on the general use of language within and between particular groups of people.

From a computational perspective, the analysis of discourse aims to model human understanding/generation of natural language and particular phenomena in discourse and dialog in computational processes. It also has the objective of providing useful natural language services concerning discourse and dialog aspects (e.g., parsing, question answering, machine translation) (TANNEN; HAMILTON; SCHIFFRIN, 2015).

Discourse analysis can be used to study inequality in society, such as racism, sexism, bias in media, and political discourses. One of the defining characteristics of discourse analysis is that it is capable of application in a myriad of settings and contexts (TANNEN; HAMILTON; SCHIFFRIN, 2015). It has evolved to include several topics ranging from formal to colloquial rhetoric, public to private language use, oratory to written and multimedia, among others. For instance, critical discourse analysis (CDA) is a type of study that primarily focuses in the way that social power abuse, dominance, and inequality occurs and are reproduced and resisted by text and speech in the social and political context. The medical language and forensic linguistics are also very fertile fields, encompassing topics such as language of the courtroom, communication between patients and physicians, response analysis in crimes, etc. By examining the context of language use instead of only considering language structures, we can understand the layers of meaning added by the social aspects such gender, racism, conflicts, etc.

2.3 HATE SPEECH AND TWITTER CHARACTERISTICS

In this dissertation we access discourse characteristics in tweets, mainly regarding hate speech and abusive language. To understand the impact of hate speech it is important to first conceptualize it. Furthermore, we propose the reflection of certain Twitter characteristics which may incite abusive language and hate speech in Twitter.

2.3.1 Hate Speech

The popularization of social media is a two-edged sword: besides being a space for social integration in which users share different thoughts and opinions, social media has also become a mean to incite hate against people, spread disinformation and manipulate opinions. Hate speech is often defined as the use of abusive/offensive language targeting certain groups of people that share a common property (e.g., gender, religion, nationality) (NOCKLEBY, 2000). Generally, hate speech is accompanied by abusive language, such as sexists slurs, to express hatred in the discourse. The occurrence of abusive language, and ultimately hate speech, is very common on the Web, where forms of racism and sexism are the most frequent(KETTREY; LASTER, 2014; WASEEM; HOVY, 2016). Identifying and analyzing such hateful speech in social media is crucial for understanding, fighting, and discouraging offensive activities towards groups of people.

Despite the fact that the term *hate speech* became popular only recently (DAVIDSON et al., 2017; WASEEM; HOVY, 2016; GITARI et al., 2015), other authors have already used terms such as *abusive* or *hostile messages* (SPERTUS, 1997), and *cyberbullying* (ZHONG et al., 2016; DINAKAR et al., 2012) to address the same issue (SCHMIDT; WIEGAND, 2017).

Hate speech identification requires awareness of the context within the sentence. Although dictionaries can be applied to detect explicit instances of hatred and slurs in text, it does not discern more subtle cases. It is a complex task to depict hate speech in a given sentence, specially if it contains language figures or does not use explicit slurs. For instance, despite the absence of slurs in the mockeries of women participants in the Australian cooking show *My Kitchen Rules* in the real tweets listed bellow, they were sent by an user that frequently posts content attacking female competitors (explicitly and implicitly).

- t_1 = *'Nikki...Kermit the frog called and he wants his voice back #MKR #MKR2015 #KillerBlondes @mykitchenrules'*
- t_2 = *'Nikki looks like the kind of girl who would ask if there's anything cheaper than a $4 skittle bomb #MKR #MKR2015'*

On the other hand, apparently hateful expressions that include reference to a particular gender, race, ethnic group or religion are broadly used in joking contexts between friends. The tweets listed below are not instances of hateful speech, regardless of the words highlighted in bold being frequently used in hate speech.

- t_1 = *'I just be wanting to talk to my* ***n*gga*** *I don't like y'all'*
- t_2 = *'I love you my fave* ***b*tch***.'

Therefore, in order to decide if a text depicts a case of hate speech, it is necessary to perform a critical analysis of its contents, taking into consideration aspects such as the usual context in which certain words and expressions are employed, besides the text itself. Consequently, the solely use of dictionaries and n-grams are not sufficient to develop a suitable approach to detect hate speech.

Another problem in hate speech detection lies in the subjectivity of the resource annotators. Although attacks aimed at women are very frequent on the internet, frequently human coders tend to classify these contents as only offensive, in contrast to racist and homophobic contents, which are viewed as hateful (DAVIDSON et al., 2017; WASEEM; HOVY, 2016).

2.3.2 Twitter Characteristics

Twitter is a microblogging platform, i.e., a form of blogging consisting of short content such as phrases, quick comments, images, or links to videos (STIEGLITZ; DANG-XUAN, 2013). Due to its post character limitation, Twitter structurally disallows the communication of detailed and sophisticated messages (OTT, 2017). Therefore, complex ideas suffer major simplifications or have the need to be transmitted via links. This trait can be considered prejudicial in two ways. First, the cycle of continuously redirecting the user's attention elsewhere via hyperlinks may cause shortening of attention span. In fact, the culture of the internet encourages "shallow" information processing behaviors through rapid attention shifting and reduced deliberations (LOH; KANAI, 2016). Second, as a result of such simplicity restrictions tweets can be posted or retweeted very fast and easily. Therefore, this easiness may influence impulsive discourses with disregard for reflection, or consideration of consequences.

These characteristics allied to the informal style of writing and the protection of messages exchanges on a non-physical medium encourages uncivil discourses. If the user is not required to cautiously think how to express an idea and does not have to suffer from real world social retaliation, it is much easy to make aggressive and offensive comments. Ultimately, the impulsiveness and

anonymity or the absence of physical presence enables a way to depersonalizes interactions, generating a disregard for consideration of how interactions will affect others.

3 RELATED WORK

The idea of exploring forms of linguistic patterns for application in NLP tasks dates back to a long time. A myriad of types of linguistic patterns may be explored in natural language. Thus, works on linguistic patterns tend vary greatly in purpose of application. Nonetheless, few recent works access linguistic pattern recognition in the literature and are mostly directed towards structurally fixed patterns discovery. Most of the published studies, especially older ones, are rule based (HEARST, 1992; BERLAND; CHARNIAK, 1999; LIN et al., 2003). These rules are handcrafted and therefore those methods are time consuming and very often devoted to a specific corpus (BÉCHET et al., 2012).

Most related to ours are the works of Mondal et al. (MONDAL; SILVA; BENEVENUTO, 2017) and Watanabe et al. (WATANABE; BOUAZIZI; OHTSUKI, 2018). In Mondal et al. (MONDAL; SILVA; BENEVENUTO, 2017) the authors propose to evaluate the measurement of hate speech using common expressions and linguistic patterns. They analyze hate speech in contents from Twitter and Whisper concerning race, sexual orientation, class, gender, among others. Their idea is to use predefined sentence structures to discover hate speech patterns. In contrast to our work, which aims to find patterns dynamically, they define sentence template, namely '$I < intensity >< user\ intent >< hate\ target >$' to evaluate data. Watanabe et al. (WATANABE; BOUAZIZI; OHTSUKI, 2018) uses unigrams and patterns extracted from a training set and use them as features to train a classifier available in Weka [1]. The authors combine features such as sentiment, semantic (e.g., punctuation, capitalized words), with unigrams, to detect explicit forms of hate speech, and pattern features, to identify longer or implicit forms of hate speech. The patterns are extracted based on sequences of the PoS-Tags of the words in the tweets. They test their approach on binary classification (hateful or clean) and ternary classification (hateful, offensive or clean).

The work of Schwartz et al. (SCHWARTZ; REICHART; RAPPOPORT, 2015) also leverages sentence structures to find patterns. However, instead of finding semantic patterns, they are interested in finding symmetric patterns (e.g, X *and* Y, X *in* Y) . The authors aim to automatically acquire such symmetric patterns from a large corpus of plain text to generate vectors, where each coordinate represents the co-occurrence in symmetric patterns of the represented word with another word of the vocabulary. Therefore, their patterns are used to build a symmetric pattern based model for word vector representation. Their model proved to achieve superior performance than six strong baselines, including the skip-gram model on SimLex999 (HILL;

[1] https://www.cs.waikato.ac.nz/ml/weka/

REICHART; KORHONEN, 2015). They also combined their approach with the skip-gram model, attaining even better results.

All the following related works explore the use of PoS-Tags to find linguistic patterns. Despite the fact that PoS-Tags are a great way of extracting patterns though morphological and syntactic structure in formal texts, PoS-Taggers perform poorly on social media texts (SORATO et al., 2016). In Béchet et al. (BÉCHET et al., 2012) the authors present a method based on data mining techniques to automatically discover linguistic patterns by matching appositive qualifying phrases. Their algorithm is capable of mining sequential patterns made of itemsets using part-of-speech tags. By using itemsets, a word can be represented by a set of features (e.g., words, PoS-Tag, lemma), combine different levels of abstraction. As the extracted sequential patterns are partially ordered, they are further organized in a data structure according to such partial order so that a user can manually validate them.

The work developed by Zouaq et al. (ZOUAQ; GASEVIC; HATALA, 2012), Tovar et al. (TOVAR et al., 2014) and Volkova et al.(VOLKOVA et al., 2010) are aimed at ontologies. The work of Volkova et al.(VOLKOVA et al., 2010) has the objective of using syntactic patterns and PoS-Tagging to obtain semantic relations in a ontology in the veterinary medicine domain. In their approach, they manually constructed an ontology for extracting veterinary biomedical entities (e.g., animal disease names, viruses). Afterwards, they apply automated ontology expansion to extract semantic relationships, namely synonymy, hyponymy and causality, between the concepts using syntactic patterns. In the expansion step, they search for such relationships between entities using both the initial ontology and raw data from the veterinary medicine domain. They define fixed syntactic structures (e.g., “is a”, “is equivalent to”, “and”, “for instance/”, “and|or other”) and use PoS-tagging to extract n-gram concepts (e.g., “swine vesicular disease”). Lastly, a new ontology is created based on the manually created one and the information extracted.

