Eduardo Santos Duarte

Licenciado em Engenharia Informática

Sentiment Analysis on Twitter for the Portuguese Language

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática

Orientador : Carlos Viegas Damásio, Prof. Associado,
Universidade Nova de Lisboa

Co-orientador : João Gouveia, CTO,
AnubisNetworks

Júri:

Presidente:	Pedro Abílio Duarte de Medeiros
Arguente:	José Miguel Gomes Saias
Vogal:	Carlos Viegas Damásio

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Dezembro, 2013

Sentiment Analysis on Twitter for the Portuguese Language

Copyright © Eduardo Santos Duarte, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Acknowledgements

First off, i would like to thank my thesis adviser, Prof. Carlos Viegas Damásio, for not only providing with this opportunity but also for introducing me into the interesting research area that is the semantic web.
I would like to thank my co-adviser, the CEO of *AnubisNetworks* João Gouveia, for the opportunity in working closer to the industry and on an interesting subject.
I would like to thank the whole *AnubisNetworks* team for all the support provided.
A special thank you to João Moura, for all the insightful discussions, thoughts and help in the development of this thesis.
Thank you every one else that helped and provided continuous support.

Abstract

With the growth and popularity of the internet and more specifically of social networks, users can more easily share their thoughts, insights and experiences with others. Messages shared via social networks provide useful information for several applications, such as monitoring specific targets for sentiment or comparing the public sentiment on several targets, avoiding the traditional marketing research method with the use of surveys to explicitly get the public opinion. To extract information from the large amounts of messages that are shared, it is best to use an automated program to process these messages.

Sentiment analysis is an automated process to determine the sentiment expressed in natural language in text. Sentiment is a broad term, but here we are focussed in opinions and emotions that are expressed in text. Nowadays, out of the existing social network websites, *Twitter* is considered the best one for this kind of analysis. *Twitter* allows users to share their opinion on several topics and entities, by means of short messages. The messages may be malformed and contain spelling errors, therefore some treatment of the text may be necessary before the analysis, such as spell checks.

To know what the message is focusing on it is necessary to find these entities on the text such as people, locations, organizations, products, etc. and then analyse the rest of the text and obtain what is said about that specific entity. With the analysis of several messages, we can have a general idea on what the public thinks regarding many different entities.

It is our goal to extract as much information concerning different entities from tweets in the Portuguese language. Here it is shown different techniques that may be used as well as examples and results on state-of-the-art related work. Using a semantic approach, from these messages we were able to find and extract named entities and assigning sentiment values for each found entity, producing a complete tool competitive with existing solutions. The sentiment classification and assigning to entities is based on the grammatical construction of the message. These results are then used to be viewed by the user in real time or stored to be viewed latter. This analysis provides ways to view and compare the public sentiment regarding these entities, showing the favourite brands, companies and people, as well as showing the growth of the sentiment over time.

Keywords: sentiment analysis, named entity recognition, opinion mining, semantic analysis, social knowledge

Resumo

Com o crescimento e a popularidade da internet e especificamente das redes sociais, utilizadores podem com facilidade partilhar os seus pensamentos, opiniões e experiências. Mensagens partilhadas pelas redes sociais contêm informação útil para diversas aplicações como monitorizar o sentimento de uma entidade específica ou comparar o sentimento público entre alvos, evitando o uso do modo tradicional para obter informação com o uso de inquéritos que pedem explicitamente a opinião ao público. Para extrair informação da grande quantidade de mensagens partilhadas, o melhor método é criar um mecanismo automático para processar estas mensagens.

A análise de sentimento é um processo automático para determinar o sentimento expresso em linguagem natural em texto. Sentimento é um termo vago, mas aqui referimos sentimento como opinião e emoção que estão expressas no texto. Das redes sociais existentes, o *Twitter* é das melhores plataformas para este tipo de análise. Utilizadores enviam e partilham sentimento sobre diversos temas e entidades, por vias de texto curto. Sendo que o texto pode ter erros gramaticais, algum tratamento do texto será necessário para que seja bem analisado.

Para determinar o alvo da mensagem primeiro temos de marcar as entidades que reconhecemos, pessoas, locais, organizações, produtos, etc., e analisamos a relação que essa entidade tem com o restante texto, atribuindo os sentimentos que esse texto contem à respectiva entidade. Com a análise de várias mensagens, podemos verificar o sentimento geral que o público pensa sobre diversas entidades.

O nosso objectivo é extrair toda a informação referente a diversas entidades de tweets em Português. Aqui serão apresentadas diferentes técnicas que podem ser usadas, alguns exemplos e resultados de trabalho relevante de ponta. Usando uma abordagem semântica, conseguimos extrair entidades relevantes da mensagem e atribuir um valor de sentimento para cada entidade encontrada, obtendo uma ferramenta completa que compete com soluções existentes. A classificação de sentimento e a atribuição a entidades tem em conta a construção gramatical da mensagem. Estes resultados são depois usados para serem visualizados em tempo real ou guardados para serem consultados mais tarde.

Esta análise fornece maneiras de visualizar e comparar o sentimento do público em relação a entidades, mostrando as preferências de marcas, empresas e pessoas, bem como mostrar o crescimento do sentimento ao longo do tempo.

Palavras-chave: análise de sentimento, reconhecimento de entidades referidas, extração de opinião, análise semântica, conhecimento social

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Background 2

1.2.1 What is Sentiment Analysis 2

1.2.2 Importance of Information 2

1.2.3 Semantic Web 3

1.2.4 Named Entity Recognition 4

1.2.5 Twitter 5

1.3 Project Integration 5

1.4 Document Structure 6

2 The Problem, Approach and Applications **7**

2.1 Problem Formulation 7

2.2 Approach 8

2.3 Application and Practical Usage 10

2.3.1 Entity Monitoring 10

2.3.2 Pooling 10

3 Related Work and State of the Art **11**

3.1 Computational Linguistics 11

3.1.1 Summarization 11

3.1.2 Language Identification 12

3.1.3 Acronyms and Other Abbreviations 12

3.1.4 Parser and Part Of Speech Tagger 12

3.1.5 Entity Recognition and Extraction 13

3.2 Sentiment Classification 14

3.2.1 Machine Learning Classification 15

3.2.2 Semantic Classification 17

3.3 Domains for Sentiment Analysis 19

3.4 Conclusions 20

4 Tools 21

4.1 Twitter API 21
4.2 Stanford Parser 22
4.3 SentiWordNet 23
4.4 DBpedia 24
4.5 Apache Jena 25
4.6 Hunspell 25
4.7 Translator 25
4.8 GSON - Java JSON Library 25

5 Implementation 27

5.1 Twitter Input Stream 29
5.2 Twitter Information Extraction 29
    5.2.1 UserTags 30
    5.2.2 HashTags 31
    5.2.3 URLs 31
    5.2.4 Emotion Icons 32
5.3 Build Parse Tree 33
5.4 Spell Checking 34
    5.4.1 Soundex 35
    5.4.2 Levenshtein Distance 36
5.5 Entity Recognition 37
    5.5.1 Entity Enumeration 38
5.6 Entity Disambiguation 38
    5.6.1 HashTag Map 39
    5.6.2 Property Similarity 39
5.7 Sentiment Classification 40
    5.7.1 Sentiment Value 41
    5.7.2 Sentiment Negations 42
    5.7.3 Sentiment Intensifiers 43
5.8 Sentiment Assigning 45
5.9 Entity Sentiment 46
5.10 Thread Pool 48
5.11 Conclusions 48

6 Evaluation and Results 49

6.1 Entity Recognition 49
    6.1.1 Making Sense of Microposts #MSM2013 49
6.2 Sentiment Particle Classification 54
    6.2.1 SentiLex-PT01 54

6.2.2 SentiLex-PT02 56
6.3 Sentiment Classification on Messages 58
6.3.1 Sentiment Classification on a Random Sample of Tweets 58
6.4 Sentiment Classification on Political Tweets 60
6.4.1 SentiTuites-01 60
6.4.2 SentiCorpus-PT 63
6.5 SemEval2013 - Sentiment Analysis in Twitter 66
6.5.1 Results 67
6.6 Main problems 71
6.6.1 Entity Recognition 71
6.6.2 Sentiment Classification 72

7 Conclusions 75
7.1 Main Contributions 75
7.1.1 Adapting for multi-language sentiment analysis 76
7.2 Future Work 77
7.2.1 Viewing Data using StreamForce 77
7.2.2 Keyword Sentiment 79
7.2.3 Topic Cloud 80
7.2.4 Entity Type Card 81

A Random Sampling of Tweets with Sentence Sentiment and Entity Sentiment 93
A.1 Correct classifications 93
A.1.1 Full agreement between the annotators 93
A.1.2 Agreement with two annotators 98
A.1.3 No Agreement 99
A.2 Incorrect Classifications 99
A.2.1 Full agreement between the annotators 99
A.2.2 Agreement with two annotators 102
A.2.3 No Agreement 104

1

Introduction

Micro-blogging and social networks are a vast and new way for billions of users to communicate and are becoming more popular every day. Users send messages about their every day life, discussing and sharing their opinions and emotions on several topics, such as opinions on products, services, religious and political views.

1.1 Motivation

With the existence of social websites, users can send and share messages that express sentiment and influence others. With the growth of these social websites, so has grown the attention given to information extraction on these messages. Millions of messages are sent every day, written in multiple languages, messages containing useful information that could be used for much more than just communication. By annotating these messages through an automatic process, statistics with the information contained in the messages can be highly useful. These messages contain personal opinions and emotions on specific topics and entities.

While most messages are written in the English language, other languages are also a big part of the social networks population. Since most of the tools are for the English language, it is important to build a model that could be expanded for other languages to process this kind of information.

Our focus will be on *Twitter* messages in the Portuguese language. Portuguese is the third most spoken language on *Twitter*, right after English and Japanese [1]. Even though Portuguese is the third most spoken language on *Twitter*, Brazil is the second country

---

[1] `http://semiocast.com/publications/2011_11_24_Arabic_highest_growth_on_Twitter`

with more *Twitter* accounts, right after the USA [2].

1.2 Background

1.2.1 What is Sentiment Analysis

Sentiment analysis is a technique used to automatically extract and determine the sentiment expressed in natural language. The term *sentiment* refers to feelings or emotions as well as senses, such as hearing, sight, touch, smell and taste. What we want to extract from the messages shared via social websites is the sentiment on emotions or opinions, expressed either as positive, negative or neutral sentiment.

Sentiment analysis is a *Natural Language Processing* (NLP) problem, dealing with message parsing, *Part Of Speech* (POS) tagging, negation and intensification handling. The *POS* tagging will give the grammatical use of words that exist in sentences, using this POS tags we can identify nouns, verbs, adjective, adverbs, etc. Negations and intensifications are features that will affect the sentiment results. These will be explained in detail further in this document. Sentiment analysis is also know as *Opinion Mining* or *Sentiment Classification*, but *Subjectivity analysis*, *review mining* and *appraisal extraction* have also been used in the literature.

Sentiment analysis has different scopes that can be analysed, varying between document level, sentence level or word level. Analysing sentiment on the word level, will simply check the polarity of that specific word, checking if it contains positive, negative or neutral sentiment. On the sentence level it will be taken into account not only the polarity of the words it contains, but will also take into consideration the relations between these words and their grammatical use. These will compose the sentiment value of the sentence. On the document level takes into consideration the full context of the document, leading to a more complex analysis on how the sentences interact with each other [Liu12].

Since our focus is the social website *Twitter*, which messages typically have no more than two sentences, our scope will be at the sentence level. A sentiment classifier is used to get the sentiment value of specific parts of the text. Some classifiers may be trained for specific topics or domains for more accurate results.

1.2.2 Importance of Information

Information is critical for every field, being information on individuals, groups, organizations, methods, etc. Information is the key component to make informed decisions and choices.

Getting information about the public's opinion can be very useful, some examples are, the general opinion of a person, the opinion on places to visit and opinion on the

---

[2] `http://semiocast.com/publications/2012_01_31_Brazil_becomes_2nd_country_on_Twitter_superseds_Japan`

best countries to travel to. The opinion on a company can help stock market investors and opinion on brands and products can help undecided customers on what they should buy or consider, and could also help marketeers view the effectiveness of their campaigns.

1.2.3 Semantic Web

Tim Berners-Lee, british engineer, inventor of the *World Wide Web* and director of the *World Wide Web Consortium* (W3C)[3], is the man that proposed the *semantic web* in an article published in the *Scientific American Magazine* [BLHL01].

The semantic web aims to be an extension of the present web, giving meaning and structure to the already available information, allowing access to users and computers to read and process this information. To develop knowledge sources and adapt existing web pages toward the semantic web, the main technologies used are *eXtensible Markup Language* (XML), *Resource Description Framework* (RDF) [LS99] and *Web Ontology Language* (OWL) [AAAHH03]. XML allows annotations on existing web pages or sections giving them structure and allows to add meaning by using RDF to describe the data. *OWL* is a language used to information expressed by graphs, known as ontologies, formed by RDF triples, composed of a subject, a predicate and an object, which are identified using an *Uniform Resource Identifier* (URI). The predicate represents the relation between the subject and the object in the triple. A *Uniform Resource Locator* (URL) is the most common type of URI, used to locate web pages. Information on the internet is mostly posted by humans and processed by humans, and this information is not easily readable by machines. The use of these technologies will lead to a structured formatted page, easily readable by machines as well as humans.

For example, if we have the information that 'John Lennon' is part of the 'Beatles' and that the 'Beatles' have a song named 'Hey Jude', using these technologies and well written rules, the computer can infer based on the rules that 'John Lennon' has plaid a song named 'Hey Jude' without this information explicitly existing.

This information does not necessarily exist on its own and may have relations to more related information. *Open Linked Data* [BHB09] focuses on the relations between the structured information and combining this data. This enriches the information even more, gaining easy access and splitting the building process between different organizations. This data uses the URIs of the RDF triple to link these organizations information. A great example of this collaboration is the linked information between DBpedia[4], Geonames[5], Freebase[6] and many more. This also allows powerful tools to be built using this open data, such as search engines and finding related content.

---

[3] `http://www.w3.org/`

[4] `http://www.dbpedia.org/`

[5] `http://www.geonames.org/`

[6] `http://www.freebase.com`

There is a strong effort by the W3C to enforce a standard for these good practices and collaborations. This structured information is an important step in the vision of a semantic web. Semantic web is not only about RDF structured information and open linked data, it also focusses in building vocabularies, ontologies and knowledge organization systems to enrich the existing data. With these vocabularies it is possible to apply some reasoning based rules. These rules can generate more data, not provided initially by people, or enriching even more the existing data. The same way information can be shared and linked, rules can also be shared using *Rule Interchange Format* (RIF) [BK07]. This allows knowledge systems to collaborate and enrich their systems.

The aim of semantic web is not only to correctly express knowledge but also to reason with it. Some technologies were built for this purpose. As previously stated, RDF triples are composed by a relation between 3 parts, a subject, an object, and a predicate or property. For example stating "John is a person" results in the triple with the subject "John", the object "is a" and the predicate or property "person". These relations can expand more and more, resulting in a graph data model. RDF triples can be based on a RDF Schema (RDFS) vocabulary definition, allowing to express simple ontologies.

While being similar to RDFS, OWL ontologies are much more than just for vocabulary definition. RDFS is more limited on their expressiveness and OWL provides a more flexible definition of the vocabulary and also can find inconsistencies in data and axioms.

SPARQL Protocol and RDF Query Language (SPARQL) [PS08] is a query language build to access RDF formatted resources. Basically these queries targets to find RDF triple patterns within the knowledge base. RDF, OWL and SPARQL are recognized as the key technologies of the semantic web, and have become a *W3C* recommendation.

These evolving technologies are useful to acquire relevant machine readable data, containing not only information on specific targets, such as entities, but as well as the relations these pieces of information have between them. These tools and information can be used for *Named Entity Recognition* (NER) to find and disambiguate entities from natural language text.

1.2.4 Named Entity Recognition

Named entity recognition, or *NER* [MZS06], is a natural language processing task, aimed to identify and extract entities (people names, geographical locations, organisations, etc.) from text written in natural language. These systems are important to find relevant information contained in text. These entities can be linked to *open linked data* by assigning them an unique *URI* for that specific entity, linking to more information than just the name or the type of entity.

Some of these entity names are ambiguous, depending on the context of the text it is possible to disambiguate to a specific entity. Ambiguous entities are when there are at least two specific entities for the text. Each specific entity is different. For example

'Henry Ford' is not ambiguous, but reducing to 'Henry' can result in an ambiguous entity, being ambiguous between several people named Henry, such as 'Thierry Henry', 'Henry Charles' and 'Henry Ford'. This disambiguation can be done several ways, being the most common one assuming a specific domain.

These *NER* systems work better when focussed in a specific domain (movie reviews, product reviews, etc.), as opposing to having no domain. The more specific the domain, the more likely the system gets better results. Is is possible to adapt these domains, but the systems will lose performance with this change [BDP07].

1.2.5 Twitter

Twitter[7] is a very popular social website, allowing users to communicate and share their opinion of anything they want. Twitter allows a fast and easy way for users to send short messages. There are more than 600 million users, sending million messages per day[8].

The messages shared by Twitter are known as *tweets*, limited messages up to 140 characters that may also contain a location and a picture. Messages can also contain hashtags, user names, links all inside the message or be a reply to other tweeted messages. A hashtag is a mark that can be shared between tweets that have something in common. Hashtags are inserted by the user, composed of words that follow the symbol '#' until a space is found or the end of the message. Clicking a hashtag will show other messages that also contain that hashtag.

Messages containing user names will show the referred users profile and his messages. User names must follow the symbol '@'.

Twitter users may send messages or follow other twitter users. Private messages can be sent to people who follow you, but the rest of the messages are public. Following an user means you can receive notifications when than user sends a new message.

1.3 Project Integration

This work was done with the collaboration of the company *AnubisNetworks*[9]. *AnubisNetworks* is a Portuguese company that develops advanced security solutions for internet service providers, businesses, corporations and institutions. *AnubisNetworks* is a leading email security provider, serving mobile operators, internet service providers, banks, universities and other companies. The result of this work will be merged with an existing project in *Anubis Networks*, namely *StreamForce*.

StreamForce is a real-time analysis and viewing platform on big data. Using a stream of data, this data is grouped by a common trait within a specific grouping time specified by the user. This grouped data is then printed in a chart to be viewed in real time and can easily be used to compare different calculated values.

---

[7] `https://www.twitter.com/`

[8] `http://twopcharts.com/twitter500million.php`

[9] `www.anubisnetworks.com`

StreamForce already supports information streams from social media, including twitter, sentiment analysis will enrich this existing information, allowing clients to view and compare the public sentiment.

1.4 Document Structure

In Chapter 1 an introduction was given on the motivations and brief background and key concepts necessary to understand the remaining content. This background is not exhaustive since the research area is large and diverse.

In Chapter 2 the problem will be presented as well as the approach taken. Some applications are given to better understand the possible goals of this problem.

In Chapter 3 work that is related to our own is discussed, showing and discussing the state of the art on this topic. Computational linguistics are analysed as well different approaches for sentiment classification.

In Chapter 4, a brief explanation on the tools that were used is given. These tools provide the information and methods that are useful for the selected approach.

In Chapter 5, it is shown a detailed explanation of the approach, showing how techniques and algorithms are used and how they are linked together, to achieve the expected results.

In Chapter 6, several benchmarks are used to evaluate our system regarding entity recognition and sentiment classification. These benchmarks for entity recognition are applied to identify people, locations, organizations and other relevant entities, and for sentiment analysis is applied for simple adjectives, selected messages and random collected tweets, and combining both entity recognition and sentiment analysis we process political tweets, finding and assigning sentiment to specific entities.

Finally Chapter 7 gives the final conclusions, contributions and future work supported by the contributions and results from this work.

2

The Problem, Approach and Applications

2.1 Problem Formulation

The main objective is to identity the general sentiment contained in messages with specific target entities. The first step is getting the messages from twitter as a stream of messages. From this stream we will process each message separately, extracting all useful information from it such as usernames, hashtags, emotion icons and URLs.

Usernames are marked with the character @ as the first of a sequence of characters, not including white spaces (e.g. @twitter, @YouTube, @BarackObama). Usernames are unique and are not case sensitive. Hashtags are marked with the character # as the first of a sequence of characters, not including white spaces (e.g. #android, #music, #news). Hashtags are unique and are not case sensitive.

Emotion icons are a sequence of characters to express emotions, usually by resembling a face (e.g. ":)", ">:(", ";p"). Emotion icons have become popular with the growth of short text based messaging in the internet.

By removing these elements, the message will be more concise and easier to parse. This slightly smaller message will be marked with all known entities and sentiment contained within it. The entities will be identified with a specific URI for that specific entity. The sentiment n-grams in these messages will be labelled either as being positive, negative or neutral. A n-gram is a sequence of n adjacent words or letters, depending on the application. Here the n-grams are used as a sequence of words forming an item with a specific sentiment. A n-gram of size one is referred to as an unigram, of size 2 is referred as a bigram and the less common n-gram of size three is referred to as a trigram. An example of an unigram is "awful" and of a bigram is "well done".

Once entities and sentiment contained in the message are identified, we assign the sentiment to the entities using the sentence grammatical structure identified by a parser. Then this information can be used to create statistics and follow certain entities, usernames and hashtags for corresponding sentiment, or saved in databases for historical data. These messages are sent from every part of the world and can be in different languages. Our focus is to extract sentiment on messages in the Portuguese language. We can not extract correct sentiment from messages in languages that we can not interpret.

2.2 Approach

The implemented solution for problem stated in Section 2.1 is a combination of a few tools and methods. All of these components will be fully explained in Chapter 4 and Chapter 5. Here a short overview of our solution is described.

Messages from Twitter will be received using the Twitter Stream API. These messages will contain information of the sending user, the sent message as well as the language classification provided by Twitter. The Twitter message will contain information that will be useful later on and will be extracted by using regular expressions, defining a search pattern to find usernames, hashtags, URLs and emotion icons. This information is saved and extracted from the message and replaced by identifiers, as to shorten and better process the message by preventing some parsing errors that could occur when this information is present.

Lets take the following tweets as an example, "a siria está muito mal, nem a onu consegue ajudar #politics #war http://example.org/?id=123456". Here the hashtag and the URL would be internally saved and replaced by a identifier marking their original position, resulting in "a siria está muito mal, nem a onu consegue ajudar /HASH1 /HASH2 /URL1".

This message is then parsed, resulting in a parse tree with classification of *Part Of Speech* within a tree structure, representing relations between each segment of the message. Using these grammatical classified words or *Part Of Speech* we can search for entities and sentiment accordingly. In the previous example we can identify as nouns "siria" and "onu", as verbs "está", "consegue" and "ajudar" and as adverbs "muito" and "mal".

Entities are searched first. Entities will be likely classified as a noun, and will be searched within certain entity types (Person, Location, Organization, etc.). In this example the only nouns that were identified and an entity is "onu", as an organization. The noun "siria" was not found as an entity because it is not spelled correctly.

Ambiguous entities can be found, having more than two specific entities as possible options for that combination of words or word, and these can be filtered to a specific entity by some context in the message. This context is ether hashtags or similarity with other entities. All entities found with messages containing hashtags could influence future entity disambiguation. Disambiguation by similarity will find the most likely entity, within the ambiguous options, that have more in common with the remaining entities found in that message.

After this initial entity recognition is the spell checking performed. The spell checker as well as the translator should not be used before entity recognition, since these can change entities to something that would not be recognized as an entity and change to something else that is similar or close to the word. For example the spell checker, in Portuguese, could change 'Barack' to 'barraca' and the translator, from Portuguese to English, could change 'Pedro Passos Coelho' to 'Peter rabbit steps'. The spell checker will correct the message, maintaining the entities previously found. After this correction a new parse tree is build and the entity recognition algorithm will be run one more time, trying to find something that could not be identified previously.

Following the same example, the noun "siria" would then be corrected to "síria" and then it would be correctly identified as an entity, as a location.

After the entity recognition is done, the sentiment classification starts. Sentiment particles are the sentiment values of small n-grams within the message, searched in the words of the message, taking into consideration their *POS*. In our example we could identify "mal" being an adverb as a negative sentiment particle and "ajudar" being a verb as a positive sentiment particle.

These sentiment particles can be influenced by sentiment changing words such as negations or sentiment intensifiers. Sentiment negations and intensifiers will influence the closest sentiment particle based on the parse tree. If no sentiment particle is found within a maximum range, then these will become new sentiment particles taking a default sentiment value accordingly.

In our example "muito" is identified as a sentiment intensifier, and will increase the closest sentiment particle value, in this case it will increase the negative value of the sentiment particle "mal". Also "nem" is identified as a negation and will change the sentiment from the closest sentiment particle "ajudar" from positive to negative. The distance between words takes into consideration the structure of the sentences and will be explained further in this document.

Each sentiment particle is then assigned to the closest entity, taking into consideration the distance within the parse tree, using a distance algorithm. These particles will then be grouped into a single sentiment value for each entity found.

Each entity will have its own sentiment value as well as a value for the whole message, as a grouped value of all sentiment particles that were found in the message.

In our example the negative sentiment of "muito" and "mal" will be assigned to the entity "síria", while the negative sentiment of "nem" and "ajudar" will be assigned to entity "onu", assigned using the same distance as the negations and sentiment intensifiers.

2.3 Application and Practical Usage

A single message shares little on the public opinion. This becomes relevant when we are able to gather and process a large number of these messages. Sentiment analysis can be applied on specific sample of messages and extract information only from these messages, or it can also be applied to a constant inflow of messages, continuously running and gathering information. With the gathered information we can obtain results and use them to create several different statistics that could be used for several applications.

2.3.1 Entity Monitoring

Considering a company or a product, using sentiment analysis to monitor the trending or recent sentiment. This can be useful to alert the company of sudden changes on the public opinion, so that it can quickly respond to these changes. It can also be useful to see the public reaction on recent changes and decisions that the company makes.

This information that is being monitored can be saved in a database with the corresponding date. This will create a benchmark to easily view the evolution of the monitored sentiment over time.

Monitoring has been used by O'Connor et al. [OBRS10] and compared with public opinion surveys from polling organizations with the political opinion in the USA from 2008 to 2009. Using a moving average of 15 and 30 days, and comparing the extracted sentiment with the values form the pooling organizations, a correlation of about 70% was found between the results.

2.3.2 Pooling

For a deeper understanding of the sentiment on entities, it is possible to focus our sentiment into groups for comparison and getting more insights on the public thought. With the user information it is possible to compare sentiment between genders and different age groups. This helps select a specific target user or group to analyse. The available information will be further explained in Section 5.1 and Section 5.2.

Pooling can also be used to compare sentiment of different entities. This type of application has been done by O'Connor et al. [OBRS10], to monitor *Twitter* sentiment for the presidential elections of the United States of America of 2008. Keeping their focus on the candidates, this approach only interpreted messages containing the entities *Obama* and *McCain*.

3

Related Work and State of the Art

Sentiment analysis of tweets is considered harder than the sentiment analysis of reviews, since we do not have a specific domain in twitter and with the use of informal and irregular language. Most of the work in the literature follow a feature classification approach, namely by obtaining n-grams and assigning to them a sentiment value [SHA12].

3.1 Computational Linguistics

3.1.1 Summarization

Most documents, such as reviews, are large in size and some work has been done on reducing the size of the review into a good summary, without losing much accuracy on the classification since there is a loss in information.

This is done by getting the most subjective sentences, and only analysing those. In some cases, the last sentences of a review make a good summary, instead of getting the most subjective sentences.

As done by Pang et al. [PL04], reducing long movie reviews into the most subjective sentences or the last sentences, did improved the classification. While using the most subjective sentences does have better results, getting the last sentences did show good results, as the conclusions are usually written last. This does speed up the summarization without losing too much accuracy.

Since *Twitter* messages are already short, this step will not be needed in our approach.

3.1.2 Language Identification

Language identification is the first thing that must be done, so that we do not try to process languages that we cannot interpret.

Our focus is to process Tweets written in the Portuguese language, therefore it is necessary to filter out all tweets that are not in that specific language. This will make sure the other steps will work correctly since they are also prepared for a specific language.

Language identification has been done with the use of n-grams that are frequently used in each language [CT94]. The message is split into equal sized n-grams to be compared with the n-grams frequently occurring in each language.

Another variant is done by searching for the most frequent and occurring short words, or unigrams, in each language [MP96; CZ99].

Language identification becomes harder with the shortness of the message, since it will have less text to give us the clues needed for a good probability on what language it is written in.

Using the Twitter API, messages have aggregated a probable language, provided by a private classification algorithm.

3.1.3 Acronyms and Other Abbreviations

Since we are dealing with short messages, acronyms and abbreviations can occur. With acronyms we do not need to worry, since it is mostly used on names and normally do not have sentiment attached to it, and if they are used as nouns, the acronyms will be treated as an entity.

Some used abbreviations are for example "FTW"[1] is used to indicate a positive sentiment, and an opposite example is "TMI"[2] used to indicate a negative sentiment. These can be mapped directly into sentiment, alongside the emotion icons, instead of replacing with their explicit meaning text in the message.

These acronyms constructed with the English language, but are used through the globe in many different languages and sharing their meaning. These acronyms are relatively infrequent, compared to the total amount messages [Bar10].

Other abbreviations types include emotion icons or emoticons, providing a way for the user to express the sentiment they want to share using text characters.

The use of these features has been the focus of some related work [PP10; KWM11] and has shown improvements in classification accuracy.

3.1.4 Parser and Part Of Speech Tagger

A parser is a tool used obtain a structured representation of a sentence, according to the grammar rules. A parsed sentence will be split into n-grams, annotating the use and information on the relations with other n-grams, resulting in a parse tree. The POS gives

---

[1] Meaning "For the win"

[2] Meaning "Too much information"

information on the grammatical use of the word, as being a noun, a verb, an adverb, etc. Even though POS taggers may exist on their own, normally they are provided as parts of a parser.

The use of POS has been explored for sentiment analysis in tweets by Gimpel et al. [GSODMEHYFS11] and Agarwal et al. [AXVRP11], combining these annotations with previously proposed unigram models, and reporting an overall improvement of 4% in accuracy. This shows that POS can give a better understanding of the sentiment present in the message.

Most of the current work in sentiment analysis does not take into consideration the relations between n-grams and the POS combination of n-grams in a sentence. Gathering the sentiment of all the words in the text, ignoring their POS and relations between the words.

3.1.5 Entity Recognition and Extraction

Entities are known n-grams that identify something or someone specific, that can be the target or the object of a sentence. These entities can be names, places, objects, etc.

Most of the work done in sentiment analysis is domain specific, and so assume the target of the review as the target entity for classification. For example, movie reviews will target the specific movie in the title as the entity and relate all the text and sentiment in the text toward that movie, even if the user is referring to the actors or directors in some sentences [BDP07; PLV02].

This assumption that the text refers to a single entity, does not take into consideration all the entities that occur in the text and assign to each one their own sentiment on that specific text.

Alternatively, other works assume hashtags present in a message are the targets of the message sentiment. The sentiment extracted from the message will be given to all hashtags present in that message. This was used to monitor the behaviour, sentiment and spread of hashtags [GSXYW11].

Other works use a semantic approach, searching for entities in sentences using *DBpedia* and *Wikipedia*. These sentences are also annotated with POS, using *OpenNLP*[3] trained with the *CINTIL* treebank[BS06]. The target of this work is not only to find entities, but to find relations between these entities. These relations between entity pairs are based on an initial set of examples and are found using a k-nearest neighbour algorithm (k-NN) applied to common n-gram patterns that occur for these relations [BFSMS13]. These relations between entity pairs are "origin", "location of death", "influenced", "partner" and "key person". Some examples are "Camilo Pessanha died in Macau" sharing the relation

---

[3] `http://opennlp.apache.org/`

"location of death" between the entities "Camilo Pessanha" and "Macau" and "Microsoft was founded by Bill Gates" sharing the relation "key person" between the entities "Microsoft" and "Bill Gates".

It is also possible to search entities in text using *DBpedia* information to build an ontology for specific entity groups. These entities are selected based on specific features. This has been done by the REACTION group (Retrieval, Extraction and Aggregation Computing Technology for Integrating and Organizing News)[4]. The ontology rules were previously done before using *DBpedia* information to populate it. This ontology is populated with Portuguese politics and political organizations, based on data extracted using SPARQL queries on *DBpedia* [MBCCS11].

The same group used this method to build their own filtered version *DBpedia* for specific Portuguese entities. This filtered version is *DBpediaEntities-PT01*[5].

Some n-grams are not specific enough to refer to a specific entity, but can be ambiguous denoting two or more possibilities. These ambiguous entities can then be disambiguated to specific entities based on some form of context. For example 'Henry Ford' is not ambiguous, but reducing to 'Henry' can result in an ambiguous entity, being ambiguous between several people named Henry, such as 'Thierry Henry', 'Henry Charles' and 'Henry Ford'.

Some research has been done in entity disambiguation of ambiguous entities by finding context in the Wikipedia pages of known entities and trying to find any mention of the ambiguous entities. Mentioned entities are likely to be related to other previously found entities. Existing wikipedia titles with these ambiguous entities can also be used and taken into consideration for disambiguations [RRDA11].

3.2 Sentiment Classification

There are several ways to classify sentiment, using machine learning algorithms or just classifying based on the semantics of the text. Classification is responsible for separating different things by classes, such as positive and negative n-grams in this specific case.

The classification is done by functions and with the use of pattern matching, enabling to determine the closest match, so words like 'hated' will be considered a negative word since it is close to another negative word known by the classifier 'hate'. Classification can also be used to perform language identification.

Studies have found that the human agreement levels on sentiment classification is about 80% [WWH05] on average. This value was achieved by comparing human manually classifying n-grams for sentiment value.

We performed a similar test on a random sample of 200 tweets, extracted and manually annotated by three volunteers and the agreement level was 78.5%. These tweets are

---

[4] `http://dmir.inesc-id.pt/project/Reaction`

[5] `http://dmir.inesc-id.pt/project/DBpediaEntities-PT_01_in_English`

attached to the Appendix A of this document.

Emotion icons are a representation of a facial expression used to express a person's feelings or mood. They can include numbers, letters and punctuation marks. Emotion icons are also widely used as features for sentiment classification [KML13], since these provide an almost direct sentiment classification provided by the user.

3.2.1 Machine Learning Classification

Machine learning classification is a set of computational algorithms that are used to classify objects with features into specific classes. These algorithms can be trained with examples, used as a guide line, that are already classified and based on those will find the best matching combination to classify new information.

Machine learning classification can be applied, having predefined specific classes, adapting to the information it receives, based on its specific training data classifications. In this case the classes are, positive, negative and neutral sentiment, and the considered features are the n-grams occurring in the message [PLV02].

3.2.1.1 Supervised Classification

With supervised classifiers, training examples are needed, either by manually building the examples or generated. These examples will serve as the base for future classification and must be correctly classified.

When training the classifier, each feature will get closer to a specific class. For sentiment classification these classes are positive, neutral and negative. Training will compare the classifier results to the training results, allowing the algorithm to evaluate their solution and improve based on these results.

Naive Bayes, Maximum Entropy and Support Vector Machines are some examples of currently used supervised machine learning algorithms [Sil; PP10; AXVRP11; BF10; KWM11; Liu12; MKZ13].

3.2.1.2 Combined Classification

Combined classification is an approach that uses both supervised and a source containing sentiment annotated n-grams for classification.

As done by Prabowo and Thelwall [PT09] that used both approaches using a supervised algorithm with the combination of other sentiment sources and assigning the combined sentiment, instead of using just one approach.

Melville et al. [MGL09] searched for specific n-grams to get the domain of the document, and used a supervised classifier to get the document polarity using a domain specific classifier with the determined domain.

3.2.1.3 Naive Bayes

Naive Bayes classifier is a probabilistic classifier based on the Bayes theorem.

$$P(c|f1...fn) = \frac{P(c)P(f1...fn|c)}{P(f1...fn)} \tag{3.1}$$

Where *c* is the class and *fn* are the features taken into consideration. This probability is calculated for each class, and the class with the largest probability *P(c|f1...fn)* is the most probable class for the classification. In the specific case of sentiment classification, the classes would be the positive, negative or neutral sentiment, and the features are the words that may occur in the text. It is called naive because it assumes that the features are independent from each other.

Most words are related, many sharing an origin and its meaning, and with this assumption that relation does not exist. An example is the word "goodness" contains and shares meaning with the word "good". Although this assumption is unrealistic, Naive Bayes still performs rather well comparing to other machine learning algorithms for classification.

3.2.1.4 Maximum Entropy

Maximum entropy classifier is like the naive bayes classifier but is based on an exponential expression.

$$P(c|f) = \frac{1}{N(fs)} exp(\sum_i \lambda_{i,c} F_{i,c}(f,c)) \tag{3.2}$$

Where *N(fs)* is normalization function for *fs* features, $\lambda_{i,c}$ is a weight function that changes with the training for a *c* class and $F_{i,c}(f,c)$ is a boolean function, indication if the feature *f* can belong to the class *c*. *i* iterations are used to get the best matching example from the training data, improving accuracy. Like Naive Bayes, the class *c* where *f* has the largest probability *P(c|f)* is the most probable class for that feature.

Unlike Naive Bayes, maximum entropy makes no assumptions on the independence between classes.

3.2.1.5 Support Vector Machines

Support vector machines, or SVMs, rather than a probabilistic classifier such as Naive Bayes and Maximum entropy, will convert the feature into a point in a hyperplane. This hyperplane is split into into the several classes, based on the training data. The position of the features point, being closer or farther from the area of a class in the hyperplane, will indicate the class that this feature is more likely to belong. There is also a margin to separate the classes and features contained in this margin are treated as neutral.

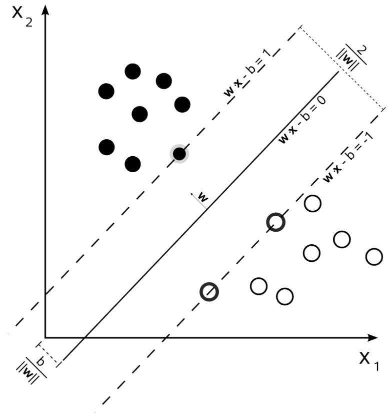


Figure 3.1: Support vector machine example with 2 classes

Of the standard machine learning algorithms, Pang et al. [PLV02; PL04] showed that Support Vector Machines outperformed Naive Bayes and Max Entropy classifications, having the best overall results, although this depends on the domain. Having a specific domain help to minimize the classification error, while having a broad domain has more features and a wider range of values, being more difficult to correctly classify [BDP07].

3.2.2 Semantic Classification

Semantic classification assigns the sentiment of found features from a source, containing n-grams already annotated with the corresponding use and polarity. This semantic classification will take into consideration the n-grams, used as features, relations with the rest of the text, proving a better context for their use and provide a more specific classification for each different use. A specific grammatical use of a n-gram could have different sentiment values.

Yan Dang et al. [DZC10] and Bruno Ohana et al. [OT09], used *SentiWordNet* for this semantic approach, providing a large set of domain independent features to classify product reviews. This approach does not require prior training, since n-grams in *SentiWordNet* are already annotated with sentiment and objectivity. His experiments show that while the machine learning approach tend to be more accurate in specific domains, the semantic approach has better results in generality, providing better results when no specific domain is used. Different types of product reviews were used to compare these results. Sentiment n-grams were also annotated with POS using the Stanford POS tagger[6].

Dan Moldovan et al. [MBTAG04] has shown that semantic classification can be achieved by finding the semantic relations between certain n-grams in a sentence. Using noun phrases as their targets, the relations between these noun phrases are found by mapping

---

[6] `http://nlp.stanford.edu/software/tagger.shtml`

the remaining content (nouns, verbs, adverbs, adjectives, etc.) to the *WordNet* online dictionary, taking into consideration their surroundings allowing to give some context for their use, and finding existing relations.

In a study by Pang et al. [PLV02] in related work, now focussing on the sentiment within text, considering movie reviews as the domain, using as features sets of commonly found n-grams considered for that specific domain, can affect the accuracy results. A experiment was done by using different n-grams as positive and negative features and their results, shown in Table 3.1.

	Positive word list	Negative word list	Accuracy	Ties
Human 1	dazzling, brilliant, phenomenal, excellent, fantastic	suck, terrible, awful, unwatchable, hideous	58%	75%
Human 2	gripping, mesmerizing, riveting, spectacular, cool, awesome, thrilling, badass, excellent, moving, exciting	bad, cliched, sucks, boring, stupid, slow	64%	39 %
Human 3 + stats	love, wonderful, best, great, superb, still, beautiful	bad, worst, stupid, waste, boring, ?, !	69%	16%

Table 3.1: Word Features for Sentiment Classification on Movie Reviews

Adapted table from Pang et al. [PLV02], Table 3.1 shows human selected n-grams (Human 1, Human 2 and Human 3) and n-grams statistically selected from test data to be used for sentiment classification. These results are using test data composed of 700 positive and 700 negative reviews and rely on the detection of their word selections.

Pang [PL04] in sentiment classification of movie reviews got result showing that taking into account the features by presence rather by frequency has given better results. This is done by using feature vectors with a binary value, indication if the feature occurred or not within the processed text.

Pang's results also indicate that using unigrams is more suitable for sentiment analysis in movie reviews. Although Dave et al. [DLP03] found that bigrams and trigrams work better than unigrams in sentiment analysis in product reviews.

On Barbosa and Feng [BF10] experiments, it is argued that with the open domain and the use of infrequent words in short messages, the classification will have low accuracy since only a low number of n-grams are recognized.

Saif et al. [SHA12] have done a similar classification using positive and negative n-gram features, adapted n-grams with POS, to classify only what is relevant and not all the n-grams that exist in the text.

A broad set of n-grams, even though may contain n-grams that are rarely used, will only take into consideration the n-grams found. With a larger set of features, the more n-grams are recognized, resulting in a better classification of the n-grams.

To create this broad set of n-grams, António Paulo-Santos et al. [PSRM11] have used online dictionaries, such as *WordNet*, to find existing semantic relations between words, building a broad set of n-grams with sentiment value, based on a few initial hand annotated set. This starting set has very few, simple commonly used unigrams that clearly are positive and negative, and based on relations between other words using the dictionary to search for synonyms, antonyms or other semantic relations, these relations will propagate sentiment from the initial set to all other words that contain any relation to them.

Denecke [Den08] with the use of translators, applied sentiment annotated n-grams in the English language, to classify sentiment in German written messages. All relevant n-grams found in the German text were translated to English, and using *SentiWordNet* to get their corresponding sentiment values. These sentiment values take into consideration the n-gram use in the message, such as their grammatical use (or POS) and their relations to other n-grams.

He concluded that the sentiment of the n-gram in English is very similar to the sentiment of the same n-gram in German. Using 50 emails in German and translated into English by hand, Denecke got results showing better accuracy on the German classifications, even though the translations could add errors.

Tawunrat Chalothorn et al. [CE13] did something similar for the Arabic language. A human translator was used to translate messages from two forums and then used *SentiWordNet* to get the sentiment of the n-grams. This was used to find the most negative of both forums.

3.3 Domains for Sentiment Analysis

Most of the work done in sentiment analysis is done with specific domains and with data sets through which the results can be verified, such as reviews, consisting of the text that will be classified and with a rank or value that the user gives, along side with the review. This assumes the domain where you extract the text from, if it's a movies review website, the assumed domain and entities are movies.

Most of the related work has been done within domains, such as movie reviews [PLV02; PL04], product reviews [DLP03; NSKCZ04; NKW05; DZC10], news blogs [GSS07; BVS08; MGL09] and politics [ST11; CSTS11].

Training the machine learning classifier and the parser for a specific domain will have more accurate results than a generic classifier and parser, since it is more focussed and all messages have a similar use of the language, but only for a specific domain [VCC12].

Some work has been done in trying to switch between domains and minimizing the accuracy loss between domains, maintaining some features and their weights. For example, "Electronics" and "Home appliances" are very close domains, being better to switch from "Electronics" than from "Books" or "Music" domains [BDP07].

3.4 Conclusions

While some different approaches exist, some important common features are to be considered.

Short, clean messages are more easily and correctly classified. Emotion icons or emoticons are important sentiment features to take into consideration. For entity recognition a good broad set of data is very useful to use, such as *DBpedia*.

For classifying the sentiment value of n-grams, the POS is an important technique to take into consideration as well as the surrounding n-grams that could influence their sentiment. For domain specific classification, the machine learning approach tends to do better, but for a domain independent classification the semantic approach appears to be more appropriate.

Translation of the n-grams can be used, since they will likely maintain their sentiment value through different languages, even though translation errors can affect the classification. If there is a clear specific domain, focussing the classification for this domain will improve results. When the domain is not clear, hashtags can provide some context for the message.

4

Tools

This chapter describes relevant tools that provide the information and methods that are useful for the selected approach presented in Chapter 2.

4.1 Twitter API

Twitter supplies an *Application Programming Interface* (API) for the community of developers. This API allows to get information by *REST* calls or streaming.

The *REST* API, based on the *REST* architecture [Fie00], allows the normal functionalities of Twitter by using *REST* calls. These normal functionalities include getting a personal feed of followers tweets, posting a tweet or getting specific user information. To post messages this API will be linked to a specific twitter account.

This API can also be used for specific queries, such as searching for a specific user sent messages, user mention or hashtags, returning limited results and ordered by sent date.

Instead of using the *REST* calls, the Twitter API also provides streaming readers. The streaming API also allows access to a low latency, real-time input of messages. Twitter provides access to public, single-user and multi-user streams. Public streams contain the public tweets sent, single-user streams contain messages from a specific user and multi-user streams contain messages from a specific user list.

Once these streams are established, they will never need any further interaction. These streams are read only, and can be filtered based on available information such as used language or user location.

The API supplies all this information in the *JavaScript Object Notation* (JSON) format[1], which is a widely used format in the web. For Twitter to provide access to this information, *Open Authentication* (OAuth) protocol[2] is used to securely grant permissions to use these APIs.

4.2 Stanford Parser

The stanford parser[3] is a statistical natural language parser that figures out the grammatical structure of sentences. In a sentence, the parser will figure out what is the most probable use of the word, if the word is used as the subject or the object of the verb, as well as determine if the word is a noun, verb, adverb and so on. Being a statistical parser, it uses as examples, sentences parsed by humans, and from those, tries to reproduce the results in new sentences.

The stanford parser also provides a POS tagger[4], that can find the most probable grammatical relations. This tagger is natively available for the Chinese and English language. Another frequently used parser, such as *FreeLing*[5] does provide POS tagging but without the parse tree with existing relations.

An adapted parser for the Portuguese language has been developed by the Natural Language and Speech Group of the University of Lisbon[6]. This parser was made by applying a tree bank, with sentences in the Portuguese language, to be used as training data for the Stanford parser.

In order to use this adapted Portuguese parser the message needs to be previously tokenized. A tokenizer detaches punctuation marks and splits the text into n-grams and symbols as tokens. A tokenizer for the Portuguese language has also been made by the same group [BS04] for use with the adapted parser. This parser returns the POS tags listed in Table 4.1.

---

[1] `http://www.json.org/`

[2] `http://oauth.net/`

[3] `http://nlp.stanford.edu/software/lex-parser.shtml`

[4] `http://nlp.stanford.edu/software/tagger.shtml`

[5] `http://nlp.lsi.upc.edu/freeling/index.php`

[6] `http://lxcenter.di.fc.ul.pt/tools/en/LXParserEN.html`

POS Tag	Grammatical Use
A	Adjective
AP	Adjective Phrase
ADV	Adverb
ADVP	Adverb Phrase
C	Complementizer
CL	Clitics
CP	Complementizer Phrase
CARD	Cardinal
CONJ	Conjuction
CONJP	Conjuction Phrase
D	Determiner
DEM	Demonstrative
N	Noun
NP	Noun Phrase
O	Ordinals
P	Preposition
PP	Preposition Phrase
PPA	Past Participles/Adjectives
POSS	Possessive
PRS	Personals
QNT	Predeterminer
REL	Relatives
S	Sentence
SNS	Sentence with null subject
V	Verb
VP	Verb Phrase

Table 4.1: Stanford Parser POS Tags

4.3 SentiWordNet

SentiWordNet[7] [ES06; BES10] is a sentiment dictionary, that is the result of annotating n-grams from WordNet, with sentiment values, according to the notions of positive, negative and neutral sentiment. The considered n-grams are in the English language.

This online platform provides the sentiment values for n-grams, including different values for different uses, as well as different POS of the n-gram. For example the word *'good'* may be used as a noun, as an adjective or as an adverb, and comparing those types it has different meanings and different values [BES10].

---

[7] `http://sentiwordnet.isti.cnr.it`

SentiWordNet also improves and evolves with the collaboration of the users community. People that find that an incorrect sentiment value, they can submit a feedback with the sentiment they agree on, possibly changing future values of that sentiment particle. This not only improves results over time but also will help adding more n-grams that users find that are missing and should exist.

For each n-gram there are several outputs for different meanings of the feature as well as different grammatical uses (POS), as shown by 4.1.

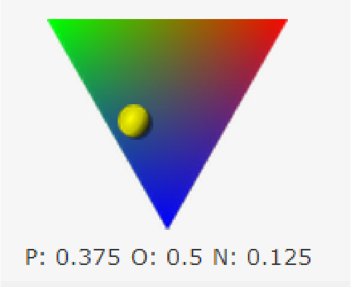


Figure 4.1: SentiWordNet output example

The output for a word contains three values, a positive, a negative and an objectivity value. Positive and negative values can be viewed from left to right, the top most left being full negative and top most right being full positive. The vertical value is the objectivity, having the most objective value at bottom. The sum of the three values is equal to 1.

The less objectivity a meaning has, the more subjective it is, having more meaning and leading to better results [ES06]. As we can see, if the objectivity value is at 1, the positive and negative values are 0.

4.4 DBpedia

DBpedia[8] is a project started by a collaboration between the *Free University of Berlin*, the *University of Leipzig* and *OpenLink Software*. *DBpedia* is the result of efforts made to convert information from *Wikipedia*[9] into structured format. *Wikipedia* is a free multi-language internet encyclopedia, gathering information on any topic built with the efforts of the community of users. Users submit and collaborate to build better and more reliable information accessible for every one in the web.

DBpedia supports all its information on a knowledge base, creating and maintaining a consistent ontology. These efforts are to build a better and more reliable information system, improving not only the navigation but adding to the initial information as well.

DBpedia will be used for our named entity recognition process, containing a large amount of annotated data with people, locations, organization etc.

---

[8] `http://dbpedia.org/`

[9] `http://www.wikipedia.org`

4.5 Apache Jena

Apache Jena is a Java framework for linked-data and semantic web tools and applications.

This framework provides an *API* for the Java programming language, allowing easy access to powerful tools, libraries and structures useful for this thesis.

Apache Jena is useful to manipulate *RDF* formatted triples, *OWL* ontologies as well as reasoning with these and *SPARQL* queries.

4.6 Hunspell

Hunspell[10] is an open source spell checker tool. Based on *MySpell*(former spell checker of OpenOffice), libraries and APIs have been made for several programming languages, such as Delphi, Java, Perl, .NET, Python and Ruby.

Hunspell is the spell checker of LibreOffice, OpenOffice, Mozilla Firefox, Thunderbird, Google Chrome, Mac OS X, InDesign, memoQ, Opera and SDL Trados.

4.7 Translator

Our approach will need to translate n-grams from Portuguese to English. The translator that is used was developed by *AnubisNetworks*. While not being as good as other translators like *Google Translate*[11] and *Bing Translator*[12], it does not have a maximum limit of translations or pricing for its use.

4.8 GSON - Java JSON Library

GSON[13] is a simple open-source Java library. This library allows to convert JSON from a String to a Java object representing that string. This Java object can be more easily read and changed. New GSON objects can also be created from scratch.

These GSON objects can easily return their string representation.

---

[10] `http://hunspell.sourceforge.net/`

[11] `http://translate.google.com/`

[12] `http://bing.com/translator`

[13] `https://sites.google.com/site/gson/`

5

Implementation

This chapter describes the implementation according to the proposed approach. The majority of the programming in this thesis is done using the Java programming language. The choice of the programming language is not only for its portability but also available APIs for some used tools, mentioned in Chapter 4.

A flow diagram showing how these tools and algorithms interact according to the proposed approach, is shown in Figure 5.1. Here it is shown how messages are going to be processed from the Twitter input stream to an output stream. This output stream is where the results will be written, to be subsequently read and processed by other applications.

Twitter information, such as hashtags, usertags, URLs and emotion icons will be stored and replaced by references, marking their original location. Then the parse tree will be built from the message, containing POS and entities are extracted and disambiguated using the entity recognition algorithm.

Only then is the message spell checked, since the spell checker can change entities to something different. The spell check will not only correct entities, but will also improve the parse tree, correcting words that the parser could not previously recognize correctly. With this correction the entity recognition algorithms is applied again, to find entities that were missed on the first run.

Then with the message corrected and entities found, the sentiment classification takes place. This sentiment classification takes into consideration the sentiment of n-grams as well with relation with negations and sentiment intensifiers within the message.

Then entity enumeration joins entities that should share sentiment in that message, and the sentiment found is assigned to each entity and entity enumeration. Each entity will have their own sentiment value, calculated based on the assigned sentiment.