In Zouaq et al.(ZOUAQ; GASEVIC; HATALA, 2012), the authors present a number of fixed syntactic patterns, based on dependency grammars, which output triples for ontology learning. They address the patterns used by OntoCmaps, an ontology learning tool developed by the same authors that takes unstructured texts about a domain of interest as input (ZOUAQ; GASEVIC; HATALA, 2011). In the knowledge extraction step, OntoCmaps utilizes these syntactic patterns to extract candidate triples from texts. Such syntactic patterns are extracted using dependency grammars and PoS-tagging, where the dependency analysis is obtained through the Stanford Parser[2]. The patterns target specific syntactic structures in a dependency representation to extract

[2] https://nlp.stanford.edu/software/lex-parser.shtml

multi-word expressions and triples that can be translated into OWL classes and properties. Tovar et al. (TOVAR et al., 2014) proposes an approach for evaluate the validity of taxonomic relations (hypernym/hyponymy or subsumption) of restricted domain ontologies using patterns extracted from the referring corpora. Their pattern structures are based in PoS-Tags, where their use Freeling[3] to perform the tagging, and regular expressions using PoS-Tags and tokens. In total, they collected 106 lexico-syntactic patterns associated with the identification of taxonomic relations (e.g., 'NP (is | are) NP', 'NP NP , is (a|an|the) NP'), from which they considered only 16 to be useful. The lexico-syntactic extracted are used for discovering evidence of the ontology taxonomic relations in its reference corpus.

Our solution relies on new methods to mine a kind of pattern (Short Semantic Pattern) that, to the best of our knowledge, has not been exploited yet. It allows the identification of common thoughts and senses used in discourses towards certain entities or groups. In relation to the mentioned related works, our approach has the following advantages: (i) the usage of semantic similarity of word vector representation allows us to uncover patterns which are structurally flexible and not tied to lexical similarity; (ii) in general, word embedding models are easy and quick to train, therefore it is possible to train a model with texts suitable to the desired scope before the pattern mining; (iii) training word embedding models on social media texts enables the capacity of depicting the context in which OOV words such as slangs are used; (iv) our discovery of patterns is dynamic and is not dependent of the results of PoS-Taggers and thus not sensible to PoS-Tag attribution errors. The Table 1 summarizes the comparison of related works.

[3] http://nlp.lsi.upc.edu/freeling/node/1

Author and Year	**Input**	**Word Representation**	**Objective**	**Dynamic Discovery of Patterns?**	**Type of Pattern**	**Approach**
Volkova et al., 2010	Veterinary articles and ontology	Tokens, ontology concepts	Improve biomedical entity extraction	No	Syntatic	Rule-based patterns and semantic relationships
Zouaq et al., 2012	Ontologies	PoS-Tags and tokens	Ontology learning	No	Syntatic	Dependency grammars
Béchet et al., 2012	Corpora of newspaper	PoS-Tags, lemmas and tokens	Information extraction	Yes	Syntatic	Itemset mining
Trovar et al., 2014	Corpus of manuals, Wikipedia pages	PoS-Tags and tokens	Evaluate taxonomic relationships	No	Lexico-syntatic	Regular Expressions
Schwartz et al., 2015	Word similarity datasets	Tokens	Improve word embedding similarity	No	Simmetric	Token template and antonyms
Mondal et al., 2017	Microblog texts	Tokens	Information extraction, text classification	No	Semantic	Sentence template
Watanabe et al., 2018	Microblog texts	PoS-Tags and tokens	Text classification	Yes	Syntactic	PoS-Tags matching
Sorato et al., 2019	**Microblog texts**	**Word Vectors**	**Information extraction, discourse analysis, text classification**	**Yes**	**Semantic**	**Context window and semantic similarity**

Table 1: Related works.

4 SHORT SEMANTIC PATTERNS

Our approach revolves around the representation of each word in a sentence as a word vector produced by some embedding model. These vectors can then be iteratively aggregated by some function (e.g., sum, average) and filtered to form a structure of arbitrary size (n-grams). Ultimately, sequences with similar meanings but distinct sizes and lexical contents can be detected, as seen in Example 1.

Example 1 *Given a set of documents* $D = \{d_1, d_2\}$*, in which*
d_1 =‘That man is **sexist**’ *and*
d_2 =‘That man **objectifies women**.’

A Short Semantic Pattern (SSP) refers to a collection of word sequences that may be distinct in terms of lexical components and size, but are semantically similar to each other and correspond to a frequent meaning. Each of these word sequences must be found in distinct documents (e.g., two or more different social media posts) and is called an instance of the respective SSP. Although distinct documents can pose thoughts and opinions by using diverse words and expressions, these are just varied ways to make similar statements. An example of SSP expressed in distinct instances taken from tweets is displayed in Figure 6. A statement like ‘*I’m not sexist, but*’ does not configure sexism by itself, but may characterize a linguistic pattern, as it is frequently used to introduce sexist commentaries such as ‘*women cannot drive*’. The text fragments comprised in the gray boxes in Figure 6 have similar meanings and display the same intended sense (prejudice against women drivers) although they do not have the same lexical and syntactic arrangement. Thus, they can be seen as instances of the same SSP. In short, SSP are represented by a collection of instances, these being text fragments present in distinct documents that are related through semantic similarity.

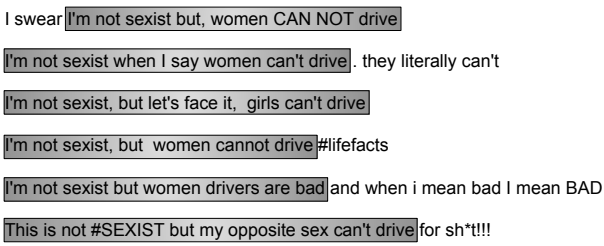

Figure 6: Examples of SSP instances.

To uncover such patterns, first it is necessary to find it’s instances.

A formal definition of a Short Semantic Pattern instance is presented in Definition 1.

Definition 1 (Short Semantic Pattern Instance) *Let D be a set of documents, $similar(s, s')$ be a semantic similarity function between word sequences s and s' of two distinct documents in D, and t be a threshold that specifies a minimum semantic similarity value between two word sequences of distinct documents d, $d' \in D$. A Short Semantic Pattern instance $SSPi$ contained in a set of Short Semantic Patterns SSP is a non-empty set of word sequences s satisfying the constraint:*

$$\exists d',\ d \in D,\ d' \neq d \mid \forall SSPi \in SSP : (s \subseteq d \wedge \exists s' \subseteq d') \wedge (similar(s, s') \geq t)$$

Of course, for the sake of efficiency and flexibility, the similarity function $similarity(s, s')$ can be calculated using word embedding representations of the word sequences s and s'.

4.0.1 SSP Mining

SSP mining aims to find SSP instances in a collection of short texts, such as tweets. In order to obtain SSP instances, it is necessary to locate word sequences that are semantically similar in different documents. As such, the mining process rely on the expansion of a Context Window (CW) to find the longest sequence. A context window is represented by a tuple $CW = (d_i, keyword, s_r, s_l)$, where $d_i \in D$ is the document that the context window is associated with. The keyword, i.e. $keyword \in d_i$, is the word in the central position of CW, while $s_r, s_l \in [0, |d_i|]$ are integers that denote the borders of CW, in terms of the number of words on its right side and left side, respectively. During the mining process, the s_r and s_l values are dynamically increased while the semantic similarity between the contents of two context windows are kept above a threshold. Figure 7 illustrates a growing context window, iteratively covering more neighboring words in a tweet. Different numbers of words to the left or to the right of the keyword are allowed.

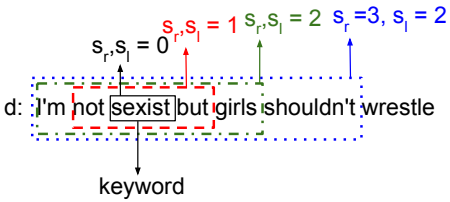

Figure 7: Examples of context window in a document d.

4.0.2 SSP using Keywords

The construction of a given instance through a context window depends on the selection of a central word $keyword$ in a given document. In this work we propose to use domain keywords. Therefore, if a document contains any of such keywords, it is marked for pattern mining and that keyword is chosen to be the word in the central position of an expanding context window. A pseudo code for the task of creation, expansion and comparison by similarity of words sequences centered in a given keyword occurring in two documents is shown in Algorithm 1. A context window object *ContextWindow* is created for each one of the two documents being analyzed. If a document contains more than one domain keyword, the matches are analyzed separately. Each context window is expanded gradually from its center $keyword$ to find the maximal word sequence in its respective document that is still similar to the other word sequence. The method *expand(document, ContextWindow)* expands the context window in both directions while testing the document boundaries. If the context window reaches the beginning or the end of the sentence, the context window is not expanded in that direction anymore. The expansion continues while the similarity value between the two word sequences being analyzed is kept above the threshold value t. When the context windows cannot be further expanded, the words contained within their boundaries are aggregated and stored as two distinct SSP instances. Afterwards, other distinct documents will pass trough the same process, in order to find more instances. An SSP is characterized by the collection of correlated instances in the analyzed documents.

Subsequently, instances that share similar semantics are grouped into SSP. This task is explained through Algorithm 2, which is called after the instance mining step (Algorithm 1). Using the same threshold value t defined to mine the SSP instances, we decide whether or not an instance should belong to an existing pattern or constitute a new pattern. As such, the SSP instances are passed as an argument to Algorithm 2 and a map marking all the instances for verification is created with the label *Pending*. While there are still pending instances to be analyzed, one will be selected to be the pivot and be compared against every other pending instance. An instance that is at least t similar to the pivot will be grouped to constitute a new pattern and they label will be changed from *Pending* to a incremental *Pattern_Id*. As a result, after analyzing every instance, they will be associated and aggregated to form patterns.

After the grouping step (Algorithm 1), the SSP that achieved only a small number of instances can be discarded. We apply a minimum support *minsupp* parameter, which can be defined by the user. The *minsupp* determines what is the minimum number of instances for a given SSP to be considered valid. This process is illustrated in Algorithm 3.

Algorithm 1: Creation, expansion and comparison of word sequences inside context windows.

Input: Two documents d and d', with their respective matching positions concerning a keyword.
Result: Two instances of an SSP, which are at least t similar.

```
instances = []; keyword_d = d[match.start:match.end];
keyword_d' = d'[match.start:match.end];
cw_d = ContextWindow(d, keyword_d);
cw_d' = ContextWindow(d', keyword_d');
similarity = similar(cw_d, cw_d');
while similarity ≥ t do
    cw_d.expand(d,keyword_d);
    cw_d'.expand(d', keyword_d');
    similarity = similar(cw_d, cw_d');
end
instance_d = (d.id, cw_d);
instance_d' = (d'.id, cw_d');
instances.append(instance_d, instance_d');
```

Figure 8 illustrates our general process for SSP mining. After mining and grouping SSP instances, the patterns are stored in a dictionary structure. The quality of the patterns must be manually analyzed by an specialist (we intend to investigate semi-automatic ways to validate the patterns in future works). Afterwards, the analysis of the pattern contents can reveal a variety of information, including words and concepts frequently attributed to a given target (group or entity), common expressions employed by users when referring to a certain subject, etc. Furthermore, the patterns can be used in other NLP tasks, such as text classification.