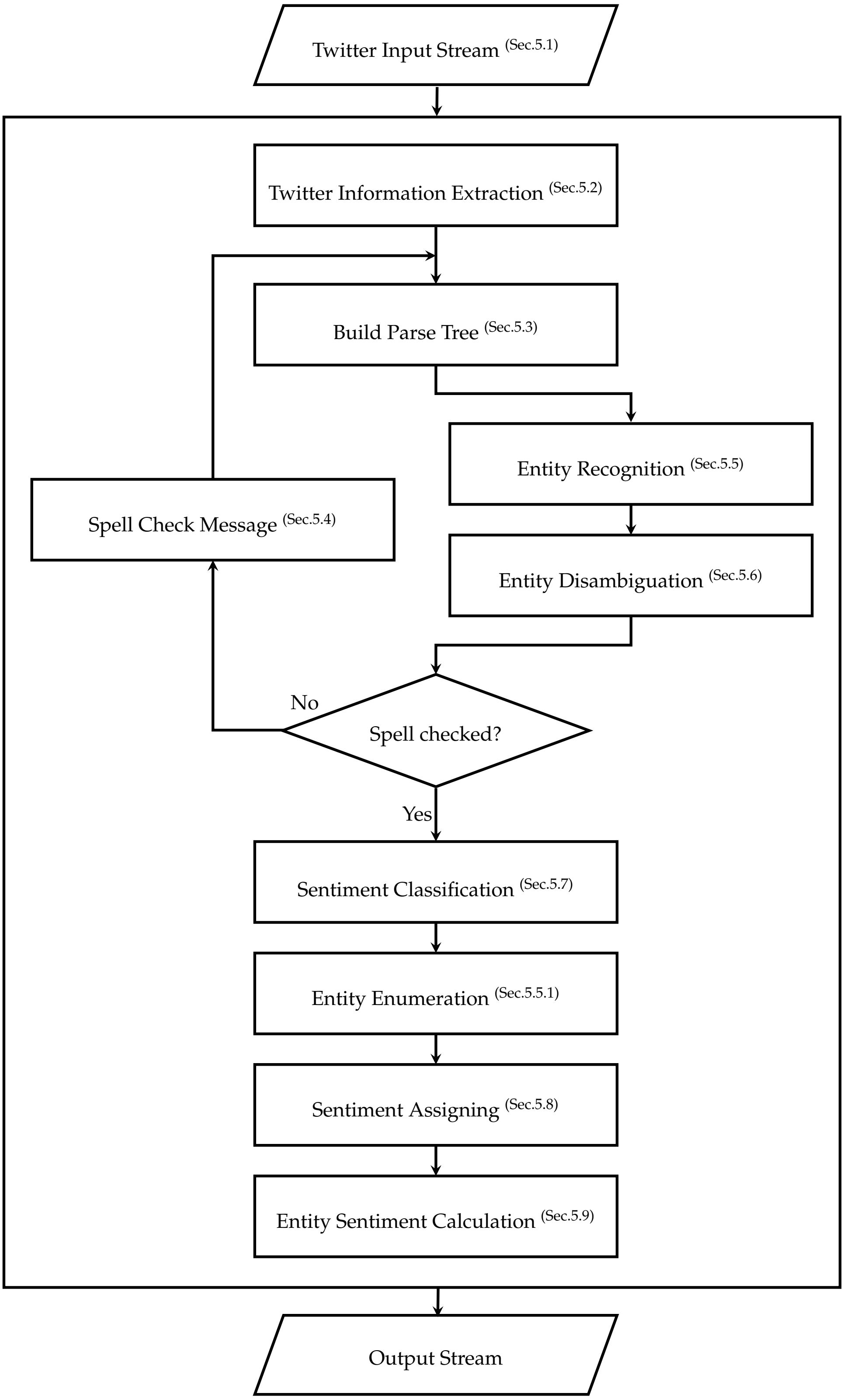

Twitter Input Stream (Sec.5.1)
Twitter Information Extraction (Sec.5.2)
Build Parse Tree (Sec.5.3)
Entity Recognition (Sec.5.5)
Spell Check Message (Sec.5.4)
Entity Disambiguation (Sec.5.6)
No
Spell checked?
Yes
Sentiment Classification (Sec.5.7)
Entity Enumeration (Sec.5.5.1)
Sentiment Assigning (Sec.5.8)
Entity Sentiment Calculation (Sec.5.9)
Output Stream

Figure 5.1: Main Information Flow Diagram

5.1 Twitter Input Stream

Twitter has a good community of developers that make and maintain several libraries for many programming languages, such as C, Java, .NET, PHP, Javascript, Python, Perl and Ruby.

To get access to real time tweets, we use the Twitter Stream API, getting a public sample of the current messages while they are being posted. These messages are stored in a queue, waiting to be processed. The information that arrives form the stream is in JSON format. Messages can appear duplicated in the stream, therefore this queue deletes duplicated messages. This way a message will be processed only once.

This stream supplies not only the message that has been sent, but also information on the sending user, such as its actual name, date of birth, location, friend count, followers count, profile image and profile colors, as well as the creation date and also a probable language of the message (supplied by Twitters private language classification). The sending location can also be available if the user permits, having the exact coordinates of the sender at that given time. Messages in the stream are filtered by language, processing only messages written in the Portuguese language.

5.2 Twitter Information Extraction

Other than the information supplied by the Twitter streaming API, the message can contain some extra information unique to social networks or to Twitter specifically, usertags, hashtags, URLs and emotion icons. Messages may also contain emotion icons, that could indicate an explicit feeling the user wants to transmit. Extracting this information from the message will also shorten the message and will ease the parsing for better results. These features, specially URLs and emotion icons will return incorrect parsing trees, normally because of non-alphabetic characters.

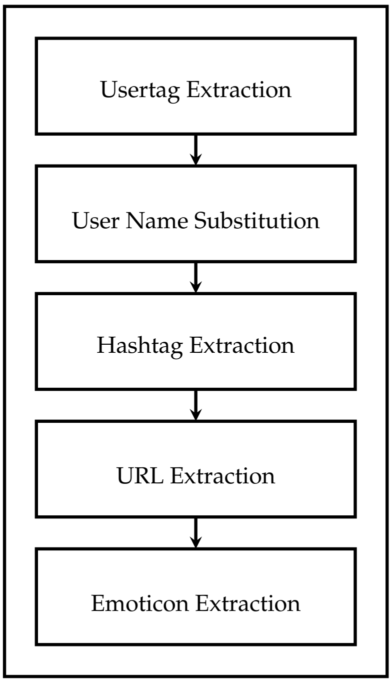


Figure 5.2: Twitter Information Extraction Flow Diagram

The mentioned information is extracted using regular expressions. This information is extracted and a reference is placed were this information was located in the message. These references will not affect the parser results and the spell checker will leave these references as they are.

5.2.1 UserTags

Usertags are employed by users to mark and share a message referring to a specific user. This tag is composed by the character ′@′, followed by the username of the referred user. These referred users may be the target of analysis and therefore it is important to identify and extract them from the message.

To identify these usertags, the following Java regular expression is used:

```
((\s)?(@[\w]+))
```

Here it is specified that we want to find the character ′@′ that may have a white space character before, represented as ′(\s)?′, and is followed by one or more word characters, represented as ′[\w]+′. The white space characters represented by ′\s′ include spaces, tabs and line feeds. The word characters represented by ′\w′ include letter characters from ′A′ to ′Z′, in upper-case and lower-case, number characters from ′0′ to ′9′ and the underscore character.

An example of an usertag is ′@Obama′ referring to the specific twitter user Barack Obama.

5.2.1.1 User Name Substitution

For the sake of better parsing and even for entity recognition, we can replace the usertag with the name given by the user to their Twitter account. The parser will most likely identify this replaced name as a noun, and the entity recognition will try to find as a relevant entity. To get the users account name, a REST call is done using the Twitter API, adding some processing time for each entity contained in the message. This option can be turned off for better processing time.

Taking as an example the president of the United States, Obama's personal twitter account, with the usertag *@Obama* would be substituted with the string 'Barack Obama', which can be identified as a named entity.

5.2.2 HashTags

Similar to usertags, hashtags mark specific events, people or anything the user wants. This tag is composed by the character '#', followed by a character string with no spaces. The hashtag can group messages that contain the same hashtag and can provide some context for those messages. This information can be very important for entity disambiguation.

To identify these hashtags, the following Java regular expression is used:

```
((\s)?(#[\w]+))
```

Here it is specified that we want to find the character '#' that may have a white space character before, represented as '(\s)?', and is followed by one or more word characters, represented as '[\w]+'. The white space characters represented by '\s' include spaces, tabs and line feeds. The word characters represented by '\w' include letter characters from 'A' to 'Z', in upper-case and lower-case, number characters from '0' to '9' and the underscore character.

An example of a hashtag is '#android' used for messages that the user may find that fit that topic.

5.2.3 URLs

URLs contained in messages can be confusing for the parser, and the removal of these is important for a good parsing of the message. While usertags and hashtags are useful latter on, the URL is only useful if we analysed the information that the URL points to or taking in consideration the domain name. This analysis of the URL could lead to no better results and will add some process time for each message containing these URLs.

To identify these URLs, the following Java regular expression is used:

```
((\s)?((http(s)?)|(ftp)://)?([\d\w-/]+)(\.)([a-zA-Z]{2,6})([^\s]*)(/)?)
```

These URLs may include the protocol ′http′, ′https′ or ′ftp′ followed by `'://'`, then finds the domain name and then can be followed by any additional URL information, such as a port number, path or query, anything that is not a whitespace, represented by ′[ˆ \s]′.

5.2.4 Emotion Icons

Emotion icons used in social networks, provide a simple and easily identifiable sentiment that the sending user supplies within the message, to show the emotions he feel or wants to show. For this purpose we detect positive and negative emotion icons by using regular expressions. These positive and negative emotion icons are given a default sentiment accordingly. The sentiment assigned to the emotion icons will be then processed the same way as other sentiment particles.

Normally these emotion icons resemble faces, with the most simple forms having only eyes and mouths. Other forms have extra features such as noses, hats, hair, etc. The icons we find are the most classical side ways icons with smiles or frowns (e.g. ′:)′ , ′:(′) and similar variations with noses or hats.

To find these emotion icons, Java regular expressions are used focussing on different parts of these emotion icons. A part of the regular expressions is focussed for varied forms of eyes. A different part is focussed for extra features such as the noses. The main part of the regular expressions is the mouth. The mouth is the main feature to indicate the sentiment expressed by the emotion icon.

These can also appear inverted, having the mouth on the right and eyes to the left but maintain the sentiment (e.g ′:)′ inverted as ′(:′). Because of these inverted forms of the emotion icons we need to be aware of the mouth and eyes position.

```
eyes = "([:;=X8])";
nose = "([-oO^>'\"*]{0,1})";

regexPos = "[\s]?"+eyes+nose+"([)}DpP3]|])"+"\s?";
regexPosInv = "[\s]?"+"([({Cc])"+nose+eyes+"\s?";

regexNeg = "[\s]?"+eyes+nose+"([({\/CcSst])"+"\s?";
regexNegInv = "[\s]?"+"([)}\/DSs]|])"+nose+eyes+"\s?";
```

These regular expressions are built to find these emotion icons in a generic way, taking into consideration the ones most commonly used, but cannot identify them all. Some users are very creative on building their emotion icons, some even use characters from other alphabets to get the most unique ones.

5.3 Build Parse Tree

After the Twitter information extraction, the message will be parsed to get the parse tree tagged with POS. To parse the message, a statistical parser is used, adapted with a treebank corpus for the Portuguese language. The parser used is the *Stanford Parser* of the *The Stanford Natural Language Processing Group*, using the Java programming language API. The treebank corpus used is the *CINTIL TreeBank* done by the *CLUL - Center of Linguistics from the University of Lisbon* [BS06].

The most important parts to be extracted are entities and sentiment particles. These must be found and extracted based on their use in the message. The spell check would improve the parsing results, but spell checking previously can change certain nouns that could be entities to other similar correctly spelled n-grams, specially since our spell checker is for the Portuguese language and entities can be in the English language (e.g. 'barack' will be corrected to 'barraca' which means shack in English or 'macintosh' will be corrected to 'acintoso' which means blatant in English meaning doing openly and unashamedly bad behaviour). These entities will be found using the results of the parse tree, searching entities only based on nouns. The parser finds the most likely relations or connections between the different n-grams and assigns a probable POS.

For this reason spell checking will only occur after entity recognition, allowing to find entities, correcting the remaining sentence and build a more correct parse tree with the correct word POS. Entities that were not recognized and were incorrectly spelled, after the spell checking, nouns are searched again for entities.

Translation will not be used on the whole message for two reasons. Translation a whole message is much difficult than translating smaller n-grams. Translation could also change nouns that could be entities.

Taking for example the tweet "a siria está muito mal, nem a onu consegue ajudar", will result with the following parse tree.

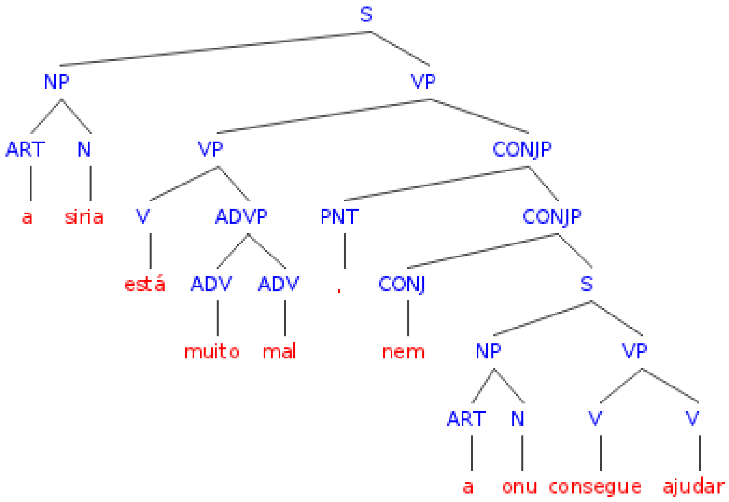


Figure 5.3: Parse tree example

In a few examples, the parser assigns the wrong POS. Any n-gram the parser does not identify, will classify as a noun. Some times these are verbs or adverbs.

5.4 Spell Checking

Before we translate the n-gram, spell checking is important to yield the best results.

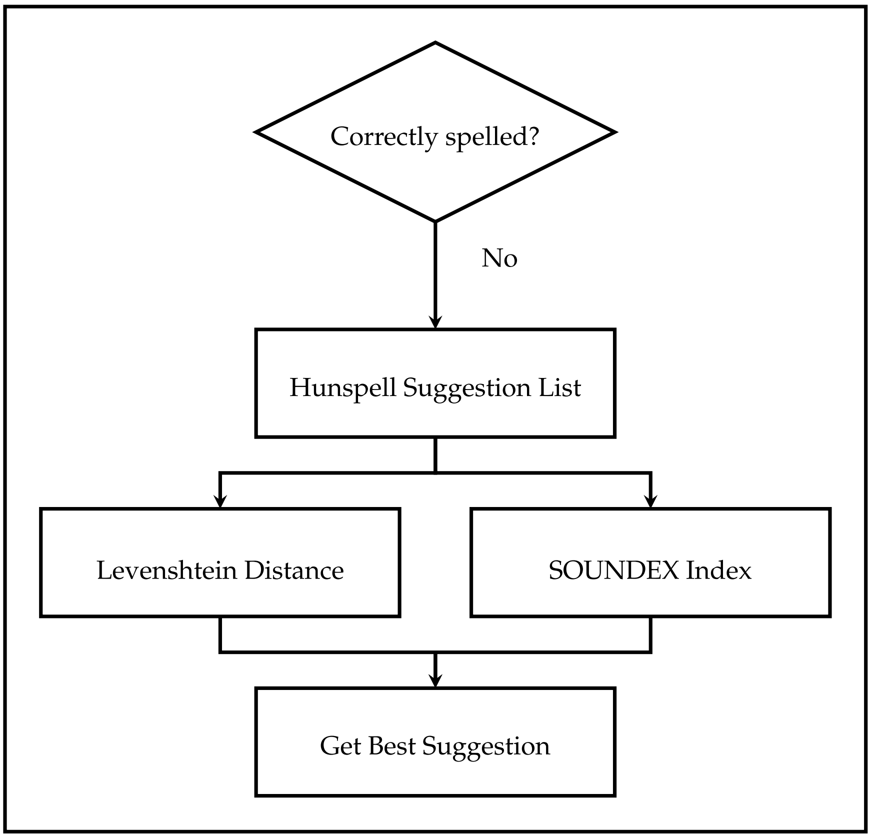


Figure 5.4: Spell Check Flow Diagram

For the spell checking the *Hunspell* spell checker is used. *Hunspell* can also search the dictionary if the words are correctly spelled. When words are misspelled, *Hunspell* does not give the correctly spelled words directly, instead it gives a list of suggestions for the incorrect words. The first entry of the suggestion list in not always the correct choice, so specific words have to be chosen from this list.

For this purpose we apply a combination of two algorithms, one focussed on the phonetic similarity (Soundex) and the other focussed on grammatical similarity. A ratio of these two algorithms is used, and since in the social networks spelling mistakes are phonetically similar to the correct word, the Soundex will be given a bigger importance in this ratio. These ratios can be manually configured but the selected default values are 40% for grammatical similarity and 60% for phonetic similarity.

5.4.1 Soundex

The Soundex is a sound indexing algorithm that will map sounds to a phonetic index. Similar indexes have similar phonetics. The more similar the Soundex indexes are, the more similar their phonetics.

The used Soundex phonetic indexes are build retaining the first letter of the word, removing the remaining vowels plus occurrences of y,h and w and replacing consonants with a value.

If two of more letters that have the same code in the index are adjacent, these are reduced to the first letter. Retain the first letter of the name and drop all other occurrences of a, e, i, o, u, y, h, w.

These are the Soundex indexes used for American English.

```
[b,f,p,v] => 1
[c,g,j,k,q,s,x,z] => 2
[d,t] => 3
[l] => 4
[m,n] => 5
[r] => 6
```

These indexes were adapted for the Portuguese language. The used Soundex is based on an existing study performed by Dimas Trevizan Chbane on converting text to phonetics in the Portuguese language, that takes into consideration the consonants articulation, point of articulation, vocal cords functions and mouth and nasal cavities, specific for consonants [Chb94].

```
[p,b,m] => 1
[f,v] => 2
[t,d,s,z,ç] => 3
[l,r,n] => 4
[x,j] => 5
[k,q,g,c] => 6
```

The resulting Soundex code is the first letter of the word followed by the first 3 index codes. The indexes do not have any relation between them, so the difference between any two different numbers is the same. These codes are then compared by their string similarity using the *Levenshtein Distance*, but normally there is an occurrence of an equal Soundex code in the spelling options.

For example result of applying the Soundex algorithm to the word 'Lisbon' is L314, 'Lissabon' is L314 and 'Lisboa' is L31.

5.4.2 Levenshtein Distance

The Levenshtein distance [Lev66] is used to calculate the minimum editing distance between two words. Adding to this distance, different letters or missing letters between the words.

Low values of the Levenshtein distance, more syntactically similar the words are.

$$lev_{a,b}(i,j) = \begin{cases} max(i,j) & \text{if } min(i,j) = 0 \\ min \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+[a_i \neq b_j] \end{cases} & \end{cases} \tag{5.1}$$

In this equation *a* and *b* are the words to be compared, *i* and *j* are indexes pointing to specific characters in the string. *i* has an initial value equal to the length of the string *a* and *j* has an initial value equal to the length of the string *b*.

This calculation has been optimized by using memoization, saving the calculated minimum values in a temporary table to be read instead of calculated for specific inputs. Since the function is recursive, the same combination of indexes *i* and *j* are often called.

5.5 Entity Recognition

Using the result of the parser, we can search for all the nouns in the message. Nouns are the most likely POS to be recognized as a named entity. Other POS, such as adverbs or adjectives, are not likely to be entities. Words starting with a capital letter are also taken into consideration to be possible named entities.

Entity recognition is done using dumps of information from DBpedia. These dumps contain information on the specific types of subjects. Information on other ways to write or specify certain entities are also available with these dumps. These are called redirects. For example "USA" will have a property redirect to "United Stated of America". Information on ambiguous entries and their possible disambiguation will also be available.

All this information will be filtered to the desired types taken into consideration.

- dbpedia.org/ontology/Person
- dbpedia.org/ontology/Place
- dbpedia.org/ontology/Company
- dbpedia.org/ontology/Organisation
- dbpedia.org/ontology/PoliticalParty
- dbpedia.org/ontology/Software
- dbpedia.org/ontology/Work

An additional entity type is taken into consideration, entries that are stated as being a product of an entity that belongs to one of the entity types, are also considered entities.

Noun phrases, composed of a series of nouns, are searched as a whole, applying an algorithm to find the largest named entity found within that noun phrase. The larger entity will be more specific, even if smaller entities can be found, most likely as ambiguous entities. Identifying a non ambiguous entity will result in an unique URI for that specific entity and the entity type as well. Ambiguous entities are entities that can be more than one named entity. These ambiguous entities will be disambiguated based on the remaining content of the message.

Recognized entities are limited by the specific types we desire to find. These are declared before running the process in order to filter unnecessary information. If the target entities are only people, no other entity type will be searched or found.

Using the same tweet example as before, "a siria está muito mal, nem a onu consegue ajudar", we can identify two nouns that will be recognized as named entities, associated with their unique URI.

For the noun "siria" we find the URI pt.dbpedia.org/resource/Síria with the URI type dbpedia.org/ontology/Place.

For the noun "onu" we find the URI pt.dbpedia.org/resource/Organização_das_Nações_Unidas with the URI type dbpedia.org/ontology/Organisation.

5.5.1 Entity Enumeration

Entity enumeration is useful to share sentiment to several entities, such as stating "I like X, Y and Z", or in Portuguese "Gosto de X, Y e Z". These entities share the sentiment particle and each will have this value assigned. The assigning process is explained in the next section.

In a sentence where several entities are found close together with no sentiment particles between them, these entities are grouped as an enumeration. The reason we do not group entities with sentiment particles in between is because this sentiment could have a different sentiment from all the other entities, it could even have a different sentiment polarity, such as stating "I like X and Y but not Z", or in Portuguese "Gosto de X e Y mas não de Z". This grouping provides a better sentiment attribution for all the entities found.

5.6 Entity Disambiguation

If a group of words may refer to different known entities, these result in being ambiguous. Some n-grams are ambiguous when they are not enough to fully specify the entity, for example 'Henry Ford' is not ambiguous, but reducing to 'Henry' can result in an ambiguous entity, being ambiguous between several people named Henry, such as 'Thierry Henry', 'Henry Charles' and 'Henry Ford'.

Entities are marked as ambiguous when they have a corresponding list of 2 or more possible specific entities. Ambiguous entities have little value since they cannot be disambiguated to a specific target.

To address this problem we apply two different approaches for disambiguation, using existing hashtags and similarity to other entities found.

5.6.1 HashTag Map

Using the previously extracted hashtags, relations can be created between found entities and hashtags. Also as previously said, hashtags provide some context for the message. Getting the relations from hashtags that appear with the message and we try to find a match with the ambiguous entity list. Since more than one of these entities can be found with the same hashtag, a counter is kept for each entity within a hashtag.

Here it will be illustrated with a different example than before. The noun "PT" can refer to one of the following entities:

- dbpedia-pt:Portugal
- dbpedia-pt:Partido_dos_Trabalhadores
- dbpedia-pt:Portugal_Telecom

Using the hashtag #countries in which a different messaged used something similar to "Love Portugal #countries" the new message with only "Love PT #countries" will be more likely to be referring to the entity pt.dbpedia.org/resource/Portugal with the type dbpedia.org/ontology/Place. This selection takes into consideration the number of times the entities have appeared with that specific hashtag. The entity with the higher count will be selected.

5.6.2 Property Similarity

A different approach to disambiguate entities is to find similarities with other entities found in the message. This uses the information supplied by *DBpedia*. This is similar to related work mentioned in Section 3.1.5, using DBpedia and SPARQL queries instead of searching for an existing Wikipedia pages of other known entities and trying to find any mention of the ambiguous entities, since mentioned entities are likely to be related to the other previously found entities.

Using a SPARQL call we can search for entities in the ambiguous entity list most similar to the entities already found. This takes into consideration if the entities are property of each other (e.g. Considering *Lisbon* and *Portugal* as entities, Lisbon is the capital of Portugal and Portugal has a city that is Lisbon), or if they share a common property (e.g. Considering *Lisbon* and *Porto* as entities, both are cities of *Portugal*). This gives a better disambiguation but as a result of the SPARQL call the processing time also increases.

Consider again the noun "PT", this noun can be one of many specific entities. Using the message "Gosto da Vodafone, mas não da PT" as an example, we find the entity pt.dbpedia.org/resource/Vodafone and the ambiguous entity "PT".

Using the following SPARQL query we find the number of similarities between two URIs. These similarities can be sharing the same predicate and object or one being the object of the other.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX : <http://dbpedia.org/resource/>
SELECT (COUNT(*) AS ?count)
WHERE {
  {
    # Share the same property and object pair
    <uri1> ?p ?o .
    <uri2> ?p ?o .
  } UNION {
    # URI1 contains URI2 as object
    <uri1> ?p <uri2> .
  } UNION {
    # URI2 contains URI1 as object
    <uri2> ?p <uri1> .
  }
}
```

In our example the entity pt.dbpedia.org/resource/Vodafone having more similarities with the entity pt.dbpedia.org/resource/Portugal_Telecom than with all the other possible entities, both being companies, organization and in the telecommunication industry.

5.7 Sentiment Classification

This section presents several steps taken to obtain a semantic sentiment classification of the existing n-grams of the message will be shown. The classification of these n-grams will result as sentiment particles in the message, maintaining their position in the parse tree and latter treated accordingly to their distance to other particles, such as entities.

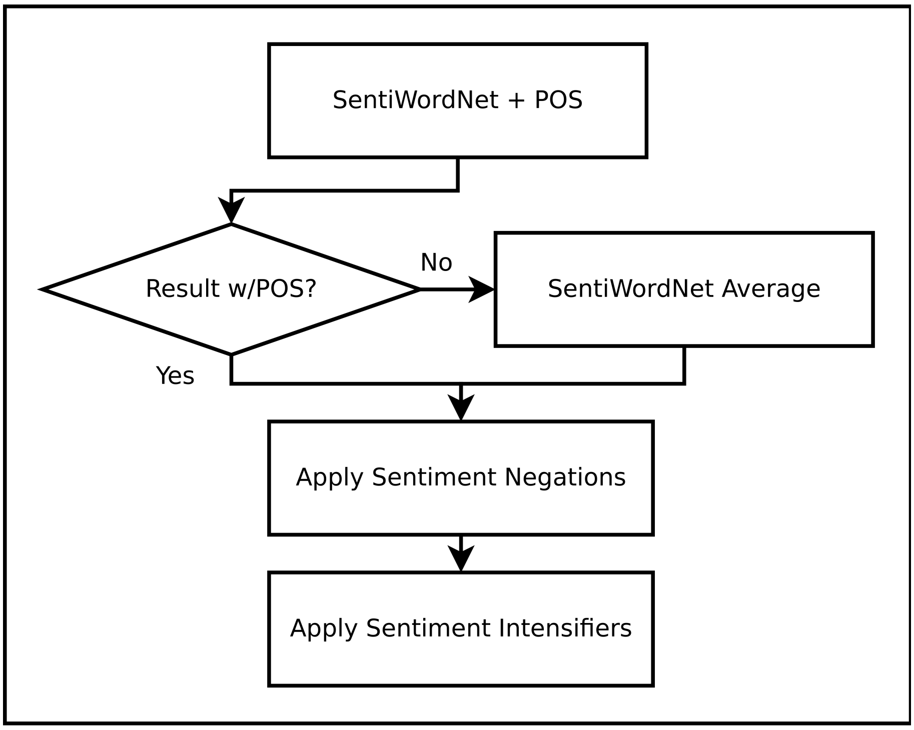


Figure 5.5: Sentiment Classification Flow Diagram

5.7.1 Sentiment Value

The sentiment value of the n-grams of a sentence are the main features for sentiment classification. For this sentiment classification the semantic classification approach will be used, described in Section 3.2.2. The sentiment value is found using *SentiWordNet* dumps, previously introduced in Section 4.3. These values are constantly evolving with the help of users that submit feedback on values that they do not agree with. This will influence future values of the existing n-grams. New n-grams are added with time, and with new POS. This provides a good evolving framework that is evaluated and maintained.

Since messages are in Portuguese and SentiWordNet is annotated for English n-grams, the translator mentioned in Section 4.7 is used. To get information from SentiWordNet, n-grams are translated and checked for their sentiment. Sentiment from SentiWordNet can be accessed as a simple dictionary, having a specific value for each POS of the existing n-grams.

For each n-gram, marked with POS in the parse tree, will be searched for their corresponding sentiment value, taking into consideration the specific POS. If the n-gram with that specific POS has no value, then an average value of all other POS for that n-gram is used instead. For example the unigram 'help' only has entries for the noun and verb as POS, if searched with a different POS, the average of the value as a noun and as a verb would be used.

The sentiment of n-grams is searched, starting from unigrams and adding words, until no sentiment is found for the last n-gram. The resulting sentiment of the n-gram is the sentiment assigned to the largest valid n-gram.

Using our previous tweet example, we can find the following sentiment particles in the tweet.

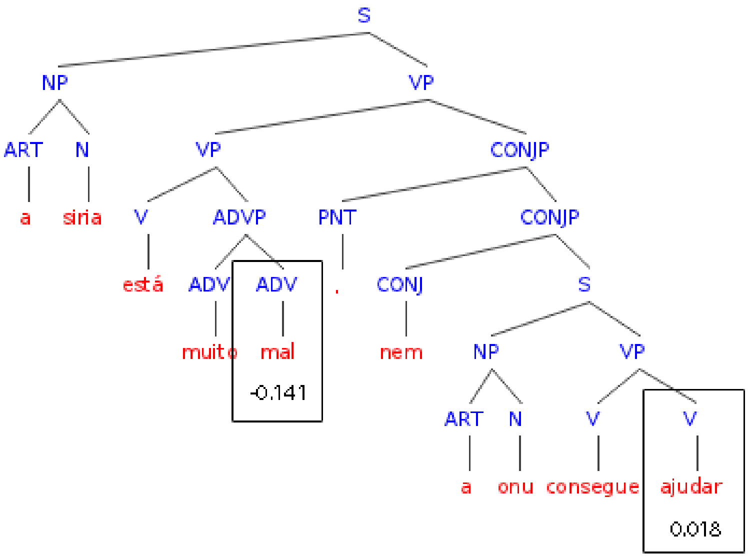


Figure 5.6: Sentiment tree example

Only two unigrams were found containing sentiment value. The unigram 'mal' was translated to the unigram 'wrong', and searching SentiWordNet for the unigram 'wrong' with the adverb POS, the sentiment is -0,141. The unigram 'ajuda' was translated to 'help', and searching SentiWordNet for the unigram 'help' with the verb POS, the sentiment is 0,018.

5.7.2 Sentiment Negations

Some sentences may contain words that counter the sentiment of other words in the same sentence, such as "not good" or "didn't like it". These key negative words will be contained in a table of known negative n-grams.

When found, these n-grams will change the polarity of the sentiment value of the closest sentiment particle, switching positive to negative and negative to positive. This is a very important step that is not mentioned in related work and can affect the sentiment value drastically.

These negative n-grams are marked and will change the sentiment value of the closest sentiment particle according to the parse tree relations. Negative n-gram will only influence sentiment particles that are within a default range. This default range has a distance value of 10 based on the parse tree, this value was established by observing some negation rules for the 'no', 'never' and 'not' adverbs, defined by Jin-Cheon Na [NKW05]. This distance is calculated by summing the number of edges connecting the nodes in the tree.

When no sentiment particles are found within the default range, either existing ones being distant or no sentiment particles exist following this n-gram, the negative n-gram will not influence other sentiment particles and is then attributed a default negative sentiment value, equal to the SentiWordNet of the n-gram value 'not' with a value of -0.625.

Double negations are applied as the grammatical Portuguese rules. When a negative n-gram that has not been applied only finds a sentiment particle that has already been the target of a different negative n-gram, the not applied negative n-gram will be attributed the default negative sentiment value.

Using our previous example, we find a negative particle that will influence the closest sentiment particle.

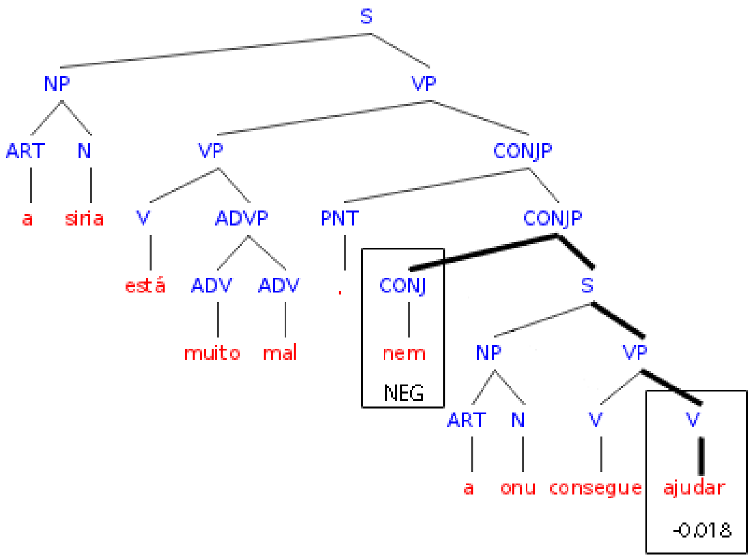


Figure 5.7: Sentiment negation tree example

In this example the distance between the negative n-gram and the following sentiment particle if equal to 4, the sum of the edges to the lowest common ancestor in the tree.

5.7.3 Sentiment Intensifiers

Similar to negations, sentiment intensifiers are n-grams that influence the value of some sentiment particles. Some examples are "more" or "less". These n-grams are kept in a table of known intensifier n-grams.

When found these n-grams will influence the sentiment value of a close particle, lowering or increasing their positive or negative values.

This new value is calculated with the following equations.

Increasing intensifier:

$$V1 = \begin{cases} \|V\|^{1/x} & \text{if } V \geq 0 \\ -\|V\|^{1/x} & \text{if } V < 0 \end{cases}, V \in [-1.0, 1.0], x > 1 \tag{5.2}$$

Lowering intensifier:

$$V1 = \begin{cases} \|V\|^{x} & \text{if } V \geq 0 \\ -\|V\|^{x} & \text{if } V < 0 \end{cases}, V \in [-1.0, 1.0], x > 1 \tag{5.3}$$

The exponential variable x = 2 by default, but can be changed to influence more or less the original value. This results in the following plots:

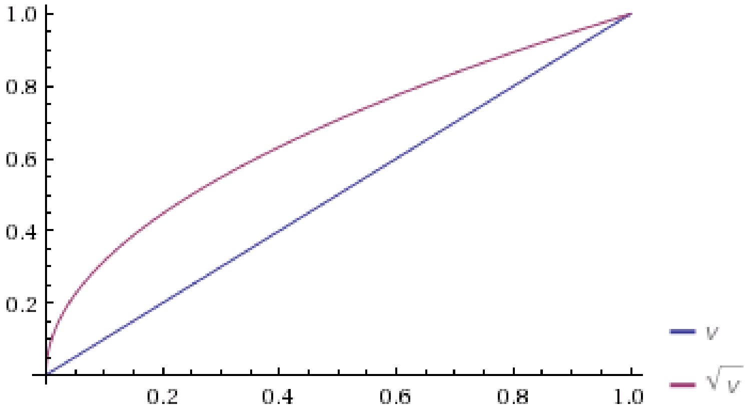


Figure 5.8: Sentiment increasing intensifier example

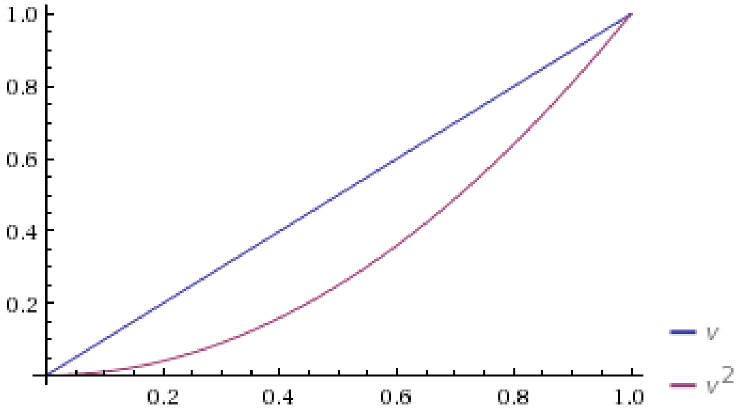


Figure 5.9: Sentiment lowering intensifier example

Intensifier n-grams will only change the sentiment value of the closest sentiment particle according to the parse tree relations. Similar to the negative n-grams, the intensifiers n-grams will only influence sentiment particles that are within a default range, the same range as the negations in Section 5.7.2 with a distance value of 10 based on the parse tree.

When no sentiment particles are found within the default range, the intensifiers n-grams are attributed a default positive sentiment value, equal to the SentiWordNet of the n-gram value 'very' with a value of 0.5, and a default negative sentiment value equal to the SentiWordNet of the n-gram value 'less' with a value of -0.5.

Using our previous example, we find an intensifier particle that will influence the closest sentiment particle.

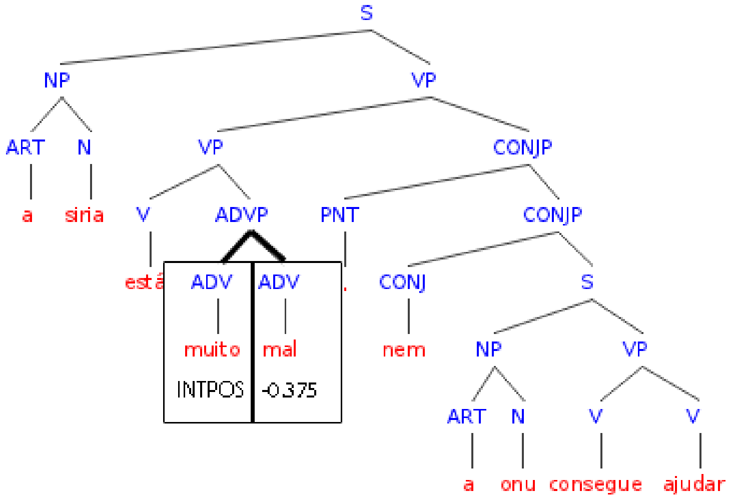


Figure 5.10: Sentiment intensifier tree example

5.8 Sentiment Assigning

After all the sentiment values and entities have been found, we need to connect these based on the relations between them in the sentence. Using the parse tree, a distance is calculated between the sentiment nodes and entities. This distance takes into consideration the direction of the sentiment to the entity, since typically in Portuguese as well as in English, it is written and read from left to right.

The distance is equal to the sum of the edges to the lowest common ancestor in the tree. This distance has obtained more consistent results than just using a linear distance between the n-grams. The entity with the lowest distance to the sentiment node, will be assigned the sentiment value of that node.

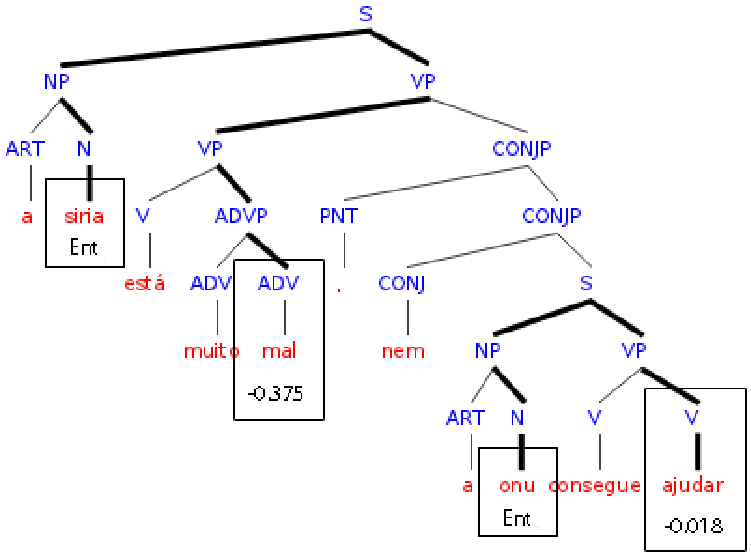


Figure 5.11: Sentiment assigning example

Considering entity enumerations, the sentiment particles will be assigned to all entities in the enumeration.

5.9 Entity Sentiment

Since entities may have several sentiment nodes assigned to them, we need to attribute a single value to that entity that remains between 1 and -1. For this purpose a T-norm or a T-conorm (or S-norm) could be used.

Both T-norms and T-conorms share the following properties:

- Commutativity

$$T(a,b) = T(b,a) \tag{5.4}$$

- Monotonicity

$$T(a,b) \leq T(c,d), a \leq c, b \leq d \tag{5.5}$$

- Associativity

$$T(a,T(b,c)) = T(T(a,b),c) \tag{5.6}$$

The identity element property diferenciates between the T-norm an the T-conorm.

- T-norm Identity element

$$T(a,1) = a \tag{5.7}$$

- T-conorm Identity element

$$T(a,0) = a \tag{5.8}$$

For this calculation the T-conorm Einstein Sum was chosen, generalized for the interval [-1,1]. This is possible since the T-conorm Einstein Sum is a symmetrical function and was obtained from Einstein's velocity addition formula in physics, applicable with any real number with the exception when the denominator is equal to zero.

$$\perp_{H2}(a,b) = \begin{cases} \frac{a+b}{1+a*b} & \text{if } 1+a*b \neq 0 \\ 0 & \text{if } a*b=-1 \end{cases}, a,b \in [-1.0, 1.0] \tag{5.9}$$

The main reason for the choice of this function, is the addition of the inverse property of addition on the numerator of the main fraction. This will result in any two values that are additive inverse will return 0. In other words a positive value with an equal negative value will result in a neutral value. This is shown in the following equation by replacing the value b with -a.

$$\perp_{H2}(a,-a) = \frac{a+(-a)}{1+a*(-a)} = \frac{0}{1-a^2} = 0, a \neq 1 \tag{5.10}$$

This function also re-enforces values, allowing two positive values to become a slightly higher or equal positive value, and the same for negative values.

The adapted Einstein sum t-conorm equation results in the following graphs.

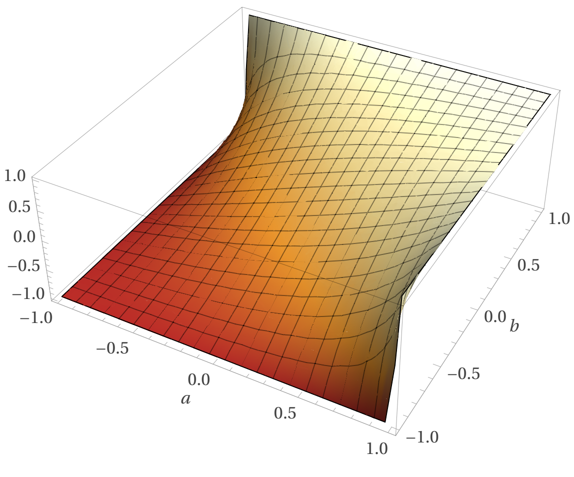


Figure 5.12: Adapted Einstein Sum T-conorm 3D Plot

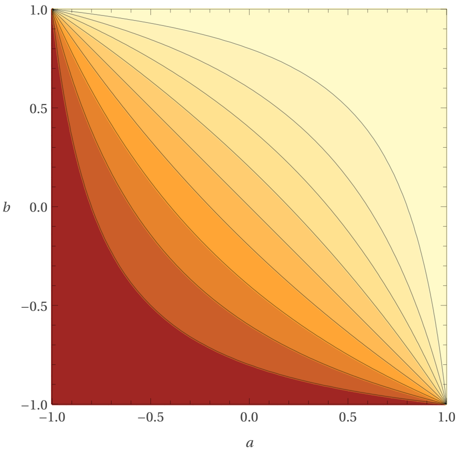


Figure 5.13: Adapted Einstein Sum T-conorm Contour Plot

In Figure 5.13 each line is represents equal subdivisions on the values 1 and -1, in this case each line adds or sums 0.2 from the line marking 0. So going from 0 to 1 each line values 0.2, 0.4, 0.6 and 0.8, leaving the top and right margin of the figure with value 1.

5.10 Thread Pool

Since the income of messages from the stream can be slightly faster than the processing time of each message, a solution to process all messages as they arrive from the stream is to have a thread pool ready to process several messages at the same time.

Each thread is responsible for processing a single message read from the stream sharing most of the tools. All shared tools have been prepared for concurrent use. Tools and methods that are not able to be used concurrently are multiplied. In this specific case each thread has its own translator process and its own parser instantiation.

5.11 Conclusions

This semantic approach provides a sentiment classification without the need for training data, using a broad set of data that is constantly evolving for the classifications. DBpedia is used for entity recognition and SentiWordNet is used for sentiment classification.

This approach assigns to each entity found in the messages their own sentiment, calculated based on the parsing tree of the message annotated with POS.

A single processed message does not supply much information, but with the use of URIs for the entities, the results of several messages can give us a better understanding on the public sentiment.

6

Evaluation and Results

To evaluate the results of this thesis, we will compare our results with similar projects using some existing benchmarks. We shall analyse the entity recognition and sentiment classification portion separately.

6.1 Entity Recognition

6.1.1 Making Sense of Microposts #MSM2013

Making Sense of Microposts is the 3rd workshop at the *International World Wide Web Conference* of 2013. The aim of this workshop is to show and discuss the approach on processing micro posts. Micro posts are messages sent through the social media (Twitter, Facebook, Google plus, etc.), with typically with no more than 140 characters.

A challenge was also hosted by this workshop with the aim of concept extraction or also known as named entity recognition (NER). The goal of this concept extraction challenge is to find and extract concepts or entities from several of messages. At least 35 teams participated in this challenge but only 14 teams were accepted. Team sizes varied from single person teams to teams of five members.

6.1.1.1 The Dataset

The concept extraction challenge dataset is formed by 4341 manually annotated sentences. The training data is composed of 2815 (roughly 65%) sentences and the test data is composed of 1526 (roughly 35%) sentences. The test data is used for the teams to evaluate their own work while they are developing their approaches.

The gold standard, composed of 1450 manually annotated sentences, is used for the evaluation. The entity types that are considered in this challenge are, *Person*, *Location*, *Organization* and *Misc*. The type *Misc*, is considered as *EntertainmentAwardEvent*, *SportsEvent*, *Movie*, *TVShow*, *PoliticalEvent* or *ProgrammingLanguage*.

6.1.1.2 Evaluation

For a valid comparison between the different approaches, the same evaluation method must be used. This evaluation was also applied to this thesis and compared to the rest of the submitted results. Precision, Recall, and F-measure are calculated for each entity type (Person, Location, Organization and Misc). The final value is the average calculated for each metric for all entity types.

For this we compare our results to a gold standard considering the dataset. This standard has the expected results for each entry in the dataset.

Entities correctly found and correctly classified are considered TP (True Positive). Entities correctly found but incorrectly classified and considered as FP (False Positive) and FN (False Negative). Other entities found that are not in the gold standard are considered FP (False Positive). Entities in the gold standard that are not found are considered FN (False Negative). The remaining words are considered TN (True Negative), but this is not relevant for the following calculations.

With these values summed for each type of entity we then proceed to calculate the precision, recall and f-measure.

The calculated precision is the ratio of relevant entities found out of all the entities returned.

$$Precision = \frac{TP}{TP + FP} \tag{6.1}$$

The calculated recall is the ratio of relevant entities found out of all known entities.

$$Recall = \frac{TP}{TP + FN} \tag{6.2}$$

The precision and recall are usually inverse proportional, meaning that when one values goes higher, the other drops. The f-measure is the harmonic mean of the precision and recall measures.

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{6.3}$$

Since the training data is not necessary for our approach, and therefore two results are shown, one using only the gold standard data, used for a better close comparison to the other projects, and one also using the training data in addition to the gold standard. This shows the difference in the results using more data.

6.1.1.3 External Knowledge Sources and Other Tools

Contestants could use external tools for their projects. These included knowledge sources to improve their available information. The used tools could help them to correctly identify entities, including POS taggers, parsers and even tools specific for NER. The main knowledge sources used are Yago[1] and DBpedia[2]. Additional knowledge sources were also used by some contestants, including CONLL-2003[3], ACE-2004, ACE2005, JRC Names corpus, Microsoft N-grams, WordNet, Wikipedia, BabelNet, Samsad, NICTA and Gazetteers(geonames, ANNIE, BALIE).

Some contestants used external tools or a combination of external tools. The tools used by some of the contestants were AIDA, Alchemy, DBpedia Spotlight, Extractiv, OpenCalais, Textrazor, Wikimeta, Stanford NER, twitter_nlp, ANNIE, OpenNLP, Illinois NET, Illinois, Wikifier, LingPipe, WikiMiner, Zemanta. These are natural language processing tools used for entity recognition and disambiguation.

6.1.1.4 Results

Three tables are provided, comparing results on the resulting precision, recall and f-measure for all the data and the gold standard. For the final classification the f-measure is taken into consideration over the others since the f-measure is a harmonic mean of the precision and recall.

Each line represents a team entry, showing their rank, team number and classification results for each entity type. The entity types that are considered in this challenge are, *Person* as 'PER', *Location* as 'LOC', *Organization* as 'ORG' and *Misc* as 'MISC'. The entry 'ALL' is the average of the results of all the entity types. The rank is ordered by the values of the entry 'ALL'. The best values for each type are in bold for better viewing, and when in draw, the one with the higher rank is chosen.

In summary, the values for all data, composed of the training data and the gold standard, achieved better results than just using the gold standard, but this result using all the data should not be taken into consideration for comparing to the other teams results, since the shown results of the teams are using the gold standard, and should only be compared to our results using only the gold standard. This allows to view the difference in the obtained results using more data.

---

[1] `http://mpi-inf.mpg.de/yago-naga/yago/`

[2] `http://dbpedia.org/`

[3] `http://www.cnts.ua.ac.be/conll2003/ner/`

Rank	Team Number	PER	ORG	LOC	MISC	ALL
1	14	**0.92**	**0.64**	0.74	0.38	**0.67**
2	21	0.91	0.61	0.72	**0.41**	0.66
3	15	0.92	0.57	**0.79**	0.36	0.66
4	20	0.83	0.61	0.62	0.38	0.61
5	25	0.83	0.49	0.74	0.30	0.59
6	03	0.87	0.56	0.74	0.19	0.59
7	29	0.76	0.54	0.59	0.36	0.56
8	28	0.81	0.41	0.71	0.24	0.54
->	all_data	0.72	0.47	0.70	0.28	0.54
9	32	0.73	0.35	0.59	0.41	0.52
->	gold_standard	0.76	0.41	0.58	0.29	0.51
10	30	0.71	0.38	0.58	0.31	0.49
11	33	0.85	0.37	0.62	0.14	0.49
12	35	0.82	0.42	0.60	0.12	0.49
13	23	0.83	0.52	0.50	0.04	0.47
14	34	0.54	0.37	0.53	0.16	0.40

Table 6.1: #MSM2013 F-Measure Results

Rank	Team Number	PER	ORG	LOC	MISC	ALL
->	all_data	0.92	0.54	0.71	**1.0**	**0.79**
1	14	**0.93**	0.67	**0.89**	0.62	0.78
2	21	0.88	0.60	0.88	0.71	0.77
3	30	0.83	0.65	0.82	0.67	0.74
4	15	0.89	0.69	0.85	0.53	0.74
5	33	0.81	**0.71**	0.75	0.64	0.73
->	gold_standard	0.93	0.44	0.53	1.0	0.73
6	23	0.72	0.42	0.62	1.0	0.69
7	29	0.79	0.60	0.82	0.55	0.69
8	25	0.77	0.61	0.82	0.55	0.69
9	03	0.82	0.70	0.80	0.43	0.69
10	28	0.77	0.67	0.71	0.50	0.66
11	20	0.81	0.64	0.75	0.34	0.63
12	32	0.71	0.43	0.69	0.43	0.57
13	35	0.74	0.53	0.72	0.14	0.53
14	34	0.41	0.55	0.67	0.38	0.50

Table 6.2: #MSM2013 Precision Results

Rank	Team Number	PER	ORG	LOC	MISC	ALL
1	21	0.94	0.61	0.61	0.29	**0.61**
2	15	**0.95**	0.48	**0.74**	0.27	0.61
3	14	0.91	0.61	0.63	0.28	0.61
4	20	0.86	0.59	0.53	**0.42**	0.60
5	03	0.93	0.46	0.69	0.12	0.55
6	25	0.89	0.41	0.68	0.20	0.54
7	23	0.97	**0.67**	0.43	0.02	0.52
8	28	0.87	0.29	0.70	0.15	0.50
9	29	0.74	0.49	0.46	0.26	0.49
10	32	0.74	0.29	0.51	0.39	0.48
11	35	0.92	0.35	0.51	0.10	0.47
->	all_data	0.59	0.41	0.67	0.16	0.46
->	gold_standard	0.63	0.38	0.63	0.17	0.46
12	33	0.88	0.25	0.52	0.08	0.43
13	34	0.79	0.28	0.43	0.10	0.40
14	30	0.62	0.27	0.45	0.20	0.39

Table 6.3: #MSM2013 Recall Results

The overall precision has good results with a value of 73% achieving the equivalent of the 6th place in the precision overall ranking, but has lower recall with a value of 46% achieving the equivalent of the 12th place in the recall overall ranking. The f-measure resulted in 51% got the equivalent of the 10th place in the overall ranking. From the results our approach performed worse for organizations and misc and better for people and locations.

While our precision obtained good results, the obtained recall is worst, indicating there were a larger occurrence of false negative observations, entities that exist in the sentence but were not identified by our approach. One cause of this is that these entities were ambiguous and could not be disambiguated. This lower recall influences the f-measure results.

While not using complicated systems for entity recognition, using all sorts of tools, combination of tools and data, our overall results were acceptable. The team that achieved the 2nd place in the overall ranking, used a combination of existing NER tools (AlchemyAPI, DBpedia Spotlight, Extractiv, Lupedia, OpenCalais, Saplo, Yahoo, Textrazor, Wikimeta, Zemanta and Stanford NER), to get the most probable entities that could be extracted. As expected some of these tools are not in sync with each other and returned different results in some occasions and return some ambiguous entities, but overall achieved good results [ERT13].

The team that achieved 1st place in the overall ranking used a hybrid approach using CRF (Conditional Random Fields) and SVM (Support Vector Machines) enriched using

Yago KB data. The AIDA tool was used for disambiguation [YHBSW11].

6.2 Sentiment Particle Classification

6.2.1 SentiLex-PT01

SentiLex-PT01[4] is a sentiment lexicon build to classify sentiment particles in the Portuguese language. This lexicon consists of adjectives in the masculine single form and their corresponding inflected form.