Algorithm 2: Grouping similar instances.

```
Input: A list of SSP instances.
Result: A dictionary of SSPs with their respective list of instances.
Pending = 0;
Discarded = 1;
mapped = 0;
current_id = 2;
pattern_map = list(map(lambda item: Pending, instances));
dictionary = dict();
while mapped_instances < length(instances) do
    for i, pivot in instances do
        if pattern_map[i] is not Pending then continue ;
        marked = 1;
        pattern_map[i] = Discarded;
        for j, instance in instances do
            if pattern_map[j] is not Pending then continue ;
            if pivot.similar(instance) ≥ t then
                pattern_map[i] = current_id;
                pattern_map[j] = current_id;
                marked += 1;
            end
        end
        if pattern_map[i] is not Discarded then
            current_id += 1;
            mapped += marked;
        end
        for i,item in pattern_map do
            if item not in dictionary then
                dictionary[item] = set();
            end
            dictionary[item].append(instances[i]);
        end
    end
end
```

Algorithm 3: Deleting SSPs bellow the *minsupp* parameter.

Input: A dictionary of SSP instances.
Result: A dictionary of SSPs instances, without SSPs bellow *minsupp*.
for *key, value in dictionary* **do**
  if *key == 1* **then** delete dictionary[key] ;
  if *length(value) < minsupp* **then** delete dictionary[key] ;
end

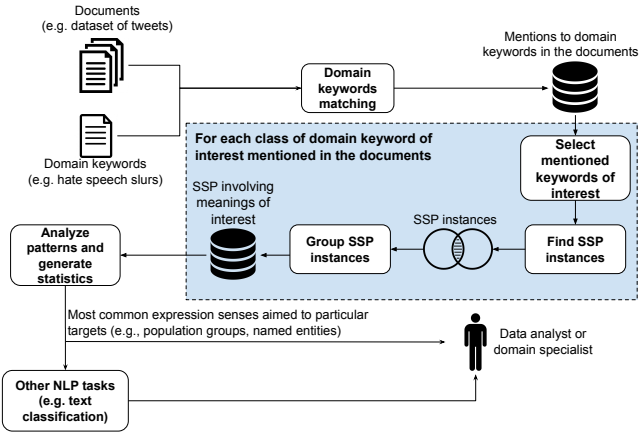

Figure 8: General process for SSP mining.

5 LINGUISTIC PATTERN MINING FOR DISCOURSE ANALYSIS AND TEXT CLASSIFICATION

In this chapter we present our experiments and results. We applied SSP mining in two distinct case studies. In the first case study we mined SSP in tweets posted by Donald Trump during the campaign for the 2016 presidential election in the United States of America. We investigate his recurring speeches about the media and certain political opponents. In the second case study we mined SSP in a hate speech dataset, in order to extract frequent hate discourses against certain groups. Afterwards, the extracted patterns are used as features in binary and ternary classification.

5.1 APPLICATION: DONALD TRUMP DISCOURSE ANALYSIS

This section describes our experiments discourse analysis using Donald Trump's tweets at the time of the campaign for the 2016 presidential election. Their goal is to show how our SSP mining technique can help to uncover frequent thoughts and discourses textually expressed by Donald Trump in tweets concerning the media and his adversaries.

5.1.1 Dataset

We selected a dataset of 6,445 tweets[1] that were posted or retweeted by Donald Trump and Hillary Clinton during the campaign for the 2016 presidential election. From these, we analyzed only the tweets posted or retweeted by Donald Trump, comprising 3,219 tweets. In this dataset we observed a incidence of tweets concerning vote intention polls, antagonism with his opponents and the media.

5.1.2 Implementation Resources

Our approach was implemented in Python3.6, using the open source libraries Spacy[2] and Gensim[3]. The Spacy Matcher module was used to find the mentioned keyword positions and Spacy's similarity function was used to calculate the cosine similarity between vectors. Gensim was used to adapt embeddings suitable to our scope (trained using twitter data [4]), instead of using the pre-trained models provided SpaCy.

[1] https://www.kaggle.com/benhamner/clinton-trump-tweets/home
[2] https://spacy.io/
[3] https://radimrehurek.com/gensim/
[4] https://nlp.stanford.edu/projects/glove/

5.1.3 Preprocessing

The preprocessing phase consists of dataset cleaning, retokenization and removal of stop words. First, we removed the URLs, RTs, emojis and general punctuation. We also retokenized occurrences of apostrophes (e.g., can't, I'm) to analyze them as single tokens. This is necessary to avoid complications in the expansion of the dynamic context window, as seen in Algorithm 1.

5.1.4 Keyword Matching

We decided to evaluate the SSP instances regarding the adversaries of Donald Trump and the media. Due to space limitation and statistical relevance, we chose to evaluate the interactions of Trump with the three most mentioned opponents. Table 2 shows the number of mentions by opponent, others being the sum of Jim Gilmore, Chris Christie, Carly Fiorina, Rick Santorum, Rand Paul, Martin O'Malley, Mike Huckabee, George Pataki, Scott Walker, Rick Perry, Bobby Jindal Lawrence Lessig, Lincoln Chafee and Jim Webb.

We also standardized the mentions to the adversaries which were selected for analysis (Hillary Clinton, Marco Rubio, Ted Cruz). The keywords used to mine SSP instances regarding Trump's adversaries and the media are presented in Table 3. When two or more matches occur in the same document, we analyze those matches separately.

Hillary Clinton	**Ted Cruz**	**Marco Rubio**
396	**235**	**95**
Bernie Sanders	Jeb Bush	John Kasich
78	65	42
Ben Carson	Lindsey Graham	Others
19	16	20

Table 2: Number of mentions per opponent.

5.1.5 Results and Discussion

We mined SSP instances with size of at least 3 words for the two aforementioned keyword groups. Through previous tests, we found that semantic similarity threshold values equal or bigger than $t = 0.8$ better represent our patterns. Therefore, we chose to evaluate threshold similarity values of $t = 0.8$ and

Topic	Keywords
Opponents	Hillary, Sanders, Rubio, Cruz, Bush
Media	conference, commercial, commercials, TV, television, story, Twitter, Facebook, broadcast, broadcasting, broadcasted, journalism, journal, journalist, report, reporter, reporting, reported, interview, interviewing, interviewed, press

Table 3: Keywords used to mine SSP instances.

$t = 0.9$. As expected, we verified that instances found using $t = 0.9$ have smaller size than instances found with $t = 0.8$ due to the less flexible threshold value. Furthermore, the number of patterns found using $t = 0.9$ were also significantly smaller in relation to $t = 0.8$. Table 4 shows the number of total matches found versus the number of distinct tweets that participated in the SSP using $t = 0.8$ and $t = 0.9$. Nonetheless, the patterns found with $t = 0.9$ were more fine grained, which in the context of this work means that the grouped instances formed patterns with better subject discernment. We also noticed that since instances mined with $t = 0.8$ have larger size, these components often carry more contextual information.

	Matches	**t=0.8**	**t=0.9**
Media	327	197	123
Opponents	Hillary		
	396	183	86
	Cruz		
	235	119	53
	Rubio		
	95	42	11

Table 4: Number of matches versus number of distinct tweets that participated in the SSP.

Patterns with less than $minsupp = 5$, i.e, five instances, were discarded due to low support. The contents of mined patterns are analyzed in the upcoming subsections. We performed analyzes regarding the most frequent words found in the SSP instances and the characterization of the target entities, i.e., Hillary Clinton, Ted Cruz, Marco Rubio and the media. Through these analyzes we aim to understand the contents of frequent discourses in the dataset, which ultimately represent user's repeated visions or thoughts about a given

subject.

5.1.5.1 The Media

The most highly supported patterns found using the media keywords displayed 3 major intended senses:

1. 'The media distorts information about me';
2. 'The media protects my opponents';
3. 'The media refuses to publish good things about me';

The word trees in Figure 9 illustrate these ideas. Usually, Trump uses generic terms to refer to the media, such as 'press' and 'media'. However, the New York Times and CNN were the most explicitly mentioned and criticized communication vehicles, frequently appearing besides the word 'failing'. Figure 10 shows the number of mentions to New York Times and CNN (above), and the most frequent words found in the mined instances (below). Demeaning adjectives (e.g., disgusting, corrupt) frequently appeared together, intensifying the idea of contempt and disdain towards the media. We also found instances encouraging people to mistrust the media, such as in *'Don't believe biased phony media quoting people who work for my campaign'* and *'No wonder @nytimes is failing who can believe what they write'*. Besides the patterns mentioned above, we found a pattern in which Trump informs the TV shows where he would be interviewed. The word tree in Figure 11 shows instances that illustrate this pattern.

5.1.5.2 Trump's Opponents

The most prominent patterns found using the opponents keywords displayed 4 major intended senses:

1. 'My opponents are incompetent and/or corrupt';
2. 'My opponents lie about me';
3. 'My opponents represent the old politics';
4. 'If my opponents are elected, it would be disastrous for the country';

The majority of the patterns had Hillary Clinton as a target and the most pejorative comments are also about her. The word *'Crooked'* is frequently accompanied by her name. Donald Trump tries to create an image about his adversaries, in which he portrays them as unqualified and dangerous.