Part of the adjectives are human annotated, while the other are machine annotated. Their machine classification obtained these results on a sample of about 10% (285) of the random machine annotated unigrams and then manually verified these results. [Sil]

Polarity	F-Measure	Precision	Recall
Positive	0.66	0.67	0.66
Neutral	0.50	0.45	0.56
Negative	0.78	0.82	0.74
All	0.65	0.65	0.65

Table 6.4: SentiLex-PT01, Their machine classification results on 10% of the lexical data used for further annotations

Table 6.4 refers to their machine classification results that will annotate the remaining adjectives. Further tables in the section refers to our system results.

6.2.1.1 Evaluation

Since our sentiment particles have real number values, these will have to be fitted to one of these 3 classes by boundaries, and translated to their corresponding -1, 0 or 1 values. For this purpose an interval is needed for each sentiment type. For this specific case, specifying the interval for the neutral sentiment will define the positive, with a maximum value of 1, and negative, with a minimum value of -1.

The interval taken into consideration was achieved using 10-fold cross-validation with the SentiLex-PT adjectives. This 10-fold cross-validation is a commonly used K-fold cross-validation. This data has also been stratified, meaning that each fold will have roughly the same number of occurrences of each sentiment class.

This validation resulted in the following intervals for the classes.

$$Positive \in]0.143, 1[\tag{6.4}$$

$$Neutral \in [-0.009, 0.143] \tag{6.5}$$

---

[4] `http://dmir.inesc-id.pt/project/SentiLex-PT_01`

$$Negative \in [-1, -0.009[\tag{6.6}$$

6.2.1.2 SentiLex-PT01 Adjectives

The lexicon is composed of 6321 positive, neutral and negative classified unigrams adjectives in the masculine single form. From the lexicon about 13.6% (859) of the unigrams were not recognized by the used translator. These unigrams will not be considered for the sentiment classification results. The remaining 5462 unigrams are marked with -1, 0 and 1 for the sentiment. These 54.3% (2968) of the adjectives are human annotated and 45.7% (2494) are machine annotated.

Table 6.5 shows the sentiment classification taking into account the human annotated adjectives.

Polarity	F-Measure	Precision	Recall
Positive	0.480	0.537	0.435
Neutral	0.454	0.353	0.633
Negative	0.654	0.791	0.558
All	0.551	0.560	0.542

Table 6.5: SentiLex-PT01 Results with human annotated data

Table 6.10 shows the sentiment classification taking into account the human and machine annotated adjectives.

Polarity	F-Measure	Precision	Recall
Positive	0.409	0.497	0.347
Neutral	0.417	0.310	0.637
Negative	0.584	0.758	0.475
All	0.504	0.522	0.487

Table 6.6: SentiLex-PT01 Results with human and machine annotated data

Considering the human and machine annotated data, the results have lower values compared to the results with only human annotated data. This decrease is of about 5% on average.

6.2.1.3 SentiLex-PT01 Inflected Form

Each of the adjectives mentioned in Section 6.2.1.2 are inflected with the combination of gender, feminine and masculine, as well as number, singular and plural. This will test the performance change with these inflected forms.

These inflected forms of the adjectives have a total of 25406 positive, neutral and negative, human and machine annotated unigrams. Out of all the inflected form unigrams, 12.6% (3203) were not recognized by the translator. These unigrams will not be considered for the sentiment classification.

The remaining 22203 unigrams are marked with -1, 0 and 1 for the unigram sentiment, 54.9% (12183) human and 45.1% (10020) machine adjectives in their inflected forms.

Polarity	F-Measure	Precision	Recall
Positive	0.396	0.492	0.332
Neutral	0.417	0.306	0.656
Negative	0.559	0.755	0.444
All	0.497	0.518	0.477

Table 6.7: SentiLex-PT01 Results with human and machine annotated data in the inflected form

6.2.1.4 Results Conclusion

These tests are made without taking into consideration the n-grams that the translation could not recognize, about 13% of data. This was made to have more reliable results for the sentiment classification. The parser correctly classified all adjectives.

The sentiment misclassification are more common to occur from positive to neutral, negative to neutral and neutral to either positive or negative, since the sentiment is turned from a real number, with values between -1 and 1 , to an integer value of 1, 0 or -1.

These close errors compose around 70% of the total errors, leaving the remaining 30% of the errors to be classifying positive unigrams as negative and vice versa.

The inflected form of the adjectives does not influence much on the classification, because these inflected forms are brought back to their root word by the translator through stemming. Most of the misclassifications done in the adjectives in the singular form are the same misclassifications done in their corresponding inflected forms.

6.2.2 SentiLex-PT02

SentiLex-PT02[5] is an updated version of the sentiment lexicon SentiLex-PT01. [SCS12] Maintaining the same structure as the previous version, but changing some of the sentiment particle values and adding more entries.

The machine annotation are done using the same machine classification from their previous version stated on Section 6.2.1.

For the same reason as the previous version, our sentiment particles have real number values and these will have to be fitted to one of these 3 classes by boundaries, and translated to their corresponding -1, 0 or 1 values.

---

[5] `http://dmir.inesc-id.pt/project/SentiLex-PT_02`

The evaluation method is the same as the previous version SentiLex-PT01, maintaining the same intervals used to distinguish positive, neutral and negative values. This will provide a direct comparison between both versions.

6.2.2.1 SentiLex-PT02 Adjectives

The lexicon is composed of 7014 positive, neutral and negative adjectives n-grams in the masculine single form. Out of all existing adjectives, 14.7% (1028) were not recognized by the translator. These n-grams will not be considered for the sentiment classification.

The remaining 5986 n-grams are marked with -1, 0 and 1 for sentiment. These classification are about 77.3% (4626) human and 22.7% (1360) machine annotated n-grams.

The table 6.8 shows the sentiment classification taking into account the human annotated adjectives.

Polarity	F-Measure	Precision	Recall
Positive	0.437	0.573	0.353
Neutral	0.221	0.139	0.540
Negative	0.647	0.818	0.535
All	0.492	0.510	0.476

Table 6.8: SentiLex-PT02 Results with human annotated data

The table 6.9 shows the sentiment classification taking into account the human and machine annotated lemmas.

Polarity	F-Measure	Precision	Recall
Positive	0.437	0.570	0.355
Neutral	0.243	0. 155	0. 562
Negative	0. 631	0. 817	0.514
All	0.495	0. 514	0.477

Table 6.9: SentiLex-PT02 Results with human and machine annotated data

Considering all the data, the results have increased values compared to results only using the human annotated data. This increase is less than 1% on each metric. This might be a result of correcting the existing data and adding more data to the previous version mentioned in Section 6.2.1.

6.2.2.2 SentiLex-PT02 Inflected Form

Each of the adjectives mentioned in Section 6.2.2.1 are inflected with the combination of gender, feminine and masculine, as well as number, singular and plural. This will test the performance change with these inflected forms.

These inflected forms of the adjectives in the singular form have a total of 82347 positive, neutral and negative n-grams. Out of all the inflected form n-grams, 17.8% (14694) not recognized by the translator. These n-grams will not be considered for the sentiment classification.

The remaining 67653 n-grams marked with -1, 0 and 1 for the n-gram sentiment, 92.6% (62663) human and 7.4% (4990) machine inflected forms corresponding to the annotated adjectives.

Polarity	F-Measure	Precision	Recall
Positive	0.269	0.431	0.195
Neutral	0.156	0.091	0.552
Negative	0.563	0.777	0.442
All	0.414	0.433	0.396

Table 6.10: SentiLex-PT02 Results with human and machine annotated data in the inflected form

6.2.2.3 Results Conclusion

Having similar results to the previous version of this data mentioned in Section 6.2.1, the conclusions remain the same as stated in Section 6.2.1.4.

Most of the existing errors were errors to a close class.

The inflected form of the adjectives does not influence much on the classification, even though in this version this difference is bigger. The misclassifications done in the adjectives in their single form are the same misclassifications done in their inflected forms.

6.3 Sentiment Classification on Messages

6.3.1 Sentiment Classification on a Random Sample of Tweets

To test sentiment classification 200 random tweets were collected. These tweets were manually annotated for sentiment by 3 volunteers. As expected, these volunteers were in agreement in 78.5% of the tweets, 19% only 2 were in agreement and in 2.5% there was no agreement.

When a majority had an agreement, the sentiment of the majority was selected. When there was no agreement between the human annotators, the selected sentiment was neutral.

These tweets are composed of 60 positive, 71 neutral and 69 negative tweets. This composition was unintentional but fortunate to be have very close number of occurrences.

6.3.1.1 Results

Polarity	F-Measure	Precision	Recall
Positive	0.577	0.627	0.533
Neutral	0.615	0.531	0.732
Negative	0.583	0.686	0.507
All	0.603	0.615	0.591

Table 6.11: Results from a Random Twitter Sample

Similar to the other results, 76% of all errors were to the closest class, misclassifying positive as neutral, negative as neutral and neutral as either positive or negative.

Taking into consideration only the tweets were the human annotators were in complete agreement has shown a slight increase in the results for positive and neutral classes.

Polarity	F-Measure	Precision	Recall
Positive	0.593	0.686	0.522
Neutral	0.662	0.550	0.830
Negative	0.560	0.667	0.483
All	0.623	0.634	0.612

Table 6.12: Results of a Random Twitter Sample with Annotators Agreement

Similar to the other results, 73.8% of all errors were to the closest class, misclassifying positive as neutral, negative as neutral and neutral as either positive or negative.

This next test was done using the Sentilex-PT02 as the source of sentiment annotated n-grams, instead of using the combination of the translator and SentiWordNet.

Polarity	F-Measure	Precision	Recall
Positive	0.406	0.609	0.304
Neutral	0.626	0.489	0.868
Negative	0.531	0.650	0.448
All	0.561	0.583	0.540

Table 6.13: Results of a Random Twitter Sample, using SentiLex-PT02

Even though Sentilex-PT02, annotated with sentiment for n-gram in the Portuguese language, it achieved worst results than using the translator and SentiwordNet. Annotating n-grams with an integer value of 1, 0 and -1 to represent positive, neutral and negative sentiment is less flexible than using real values between 1 and -1.

In this next test, all the messages were translated to the English language before processing, using *Google Translate*[6]. These messages were processed according to the language, using the same parser but for the English language, the spell checker uses an English dictionary and since the translator is not needed, it will be disabled for messages in English.

Polarity	F-Measure	Precision	Recall
Positive	0.641	0.781	0.543
Neutral	0.636	0.490	0.906
Negative	0.471	0.741	0.345
All	0.632	0.671	0.598

Table 6.14: Results of a Random Twitter Sample Translated to English

The difference in the results is a good indicator that the parser performs better for messages in English than with messages in Portuguese. Google translator also outperforms the translator previously used.

6.4 Sentiment Classification on Political Tweets

6.4.1 SentiTuites-01

SentiTuites-PT is a corpus of tweets posted by Portuguese users during the campaign for the 2011 Portuguese legislative elections, collected from 29th of April to the 3rd of June of 2011. This corpus was produced by the *LASIGE* group from the *University of Lisbon* [CSTS11].

This corpus is composed of 11,376 manually annotated sentences. Each sentence will have information on the specific entity target and a corresponding sentiment value for that specific target. Some messages are duplicated, but have a different entity target with the sentiment value for this new target. Each sentence is annotated for sentiment with the values -1, 0 and 1, corresponding to negative, neutral and positive sentiment.

This corpus is composed of 1474 positive, 3721 neutral and 6181 negative sentences.

The entity targets for these messages are the Portuguese politicians Pedro Passos Coelho, José Sócrates, Paulo Portas, Jerónimo de Sousa and Francisco Louçã.

6.4.1.1 Evaluation

Similar to previous evaluations, since our results are real values and the annotated data has their results with 1, 0 and -1 to represent positive, neutral and negative sentiment, these real values must be converted by the use of intervals.

---

[6] `http://translate.google.com/`

The interval taken into consideration was achieved using 10-fold cross-validation with the SentiTuites gold standard. This 10-fold cross-validation is a commonly used K-fold cross-validation. This data has also been stratified, meaning that each fold will have roughly the same number of occurrences of each class.

This validation resulted in the following intervals for the classes.

$$Positive \in]0.33, 1[\tag{6.7}$$

$$Neutral \in [-0.35, 0.33] \tag{6.8}$$

$$Negative \in [-1, -0.35[\tag{6.9}$$

Surprisingly the 10-fold cross-validation split the intervals almost equally for each class, each class having approximately 1/3 of the interval between -1 and 1.

With our results the sentiment values of the messages will be aggregated for each entity and will be compared to the other entities considered. For each entity we will calculate two values to compare the sentiment between the entities. These calculations are based on the messages regarding the specific entity. The first value is our twitter prediction calculation, calculated by the proportion of sum of all sentiment extracted and message count for each entity. The other value takes is the number of positive tweets for each entity, in this specific case of political elections, a positive message may imply a vote for that politician.

Based on these calculations a proportion is calculated between these entities and will represent the poll percentage for each entity using this data.

To compare our proportion results to the election final results we calculate the Pearson's correlation coefficient between these values, using Equation 6.10.

$$Corr(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2 \sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{6.10}$$

The more similar the values are, the closer the value of the correlation coefficient is to 1. A correlation coefficient equal to 1 means that the prediction data is equal to the results.

The mentioned calculations are simple examples to show what can be achieved and more complex calculations can be used.

6.4.1.2 Results and Conclusions

This data has specific domain, regarding politics, as opposed to the random sampling that had none. The main problem with the political domain is that sarcasms and irony are very common, unlike product reviews, which make political opinions harder to deal with [Liu12].

Polarity	F-Measure	Precision	Recall
Positive	0.209	0.216	0.201
Neutral	0.500	0.357	0.835
Negative	0.230	0.662	0.139
All	0.402	0.412	0.392

Table 6.15: SentiTuites-01 Results with human annotated data

Similar to other results, about 87% of the sentiment classification errors are misclassified to the closest class, in this case misclassifying positive as neutral, negative as neutral and neutral as either positive or negative.

These results cannot be directly compared to the results on the original team, since they only used 881 tweets, around 8% of the manually annotated data [ST11]. The selected 8% is not clear, not specifying how these were selected or if they were randomly selected.

Using this data we can try to predict the results of the 2011 Portuguese legislative elections.

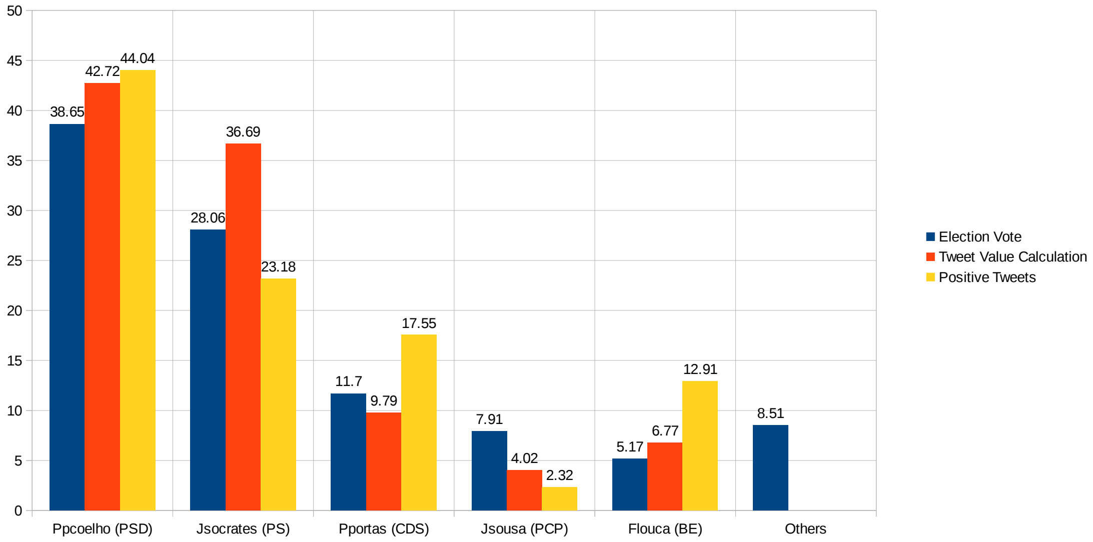


Figure 6.1: 2011 Portuguese Legislative Elections, Twitter Prediction and Positive Tweet Proportion

Figure 6.1 shows the proportions of the election results, our prediction calculation and the positive tweet proportion. The twitter prediction calculation is equal to proportion of the added sentiment of all messages and also takes into consideration the message count proportion of each entity. The positive tweet proportion focusses on the positive messages for that entity, that could indicate a vote in favour.

These are simple calculation examples and other methods to calculate results can be used, giving a bigger significance on the number of messages mentioning the entities or to the positive sentiment in the messages.

The correlation coefficient of between the vote percentage and the twitter proportion calculation is 98.1%. The correlation coefficient between the vote percentage and the positive tweets proportion is 91%.

A comparison is done to the other main polling companies with results near our sample capture dates.

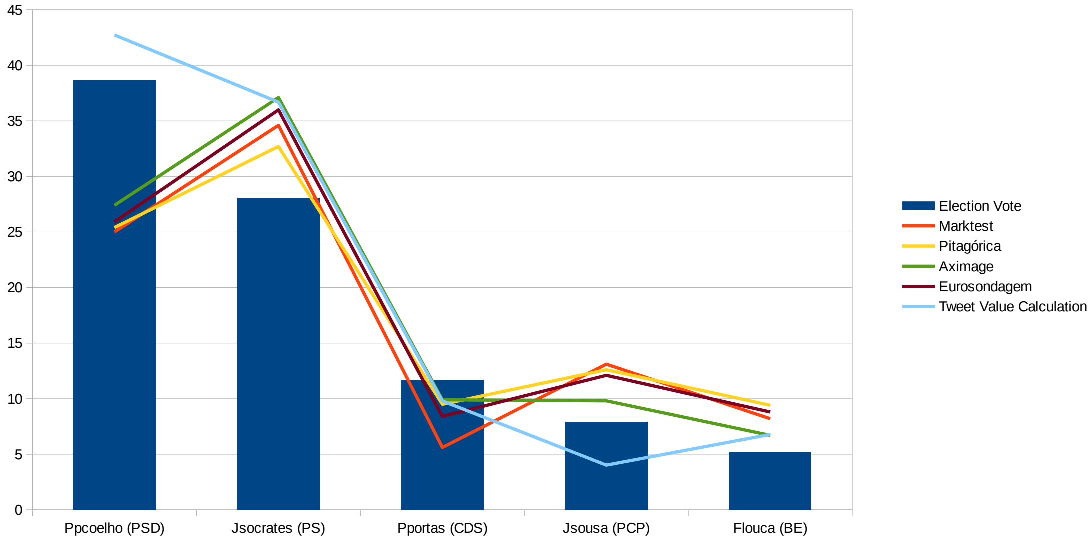


Figure 6.2: 2011 Portuguese Legislative Elections Poll Comparison

As previously stated, our calculation achieved a correlation of 98.1% while *Marktest* achieved 80.3%, *Pitagórica* achieved 85.1%, *Aximage* achieved 86.2% and *EuroSondagem* achieved 82.5%.

6.4.2 SentiCorpus-PT

SentiCorpus-PT is a corpus composed of 3888 manually annotated sentences concerning the 2009 Portuguese legislative elections, collected from the 2^{nd} to the 12^{th} of September of 2009. This corpus was produced by the *LASIGE* group from the *University of Lisbon*.

Each sentence is manually annotated with sentiment and if the sentence is considered ironic by the annotators. Some messages are duplicated, but have a different entity target with the sentiment value for this new target.

The entity targets for these messages are the Portuguese politicians José Sócrates, Manuela Ferreira Leite, Paulo Portas, Francisco Louçã and Jerónimo de Sousa.

Paula Carvalho et al. [CSTS11] found that about 11% of the annotated sentences were considered ironic. This irony is proportional to the number of negative sentences, distributed through the considered entities.

For each of these entities, the number of ironic messages concerning these entities is about 7.2% of the number of non ironic negative messages.

6.4.2.1 Evaluation

Similar to previous evaluations, since our results are real values and the annotated data has their results with 1, 0 and -1 to represent positive, neutral and negative sentiment, these real values must be converted to these by the use of intervals.

The interval taken into consideration is the same achieved, in Section 6.4.1, by using 10-fold cross-validation with the SentiTuites gold standard.

$$Positive \in]0.33, 1[\tag{6.11}$$

$$Neutral \in [-0.35, 0.33] \tag{6.12}$$

$$Negative \in [-1, -0.35[\tag{6.13}$$

6.4.2.2 Results and Conclusions

Polarity	F-Measure	Precision	Recall
Positive	0.332	0.330	0.334
Neutral	0.276	0.174	0.674
Negative	0.302	0.741	0.190
All	0.407	0.415	0.399

Table 6.16: SentiCorpus-PT Results with literal human annotated data

Using this data we can try to predict the results of the 2009 Portuguese legislative elections.

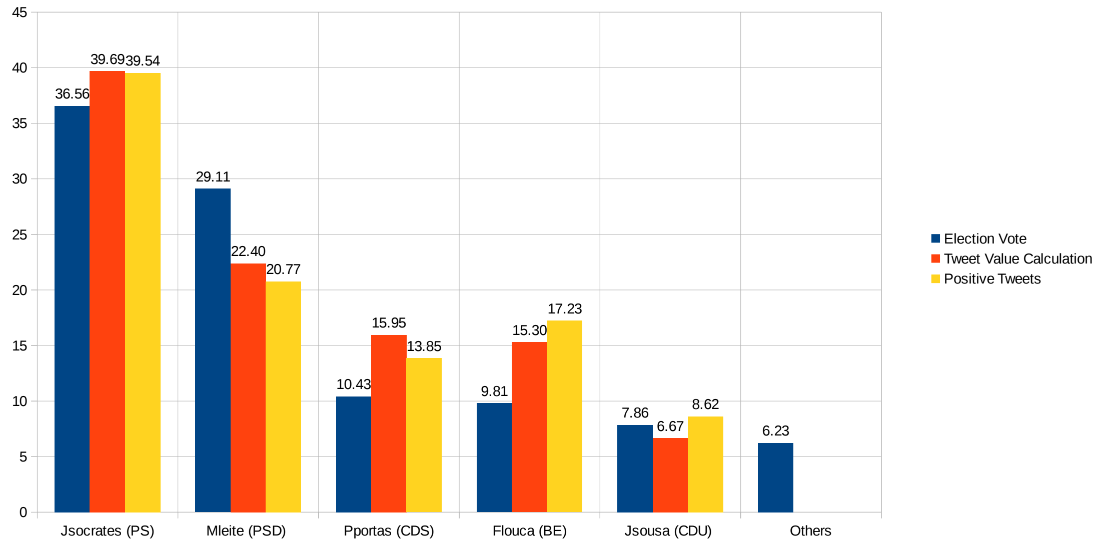


Figure 6.3: 2009 Portuguese Legislative Elections, Twitter Prediction and Positive Tweet Proportion

Figure 6.3 shows the proportions of the election results, our prediction calculation and the positive tweet proportion. The twitter prediction calculation is equal to proportion of the added sentiment of all messages for each entity and also takes into consideration the message count proportion of each entity. The positive tweet proportion focusses on the positive messages for each entity. A positive message could indicate a vote in favour of the targeted entity.

The correlation coefficient of between the vote percentage and the twitter calculation proportion is 91.8%, and between the vote percentage and the positive tweets proportion is 89.5%, slightly lower than the correlation achieved by Paula Carvalho et al. [CSTS11] of 91.7% using the positive number of messages considering the human annotated data.

A comparison is done to the other main polling companies with results near our sample capture dates.

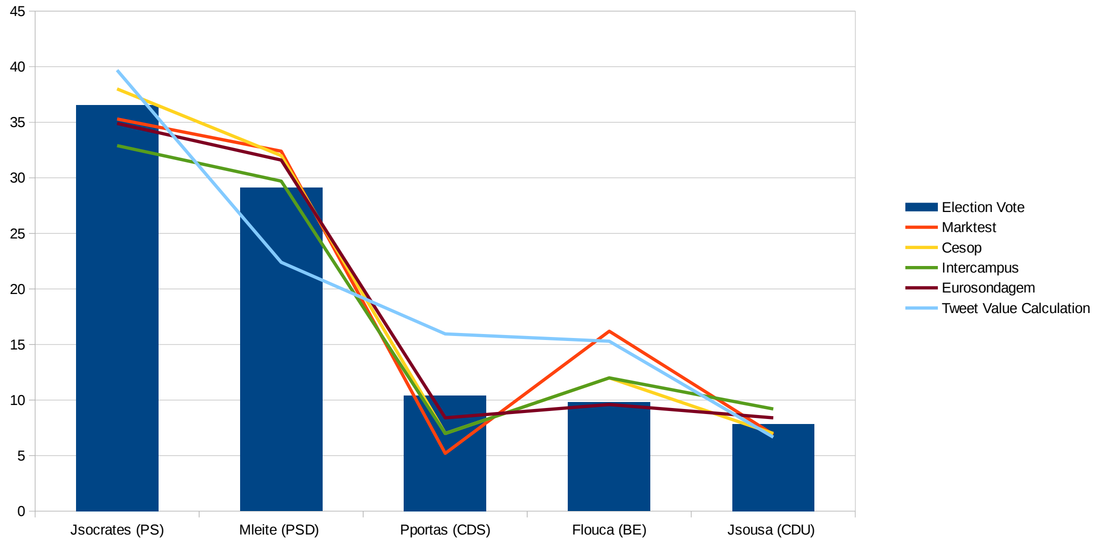


Figure 6.4: 2009 Portuguese Legislative Elections Poll Comparison

As previously stated, our calculation achieved a correlation of 91.8% while *Marktest* achieved 94.8%, *Cesop* achieved 98.9%, *Intercampus* achieved 97.9% and *EuroSondagem* achieved 99.1%.

6.5 SemEval2013 - Sentiment Analysis in Twitter

SemEval is an international workshop on semantic evaluation. One of the existing tasks on this workshop was sentiment analysis in twitter. This workshop on sentiment analysis had two separate tasks, contextual polarity disambiguation and message polarity classification.

Contextual polarity focusses on classifying sentiment in a specific part of a sentence. Message polarity focusses on the sentiment of the whole message.

In this workshop a total of 44 teams participated, 23 in the contextual polarity task and 34 for the message polarity task, having 13 teams participating in both tasks.

Two datasets were used for both tasks, one composed of 2094 SMS and the other composed of 3813 Tweets. For the contextual polarity from these messages, more tests are generated.

Since no actual tweets can be stored in the datasets, twitter only allows to keep and share the tweet id. For this reason the number of existing elements on the dataset are not the same since some tweets are deleted and no longer available when extracted by the id.