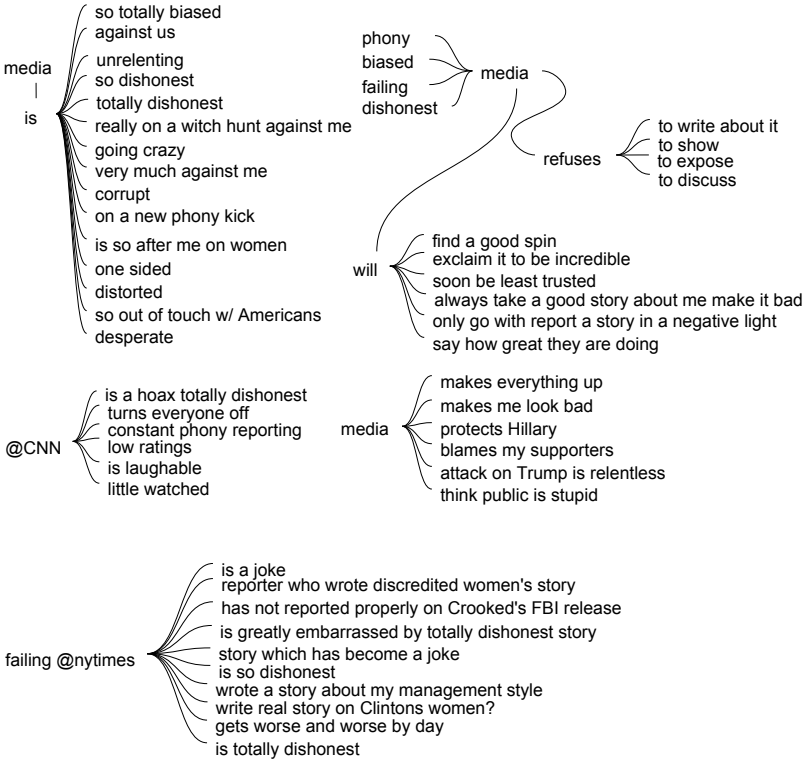

Figure 9: Word trees showing Trump's frequent discourses about the media.

Demeaning words such as 'liar', 'dishonest', 'corrupt' are very frequent in his discourse. Therefore, by spreading such concepts about their opponents, he would arise as the only viable solution.

In certain instances he also affirms that the other candidates spend a lot of money on ads against him and that they are being protected by the media. The word trees in Figure 12 depict the image that Donald Trump created about his adversaries.

5.1.5.3 Discourse Analysis

The discourse of Donald Trump fits the characteristics mentioned in Chapter 2. In general, his speech is simple, containing accessible words and even typing errors, impulsive and highly uncivil. Trump expresses his opinions frequently through disdain and degrading words. He often uses the words 'so', 'totally'

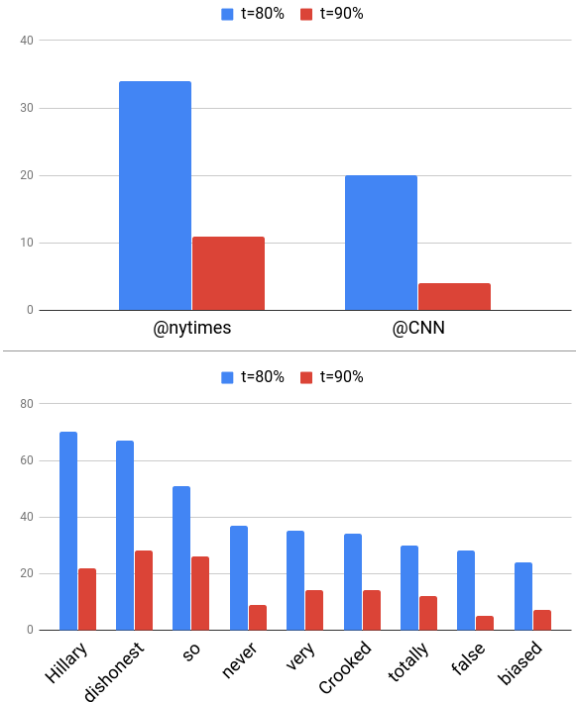

Figure 10: Most mentioned communication vehicles and words count.

and 'very' to give emphasis to bad characteristics of his targets. Beyond that, the patterns suggest that Trump chooses certain demeaning adjectives to his adversaries (Crooked, Lying, Little/lightweight), reducing people to a single bad characteristic. Figure 13 shows which where the most frequent words used by Trump to describe the media and their political adversaries in the mined instances.

Furthermore, he incites an atmosphere of mistrust and paranoia, saying the media and his adversaries are doing plots and lying about him. By making such affirmations, Trump refutes the racist and sexist accusations pointed by the media and his opponents. He constantly bullies his targets creating and reinforcing negative images about them.

5.2 APPLICATION: HATE SPEECH DETECTION

This section describes our experiments on discourse analysis and text classification using a hate speech dataset. The goal of this experiments is to

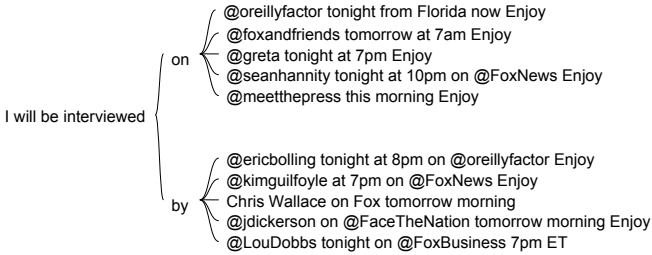

Figure 11: 'I will be interviewed on/by' pattern.

show how our SSP mining technique can help to uncover frequent thoughts and discourses textually expressed in Twitter posts containing sexist and racist content. We also aim to investigate if our patterns can improve classification performance by being incorporated as features.

5.2.1 Data

In this work we combined three distinct datasets:

- Waseem et al. (WASEEM; HOVY, 2016)[5]: This dataset contains a total of 16K tweets, of which 3.378 are labeled as *sexist*, 1.970 as *racist* and the remaining as *neither*, in accordance with their textual contents.

- SemEval 2019 Task 5 - Shared Task on Multilingual Detection of Hate (hatEval) (BASILE et al., 2019): Tweets collected from Twitter and manually annotated mainly via the Figur8 [6] crowdsourcing platform. The tweets from the task A, which were used in this work, are labeled regarding containing or not hate against women and immigrants. We used the train and test parts of the dataset, which contains a total of 10k tweets, of which 4.210 are labeled as *hate speech* and 5.790 as *clean*.

- Davidson et al. (DAVIDSON et al., 2017)[7]: The contents in this dataset were selected by searching tweets that contained lexicons from *Hatebase.org*. Afterwards, they were manually coded by CrowdFlower (CF) workers, who were asked to label each tweet as hate speech, offensive but not hate speech, or neither offensive nor hate speech. The dataset contains a total of 24.783 tweets of which 1.430 are labeled as *hate speech*, 4.163 as *offensive language* and 19.190 as *clean*.

[5] https://github.com/ZeerakW/hatespeech
[6] https://www.allcloud.io/figur8-is-now-allcloud/
[7] https://github.com/t-davidson/hate-speech-and-offensive-language

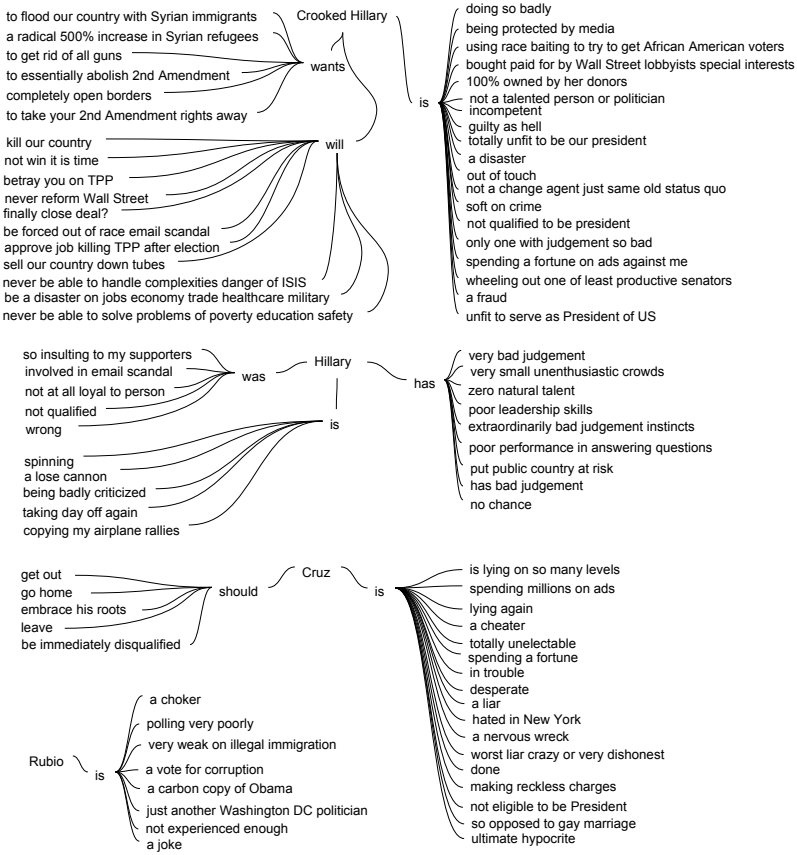

Figure 12: Word trees showing Trump's frequent discourses about Hillary Clinton, Ted Cruz and Marco Rubio.

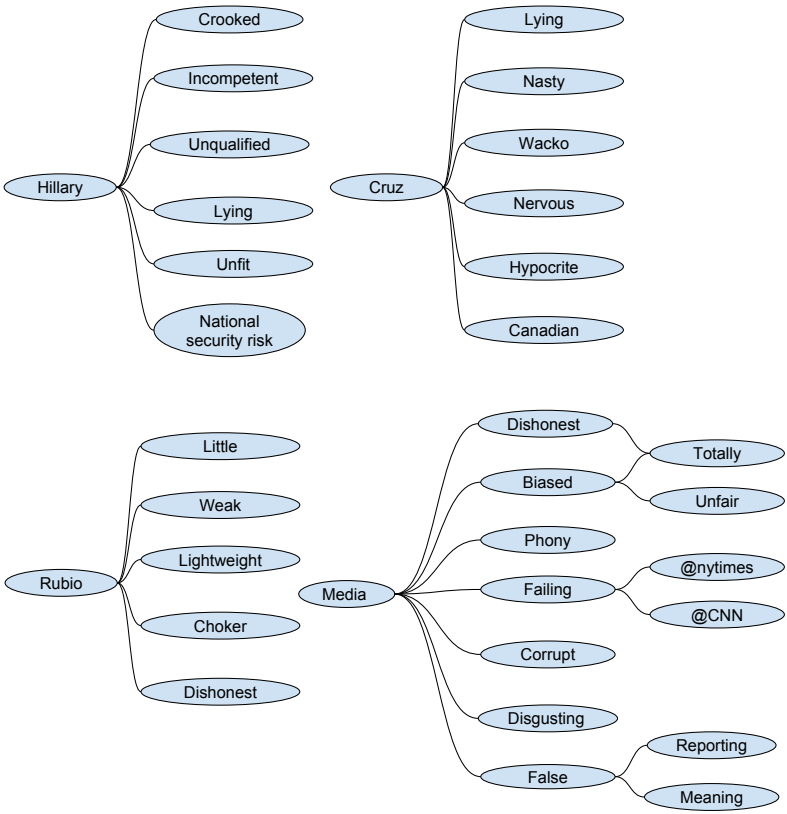

Figure 13: Most frequent words used by Trump to describe the media and his opponents in the mined instances.