6.5.1 Results

Similar to our previous evaluations, since our results are real values and the annotated data classifies using 'positive', 'neutral' and 'negative' strings, these real values must be converted to these by the use of intervals. The interval used in the first task is the same interval used in Section 6.2.1, and in the second task is the same used in Section 6.4.1.

Our results are going to be compared to the official results of the workshop[NRKSRW13]. According to the evaluation rules, our approach is constrained, it does not use additional tweets or SMSs and requires no learning for the algorithm.

Team	Constrained	Unconstrained
NRC-Canada	88.93	
AVAYA	86.98	87.38
BOUNCE	86.79	
LVIC-LIMSI	85.70	
FBM	85.50	
GU-MLT-LT	85.19	
UNITOR	84.60	
USNA	81.31	
Serendio	80.04	
ECNUCS	79.48	80.15
TJP	78.16	
columbia-nlp	74.94	
teragram		74.89
sielers	74.41	
KLUE	73.74	
OPTWIMA	69.17	36.91
swatcs	67.19	63.86
Kea	63.94	
senti.ue-en	62.79	71.38
uottawa	60.20	
IITB	54.80	
SenselyticTeam	53.88	
->	**53.48**	
SU-sentilab		34.73

Table 6.17: SemEval2013, Contextual Polarity Task Using Twitter Data

Team	Constrained	Unconstrained
GU-MLT-LT	88.37	
NRC-Canada	88.00	
AVAYA	83.94	85.79
UNITOR	82.49	
TJP	81.23	
LVIC-LIMSI	80.16	
USNA	79.82	
ECNUCS	76.69	77.34
sielers	73.48	
FBM	72.95	
teragram	72.83	72.83
KLUE	70.54	
columbia-nlp	70.30	
senti.ue-en	66.09	74.13
swatcs	66.00	67.68
Kea	63.27	
uottawa	55.89	
SU-sentilab		55.38
->	**54.44**	
SenselyticTeam	51.13	
OPTWIMA	37.32	36.38

Table 6.18: SemEval2013, Contextual Polarity Task Using SMS Data

Team	Constrained	Unconstrained
NRC-Canada	69.02	
GU-MLT-LT	65.27	
teragram	64.86	64.86
BOUNCE	63.53	
KLUE	63.06	
AMI&ERIC	62.55	61.17
FBM	61.17	
AVAYA	60.84	64.06
SAIL	60.14	61.03
UT-DB	59.87	
FBK-irst	59.76	
nlp.cs.aueb.gr	58.91	
UNITOR	58.27	59.50
LVIC-LIMSI	57.14	
Umigon	56.96	
NILC USP	56.31	
DataMining	55.52	
ECNUCS	55.05	58.42
nlp.cs.aueb.gr	54.73	
ASVUniOfLeipzig	54.56	
SZTE-NLP	54.33	53.10
CodeX	53.89	
Oasis	53.84	
NTNU	53.23	50.71
UoM	51.81	45.07
SSA-UO	50.17	
SenselyticTeam	50.10	
UMCC DLSI (SA)	49.27	48.99
bwbaugh	48.83	54.37
senti.ue-en	47.24	47.85
->	**47.03**	
SU-sentilab		45.75
OPTWIMA	45.40	54.51
REACTION	45.01	
uottawa	42.51	
IITB	39.80	
IIRG	34.44	
sinai	16.28	49.26

Table 6.19: SemEval2013, Message Polarity Task Using Twitter Data

Team	Constrained	Unconstrained
NRC-Canada	68.46	
GU-MLT-LT	62.15	
KLUE	62.03	
AVAYA	60.00	59.47
teragram		59.10
NTNU	57.97	54.55
CodeX	56.70	
FBK-irst	54.87	
AMI&ERIC	53.63	52.62
ECNUCS	53.21	54.77
->	**52.86**	
UT-DB	52.46	
SAIL	51.84	51.98
UNITOR	51.22	48.88
SZTE-NLP	51.08	55.46
SenselyticTeam	51.07	
NILC USP	50.12	
REACTION	50.11	
SU-sentilab		49.57
nlp.cs.aueb.gr	49.41	55.28
LVIC-LIMSI	49.17	
FBM	47.40	
ASVUniOfLeipzig	46.50	
senti.ue-en	44.65	46.72
SSA UO	44.39	
UMCC DLSI (SA)	43.39	40.67
UoM	42.22	35.22
OPTWIMA	40.98	47.15
uottawa	40.51	
bwbaugh	39.73	43.43
IIRG	22.16	

Table 6.20: SemEval2013, Message Polarity Task Using SMS Data

The Tables 6.17, 6.18, 6.19 and 6.20 show the various results of the workshop and the results obtained using our approach adapted for the English language.

Our results surpassed the established baselines of the workshop, limiting the minimum acceptable results that are approved. The baselines refer to the f-measure obtained and are 38.10 for contextual polarity using the twitter data and we obtained 53.48, 31.50

for contextual polarity using the SMS data and we obtained 54.44, 29.19 for message polarity using the twitter data and we obtained 47.03 and finally 19.03 for message polarity using the SMS data and we obtained 52.86.

Similar to previous tests, most errors are close errors, having 91.57% in contextual polarity using the twitter data, 91.98% in contextual polarity using the sms data, 91% in message polarity using the twitter data and 93.4% in message polarity using the sms data, achieving an average of approximate 92%. While some of these close errors are n-grams that do not result in sentiment values are considered as neutral, most of these errors are with low positive and negative sentiment values that are considered as neutral since they remain within the neutral interval.

While our results are not the best is this competition, this approach does outperform the REACTION team, belonging to the LASIGE group from the University of Lisbon, the same group that created SentiLex-PT, SentiTuites and SentiCorpus-PT.

Our results does perform quite well for the message polarity task using SMS data. Is is also important to mention that our algorithm does not require any learning, in comparison to most competing teams.

6.6 Main problems

In this section some problem are discussed and possible improvements are going to be proposed.

A possible improvement that has been observed in other work, in the entity recognition task and in the sentiment classification, is the use of several approaches and using a collective intelligence between several approaches. The use of collective intelligence of several approaches does seems to get better results but does bring forth some more problems. In the entity recognition the disambiguation task will be even harder, having several results to work with.

The use of several approaches was observed in the top ranking teams in the #MSM2013 and SemEval2013 workshops mentioned in Section 6.1.1 and Section 6.5. These approaches include using several tools such as Rizzo et al. [ERT13] in #MSM2013 and the use of several algorithms and several training lexicons such as Saif et al.[MKZ13] in SemEval2013.

6.6.1 Entity Recognition

Taking into consideration the #MSM2013 gold standard, mentioned in Section 6.1.1, there are some classification types that are not consistent, occurring on both the standard and in our results. While the gold standard considers Netherlands and China as organizations, our results consider them as locations. The same occurs in messages containing a sports team that has the same name as a location will be classified as a locations instead of the intended organization. For example Pittsburgh, Denver, Blackpool, Birmingham, Liverpool, Manchester, Valencia, Malaga, Milan, etc. An example of a message containing a sports team classified as a location, "Liverpool boss Roy Hodgson is hopeful Steven Gerrard will make his return".

Our results for entity recognition are directly influenced by the parser, since we are using nouns for the entity classification.

To check if our results would improve, by using a different *POS* tagger, *LingPipe*[7] POS tagger for the Portuguese language was used, but got worse results using a sample of SentiTuites mentioned in Section 6.4.1.

Following the tagging errors, some nouns by the parser are found as entities, such as 'Ajuda' classified as a location. Some more examples exist.

A different problem is by of products of companies in our considered entity types. Some companies have defined with their data generic products, e.g. banks that have listed products as United_States_dollar or Euro. Other examples of generic products include Corn, Sugar, etc.

These can be fixed by using a exception list to exclude these entities from being identified or removing the entity type from the selection data, such as removing all products, and then adding the ones we want to include manually.

The more generic the selection data is, the more common these errors will become. By including generic types may include several subtypes, e.g. the type Work includes the type Artwork, Film, Software, Television Episode, Television Season, Website, Written Work, Television Show, Musical Work, Musical, Cartoon, Radio Program, Line of Fashion and Collection of Valuables.

More data to be considered in our results, more ambiguities between the existing information will occur.

6.6.2 Sentiment Classification

Similar to the entity recognition, the parsing errors influence the sentiment classification.

Nouns identified as entities are not classified with sentiment. For example 'Ajuda' if classified as a noun by the POS tagger, will be identified as the location 'Ajuda' and will not be assigned with sentiment, instead of identifying as the verb 'ajuda', meaning 'help' in English.

When a n-gram is misclassified and the sentiment is not found for that POS, an average of all existing POS is used instead. While not being the most correct sentiment it is a better to have some result than no result.

As mentioned in some results, most classification errors are misclassifications to a close class, classifying positive as neutral, negative as neutral or neutral as either positive or negative. This occurs because annotations are strings or integer values representing these classes and when converting our real value to one of these classes, results will occur near the boundaries of these classes. A low positive sentence can be considered as being

---

[7] `http://www.alias-i.com/lingpipe/`

neutral and a neutral but slightly positive sentence can be considered as being positive. The same happens for negative values.

The reason this problem may not influence our results much is because the interval limit near neutral values for positive and negative values are equivalent and that neutral, low positive and low negative values apply a small influence on the final results of a large data set, for example our results on political tweets in Section 6.4.

7

Conclusions

In this work, a proposal is presented of sentiment analysis tool for the Portuguese language in short messages present in the social website *Twitter* and using SMSs. In this chapter we will show the main contribution on our approach as well as some practical uses of sentiment analysis.

7.1 Main Contributions

Twitter gives us the opportunity to access and analyse, messages that people share all over the globe. These people write in many languages and the more languages we can process and analyse, the more rich the information we can gather.

While related work focuses mostly on the English language, since there are more and better resources available, the presented proposal focusses on the Portuguese language.

Using the process stated in Chapter 5, this approach can be used for other languages, adapting some tools for the specific language, such as a parser capable of supplying a parse tree and part of speech or adapting the Stanford parser for the target language with the use of a treebank, a dictionary for the spell checker and a translator from the target language to English, unless the target language is already English and if so the translator is no longer needed.

Most of the related work assumes a single entity, and in this approach, each entity is given its own sentiment in the message. The sentiment classification is done taking into account the part of speech and its relations with the entities.

Entity recognition can be focussed for specific entity types, identifying only people or locations, or it can be extended to other types of entities such as animals, events or even television shows. Disambiguating entities by similarity to other entities give us a good alternative when no context for disambiguating is available.

Some of the presented steps in this approach are used in related work, but not in the combination as presented, with specific sentiment for each entity found in the text. With each entity having its own sentiment, negative messages that contain the said entity but that sentiment is not its target, will allow for better results to be extracted. As shown by the results this analysis can be used to predict election outcomes, showing the public's favouritism or showing the one people most talk about.

A single piece of a processed message provides little information, but aggregated this information can answer bigger questions. Getting the sentiment expressed in a message will tell you about that person on that topic about an entity, aggregated you can gather the public's opinion, not on just one topic, not on one entity, but as many as they want to talk about.

While our results may not be the best, they prove to be competitive, with a simple approach and require no previous machine learning. In #MSM2013 and SemEval2013, the participating teams are composed of several members, from investigation groups with some years invested in this field of research. Our approach may be an useful base to start from when dealing with these problems, entity recognition and sentiment classification.

Our approach provides more than just entity recognition and sentiment classification. While dealing with entity recognition and disambiguation, all entities will have a unique URI with more information on that specific entity, allowing to search for more detailed information for each entity type considered. Each entity will also have their own sentiment depending on the construction of the sentence. This was useful when dealing with political tweets, mentioned in Section 6.4, having in some messages several entities with different sentiment values assigned to them. This provides more valuable information that just getting the sentiment of a sentence or a message.

7.1.1 Adapting for multi-language sentiment analysis

The main language target of this thesis is the Portuguese language, but this approach can be adapted to process other languages.

Taking into account language classification of the message, in this case we use the language classification provided by the Twitter API, we can use the same system to classify messages in different languages.

To adapt this approach to work in English written messages, the tools necessary are the parser and a dictionary for the spell checker. The translator will not be needed for the English messages since the annotated data is already in English. A set of negations and sentiment intensifiers will also have to be defined, for these to be identified and processed accordingly.

This adaptation works for entity recognition and sentiment classification as well as the combination of both, assigning sentiment to a single entity. This adaptation was used in some of our results, more specifically on entity recognition with #MSM2013 in Section

6.1.1 and on sentiment classification with SemEval2013 in Section 6.5.

A possible adaptation is to have language specific thread pools. This allows processing messages for both languages, just by having the necessary tools for both languages and processing according to the specific language of the message.

Each thread pool will have its specific tools and will process messages using that specific language. This can be easier to adapt each thread pool to the message income rate of each specific language, in this case the message rate of messages in English is higher than the rate of messages in Portuguese.

7.2 Future Work

In this section it is shown possible near future work that can be achieved using the results of this thesis.

7.2.1 Viewing Data using StreamForce

As previously explained in Section 1.3, the results of this thesis are to be merged with an existing product of AnubisNetworks, StreamForce.

StreamForce is a real-time analysis and viewing platform. This provides a way to view and follow the public's sentiment on certain entities. Adding these features will enrich SteamForce, supplying a way for clients to follow their own popularity, finding out if the public is unhappy and can address this matter accordingly or even compare their popularity to other companies.

Here are two graphic examples for viewing the general sentiment on twitter.

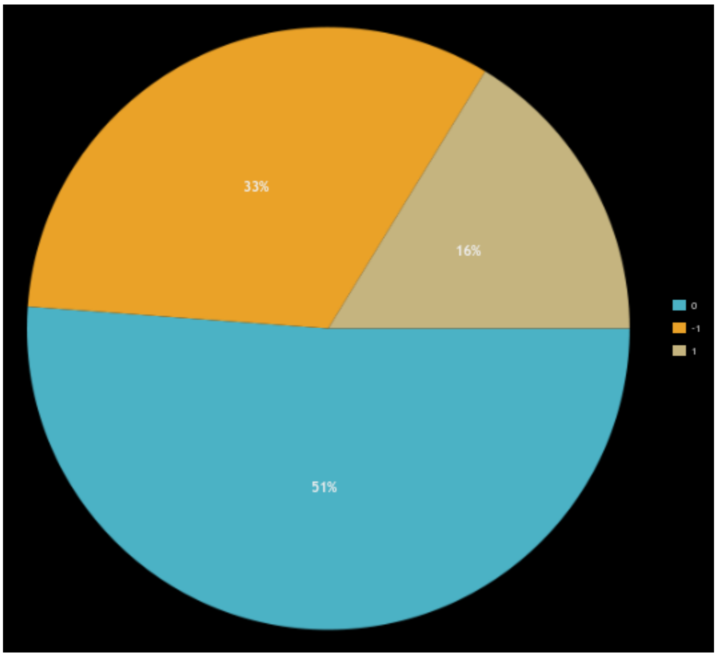


Figure 7.1: StreamForce Bar Graph showing General Twitter Sentiment

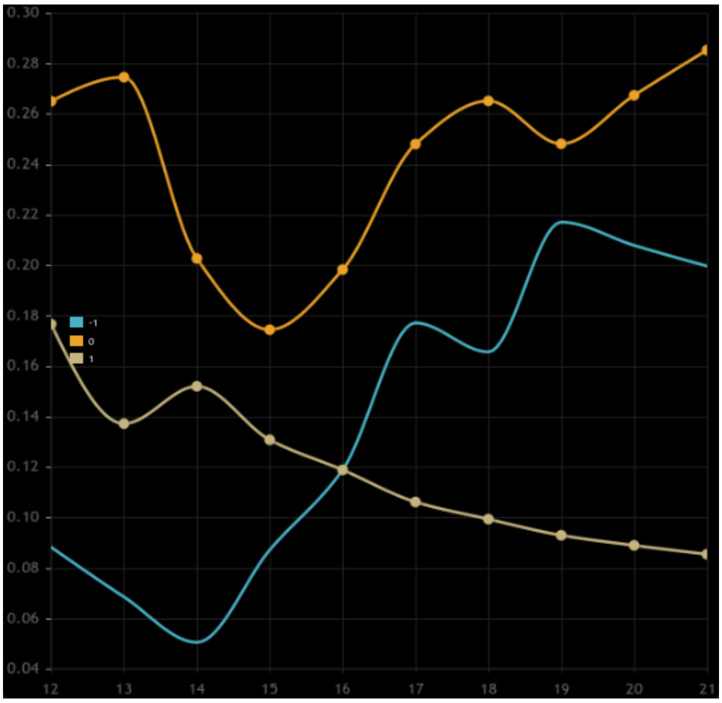


Figure 7.2: StreamForce Multi Line Graph showing General Twitter Sentiment

These examples show the general sentiment of all tweets. These graphs will be refreshed based on a specified time. While this graph is not refreshing, data will be joined by the different types, in this case positive, neutral and negative sentiment. This refresh time can be specified to view data by the minute, hour or day. This joined data will be the next result to be presented next to the existing data. Data will be saved and presented based on a viewing mirror, hidden in the case of the pie graph. Old data that will leave this viewing window will be discarded.

These graphs can be focussed on the sentiment of a specific user, hashtag, found entity or specific text that appears in the message. This provides a close real-time viewing and monitoring of these entities.

A different view option is viewing the geographical location from where the tweet was sent. This is only possible for users that have their geographical location options available when sending their tweet.

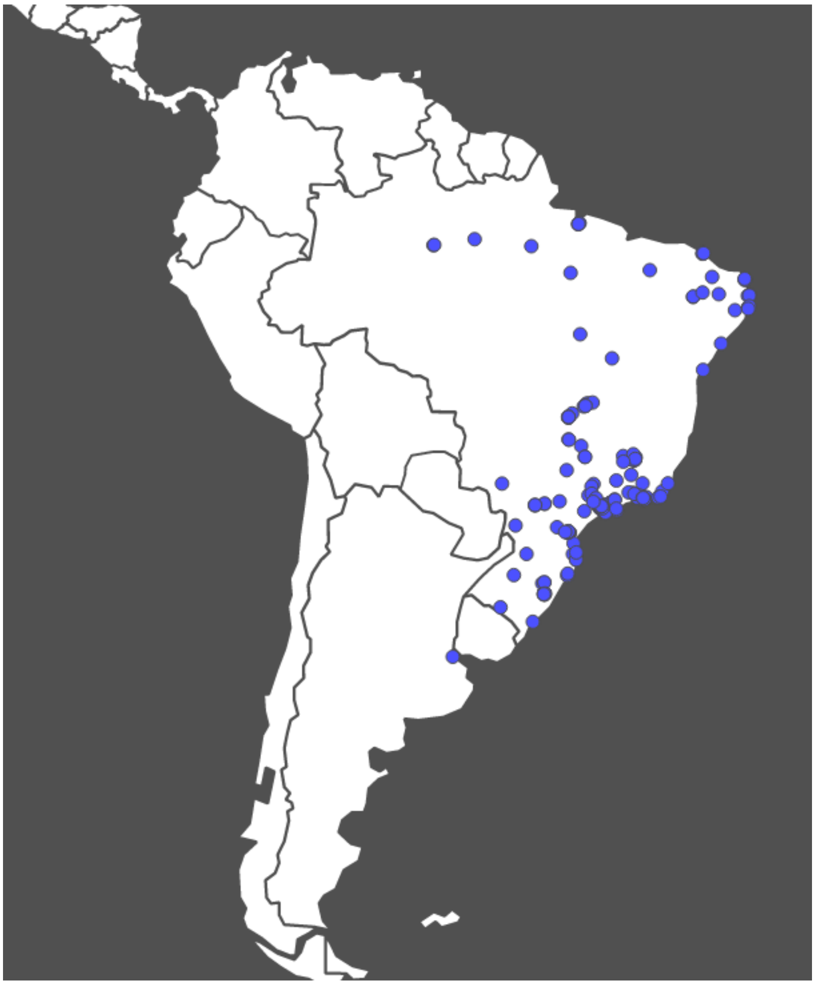


Figure 7.3: StreamForce Viewing Twitter Location

In this example we show tweets sent from Brazil in a short timespan. Filtering these messages for a specific target entity or hashtag, it is possible to find the sentiment over a specific location. Knowing these locations can be useful, easily identifying regions or countries with negative sentiment or positive sentiment for the targets.

7.2.2 Keyword Sentiment

If the sentiment toward a specific target is wanted and that target is not a named entity, such as cat or dog, the overall sentiment of the message will not ensure that it is the same result as the sentiment for that specific target. For this type of result, these keywords that need to be specified will be treated as entities, but since they are only keywords, these will not have a corresponding URI with more information on them.

This will allow to check and/or compare results to things that are normally not entities, such as checking is people like cats over dogs, or phones over tablets.

7.2.3 Topic Cloud

For real time viewing of trending topics, a cloud of topic can be built. This topic cloud was built using *GraphStream*[1], a Java library for graph viewing and manipulation.

This graph is composed of nodes and edges. Each node represents an user, hashtag or an entity recognized in the message, and edges are the connections linking these nodes. A sender that mentions anything that will result in another node will result in a edge from that sender to the node.

Entities found in the message are marked with the gray color, hashtags are marked with brown and users are marked with blue. The higher relation count, the bigger the node becomes, becoming more visible and getting more attention from the user.

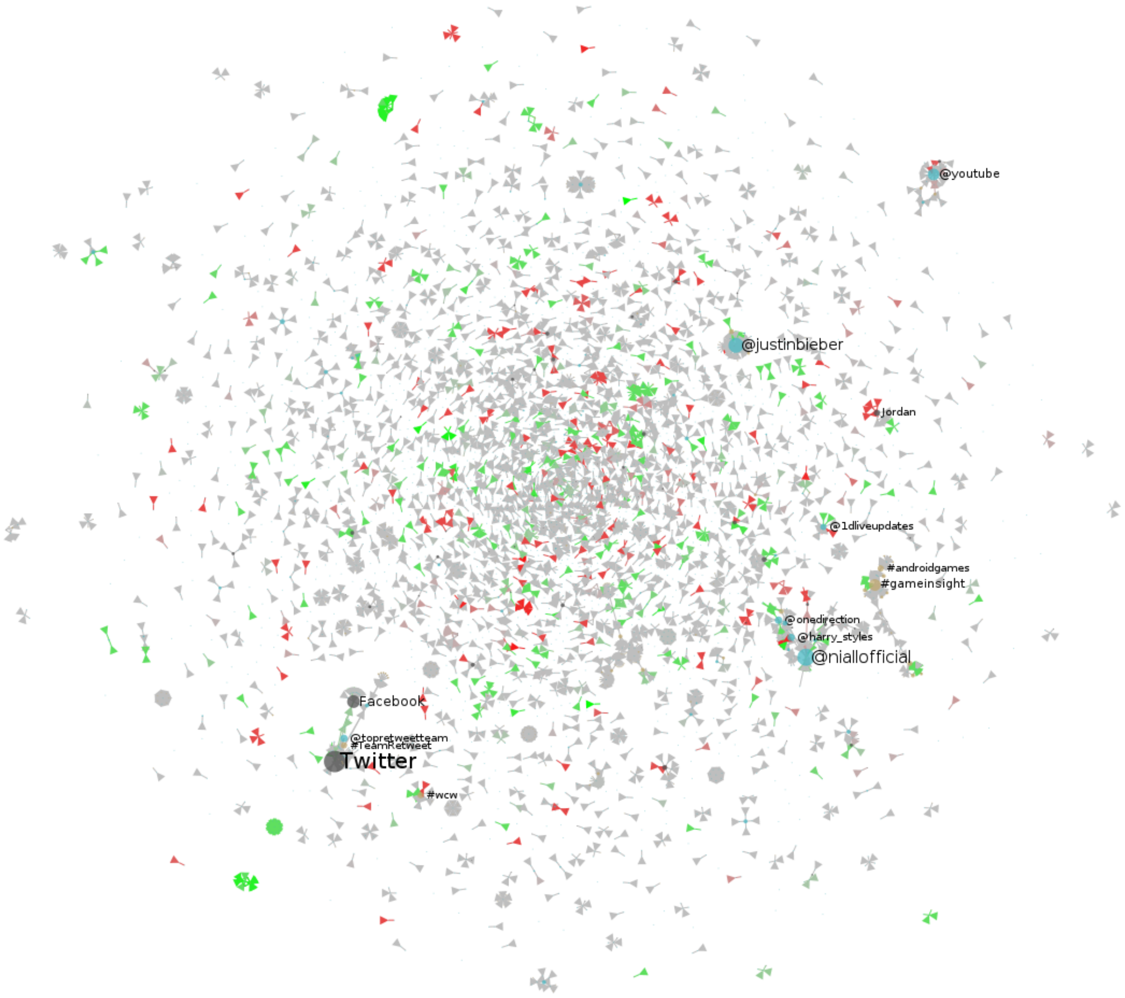


Figure 7.4: Topic Cloud Example

This cloud can be used to show the trending topics, users and hashtags in twitter as

---

[1] `http://graphstream-project.org/`

they are processed. The directional arrows will change their color from red to gray to green, representing negative, neutral and positive messages for those messages.

This graph will evolve within a certain limit established beforehand, replacing older and less important nodes from the graph and giving more value to the nodes that keep having new relations, while still allowing new nodes and edges to be created.

Even though this application is somewhat done, as shown by Figure 7.4, some work is still needed to achieve a better viewing experience.

7.2.4 Entity Type Card

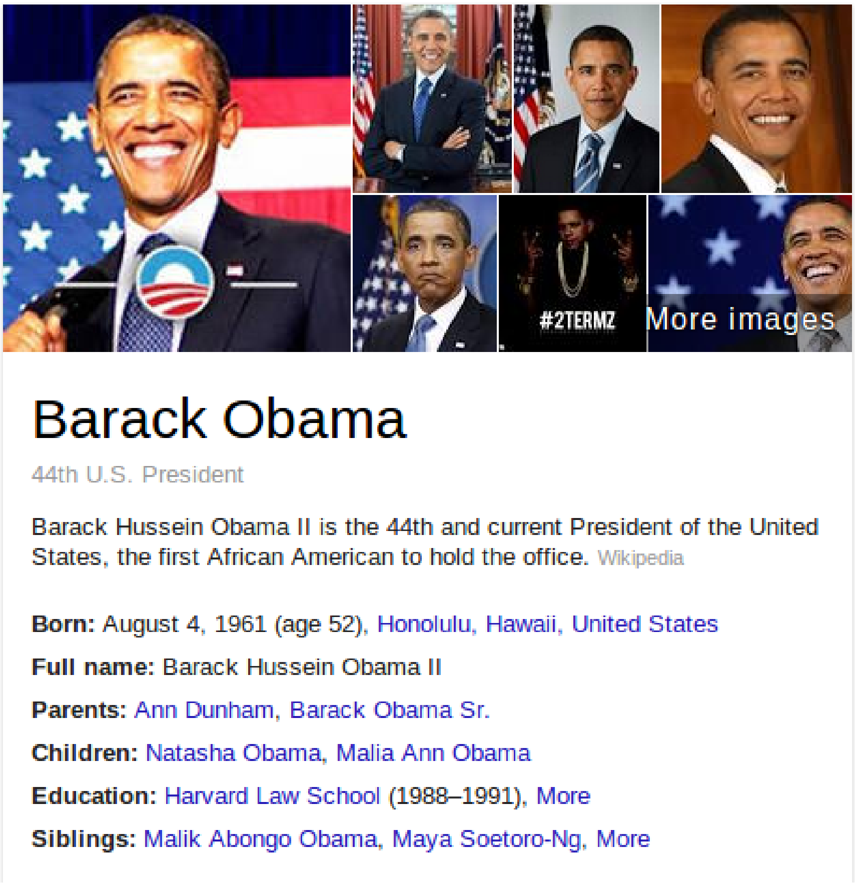
Entity cards are not connected to the sentiment of an entity, but focusses on how we view the information of an entity. This will enrich entities with relevant information, providing a simple way to view and understand what are those entities. When using the other applications and we want to explore what is happening currently, we can explore the data to find what is trending at the moment. Entities have a URI, but the information regarding entities is extent. Entity cards will show the most important information concerning that entity.

Information cards can be build for each entity type. This information will show the main fields of each type of entity. These fields are chosen appropriately and are commonly relevant for each entity type.

The *Person* type will show the person name, date of birth, living location, etc. A short description and a image can also be shown if available or found.

This is being done by *Google* when searching in their search engine for entities that they can recognize and extract information, such as people, locations and organizations.

When *Google* identifies this entity, then it uses data from *DBpedia* to get relevant information on that entity type and create a card for faster viewing.



Barack Obama

44th U.S. President

Barack Hussein Obama II is the 44th and current President of the United States, the first African American to hold the office. Wikipedia

Born: August 4, 1961 (age 52), Honolulu, Hawaii, United States

Full name: Barack Hussein Obama II

Parents: Ann Dunham, Barack Obama Sr.

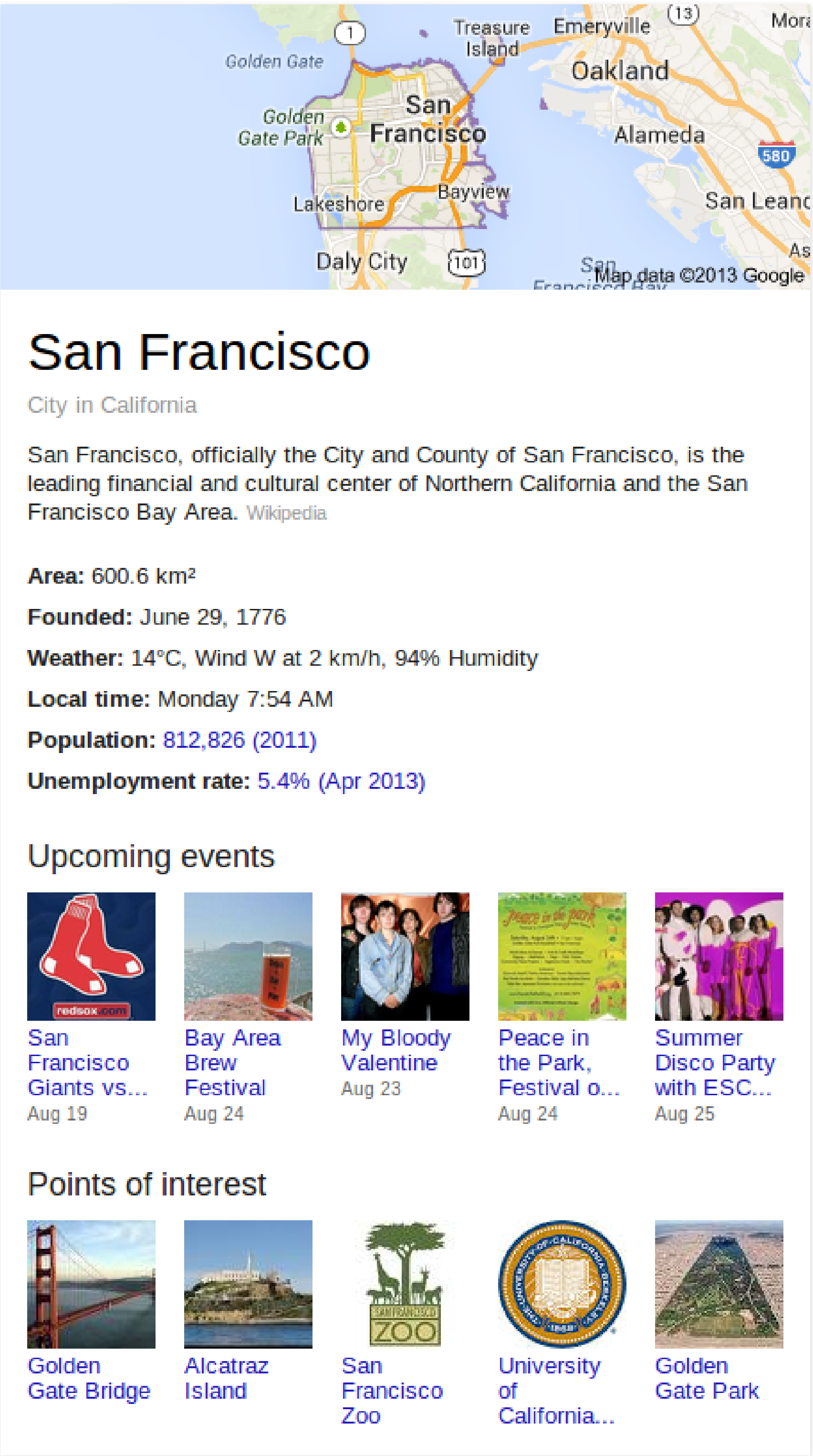
Children: Natasha Obama, Malia Ann Obama

Education: Harvard Law School (1988–1991), More

Siblings: Malik Abongo Obama, Maya Soetoro-Ng, More

Figure 7.5: Google Entity Card example for the type Person

These provide a simple and fast way of getting the most relevant information for specific entities. These cards are specific for each entity type getting different information for different types.



San Francisco

City in California

San Francisco, officially the City and County of San Francisco, is the leading financial and cultural center of Northern California and the San Francisco Bay Area. Wikipedia

Area: 600.6 km²

Founded: June 29, 1776

Weather: 14°C, Wind W at 2 km/h, 94% Humidity

Local time: Monday 7:54 AM

Population: 812,826 (2011)

Unemployment rate: 5.4% (Apr 2013)

Upcoming events

San Francisco Giants vs... Aug 19

Bay Area Brew Festival Aug 24

My Bloody Valentine Aug 23

Peace in the Park, Festival o... Aug 24

Summer Disco Party with ESC... Aug 25

Points of interest

Golden Gate Bridge

Alcatraz Island

San Francisco Zoo