To train our FastText model (used to mine SSP instances) we used the Davidson et al. and SemEval dataset. The SSP mining and classification task were performed using the Waseem et al. dataset. To mine SSP instances we used the entire dataset, while for the classification we reserved 10% for validation. To define the keywords used in the SSP mining process, we searched for English words related to racism and sexism in *HateBase.org* and HurtLex (BASSIGNANA; BASILE; PATTI, 2018). We also added words that could indicate that a woman is the target of a given sentence, or that racism/sexism is the subject of the message. These words are listed in Table 5.

Other included keywords
woman, women, girl, girls, female, females, feminist, feminists, fem, fems, sexist, sexists, sexism, feminazi, feminazis, racism, racist

Table 5: Other words regarding women and sexism/racism that were used as keywords.

In total, we selected 275 words referring to racism and 131 words concerning sexism to use as keywords.

5.2.2 Pre-processing

In a first step, we clean the tweets. Each tweet passed trough pre-processing, in order to remove stop words, punctuation, emojis, url and username mentions. We then tokenized and stemmed the cleaned tweets, using NLTK Twitter tokenizer[8] and Porter stemmer[9].

5.2.3 Classification Model

We implemented and tested models for binary and ternary classification. In order to select the best features, first we applied logistic regression with L1 regularization. Afterwards, we tested some models to predict the classes, namely linear SVM, logistic regression, random forests, decision trees and multi-layer perceptron (MLP). We used the scikit-learn[10] library in order to implement and compare models. These models were then tested using 10-fold cross validation, with 10% of the sample reserved for evaluation to avoid over-fitting. We applied a grid-search strategy to compare the models. The logistic regression and the linear SVM achieved the best performances. Due to slightly better results presented by the logistic regression, we selected the logistic regression with L2 regularization to compare the impact of the set of features.

5.2.3.1 Features

We used TF-IDF weighted bigrams, unigrams, trigrams, sequences of up to 3 Part-of-Speech tags. The presence of SSP instances (either fully or partially) was incorporated by using a count vectorizer, i.e. a matrix of

[8] https://www.nltk.org/api/nltk.tokenize.html
[9] https://www.nltk.org/_modules/nltk/stem/porter.html
[10] https://scikit-learn.org/stable/

token counts. For each tweet, we searched for the longest pattern instance fit, being that the tweet must have at least three words in common with the SSP instance to configure a partial match. Negative, positive, neutral and overall sentiment scores for each tweet were measured through VADER sentiment[11]. The number of syllables, words, characters and unique words were also tested as features but didn't impact the model performance. We also used an adapted list of refined n-grams provided by Davidson et al. (DAVIDSON et al., 2017) to check for the presence of expressions that may indicate hate speech, such as *f*cking hate you*.

5.2.4 Results and Discussion

We emphasize that the examples in this work are included to illustrate the severity of the hate speech problem and in no way reflect the opinion of the authors. We extracted SSP instances from the Waseem et al. (WASEEM; HOVY, 2016) dataset with size of at least 3 words for the aforementioned key using two distinct word embedding models. The first model was our custom FastText, trained on the above-mentioned datasets with vectors of 200 dimensions. The second model was a pre-trained GloVe model[12] trained with 2B tweets, also with vectors of 200 dimensions. Through previous empirical tests, we noticed that our method retrieves better quality patterns for t values equal or bigger than 80%. We tuned the t parameter for both the embedding model we trained and the pre-trained, finding that the values $t = 0.86$ and $t = 0.92$ better represent our patterns for the respective models. The quality of the patterns was investigated manually. We set the minimum support value to $minsupp = 5$, therefore patterns with less than 5 instances were discarded due to low support.

5.2.4.1 SSP Concerning Sexism

Using our custom word embeddings model we found a total of 25 distinct SSP, comprising 383 SSP instances. Meanwhile, with the pre-trained GloVe model we found 27 distinct patterns, concerning 1138 instances. The pre-trained model had better performance in the SSP mining strategy, allowing us to retrieve more instances for the same SSP and more diversity of subjects. Due to the better performance of the pre-trained model, we opted for using the instances mined with such model.

The vast majority of instances extracted concerning sexism contained the expressions *'I'm not sexist but'* and *'call me sexist but'* accompanied of

[11] https://github.com/cjhutto/vaderSentiment
[12] https://nlp.stanford.edu/projects/glove/

Figure 14: Word tree from a sample of *'I'm not sexist but'* instances.

sexist comments. The contents of the patterns revealed a great number of things that the users think that women are not suitable for, such as drive, play or commentate on sports, as seen in Figures 14 and 15.

In addition, one mined pattern revealed users talking explicitly about women being inferior, stupid, etc. The following instances are examples of such cases: *{'i'm', 'not', 'sexist', 'but', 'do', 'believe', 'women', 'are', 'inferior'}*, *{'i'm', 'not', 'sexist', 'but', 'out', 'females', 'managers', 'are', 'fucking', 'retarded', 'overall'}*, *{'i'm', 'not', 'sexist', 'but', 'women', 'are', 'really', 'stupid'}*, *{'i'm', 'not', 'sexist', 'but', 'women', 'are', 'seriously', 'awful'}*. Although most of the SSP instances are still those with the expressions *'I'm not sexist but'* and *'call me sexist but'*, we discovered instances with insulting content directed towards female participants of the MRK show, such as in the instances *{'scoring', 'like', 'cunt', 'you', 'can', 'not', 'cook', 'shit', 'fighting', 'hard', 'kat', 'you', 'stupid', 'mole'}*, *{'fucken', 'hate', 'you', 'kat',*

Figure 15: Word tree from a sample of *'Call me sexist but'* instances.

'see', 'you', 'street', "i'm", 'going', 'spit', 'you', 'selfish', 'ugly', 'bibitch'} and *{'mkr', 'dumb', 'blondes', 'pretty', 'faces', 'well', 'you', 'got', 'half', 'right'}*.

We also found 2 and 1 false sexism patterns in the *neither* and *racism* classifications respectively. The pattern found in the *racism* class retrieved instances which talked about prophet Muhammad, from the Islamic religion. The contents depicted the prophet as an immoral and sexist person, as seen in the instance *{'mohammed', 'was', 'pedophile', 'murderer', 'rapist', 'slave', 'trader', 'bigot', 'sexist'}*. Although the pattern presents the accusation of someone being sexist, the content is not sexist by itself. The SSP encountered in the *neither* class are responses to other sexists tweets. For example, the instance *{'you', 'not', 'seriousness', 'tweet', 'can', 'not', 'sexist', 'mother', 'is', 'woman'}* was originated from the response *'You did not just, in all seriousness, tweet "I can't be sexist, my mother is a woman"????'*. Although the subject of the message is sexism, the intended sense of the user was not sexist.

5.2.4.2 SSP Concerning Racism

Again, the pre-trained model achieved better results in retrieving SSP instances. Applying our custom word embeddings model we mined 9 distinct SSP, comprising only 75 SSP instances. In contrast, with the pre-trained GloVe model we found 28 distinct patterns, concerning 468 instances. Despite the low statistic relevance, we found some interesting discourses in the SSP instances.

Most of the contents of the instances referring to racism are forms of Islamophobia. Prophet Muhammad (founder of Islam) is depicted as an criminal and violent person, accused of a murderer and pedophile among others, as seen in the word tree in Figure 16. We found that a number of instances showed this kind of discourse is then further extended to other Islamic organizations and Muslims. Some instances explicitly attributes the guilty of Islam, Muslims and ISIS supposed violent and immoral acts to Muhammad such as in *{'point', 'is', 'nothing', 'isis', 'does', 'mohammed', 'not', 'also', 'do', 'are', 'sick', 'disgusting'}*, *{'following', 'example', 'prophet', 'muslims', 'are', 'still', 'commonly', 'marrying', 'children'}*, *{'fact', 'is', 'mohammed', 'was', 'every', 'bit', 'vile', 'sick', 'isis', 'is'}* and *{'prophet', 'rape', 'slaves', 'like', 'isis', 'yes', 'prophet', 'tell', 'women', 'cover', 'up', 'stay', 'home', 'like', 'isis', 'yes'}*. The word trees in Figures 17 and 18 shows a sample of the statements made about Islam an Muslims. These allegations illustrate seriously violent ideas and hatred incitement against Muslims and Islam, as seen in instances such as *{'islam', 'declared', 'war', 'humanity', 'years', 'ago', 'time', 'respond'}*, *{'are', 'you', 'crying', 'microbrain', 'islam', 'declared', 'war', 'humanity', 'years', 'ago'}*, *{'you', 'are', 'idiot', 'muslims', 'were', 'thugs', 'quickly', 'figured', 'out', 'forced', 'everyone'}* and *{'muslims', 'want', 'exterminate', 'everyone', 'is', 'not', 'muslim', 'are', 'around"}*.

Terrorist accusations were also retrieved in the SSP, such as *{'maybe', 'you', 'are', 'stupid', 'see', 'muslim', 'terrorist', 'palestine', 'murder', 'civilians'}*, *{'probably', 'bcus', 'fools', 'idiots', 'take', 'seriously', 'blame', 'someone', 'else', 'terrorism', 'industry' , 'islam', 'is'}* and *{'muslim', 'terrorists', 'murder', 'people', 'paris', 'terrorist', 'attacks', 'since', 'not', 'aberration', 'real'}*. ISIS is also mentioned in some instances straightforwardly connected to terrorism, as in *{'isis', 'using', 'retarded', 'kids', 'suicide', 'bombers', 'is', 'nothing', 'new'}*, *{'islam', 'wants', 'women', 'stay', 'houses', 'cover', 'isis', 'makes', 'sure'}* and *{['isis', 'mosul', 'executed', 'people', 'wabitching', 'soccer', 'game'}*.