University of California...

Golden Gate Park

Figure 7.6: Google Entity Card example for the type Location

We can also see linked data being used to get more information related to this entity. In this specific case we can find events and points of interest for a specific location.

Bibliography

[AXVRP11] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau. "Sentiment analysis of Twitter data". In: *Proceedings of the Workshop on Languages in Social Media*. LSM '11. Portland, Oregon: Association for Computational Linguistics, 2011, pp. 30–38. ISBN: 978-1-932432-96-1. URL: `http://dl.acm.org/citation.cfm?id=2021109.2021114`.

[AAAHH03] G. Antoniou, G. Antoniou, G. Antoniou, F. V. Harmelen, and F. V. Harmelen. "Web Ontology Language: OWL". In: *Handbook on Ontologies in Information Systems*. Springer, 2003, pp. 67–92.

[Sil] "Automatic Expansion of a Social Judgment Lexicon for Sentiment Analysis". In: LASIGE, University of Lisbon, Faculty of Sciences (2010).

[BES10] S. Baccianella, A. Esuli, and F. Sebastiani. "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining". In: *LREC*. Ed. by N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias. European Language Resources Association, 2010. ISBN: 2-9517408-6-7. URL: `http://nmis.isti.cnr.it/sebastiani/Publications/LREC10.pdf`.

[BF10] L. Barbosa and J. Feng. "Robust sentiment detection on Twitter from biased and noisy data". In: *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. COLING '10. Beijing, China: Association for Computational Linguistics, 2010, pp. 36–44. URL: `http://dl.acm.org/citation.cfm?id=1944566.1944571`.

[Bar10] S. N. Baron. “Discourse structures in Instant Messaging: The case of utterance breaks”. In: *Language@Internet* 7.4 (2010). ISSN: 1860-2029. URL: `http://nbn-resolving.de/urn:nbn:de:0009-7-26514`.

[BFSMS13] D. S. Batista, D. Forte, R. Silva, B. Martins, and M. Silva. “Extracção de Relações Semânticas de Textos em Português Explorando a DBpédia e a Wikipédia”. In: *linguamatica* 5.1 (2013), pp. 41–57. URL: `http://www.linguamatica.com/index.php/linguamatica/article/view/157`.

[BVS08] M. Bautin, L. Vijayarenu, and S. Skiena. “International sentiment analysis for news and blogs”. In: *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*. 2008.

[BLHL01] T. Berners-Lee, J. Hendler, and O. Lassila. “The Semantic Web”. In: *Scientific American* 284.5 (May 2001), pp. 34–43. URL: `http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21`.

[BHB09] C. Bizer, T. Heath, and T. Berners-Lee. “Linked data - the story so far”. In: *Int. J. Semantic Web Inf. Syst.* 5.3 (2009), 1–22.

[BDP07] J. Blitzer, M. Dredze, and F. Pereira. “Biographies, Bollywood, Boomboxes and Blenders: Domain Adaptation for Sentiment Classification”. In: *Proceedings of the Association for Computational Linguistics (ACL)*. 2007.

[BK07] H. Boley and M. Kifer. *RIF Basic Logic Dialect*. W3C Working Draft. http://www.w3.org/TR/2007/WD-rif-bld-20071030. W3C, 2007. URL: `http://www.w3.org/TR/rif-bld/`.

[BS04] A. Branco and J. a. Silva. “Evaluating Solutions for the Rapid Development of State-of-the-Art POS Taggers for Portuguese.” In: *LREC2004*. Ed. by M. T. Lino, M. F. Xavier, F. Ferreira, R. Costa, and R. Silva. Paris, 2004, pp. 507–510. ISBN: 2-9517408-1-6.

[BS06] A. Branco and J. a. R. Silva. “A suite of shallow processing tools for Portuguese: LX-suite”. In: *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Posters &#38; Demonstrations*. EACL ’06. Trento, Italy: Association for Computational Linguistics, 2006, pp. 179–182. URL: `http://dl.acm.org/citation.cfm?id=1608974.1609003`.

[CSTS11] P. Carvalho, L. Sarmento, J. Teixeira, and M. J. Silva. “Liars and saviors in a sentiment annotated corpus of comments to political debates”. In: *Proceedings of the 49th Annual Meeting of the Association* *for Computational Linguistics: Human Language Technologies: short papers - Volume 2*. HLT '11. Portland, Oregon: Association for Computational Linguistics, 2011, pp. 564–568. ISBN: 978-1-932432-88-6. URL: `http://dl.acm.org/citation.cfm?id=2002736.2002847`.

[CT94] W. B. Cavnar and J. M. Trenkle. "N-Gram-Based Text Categorization". In: *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*. Las Vegas, US, 1994, pp. 161–175. URL: `/brokenurl#citeseer.ist.psu.edu/68861.html`.

[CE13] T. Chalothorn and J. Ellman. "Affect Analysis of Radical Contents on Web Forums Using SentiWordNet". In: *International Journal of Innovation, Management and Technology* 4.1 (2013), pp. 122–124. ISSN: 1541-1672.

[Chb94] D. T. Chbane. "Desenvolvimento de Sistema para Conversão de Textos em Fonemas no Idioma Português". In: (1994). URL: `http://www.decampos.net/textos/disdimas.pdf`.

[CZ99] L. Cowie and Zacharski. "Language recognition for mono and multilingual documents". In: Proceedings of the Vextal Conference (1999).

[DZC10] Y. Dang, Y. Zhang, and H. Chen. "A Lexicon-Enhanced Method for Sentiment Classification: An Experiment on Online Product Reviews". In: *IEEE Intelligent Systems* 25.4 (July 2010), pp. 46–53. ISSN: 1541-1672. DOI: `10.1109/MIS.2009.105`. URL: `http://dx.doi.org/10.1109/MIS.2009.105`.

[DLP03] K. Dave, S. Lawrence, and D. M. Pennock. "Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews". In: *Proceedings of WWW*. 2003, pp. 519–528.

[Den08] K. Denecke. "Using SentiWordNet for multilingual sentiment analysis." In: *ICDE Workshops*. IEEE Computer Society, May 5, 2008, pp. 507–512. URL: `http://dblp.uni-trier.de/db/conf/icde/icdew2008.html#Denecke08`.

[ERT13] M. Van Erp, G. Rizzo, and R. Troncy. "Learning with the Web: Spotting named entities on the intersection of NERD and machine learning". In: *WWW 2013, 3rd International Workshop on Making Sense of Microposts (#MSM'13), Concept Extraction Challenge, May 13, 2013, Rio de Janeiro, Brazil*. Rio de Janeiro, BRAZIL, May 2013. URL: `http://www.eurecom.fr/publication/3968`.

[ES06] A. Esuli and F. Sebastiani. “SENTIWORDNET: A publicly available lexical resource for opinion mining”. In: *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC’06*. 2006, pp. 417–422.

[Fie00] R. T. Fielding. “Architectural styles and the design of network-based software architectures”. AAI9980887. PhD thesis. 2000. ISBN: 0-599-87118-0.

[GSODMEHYFS11] K. Gimpel, N. Schneider, B. O’Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith. “Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments.” In: *ACL (Short Papers)*. The Association for Computer Linguistics, 2011, pp. 42–47. ISBN: 978-1-932432-88-6. URL: `http://dblp.uni-trier.de/db/conf/acl/acl2011s.html#GimpelSODMEHYFS11`.

[GSS07] N. Godbole, M. Srinivasaiah, and S. Skiena. “Large-Scale Sentiment Analysis for News and Blogs”. In: *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*. 2007.

[GSXYW11] S. Gupta, B. Slawski, D. Xin, W. Yao, and M. A. Wierman. “The Twitter Rumor Network: Subject and Sentiment Cascades in a Massive Online Social Network”. In: (2011).

[KWM11] E. Kouloumpis, T. Wilson, and J. Moore. “Twitter Sentiment Analysis: The Good the Bad and the OMG!” In: *ICWSM*. Ed. by L. A. Adamic, R. A. Baeza-Yates, and S. Counts. The AAAI Press, 2011. URL: `http://dblp.uni-trier.de/db/conf/icwsm/icwsm2011.html#KouloumpisWM11`.

[KML13] S. Kumar, F. Morstatter, and H. Liu. *Twitter Data Analytics*. New York, NY, USA: Springer, 2013.

[LS99] O. Lassila and R. R. Swick. *Resource Description Framework (RDF) Model and Syntax Specification*. W3C Recommendation. W3C, 1999. URL: `http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/`.

[Lev66] V. Levenshtein. “Binary Codes Capable of Correcting Deletions, Insertions and Reversals”. In: *Soviet Physics Doklady* 10 (1966), p. 707.

[Liu12] B. Liu. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2012.

[MP96] Martino and Paulsen. “Natural language determination using partial words”. In: (1996).

[MGL09] P. Melville, W. Gryc, and R. D. Lawrence. "Sentiment analysis of blogs by combining lexical knowledge with text classification". In: *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. Paris, France: ACM, 2009, pp. 1275–1284. ISBN: 978-1-60558-495-9. DOI: `http://doi.acm.org/10.1145/1557019.1557156`. URL: `http://portal.acm.org/citation.cfm?id=1557156`.

[MKZ13] S. Mohammad, S. Kiritchenko, and X. Zhu. "NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets". In: *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*. Atlanta, Georgia, USA, 2013.

[MBTAG04] D. Moldovan, A. Badulescu, M. Tatu, D. Antohe, and R. Girju. "Models for the semantic classification of noun phrases". In: *In HLT-NAACL 2004: Workshop on Computational Lexical Semantics*. 2004, pp. 60–67. URL: `http://acl.ldc.upenn.edu/hlt-naacl2004/CLS/pdf/moldovan.pdf`.

[MZS06] D. Mollá, M. van Zaanen, and D. Smith. "Named Entity Recognition for Question Answering". In: *Proceedings ALTW 2006*. 2006, pp. 51–58.

[MBCCS11] S. Moreira, D. Batista, P. Carvalho, F. Couto, and M. Silva. "POWER - Politics Ontology for Web Entity Retrieval". In: *Advanced Information Systems Engineering Workshops*. Ed. by C. Salinesi and O. Pastor. Vol. 83. Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, 2011, pp. 489–500. ISBN: 978-3-642-22055-5. DOI: `10.1007/978-3-642-22056-2_51`. URL: `http://dx.doi.org/10.1007/978-3-642-22056-2_51`.

[NKW05] J.-C. Na, C. Khoo, and P. H. J. Wu. "Use of negation phrases in automatic sentiment classification of product reviews". In: *Library Collections, Acquisitions, and Technical Services* 29.2 (2005), pp. 180 – 191. ISSN: 1464-9055. DOI: `http://dx.doi.org/10.1016/j.lcats.2005.04.007`. URL: `http://www.sciencedirect.com/science/article/pii/S146490550500031X`.

[NSKCZ04] J.-C. Na, H. Sui, C. Khoo, S. Chan, and Y. Zhou. "Effectiveness of Simple Linguistic Processing in Automatic Sentiment Classification of Product Reviews". In: *Conference of the International Society for Knowledge Organization (ISKO)*. 2004, pp. 49–54.

[NRKSRW13] P. Nakov, S. Rosenthal, Z. Kozareva, V. Stoyanov, A. Ritter, and T. Wilson. "SemEval-2013 task 2: sentiment analysis in Twitter". In: 2013.

[OBRS10] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith. "From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series." In: *ICWSM*. Ed. by W. W. Cohen and S. Gosling. The AAAI Press, 2010. URL: `http://dblp.uni-trier.de/db/conf/icwsm/icwsm2010.html#OConnorBRS10`.

[OT09] B. Ohana and B. Tierney. "Sentiment Classification of Reviews Using SentiWordNet". In: ed. by D. I. of Technology. 2009.

[PP10] A. Pak and P. Paroubek. "Twitter as a corpus for sentiment analysis and opinion mining". In: *Proceedings of LREC 2010* (2010).

[PL04] B. Pang and L. Lee. "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts". In: *In Proceedings of the ACL*. 2004, pp. 271–278.

[PLV02] B. Pang, L. Lee, and S. Vaithyanathan. "Thumbs up? Sentiment Classification Using Machine Learning Techniques". In: *emnlp2002*. Philadelphia, Pennsylvania, 2002, pp. 79–86.

[PSRM11] A. Paulo-Santos, C. Ramos, and N. Marques. "Determining the Polarity of Words through a Common Online Dictionary". In: *Progress in Artificial Intelligence*. Ed. by L. Antunes and H. Pinto. Vol. 7026. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, pp. 649–663. ISBN: 978-3-642-24768-2. DOI: `10.1007/978-3-642-24769-9_47`. URL: `http://dx.doi.org/10.1007/978-3-642-24769-9_47`.

[PT09] R Prabowo and M Thelwall. "Sentiment analysis: A combined approach". In: *JOURNAL OF INFORMETRICS* 3.2 (Apr. 2009), pp. 143–157. ISSN: 1751-1577. DOI: `10.1016/j.joi.2009.01.003`. URL: `http://apps.isiknowledge.com.libproxy.unm.edu/full_record.do?product=WOS&colname=WOS&search_mode=RelatedRecords&qid=533&SID=1BFE94Ekeg2KHDJkJJ8&page=1&doc=4`.

[PS08] E. Prud'hommeaux and A. Seaborne. *SPARQL Query Language for RDF*. W3C Recommendation. W3C, Jan. 2008. URL: `http://www.w3.org/TR/rdf-sparql-query/`.

[RRDA11] L.-A. Ratinov, D. Roth, D. Downey, and M. Anderson. "Local and Global Algorithms for Disambiguation to Wikipedia." In: (2011). Ed. by D. Lin, Y. Matsumoto, and R. Mihalcea, pp. 1375–1384. URL: `http://dblp.uni-trier.de/db/conf/acl/acl2011.html#RatinovRDA11`.

[SHA12] H. Saif, Y. He, and H. Alani. “Semantic Sentiment Analysis of Twitter.” In: *International Semantic Web Conference (1)*. Ed. by P. Cudré-Mauroux, J. Heflin, E. Sirin, T. Tudorache, J. Euzenat, M. Hauswirth, J. X. Parreira, J. Hendler, G. Schreiber, A. Bernstein, and E. Blomqvist. Vol. 7649. Lecture Notes in Computer Science. Springer, 2012, pp. 508–524. ISBN: 978-3-642-35175-4. URL: `http://dblp.uni-trier.de/db/conf/semweb/iswc2012-1.html#SaifHA12`.

[ST11] M. J. Silva and R. TEAM. *Notas sobre a Realização e Qualidade do Twitómetro*. Tech. rep. University of Lisbon, Faculty of Sciences,LASIGE, 2011.

[SCS12] M. J. Silva, P. Carvalho, and L. Sarmento. “Building a Sentiment Lexicon for Social Judgement Mining”. In: *PROPOR*. Ed. by H. de Medeiros Caseli, A. Villavicencio, A. J. S. Teixeira, and F. Perdigão. Vol. 7243. Lecture Notes in Computer Science. Springer, 2012, pp. 218–228. ISBN: 978-3-642-28884-5. URL: `http://dblp.uni-trier.de/db/conf/propor/propor2012.html#SilvaCS12`.

[VCC12] A. Varga, A. E. Cano, and F. Ciravegna. “Exploring the Similarity between Social Knowledge Sources and Twitter for Cross-domain Topic Classification”. In: Proceedings of the International Semantic Web Conference (ISWC) (2012).

[WWH05] T. Wilson, J. Wiebe, and P. Hoffmann. “Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis”. In: *Proceedings of Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*. Vancouver, CA, 2005. URL: `http://www.cs.pitt.edu/~twilson/pubs/hltemnlp05.pdf`.

[YHBSW11] M. A. Yosef, J. Hoffart, I. Bordino, M. Spaniol, and G. Weikum. “AIDA: An Online Tool for Accurate Disambiguation of Named Entities in Text and Tables”. In: *PVLDB* 4.12 (2011), pp. 1450–1453. URL: `http://dblp.uni-trier.de/db/journals/pvldb/pvldb4.html#YosefHBSW11`.

A

Random Sampling of Tweets with Sentence Sentiment and Entity Sentiment

Format: Message; Sentence Sentiment; Entity Sentiment List

A.1 Correct classifications

A.1.1 Full agreement between the annotators

A.1.1.1 Positive

@2dgrajamaica bate, bate palma que é o hr de cantar, agr todos juntos vamos lá, PARABÉNS UHUUUUL, amanhã é teu dia meu fiel escudeiro 0.6996345591137876 [null -> 0.6996345591137876]

@cesinhajrs Adoramos o elogio! Muito obrigado e seja sempre bem vindo! 0.9651574335250973 [null -> 0.9651574335250973]

Circulando boas novas para Saquarema, parabéns! http://t.co/6LoqOXGb16 1.0 [[Boas_Novas, http://pt.dbpedia.org/resource/Boas_Novas, http://dbpedia.org/ontology/Organisation] -> 1.0, [Saquarema, http://pt.dbpedia.org/resource/Saquarema, http://dbpedia.org/ontology/Place] -> 1.0]

Valeu... Cada momento que eu pude te abraçar A gente fez amor, eu sempre vou lembrar Está tudo guardado no meu pensamento.. 0.5137320243233146 [null -> 0.5137320243233146]

"Um exercício de cidadania. Hoje o Brasil acordará mais rico, quero dizer,o Brasil acordará mais livro!... http://t.co/h8D5qQmk40 0.4966830760843944 [[Brasil, http://pt.dbpedia.org/resource/Brasil, http://dbpedia.org/ontology/Place] -> 0.4966830760843944]

falte a clases :))) 0.464 [null -> 0.464]

Respeito quem me respeita. Sou educado com quem merece. Agradeço a Deus todos os dias por tudo que tenho, e por... http://t.co/tLQla4iLSP 0.9971236150503261 [null -> 0.9971236150503261]

Gostei de um vídeo @YouTube de @artico65 http://t.co/nZyrc6ZQ05 Minecraft:HardcoreGames PVPFinal com Leonmonk e o Inscrito! 0.47058823529411764 [[Minecraft, http://pt.dbpedia.org/resource/Minecraft, http://dbpedia.org/ontology/Software] -> 0.25, [Youtube, http://dbpedia.org/resource/YouTube, http://dbpedia.org/ontology/Organisation] -> 0.25]

Somos entregues a morte Todo dia Por amor a Ti Somos a geração que se levanta E nunca vai desistir (8 0.7501019639271865 [[Morte, http://dbpedia.org/resource/Morte, http://xmlns.com/foaf/0.1/Person] -> 0.7501019639271865]

Agora sim eu sinto que dormi o suficiente. 0.3259986572165497 [null -> 0.3259986572165497]

hj o dia ta liiindo ;3 0.5194805194805195 [null -> 0.5194805194805195]

Faça sol ou chuva um lindo dia vai nascer 0.6887822933602015 [null -> 0.6887822933602015]

Ontem tive a ver os meus albuns de fotos de quando era pequena, tinha um sorriso lindo e verdadeiro. 0.42937629901918684 [[Sorriso, http://dbpedia.org/resource/Sorriso, http://dbpedia.org/ontology/Place] -> 0.42937629901918684]

Bom dia! O fds tá chegando, aproveitem para viajar =) http://t.co/mVjZd6HkAx http://t.co/LQmIqh7SwK 0.7218425369054128 [null -> 0.7218425369054128]

Ta tudo dando certo. Hum viva para todos. Só falta uma semana... http://t.co/DOydFwUBdY 0.683521078001011 [null -> 0.683521078001011]

Quais seus planos para fim de semana?? :D — Namorar , Sair com os amigos , ver filmes e tal kk' http://t.co/E44hwJrO1G 0.4591127739176909 [[Sair, http://dbpedia.org/resource/Sa'ir, http://dbpedia.org/ontology/Place] -> 0.4591127739176909]

Bom dia, mesmo não tendo muita coisa de bom :) 0.8136938698968199 [null -> 0.8136938698968199]

RT @tricolornaveia8: o galaxy s4 é o melhor celular do mundo 0.3448275862068966 [[Wanderson, http://pt.dbpedia.org/resource/Wanderson_de_Paula_Sabino, http://xmlns.com/foaf/0.1/Person, Wanderson] -> 0.3448275862068966]

@melfronckowiak seu livro ta tão perfeito q as palavras escritas nele não sai da minha mente. Ta lindo demais, Parabéns! 1.0 [[Mel_Fronckowiak, http://pt.dbpedia.org/resource/Melanie_Fronckowiak, http://xmlns.com/foaf/0.1/Person] -> 1.0]

GENTE OLHA O BRUNO QUE LINDO NO SITE DA RADIO DA NOVA ONDA TER 1 HORA DE BRUNO MARS SÁBADO A NOITE A PARTIR DAS 19:00 http://t.co/6jb2BClSsz 0.8020328139162866 [[Onda, http://dbpedia.org/resource/Onda, http://dbpedia.org/ontology/Organisation] -> 0.8020328139162866, [Bruno_Mars, http://pt.dbpedia.org/resource/Bruno_Mars, http://xmlns.com/foaf/0.1/Person] -> 0.0, [Das, http://pt.dbpedia.org/resource/Das, http://dbpedia.org/ontology/Place] -> 0.0]

ajudando meu irmão a desenha :) 0.5259660590004452 [null -> 0.5259660590004452]

Gostei de um vídeo @YouTube http://t.co/gwjeHe0uBI (29/08) - Nintendo 2DS, trailer de GTA V, Need for Speed e Battlefield 0.40041493775933606 [[Nintendo, http://pt.dbpedia.org/resource/Nintendo, http://dbpedia.org/ontology/Organisation] -> 0.5, [Gostei, http://pt.dbpedia.org/resource/Gostei, http://dbpedia.org/ontology/Place] -> -0.0625, [Gta_v, http://dbpedia.org/resource/Grand_Theft_Auto_V, http://dbpedia.org/ontology/Software] -> 0.0, [Youtube, http://dbpedia.org/resource/YouTube, http://dbpedia.org/ontology/Organisation] -> -0.0625]

"Você tem um bom coração, entregue-o para alguém que se importe." - Gossip Girl (via renunci-ar) http://t.co/Nt0Cqv8E8s 0.3864417991541779 [null -> 0.3864417991541779]

hoje o @The_Mesini vai la ver a gent :3 0.6156848169485225 [[La, http://dbpedia.org/resource/Los_Angeles, http://dbpedia.org/ontology/Place] -> 0.6156848169485225]

RT @NeiSampaio08: @GBarbosaOficial parabéns mulek doido, tu é SANTOS FC de coração e a nação SANTISTA sempre estará com você...felicidades … 1.0 [[Gabriel, http://dbpedia.org/resource/Gabriel, http://xmlns.com/foaf/0.1/Person] -> 1.0]

@JuanRG4L_ Esta foto me encanta *.* http://t.co/d2mVoualZp 0.375 [null -> 0.375]

Bom dia a todos :) 0.7963981990995498 [null -> 0.7963981990995498]

RT @princessmiiy: Eu AMO Havaianas &lt;3 Veja: http://t.co/uYehqy9bxM 0.722904745876809 [[Havaianas, http://pt.dbpedia.org/resource/Havaianas, http://dbpedia.org/ontology/Organisation] -> 0.722904745876809]

O senso de humor do Hawking é muito bom. O cara consegue fazer sempre uma boa piada, mesmo em um livro desse nível. 0.8758245073507313 [[Hawking, http://pt.dbpedia.org/resource/Stephen_Hawking, http://xmlns.com/foaf/0.1/Person] -> 0.8758245073507313]

A.1.1.2 Neutral

Corte de pelo. http://t.co/IiGhEMKhsa -0.0015432098765432098 [null -> -0.0015432098765432098]

Helton Lima manda repertório novo no Bug's Country http://t.co/gS3zFS6pcg #CelebsPE 0.02856576076175362 [[Lima, http://dbpedia.org/resource/Lima, http://dbpedia.org/ontology/Place] -> 0.02856576076175362]

RT @babipsoares: "O fato de o mar esta calmo na superficie nao significa que algo nao esteja acontecendo nas profundezas" 0.21270974320941877 [null -> 0.21270974320941877]

9h25 saindo de trem da estação Unisinos! 0.006944444444444444 [[Unisinos, http://pt.dbpedia.org/resource/Universidade_do_Vale_do_Rio_dos_Sinos, http://dbpedia.org/ontology/Organisation] -> 0.006944444444444444]

#Fato http://t.co/UzUxljIYcU 0.0 []

RT @exquadrilhaa: Postam print da tela do celular e eu fico olhando as horas, quanto tem de bateria, a operadora, se tem sinal, etc 0.21081809302148377 [[Tela, http://dbpedia.org/resource/Tela, http://dbpedia.org/ontology/Place] -> 0.10660702644733103, [Do, http://pt.dbpedia.org/resource/Do, http://xmlns.com/foaf/0.1/Person] -> 0.10660702644733103]

@eduardomps ele não divulgou apenas, ele criou um desafio para que pessoas postassem, mandassem por IMs, etc 0.06965643182610472 [null -> 0.06965643182610472]

RT @tessa0032: Elsa,eres mi gemela o k 0.0 [null -> 0.0]

RT @_anaxl: vida loka bjs @gicabaladobr http://t.co/xl5GEdQoKl 0.010869565217391304 [null -> 0.010869565217391304]

POR FAVOR, QUE ME MEO. http://t.co/SyKSbdKbts 0.140625 [[Meo, http://pt.dbpedia.org/resource/Meo, http://dbpedia.org/ontology/Organisation] -> 0.140625]

RT @TimBetaLab_1: Informações sobre preço do convite, prazo de recebimento do chip #timbeta? Email para perfistimbeta@gmail.com que explico… 0.2276759804327999 [null -> 0.2276759804327999]

Eu publiquei uma nova foto no Facebook http://t.co/Ik0wba8CDL 0.011904761904761904 [[Facebook, http://pt.dbpedia.org/resource/Facebook, http://dbpedia.org/ontology/Organisation] -> 0.011904761904761904]

Representantes de cidades discutem ações de empreendedorismo http://t.co/OEs77nR9Js @sebraesp 0.027777777777777776 [null -> 0.027777777777777776]

x o x o - | via Tumblr http://t.co/zC6meZY44k 0.0 [[Tumblr, http://dbpedia.org/resource/Tumblr, http://dbpedia.org/ontology/Organisation] -> 0.0]