We found a false positive SSP in the *sexism* category. The pattern was generated by the keyword *racist*, comprising 5 instances with contents such as *{'i'm', 'not', 'sexist', 'racist', 'okay', 'maybe'}* and *{'i'm', 'not', 'racist', 'i'm', 'not', 'sexist', 'i'm', 'tired', 'self'}*. There was also one pattern found in the neither class. The contents of the pattern are about terrorism attacks, represented by instances like *{'were', 'struck', 'airstrike', 'local', 'sources',*

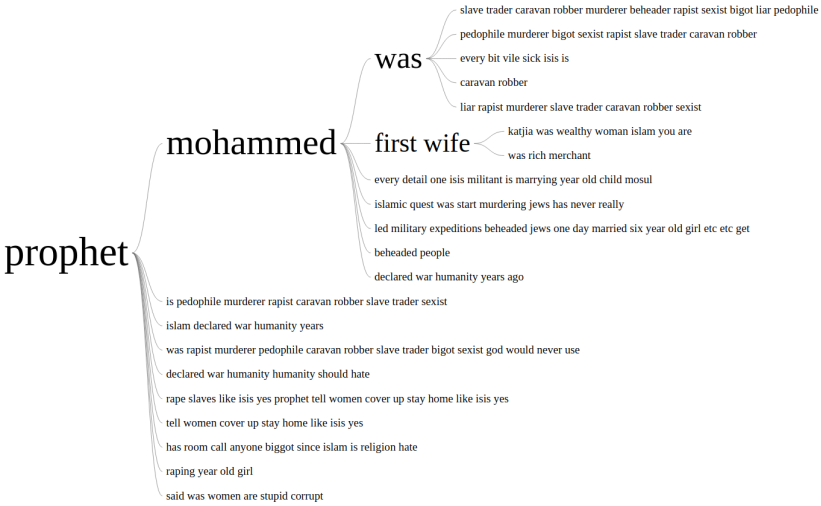

Figure 16: Word tree from a sample of instances referring to prophet Muhammad.

Figure 17: Word tree from a sample of instances referring to Islam.

Figure 18: Word tree from a sample of instances referring to Muslims.

'claims', 'isis', 'terrorist', 'were', 'killed', 'mosul'} and *{'muslims', 'die', 'terrorist', 'attacks', 'anyone', 'else', 'example', 'afghan', 'civilians', 'killed'}*.

5.2.4.3 Classification Results

The overall precision and recall for our best performing model in the binary classification were 79,1% and 80,2%, respectively. The average of 10 executions for the precision, recall and F1-scores for each configuration (SSP + *n*-grams + PoS-tags + sentiment scores and *n*-grams + PoS-tags + sentiment scores) are presented in Table 6. Our model outperformed Waseem et al.(WASEEM; HOVY, 2016) approach best configuration (char *n*-grams + gender: precision 72,9%, recall 77,7% and F1-score 73,9%). Despite the performance gain of adding SSP as features being minimal, in all executions the scores using SSP as a feature were concisely higher. We believe that the marginal gains are due to the fact that we were able to extract few pattern instances compared to the total number of tweets in the dataset.

The confusion matrix in Figure 20 shows that about 25% of hate speech tweets were misclassified. Most of the misclassfication is given by tweets containing hate speech that were predicted as clean. Instances of tweets containing veiled forms of sexism/racism and sarcasm were often wrongly predicted as clean, e.g., *"Deconstructed tart by lazy tarts #MKR"*, *"So Drasko*

SSP	Class	Precision	Recall	F1
Yes	Clean	0.883	0.850	0.867
	Hate Speech	0.699	0.754	0.724
	Overall	**0.791**	**0.802**	**0.796**
No	Clean	0.871	0.842	0.857
	Hate Speech	0.683	0.736	0.707
	Overall	0.777	0.789	0.782

Table 6: Binary classification precision, recall and F1-scores with and without SSP.

just said he was impressed the girls cooked half a chicken.. They cooked a whole one #MKR", *"#MKR #killerblondes MODEL? Puhlease. Did the local thrift shop put on a 'fashion' show. And should we be able to see her bad underwear?"* and *"#bizarresexism @LifeOfStrife I don't know why buy I find a hearty handshake between two women to be very awkward. #notsexist"*. Tweets containing explicit and strong slurs tend to be accurately classified as hate speech, like *"@anjemchoudary Your prophet was a rapist, murderer, pedophile, caravan robber, slave trader, bigot and sexist. God would never use the scum."*, *"BULLSHIT! Kat & Andre you are deadset CUNTS" "These bitches are full fucking disgusting in every conceivable way. #mkr"*.

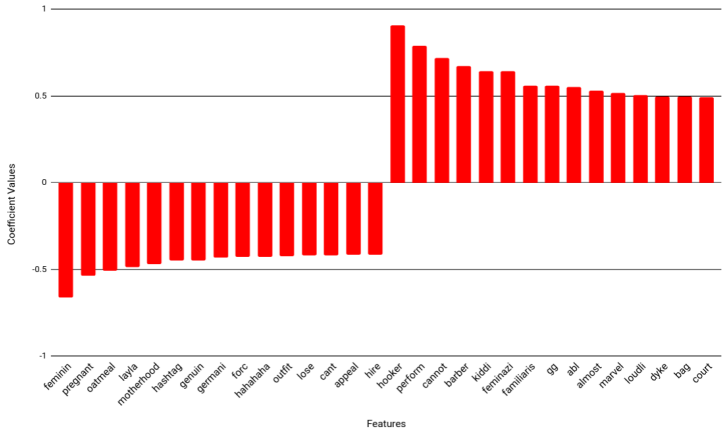

Figure 19: The 30 most important features in binary classification.

Our model also classified certain hate tweets that apparently should be in the hate speech classification, but were classified as clean by the human

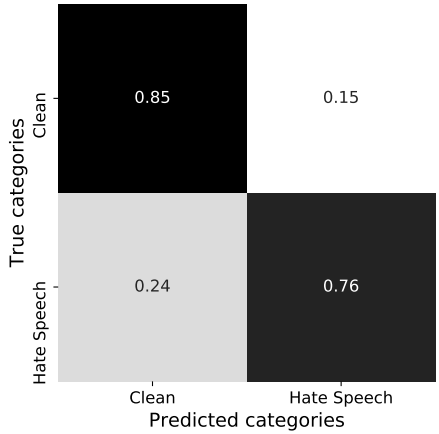

Figure 20: Confusion matrix: true versus predicted categories in binary classification.

coders, such as *"@adelaidemale: Kat = Cunt #mkr"*, *"Umm that's actually so fucked! Kat and Andre suck hair balls they should of gone, they can't cook & they are fucking bitches #mkr"* and *"According to Islam, Muhammad is the perfect man and an example to be followed. He raped, robbed and murdered, whilst sleeping with a six year old kid"*. Figure 19 shows the 30 most important features for the binary classifier, where positive and negative values refer to hate speech and clean outcomes, respectively.

The overall precision and recall of model in the ternary classification were 76,7% and 77,8%, respectively. The average of the precision, recall and F1-scores of 10 executions for each configuration are presented in Table 7. Like in the binary classification, the gains of using SSP as features for ternary classification were marginal. The confusion matrix in Figure 20 shows that most of the misclassification occurs in the lower triangle of the matrix. Almost no tweets that were originally labelled as clean were attributed to the racism and sexism classes. This implies that the classifier tends to classify tweets as less racist/sexist, reaffirming the results obtained in the binary classification.

Concealed cases of sexism and racism such as *"#Muslim #Islam Welcome to the Hotel Islamfornia. May check out any time but never leave."* and *"I can barely watch the #MKR episode of Katie and Nikki or whatever. Like my skin is crawling. They have thattt many tickets on themselves"* persisted being incorrectly classified as clean. Also, tweets that where apparently a

SSP	Class	Precision	Recall	F1
Yes	Clean	0.870	0.874	0.871
	Sexism	0.716	0.684	0.695
	Racism	0.715	0.776	0.744
	Overall	**0.767**	**0.778**	**0.770**
No	Clean	0.861	0.871	0.867
	Sexism	0.717	0.689	0.689
	Racism	0.709	0.761	0.728
	Overall	0.762	0.766	0.761

Table 7: Ternary classification precision, recall and F1-scores with and without SSP.

discussion thread, such as *"@Assiye61 That is a lot of disinformation. For example, there were never more than 5 million North American Indians"* tended to misclassify, except when containing slurs as in *"@ummayman90 Again, your entire concept of god corresponds to a tyrannical earthly egomaniac because you are simple and stupid. #Islam"*. Again, most of correctly predicted tweets contained multiple slurs. However, some tweets that were against sexism/racism, but contained slurs were incorrectly classified, like *"Halloween is a busy day for sexist assholes"*, *"Bitch, whore, slut, cunt, I am sick of these words. Change your speech, change your mind"*. As previously stated, in order to properly predict these situations, the context has to be taken into account.

The majority of tweets containing containing expressions *I'm not sexist but*, *Call me sexist but* and *ISIS/Muslims/Islam declared war on humanity* were correctly classified even in more subtle cases (e.g., *"RT @waken_jake23 I'm not sexist... but seriously if you're female you need to be able to cook. it's in your DNA"*), probably due to a large number of tweets containing similar messages in its own category.

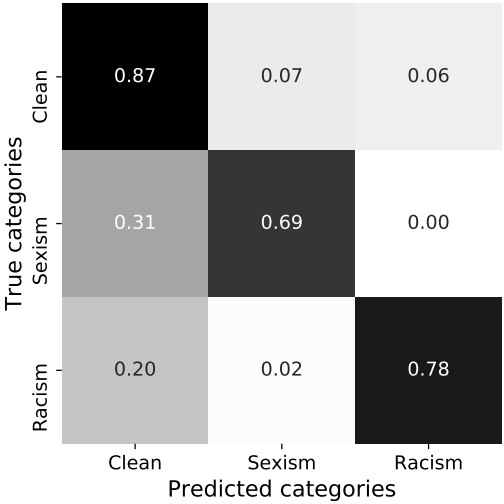

Figure 21: Confusion matrix: true versus predicted categories in ternary classification.

6 CONCLUSION AND FUTURE WORK

In this dissertation we developed an NLP approach to extract recurrent fragments of text that display the same intended meaning, which we called Short Semantic Patterns (SSP). We defined SSP, proposed our approach for SSP mining and analyzed them in two distinct case studies, with real tweets: (i) Donald Trump's tweets posted during the campaign for the 2016 presidential election in the United States and (ii) hate speech detection in tweets, focusing in racism and sexism. Experimental results show patterns containing a high preponderance of some thoughts of Twitter users towards certain groups and expressions that often appear in abusive contents.

The analysis of the SSP instances regarding Donald Trump's tweets showed that his campaign strategy consisted in systematically defame the media and their opponents. In the discourses propagated via tweets, he tries to portray the media and his adversaries as incompetent and not trustworthy. Thus, he arises as the only viable solution and source of correct information. The analysis of the SSP instances found in tweets regarding sexism revealed that a large number of sexist tweets began with the introduction *'I'm not sexist but'* and *'Call me sexist but'*. Meanwhile, SSP instances found in tweets regarding racism revealed a prominence of discourses against the Islamic religion, associated entities and organizations. The SSP pattern mining technique was useful to extract recurrent discourses that appeared in the dataset without having to revise manually the whole dataset. We analyzed discourses against groups such as women and Muslims and Trump's predominant discourses that concerning his adversaries and the media.

We also experimented using features extracted from the mined patterns for binary and ternary classification of text documents, using a number of different classifiers. Despite the marginal gains of performance obtained by using the SSP features for the classifiers, we did notice consistent improvements. We intend to investigate if the expansion of keywords would lead to the discovery of more SSP instances, to increase the statistical relevance of the patterns. We believe that an increase of the number of SSP could have more impact in the classifier results.

The scientific contributions resulting from this work are: (i) a formal definition of SSP; (ii) an algorithm to mine SSP instances in short textual documents, such as microblogs posts; (iii) a comprehensive analysis of the mined SSP in both case studies; (iv) binary classifier to detect hate speech in tweets; and (v) ternary classifier that distinguishes between sexist, racist and clean tweets.

The proposed method still suffers from certain limitations. There is a granularity restriction, since the suspicious documents are matched though

predefined keywords. Also, we still do not have a method to validate the mined patterns, however a human user can easily validate the mined patterns as relevant linguistic patterns. Furthermore, we don't have an automatic way to measure optimal values for the semantic similarity threshold and minimum support parameter. Due to this, these parameters must be adjusted manually through empirical observation. Additionally, it's necessary to evaluate if treating compound words (e.g. São Paulo), which aren't in this work, would cause a beneficial impact in instances mining.

As future works, we aim to create heuristics to extract keywords from a given dataset dynamically. We also believe that integrating other techniques such as semantic expansion of keywords and character *n*-grams in the document identification phase would be beneficial for better identification of documents containing suspicious content. In addition, also aiming better results, we plan to investigate semi-automatic ways to validate the mined patterns and eliminate badly matched instances. We also aim to investigate other feature designs that can be incorporated to take advantage of SSP in the classifier. We believe that our mining technique is applicable to other domains as well, by replacing the keywords used in our case studies with words regarding the desired domain (e.g., diseases and symptoms, products, drug names) and intend to further investigate this feature. Beyond that, we also want to evaluate the applicability of SSP to built pipelines that feed the mined patterns to solve other linguistic tasks, such as Word Sense Induction.

BIBLIOGRAPHY

BAHDANAU, Dzmitry; CHO, Kyunghyun; BENGIO, Yoshua. Neural machine translation by jointly learning to align and translate. **arXiv preprint arXiv:1409.0473**, 2014.

BANERJEE, Satanjeev; PEDERSEN, Ted. An adapted Lesk algorithm for word sense disambiguation using WordNet. In: SPRINGER. INTERNATIONAL Conference on Intelligent Text Processing and Computational Linguistics. 2002. pp. 136–145.

BASILE, Valerio et al. SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter. In: PROCEEDINGS of the 13th International Workshop on Semantic Evaluation (SemEval-2019). Association for Computational Linguistics", location = "Minneapolis, Minnesota, 2019.

BASSIGNANA, Elisa; BASILE, Valerio; PATTI, Viviana. Hurtlex: A multilingual lexicon of words to hurt. In: CEUR-WS. 5TH Italian Conference on Computational Linguistics, CLiC-it 2018. 2018. vol. 2253, pp. 1–6.

BÉCHET, Nicolas et al. Discovering linguistic patterns using sequence mining. In: SPRINGER. INTERNATIONAL Conference on Intelligent Text Processing and Computational Linguistics. 2012. pp. 154–165.

BERLAND, Matthew; CHARNIAK, Eugene. Finding parts in very large corpora. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. PROCEEDINGS of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics. 1999. pp. 57–64.

BOJANOWSKI, Piotr et al. Enriching word vectors with subword information. **arXiv preprint arXiv:1607.04606**, 2016.

BOLLACKER, Kurt et al. Freebase: a collaboratively created graph database for structuring human knowledge. In: ACM. PROCEEDINGS of the 2008 ACM SIGMOD international conference on Management of data. 2008. pp. 1247–1250.

BORDES, Antoine; CHOPRA, Sumit; WESTON, Jason. Question answering with subgraph embeddings. **arXiv preprint arXiv:1406.3676**, 2014.

BRAUD, Chloé; DENIS, Pascal. Comparing word representations for implicit discourse relation classification. In: PROCEEDINGS of the 2015 Conference on Empirical Methods in Natural Language Processing. 2015. pp. 2201–2211.

BRODY, Samuel; LAPATA, Mirella. Bayesian word sense induction. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. PROCEEDINGS of the 12th Conference of the European Chapter of the Association for Computational Linguistics. 2009. pp. 103–111.

CARLSON, Andrew et al. Coupled semi-supervised learning for information extraction. In: ACM. PROCEEDINGS of the third ACM international conference on Web search and data mining. 2010. pp. 101–110.

CARPUAT, Marine; WU, Dekai. Word sense disambiguation vs. statistical machine translation. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. PROCEEDINGS of the 43rd Annual Meeting on Association for Computational Linguistics. 2005. pp. 387–394.

CHEN, Jifan et al. Implicit discourse relation detection via a deep architecture with gated relevance network. In: PROCEEDINGS of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016. vol. 1, pp. 1726–1735.

CONROY, Niall J; RUBIN, Victoria L; CHEN, Yimin. Automatic deception detection: Methods for finding fake news. In: AMERICAN SOCIETY FOR INFORMATION SCIENCE. PROCEEDINGS of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community. 2015. p. 82.

DAVIDSON, Thomas et al. Automated hate speech detection and the problem of offensive language. **arXiv preprint arXiv:1703.04009**, 2017.

DINAKAR, Karthik et al. Common sense reasoning for detection, prevention, and mitigation of cyberbullying. **ACM Transactions on Interactive Intelligent Systems (TiiS)**, ACM, vol. 2, no. 3, p. 18, 2012.

DJURIC, Nemanja et al. Hate speech detection with comment embeddings. In: ACM. PROCEEDINGS of the 24th international conference on world wide web. 2015. pp. 29–30.

DRAGONI, Mauro; PETRUCCI, Giulio. A neural word embeddings approach for multi-domain sentiment analysis. **IEEE Transactions on Affective Computing**, IEEE, vol. 8, no. 4, pp. 457–470, 2017.

ELSNER, Micha; CHARNIAK, Eugene; JOHNSON, Mark. Structured generative models for unsupervised named-entity clustering. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. PROCEEDINGS of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. 2009. pp. 164–172.

ETZIONI, Oren et al. Unsupervised named-entity extraction from the web: An experimental study. **Artificial intelligence**, Elsevier, vol. 165, no. 1, pp. 91–134, 2005.

FASOLD, Ralph W; CONNOR-LINTON, Jeff. **An introduction to language and linguistics**. Cambridge University Press, 2014.

GITARI, Njagi Dennis et al. A lexicon-based approach for hate speech detection. **International Journal of Multimedia and Ubiquitous Engineering**, vol. 10, no. 4, pp. 215–230, 2015.

GUARINO, Nicola. **Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy**. IOS press, 1998. vol. 46.

HEARST, Marti A. Automatic acquisition of hyponyms from large text corpora. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. PROCEEDINGS of the 14th conference on Computational linguistics-Volume 2. 1992. pp. 539–545.

HILL, Felix; REICHART, Roi; KORHONEN, Anna. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. **Computational Linguistics**, MIT Press, vol. 41, no. 4, pp. 665–695, 2015.

HO, Tin Kam; HULL, Jonathan J.; SRIHARI, Sargur N. Decision combination in multiple classifier systems. **IEEE Transactions on Pattern Analysis & Machine Intelligence**, IEEE, no. 1, pp. 66–75, 1994.

IACOBACCI, Ignacio; PILEHVAR, Mohammad Taher; NAVIGLI, Roberto. Embeddings for word sense disambiguation: An evaluation study. In: PROCEEDINGS of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016. vol. 1, pp. 897–907.

_____. Sensembed: Learning sense embeddings for word and relational similarity. In: PROCEEDINGS of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2015. vol. 1, pp. 95–105.

JOULIN, Armand et al. Bag of tricks for efficient text classification. **arXiv preprint arXiv:1607.01759**, 2016.

JURAFSKY, Dan; MARTIN, James H. **Speech and language processing**. Pearson London, 2014. vol. 3.

KETTREY, Heather Hensman; LASTER, Whitney Nicole. Staking territory in the "World White Web" an exploration of the roles of overt and color-blind racism in maintaining racial boundaries on a popular web site. **Social Currents**, SAGE Publications Sage CA: Los Angeles, CA, vol. 1, no. 3, pp. 257–274, 2014.

KHAN, Aamera ZH; ATIQUE, Mohammad; THAKARE, VM. Combining lexicon-based and learning-based methods for Twitter sentiment analysis. **International Journal of Electronics, Communication and Soft Computing Science & Engineering (IJECSCSE)**, International Journal of Electronics, Communication, Soft Computing Science, and Engineering, p. 89, 2015.

KLEMENTIEV, Alexandre; ROTH, Dan. Weakly supervised named entity transliteration and discovery from multilingual comparable corpora. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. PROCEEDINGS of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics. 2006. pp. 817–824.

LAMPLE, Guillaume et al. Phrase-based & neural unsupervised machine translation. **arXiv preprint arXiv:1804.07755**, 2018.

LANDAUER, Thomas K; DUMAIS, Susan T. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. **Psychological review**, US: American Psychological Association, vol. 104, no. 2, p. 211, 1997.

LEI, Wenqiang et al. SWIM: A Simple Word Interaction Model for Implicit Discourse Relation Recognition. In: IJCAI. 2017. pp. 4026–4032.

LESK, Michael. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In: ACM. PROCEEDINGS of the 5th annual international conference on Systems documentation. 1986. pp. 24–26.

LEVELT, Pim; CARAMAZZA, Alfonso. **The Oxford handbook of psycholinguistics**. Oxford University Press, USA, 2007.

LIN, Dekang et al. Identifying synonyms among distributionally similar words. In: IJCAI. 2003. vol. 3, pp. 1492–1493.

LIU, Yefeng; ALEXANDROVA, Todorka; NAKAJIMA, Tatsuo. Using stranger as sensors: temporal and geo-sensitive question answering via social media. In: ACM. PROCEEDINGS of the 22nd international conference on World Wide Web. 2013. pp. 803–814.

LOH, Kep Kee; KANAI, Ryota. How has the Internet reshaped human cognition? **The Neuroscientist**, Sage Publications Sage CA: Los Angeles, CA, vol. 22, no. 5, pp. 506–520, 2016.

LU, Chunliang; LAM, Wai; ZHANG, Yingxiao. Twitter user modeling and tweets recommendation based on wikipedia concept graph. In: WORKSHOPS at the Twenty-Sixth AAAI Conference on Artificial Intelligence. 2012. pp. 33–38.

MANANDHAR, Suresh et al. SemEval-2010 task 14: Word sense induction & disambiguation. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. PROCEEDINGS of the 5th international workshop on semantic evaluation. 2010. pp. 63–68.

MCCALLUM, Andrew; LI, Wei. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. PROCEEDINGS of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4. 2003. pp. 188–191.

MEEHAN, Kevin et al. Context-aware intelligent recommendation system for tourism. In: IEEE. PERVASIVE Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on. 2013. pp. 328–331.

MIHALCEA, Rada; MOLDOVAN, Dan I. A method for word sense disambiguation of unrestricted text. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. PROCEEDINGS of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics. 1999. pp. 152–158.

MIKOLOV, Tomas; CHEN, Kai, et al. Efficient estimation of word representations in vector space. **arXiv preprint arXiv:1301.3781**, 2013.

MIKOLOV, Tomas; YIH, Wen-tau; ZWEIG, Geoffrey. Linguistic regularities in continuous space word representations. In: PROCEEDINGS of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2013. pp. 746–751.

MONDAL, Mainack; SILVA, Leandro Araújo; BENEVENUTO, Fabrício. A measurement study of hate speech in social media. In: ACM. PROCEEDINGS of the 28th ACM Conference on Hypertext and Social Media. 2017. pp. 85–94.

MONTEIRO, Rafael A et al. Contributions to the Study of Fake News in Portuguese: New Corpus and Automatic Detection Results. In: SPRINGER. INTERNATIONAL Conference on Computational Processing of the Portuguese Language. 2018. pp. 324–334.

NAVIGLI, Roberto et al. Two birds with one stone: learning semantic models for text categorization and word sense disambiguation. In: ACM. PROCEEDINGS of the 20th ACM international conference on Information and knowledge management. 2011. pp. 2317–2320.

NOCKLEBY, John T. Hate speech. **Encyclopedia of the American constitution**, Detroit: Macmillan Reference USA, vol. 3, pp. 1277–79, 2000.

O'CONNOR, Brendan et al. From tweets to polls: Linking text sentiment to public opinion time series. **Icwsm**, vol. 11, no. 122-129, pp. 1–2, 2010.

OTT, Brian L. The age of Twitter: Donald J. Trump and the politics of debasement. **Critical Studies in Media Communication**, Routledge, vol. 34, no. 1, pp. 59–68, 2017.

PAŞCA, Marius. Organizing and searching the world wide web of facts–step two: harnessing the wisdom of the crowds. In: ACM. PROCEEDINGS of the 16th international conference on World Wide Web. 2007. pp. 101–110.

PELLETIER, Francis Jeffry. The principle of semantic compositionality. **Topoi**, Springer, vol. 13, no. 1, pp. 11–24, 1994.

PENNINGTON, Jeffrey; SOCHER, Richard; MANNING, Christopher. Glove: Global vectors for word representation. In: PROCEEDINGS of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014. pp. 1532–1543.

RITTER, Alan; CLARK, Sam; ETZIONI, Oren, et al. Named entity recognition in tweets: an experimental study. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. PROCEEDINGS of the conference on empirical methods in natural language processing. 2011. pp. 1524–1534.

ROTHE, Sascha; SCHÜTZE, Hinrich. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. **arXiv preprint arXiv:1507.01127**, 2015.

SCHMIDT, Anna; WIEGAND, Michael. A survey on hate speech detection using natural language processing. In: PROCEEDINGS of the Fifth International Workshop on Natural Language Processing for Social Media. 2017. pp. 1–10.

SCHWARTZ, Roy; REICHART, Roi; RAPPOPORT, Ari. Symmetric pattern based word embeddings for improved word similarity prediction. In: PROCEEDINGS of the Nineteenth Conference on Computational Natural Language Learning. 2015. pp. 258–267.

SIDOROV, Grigori et al. Empirical study of machine learning based approach for opinion mining in tweets. In: SPRINGER. MEXICAN international conference on Artificial intelligence. 2012. pp. 1–14.

SINHA, Ravi; MIHALCEA, Rada. Unsupervised graph-basedword sense disambiguation using measures of word semantic similarity. In: IEEE. SEMANTIC Computing, 2007. ICSC 2007. International Conference on. 2007. pp. 363–369.

SORATO, Danielly et al. Seleção e Avaliação Experimental de Ferramentas para Anotação Morfossintática Automática. Florianópolis, SC., 2016.

SPERTUS, Ellen. Smokey: Automatic recognition of hostile messages. In: AAAI/IAAI. 1997. pp. 1058–1065.

STIEGLITZ, Stefan; DANG-XUAN, Linh. Emotions and information diffusion in social media—sentiment of microblogs and sharing behavior. **Journal of management information systems**, Taylor & Francis, vol. 29, no. 4, pp. 217–248, 2013.

TAGHIPOUR, Kaveh; NG, Hwee Tou. Semi-supervised word sense disambiguation using word embeddings in general and specific domains. In: PROCEEDINGS of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2015. pp. 314–323.

TANG, Duyu et al. Sentiment embeddings with applications to sentiment analysis. **IEEE Transactions on Knowledge and Data Engineering**, IEEE, vol. 28, no. 2, pp. 496–509, 2016.

TANNEN, Deborah; HAMILTON, Heidi E; SCHIFFRIN, Deborah. **The handbook of discourse analysis**. John Wiley & Sons, 2015.

TJONG KIM SANG, Erik F; DE MEULDER, Fien. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. PROCEEDINGS of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4. 2003. pp. 142–147.

TOUTANOVA, Kristina et al. Feature-rich part-of-speech tagging with a cyclic dependency network. 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. 2003. pp. 173–180.

TOVAR, Mireya et al. Use of lexico-syntactic patterns for the evaluation of taxonomic relations. In: SPRINGER. MEXICAN Conference on Pattern Recognition. 2014. pp. 331–340.

TRASK, Andrew; MICHALAK, Phil; LIU, John. sense2vec-A fast and accurate method for word sense disambiguation in neural word embeddings. **arXiv preprint arXiv:1511.06388**, 2015.

VASILESCU, Florentina; LANGLAIS, Philippe; LAPALME, Guy. Evaluating Variants of the Lesk Approach for Disambiguating Words. In: LREC. 2004.

VÉRONIS, Jean. Hyperlex: lexical cartography for information retrieval. **Computer Speech & Language**, Elsevier, vol. 18, no. 3, pp. 223–252, 2004.

VICKREY, David et al. Word-sense disambiguation for machine translation. In: PROCEEDINGS of human language technology conference and conference on empirical methods in natural language processing. 2005.

VOLKOVA, Svitlana et al. Boosting biomedical entity extraction by using syntactic patterns for semantic relation discovery. In: IEEE. WEB Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on. 2010. vol. 1, pp. 272–278.

VOORHEES, Ellen M. Using WordNet to disambiguate word senses for text retrieval. In: ACM. PROCEEDINGS of the 16th annual international ACM SIGIR conference on Research and development in information retrieval. 1993. pp. 171–180.

VOUTILAINEN, Atro. Part-of-speech tagging. **The Oxford handbook of computational linguistics**, Oxford University Press Oxford, pp. 219–232, 2003.

WANG, Hao et al. A system for real-time twitter sentiment analysis of 2012 us presidential election cycle. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. PROCEEDINGS of the ACL 2012 System Demonstrations. 2012. pp. 115–120.

WANG, Zhen et al. Knowledge graph and text jointly embedding. In: PROCEEDINGS of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014. pp. 1591–1601.

WASEEM, Zeerak; HOVY, Dirk. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In: PROCEEDINGS of the NAACL student research workshop. 2016. pp. 88–93.

WATANABE, Hajime; BOUAZIZI, Mondher; OHTSUKI, Tomoaki. Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection. **IEEE Access**, IEEE, vol. 6, pp. 13825–13835, 2018.

WU, Changxing et al. Improving implicit discourse relation recognition with discourse-specific word embeddings. In: PROCEEDINGS of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2017. vol. 2, pp. 269–274.

XU, Huijuan; SAENKO, Kate. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In: SPRINGER. EUROPEAN Conference on Computer Vision. 2016. pp. 451–466.

YU, Liang-Chih et al. Refining word embeddings for sentiment analysis. In: PROCEEDINGS of the 2017 Conference on Empirical Methods in Natural Language Processing. 2017. pp. 534–539.

YU, Liyang. **A developer's guide to the semantic Web**. Springer Science & Business Media, 2011.

ZHONG, Haoti et al. Content-Driven Detection of Cyberbullying on the Instagram Social Network. In: IJCAI. 2016. pp. 3952–3958.

ZHOU, Guangyou et al. Learning continuous word embedding with metadata for question retrieval in community question answering. In: PROCEEDINGS of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2015. vol. 1, pp. 250–259.

ZOU, Will Y et al. Bilingual word embeddings for phrase-based machine translation. In: PROCEEDINGS of the 2013 Conference on Empirical Methods in Natural Language Processing. 2013. pp. 1393–1398.

ZOUAQ, Amal; GASEVIC, Dragan; HATALA, Marek. Linguistic patterns for information extraction in ontocmaps. In: CEUR-WS. ORG. PROCEEDINGS of the 3rd International Conference on Ontology Patterns-Volume 929. 2012. pp. 61–72.

_____. Towards open ontology learning and filtering. **Information Systems**, Elsevier, vol. 36, no. 7, pp. 1064–1081, 2011.