Começando as atividades do dia em Foz do Iguaçu! #Capacitação #fitoterápicos #plantasmedicinais http://t.co/ORhp10oS2c 0.25764030889695966 [[Foz, http://dbpedia.org/resource/Foz, http://dbpedia.org/ontology/Place] -> 0.13103191438890213, [Do, http://pt.dbpedia.org/resource/Do, http://xmlns.com/foaf/0.1/Person] -> 0.13103191438890213]

Olá vocêsssss http://t.co/TpIE2sK2Yk 0.0 [null -> 0.0]

RT @fxtwolf: pq faz isso http://t.co/z4LFpBu264 0.0 [null -> 0.0]

@sounegativa eu acho que eu vou de novo se eles derem o replay 0.22157809332296016 [null -> 0.22157809332296016]

RT @alexandraalsm: #AquelaDorQuando o benfica joga em casa 0.06474708411439996 [[Benfica, http://dbpedia.org/resource/S.L._Benfica, http://dbpedia.org/ontology/Organisation] -> 0.032407542330894996, [Martins, http://pt.dbpedia.org/resource/Martins, http://dbpedia.org/ontology/Place, Martins] -> 0.032407542330894996]

@samantassb hahaha mas qual é o rolee? u.u 0.0 [null -> 0.0]

Casamento em Mykonos! Decoraçao simples, mas tudo a ver com o clima daqui!!! #mykonos #wedding #decor #flower... http://t.co/iNACe1dEEc 0.31204396853975175 [[Mykonos, http://dbpedia.org/resource/Mykonos, http://dbpedia.org/ontology/Place] -> 0.31204396853975175]

@tecomedina @samershousha @drunkeynesian Agüentem o Mantega hoje. Vai dizer que a nova matriz econômica está mostrando o seu valor (MM) 0.30638097994701347 [[Samer, http://pt.dbpedia.org/resource/Samer, http://dbpedia.org/ontology/Place] -> 0.30638097994701347]

Photo: Abs http://t.co/bJ3MnUlfXq 0.0 [null -> 0.0]

RT @Milanello: OFFICIAL: Matri joins Milan. http://t.co/hQmKLfBvcO #BentornatoMatri #MatriRossonero @Ale_Matri 0.07333994053518333 [[Alessandro_Matri, http://pt.dbpedia.org/resource/Alessandro_Matri, http://xmlns.com/foaf/0.1/Person] -> 0.0, [Milan, http://dbpedia.org/resource/Milan, http://dbpedia.org/ontology/Place] -> 0.07333994053518333]

RT @leo30ortiz: https://t.co/Fo4Gfxvl7o sangue meu! @ortiinsz 0.0 [null -> 0.0]

Se tiver 30 pessoas em todo o colegio eh mt 0.3125 [null -> 0.3125]

Ego 01h21 If Were a Boy 07h07 #MixFmBrasil 3532 0.1 [null -> 0.1]

#QuieroMiChulengo ___ ___() / / @Tarjeta_Naranja http://t.co/hjgGzcwLXT 0.0 [[Naranja, http://dbpedia.org/resource/Naranja,_Florida, http://dbpedia.org/ontology/Place] -> 0.0]

Tumblr http://t.co/G5Ah3nexGw 0.0 [[Tumblr, http://dbpedia.org/resource/Tumblr, http://dbpedia.org/ontology/Organisation] -> 0.0]

RT @WorldKitKat: ” eu no começo do fc ” http://t.co/WNcBLUindQ 0.16354816354816354 [null -> 0.16354816354816354]

@dmarcelocoelho vens a Paços agora? -0.10612563402262974 [null -> -0.10612563402262974]

@welovesophiaa @lovetruesophiaa Mudanças comercial e residencia http://t.co/swt3iGeEZc 11 5816-7145 -0.047081902893575285 [null -> -0.047081902893575285]

Claqué *O* 0.0 [null -> 0.0]

O austin parece, até o justin quando tinha 16 anos awn -0.004807692307692308 [[Austin, http://pt.dbpedia.org/resource/Austin, http://dbpedia.org/ontology/Place] -> -0.004807692307692308]

Bom Dia... Céu nublado aqui. E aí como está ? 0.18877005347593587 [null -> 0.18877005347593587]

@bandaRAISED eiei olha la rapidão -0.020387359836901122 [[La, http://dbpedia.org/resource/Los_Angeles, http://dbpedia.org/ontology/Place] -> -0.020387359836901122]

@nandinhabonitin pra ver o Maloka no Encontro. 0.00972492358988608 [null -> 0.00972492358988608]

@sennenka_ machão 0.0328125 [null -> 0.0328125]

Curso de Gestão Escolar Confira agora ACESSE: http://t.co/BSgQlF2xAI 0.01250390747108471 [null -> 0.01250390747108471] a fic broken fala sobre oque ? -0.10448163462026282 [null -> -0.10448163462026282]

Álbum de fotos: http://t.co/rmymkEmlc9 0.0 [[Fotos, http://dbpedia.org/resource/Fotos, http://dbpedia.org/ontology/Organisation] -> 0.0]

oi oi oi &gt;&lt; http://t.co/kFSjc1eBiJ 0.0 [null -> 0.0]

@jctransito Essa informação procede? http://t.co/5fbs1lezyG 0.0 [null -> 0.0]

A.1.1.3 Negative

RT @GigiRavaglia: Eu ja sai , ja voltei pra Ez e ainda não conegui minha foto com o @MathInglada -0.7460428258564784 [null -> -0.7460428258564784]

em nome de jesus, que o erie nau leia isso, pois morro de ciumes dele falando com quem ja ficou -0.20431586399629167 [[Jesus, http://pt.dbpedia.org/resource/Jesus, http://xmlns.com/foaf/0.1/Person] -> -0.20431586399629167]

quando eu assim meia doente eu fico mais sentimental do que o normal '-' aosl'auhsijoklas -0.5545454545454547 [null -> -0.5545454545454547]

RT @claudippinheiro: @sofiavieiraa3 VACA É A TUA TIA -0.1875 [null -> -0.1875]

RT @gilhogybe: #AquelaDorQuando os teus amigos têm todos iPhone e tu não -0.36749770470029136 [null -> -0.36749770470029136]

@alexiafpaula Nossa qe raiva mo to morrendo aquii.. :| puta q prova treta... -0.6435656843409276 [[Barbie, http://pt.dbpedia.org/resource/Barbie, http://xmlns.com/foaf/0.1/Person] -> -0.6435656843409276]

"A alma distraída é facilmente enganada." (Santo Padre Pio de Pietrelcina) -0.5836701359076916 [[Padre_Pio, http://pt.dbpedia.org/resource/Padre_Pio, http://xmlns.com/foaf/0.1/Person] -> -0.32211538461538464, [Pietrelcina, http://pt.dbpedia.org/resource/Pietrelcina, http://dbpedia.org/ontology/Place] -> -0.32211538461538464]

Então... Tem um trouxa, e ainda quer colocar fogo no relacionamento da gente pra não bastar.. #PORQ #SERÁ ?... http://t.co/jn955syd9x -0.2978220942469701 [null -> -0.2978220942469701]

RT @suckdemi: odiei essa atualização do twitter de mostrar a conversa dos outros na tl -0.7853260869565216 [null -> -0.7853260869565216]

RT @itguels: A pior coisa no inverno não é estar solteira, é ter q lavar louça -0.5039479130321799 [null -> -0.5039479130321799]

ODEIO o DETRAN, ODEIO essas filas -0.75 [null -> -0.75]

RT @instagranzim: a pessoa acha q só pq não gostam dela é pq tem inveja, ninguem gosta do esgoto mas nem por isso todo mundo tem inveja -0.4380254710721983 [[Instagram, http://pt.dbpedia.org/resource/Instagram, http://dbpedia.org/ontology/Software] -> -0.4380254710721983]

RT @apenasperf: ew odeio o meu cabelo -0.75 [null -> -0.75]

RT @MasterColorado: Não tentem achar desculpas, NÃO EXISTEM. Elenco milionário contra um time de SEI LÁ QUE DIVISÃO. Até o sub-17 tinha obr… -0.23601416084240945 [[Colorado, http://dbpedia.org/resource/Colorado, http://dbpedia.org/ontology/Place] -> -0.23601416084240945]

@lucaslealfo @brunnobarcellos nada não, é só esse Brunno que quer levar uma surra. -0.16102408237224217 [null -> -0.16102408237224217]

mas o que me deixa mais puto é o trânsito velho. É muito desrespeito com o trabalhador! Empresa nao entende nao. -0.9080489938507181 [null -> -0.9080489938507181]

RT @katiaelias0: Se a outra pessoa seguiu em frente, te ignorou e te fez sentir uma merda, faz-lhe o mesmo, não há outra solução. -0.16383992836128708 [[Frente, http://dbpedia.org/resource/Frente!, http://dbpedia.org/ontology/Organisation] -> -0.16383992836128708]

E destes dias tão estranhos, fica a poeira se escondendo pelos cantos -0.1514138968919065 [null -> -0.1514138968919065]

Mais uma noite mal dormida e vou passar o dia no posto -0.22463347417619917 [null -> -0.22463347417619917]

e o pior, metade são "colegas" do meu namorado euheiuhfuehfieh fico pretérita -0.5230769230769231 [null -> -0.5230769230769231]

o site onde eu posso ver os valores não carrega, uhuu -0.6068802734745924 [null -> -0.6068802734745924]

RT @PakiStonedMan: Um puto que fuma ganzas porque é fixe não é um stoner seus caras de merda -0.4324345067539909 [null -> -0.4324345067539909]

@wtfabel desejar o mal dos próprios irmãos é muito errado, vai dizer que não sabia agora né -0.623141003717845 [null -> -0.623141003717845]

Caralho, iPhone maldito! Passa logo a porra dessas fotos! :(-0.7608695652173912 [null -> -0.7608695652173912]

RT @estrondeira: Vocês são uns inconstantes, nunca sabem o que querem e depois fodem os outros -0.6704131227217497 [null -> -0.6704131227217497]

ninguém ta votando mais não é #MPN #LarissaManoela -0.8098045191883961 [null -> -0.8098045191883961]

To com muita dor :// -0.7114093959731543 [null -> -0.7114093959731543]

Crllllll não tem nada pra fazerrrrrrr -0.625 [null -> -0.625]

RT @nearselala: @brancaden3ve nao vai da pra ir na sua casa pq minha mae qr que eu ajude-a a arrumar aqui srry): -0.4758525337360898 [null -> -0.4758525337360898]

A.1.2 Agreement with two annotators

A.1.2.1 Positive

Hoje eu só quero que o dia termine bem. 0.33104690093039474 [null -> 0.33104690093039474]

Ei @gabesimas, quero conhecer o Emblem3 no show do RJ #ZFestival2013! http://t.co/oJ2iHKVaLM 166 0.6106730661147832 [[Do, http://pt.dbpedia.org/resource/Do, http://xmlns.com/foaf/0.1/Person] -> 0.34079966506175424, [Rj, http://pt.dbpedia.org/resource/Rio_de_Janeiro, http://dbpedia.org/ontology/Place] -> 0.34079966506175424]

RT @kathegldm: Ummm mas lindas http://t.co/RggWHnyxGd 0.6875 [null -> 0.6875]

A.1.2.2 Neutral

RT @Shopping10Natal: Bom dia, estamos aberto nesta terça até ás 19hs. https://t.co/nvXzJ835e5 0.18373646643516908 [null -> 0.18373646643516908]

@LOLGAMESPL ja viram os novos vídeos do canal Tree House? chega mais http://t.co/EDD2MOPNbq SE INSCREVA, NOS DIGA O QUE ACHOU, OBRIGADO 0.0012829757387263336 [null -> 0.0012829757387263336]

Netshoes lança loja da NBA no México http://t.co/Zjng1TCRpR -0.035702900860694935 [[Nba, http://dbpedia.org/resource/National_Basketball_Association, http://dbpedia.org/ontology/Organisation] -> -0.017857142857142856, [Netshoes, http://pt.dbpedia.org/resource/Netshoes, http://dbpedia.org/ontology/Organisation] -> 0.0, [México, http://dbpedia.org/resource/Mexico, http://dbpedia.org/ontology/Place] -> -0.017857142857142856]

Publiquei 7 fotos no Facebook no álbum "3 meses! uhuuuuu" http://t.co/cVYwFnHx91 0.0 [[Facebook, http://pt.dbpedia.org/resource/Facebook, http://dbpedia.org/ontology/Organisation] -> 0.0, [Fotos, http://dbpedia.org/resource/Fotos, http://dbpedia.org/ontology/Organisation] -> 0.0]

Acordei agora. Hoje não tive prova 0.22090589230837746 [null -> 0.22090589230837746]

RT @LivroDeFrases: "Quando o vento está contra você significa que está na hora de mudar de direção." 0.2902985988000644 [null -> 0.2902985988000644]

Gente, é o alambique que ta pegando fogo?? :O -0.03448275862068967 [null -> -0.03448275862068967]

A.1.2.3 Negative

@fontes_mel @igordomingos @_ingridemmily Não. Ai, amiga. Desculpa! Mas é feio pra caralho. kkkkkkkkkkkkk. -0.20632766316805404 [[Natiruts, http://pt.dbpedia.org/resource/Natiruts, http://dbpedia.org/ontology/Organisation] -> -0.20632766316805404]

não consigo fazer aquele "muahah" maléfico -0.6091793623894053 [null -> -0.6091793623894053]

Agora tô escutando a radio ultra do Funck :S -0.4925955012060358 [null -> -0.4925955012060358]

"Existem pessoas que passam pelas nossas vidas justamente para nos ensinar a não sermos como elas" -0.43863179074446673 [null -> -0.43863179074446673]

Mari Alexandre relembra vergonha dos seios fartos na adolescência - http://t.co/qVraj2LhhG -0.2584745762711864 [[Mari_Alexandre, http://pt.dbpedia.org/resource/Mari_Alexandre, http://xmlns.com/foaf/0.1/Person] -> -0.2584745762711864]

To tãaao confusa :s -0.6329113924050632 [null -> -0.6329113924050632]

A.1.3 No Agreement

A.1.3.1 Neutral

@MauroBambil AHUHAUHAU não faça isso vive caindo e não quero dar suporte pra vc HUAHUAHUAHU 0.02514045073876008 [null -> 0.02514045073876008]

RT @horanwade: hj os 5 estavam juntos jogando bola pelo amor de deus alguém me ajuda http://t.co/kgGEEaPko0 0.31122113691150843 [[Ajuda, http://dbpedia.org/resource/Ajuda_(Lisbon), http://dbpedia.org/ontology/Place] -> 0.31122113691150843]

A.2 Incorrect Classifications

A.2.1 Full agreement between the annotators

A.2.1.1 Positive

Gostei de um vídeo @YouTube de @iamgak http://t.co/TqKCLegg0i Back To School Bully -0.7035415248123978 [[Gostei, http://pt.dbpedia.org/resource/Gostei, http://dbpedia.org/ontology/Place] -> -0.4112704629673062, [Youtube, http://dbpedia.org/resource/YouTube, http://dbpedia.org/ontology/Organisation] ->

-0.4112704629673062]

Aaaaadoro questa foto *–* http://t.co/YmCt0knj0e 0.0 [[Questa, http://dbpedia.org/resource/Questa,_New_Mexico, http://dbpedia.org/ontology/Place] -> 0.0]

Angela me deu uva , que auxiliou a nao dormir na aula de bio hahahha 0.04193971166448229 [null -> 0.04193971166448229]

o amor não se vê com os olhos apenas com o coração... 0.09810546639314591 [null -> 0.09810546639314591]

@quadrophenia__ não tem como você deixar de ser especial, é uma amiga 0.02398956975228162 [null -> 0.02398956975228162]

Bom dia. Nada melhor do que um café da manha reforçado pra começar o dia. #amor #ferias #taacabando… http://t.co/UPtQvfDvn1 0.22117500928662687 [null -> 0.22117500928662687]

"... te farei as vontades. Direi meias verdades sempre à meia luz. E te farei, vaidoso, supor que és o maior e que me possuis..." - Chico -0.754287812800933 [null -> -0.754287812800933]

RT @fanaticanolubs: FEEEELIZ *-* NÃO TEM AULA HJ U.U -0.625 [null -> -0.625]

Mee, amanhã já é o Cacio e Marcos, mas que belezaaa. *—* -0.039859951521680585 [null -> -0.039859951521680585]

Para começar bem a sexta-feira: Jess Greenberg, cantando Highway to hell - AC/DC (cover) http://t.co/rxlBF5U9Wt -0.03681675628332618 [[Ac/dc, http://pt.dbpedia.org/resource/AC/DC, http://dbpedia.org/ontology/Organisation] -> -0.03681675628332618]

Ei @gabesimas quero conhecer o Emblem3 no show do Rio de Janeiro. #ZFestival2013! http://t.co/E8pYrllNJd ser fã é torturador e gratificante -0.9521094888122736 [[Rio, http://pt.dbpedia.org/resource/Rio_de_Janeiro, 13] -> -0.5497047244094488, [Do, http://pt.dbpedia.org/resource/Do, http://xmlns.com/foaf/0.1/Person] -> -0.5497047244094488, [Janeiro, http://pt.dbpedia.org/resource/Rio_de_Janeiro, 13] -> -0.5497047244094488]

Ainda assim, a Solange tem estilo. -0.14514145141451412 [[Solange, http://dbpedia.org/resource/Solange, http://xmlns.com/foaf/0.1/Person] -> -0.14514145141451412]

RT @mariagandinmore: @yerai_toloko es perfectamente perfecto asfkjfhx http://t.co/25zDBeriQO 0.0 [null -> 0.0]

Gostei de um vídeo @YouTube de @kronosplaying http://t.co/7W6EzqYSLv Minecraft 1.5.2 - Mo' Creatures com pasta .Minecraft Mods &amp; -0.3227340422752332 [[Gostei, http://pt.dbpedia.org/resource/Gostei, http://dbpedia.org/ontology/Place] -> -0.16580310880829016, [Minecraft, http://pt.dbpedia.org/resource/Minecraft, http://dbpedia.org/ontology/Software] -> 0.0, [Youtube, http://dbpedia.org/resource/YouTube, http://dbpedia.org/ontology/Organisation] -> -0.16580310880829016]

Vamos com força tottais! #MPN #ClaudiaLeitte 0.03398191285283639 [null -> 0.03398191285283639]

seu maior sonho http://t.co/p2rEdYsEST -0.14433756729740643 [null -> -0.14433756729740643]

RT @BrunoMotta_3: Porra, mas não há nada melhor que pele na pele hehehehehe -0.12405929304446976 [null -> -0.12405929304446976]

A.2.1.2 Neutral

DESMANCHE: Vendo para desmanche, parati batida ano 2000 completa sem documentação, motor funcionando e caixa n... http://t.co/TD5yWZT6aw -0.3527720418849577 [null -> -0.3527720418849577]

que isso sem or -0.625 [null -> -0.625]

Acreditam-se se eu disser que ainda não vi o anuncio do continente este ano??????? -0.204398447606727

[null -> -0.204398447606727]

To assistindo Mais Você porque é sobre comida. 0.4881831610044314 [null -> 0.4881831610044314]

Estamos presentando #Deportes de @NTN24 | | 12:30 GMT. http://t.co/44F0eapOWa 0.45714285714285713 [null -> 0.45714285714285713]

Va o autocarro deve tar a chegar. Bisou 0.3434389314177303 [null -> 0.3434389314177303]

Que onda entonces? O.o 0.375 [[Onda, http://dbpedia.org/resource/Onda, http://dbpedia.org/ontology/Organisation] -> 0.375]

RT @daicristinac: Queria saber se é normal carregar o celular três vezes em um dia. 0.46143069671442416 [null -> 0.46143069671442416]

Bom, se vc fez uma cirurgia, n tome o primeiro banho pós operatório sozinho De vdd... 0.3151573046970938 [null -> 0.3151573046970938]

@gguidorizzi mas isso é de forma geral ne...NÃO É SÓ O SBT -0.619895020681711 [null -> -0.619895020681711]

A.2.1.3 Negative

RT @Expectohoran: Quem não sabe amar o primeiro não deve amar o segundo... http://t.co/9xMVkvqIfC 0.2180207387513463 [null -> 0.2180207387513463]

PEDREIRO MORRE APÓS LEVAR DESCARGA DE 13.000 VOLTS - PE: Um pedreiro morreu eletrocutado no fim da tarde de qu... http://t.co/DdSatCUsW7 -0.052018832819776754 [[Morre, http://pt.dbpedia.org/resource/Morre, http://dbpedia.org/ontology/Place] -> -0.052018832819776754]

@mefeatliam porq toda hora vc muda o user e eu n lembro na hora qual vc ts 0.39221025344994714 [null -> 0.39221025344994714]

@naielylemos Bah, que saco isso aí, já pagamos um absurdo pra essa merda não trabalhar ainda 0.30743451213636336 [[Bah, http://pt.dbpedia.org/resource/Bah, http://dbpedia.org/ontology/Place] -> 0.30743451213636336]

A @unijorge tá supe vazia... 0.127208480565371 [null -> 0.127208480565371]

@PutaQuePariuMah Né, não sabe respeitar as meninas af. 0.416144771337023 [null -> 0.416144771337023]

A unica coisa que eu gosto na sexta é a aula de Lu, e eu faltei /: 0.31247931148626285 [[Lu, http://pt.dbpedia.org/resource/Lu, http://dbpedia.org/ontology/Place] -> 0.31247931148626285]

Legal, a porra do vinagre vazou no caralho da minha mochila... bendita experiência de química!!!! -0.12474132146214578 [null -> -0.12474132146214578]

"A CBF não faz obras de infraestrutura. Somos auditados". Palavras de José Maria Marin, Presidente da CBF, no evento da Lide de @jdoriajr -0.007102272727272728 [null -> -0.007102272727272728]

Vou corta o alemão da foto 0.0581591177264273 [null -> 0.0581591177264273]

Estou triste com tudo isso que anda acontecendo , mais vou fazer oque ne ? Mais eh bom saber que posso continuar com a cabeça erguida 0.5767187372032078 [null -> 0.5767187372032078]

eu não aguento mais esses pedreiros aqui em casa puta merda 0.29709223279680713 [null -> 0.29709223279680713]

tenho que trabalhar hoje mais eu to morrendo d preguiça bosta kkk 0.10122561714413263 [null -> 0.10122561714413263]

VELHO eu coloquei o chip no outro celular e ele também travou -0.11484993555514637 [null -> -0.11484993555514637]

Eu não quero lembrar que tudo acabou pra mim 0.40169226001928104 [null -> 0.40169226001928104]

Crianças são sádicas. Os bulling são mais forte na fase escolar. Eu fui vítma disso, e tenho nojo de mim mesmo ao... http://t.co/LaCKR2G2TG 0.2681275961174492 [null -> 0.2681275961174492]

a france tem mais defeitos que qualidades? — Hoje eu não sei ... http://t.co/HHtbUoEblh 0.012793728537006761 [[France, http://dbpedia.org/resource/France, http://dbpedia.org/ontology/Place] -> 0.012793728537006761]

RT @PedroHenrikeSM: As pessoas reclamam dos seus problemas quando na verdade os seus problemas são elas mesmas. 0.07584056705295154 [null -> 0.07584056705295154]

pq que a minha mãe acha que eu tenho que ir pra todo lugar com ela,só pq eu tenho quinze anos 0.301126935840708 [[Anos, http://pt.dbpedia.org/resource/Anos, http://dbpedia.org/ontology/Place] -> 0.301126935840708]

Larga tudo e vem correndo vem matar minha vontade já faz tempo que eu to sofrendo mereço um pouco de felicidade... 0.7721947149546377 [null -> 0.7721947149546377]

Mó preguiça de levantar da cama #credo 0.03195043186106807 [null -> 0.03195043186106807]

já que nao tenho namorado para me comprar uma hello kitty nova, acho que hoje vou perder a cabeça e comprar 0.25599663327536676 [null -> 0.25599663327536676]

O primeiro a ser preso foi Carlos Humberto Peluchera de Abreu, 32 anos. 0.03573875626935621 [null -> 0.03573875626935621]

Detesto caps em gajos, nao conheço um que fique bem, só gajas -0.04798516067724327 [null -> -0.04798516067724327]

RT @CharlesPantanal: "@AneliseMossmann: Lobão conseguiu estragar vida louca do Cazuza –'" #taqueospariu #takarai #tamerda -0.11182489050049217 [[Cazuza, http://pt.dbpedia.org/resource/Cazuza, http://xmlns.com/foaf/0.1/Person] -> -0.11182489050049217]

CARA, Vou ter que entrar no procon não é possivel essa porra desse atendimento derruba sua ligação :@ -0.045423527060748425 [null -> -0.045423527060748425]

RT @AndreRadunz: To irritado com esse negocio de mostrar a conversa toda ligada por pontinhos 0.4699711703966665 [null -> 0.4699711703966665]

Em dois meses, mais de 40 morrem à espera de transplante de rim em Sergipe http://t.co/ZsjhwZrOGH 5.173448093423734E-4 [[Sergipe, http://pt.dbpedia.org/resource/Sergipe, http://dbpedia.org/ontology/Place] -> 5.173448093423734E-4]

Mas meu @Deus me ajuda a destruir a macumba das inimigas. -0.08647036898993664 [[Meu, http://pt.dbpedia.org/resource/Meu, http://dbpedia.org/ontology/Place] -> -0.08647036898993664]

A.2.2 Agreement with two annotators

A.2.2.1 Positive

Vale lembrar.@marinasantanna ficha limpa,tem varios projetos camara da sua autoria para http://t.co/XJmFUXXGeU malandragem em nosso brasil 0.0808554343648198 [[Marina_Sant'anna, http://pt.dbpedia.org/resource/Marina_Sant'anna, http://xmlns.com/foaf/0.1/Person] -> 0.0808554343648198]

RT @DavidLucass: To zuando...só falei isso pra ver se vcs iam se importar. hajahhajajajaj e pra dar uma pitadinha de adrenalina nessa noite. . . 0.027060245734418616 [[David_Lucas, http://pt.dbpedia.org/resource/David_Lucas, http://xmlns.com/foaf/0.1/Person] -> 0.027060245734418616]

Dá pra sentir daí mesmo a maciez do tapete Nardo, em couro natural.... http://t.co/GVpJCMomFH 0.17373762064883527 [null -> 0.17373762064883527]

@PiratinhaEXIT ver o video da minha equipe? PMD ft. Friends - Um Dia Feliz #2 [FREE STEP] http://t.co/HyOKZQTASr agredeço e retribuo. 0.057473967677002205 [[Feliz, http://pt.dbpedia.org/resource/Feliz, http://dbpedia.org/ontology/Place] -> 0.057473967677002205]

no meu instagram as unicas menin(A)s que são bunita é a Carol a Laine a hellen e a Emilly por que o resto pode joga fora .-. 0.03211496020055761 [null -> 0.03211496020055761]

Por fora somos comuns. É por dentro que está a diferença de cada um. -0.22132331696041604 [null -> -0.22132331696041604]

RT @Junys_: Acho engraçado sexta feira, o povo falando "Uhul, chegou sexta! Vou beber todas" e quando vc vai ver, a pessoa nao sai nem da c... -0.5348968300212289 [[Feira, http://dbpedia.org/resource/Feira_(Santa_Maria_da_Feira), http://dbpedia.org/ontology/Place] -> -0.5348968300212289]

Kkkkk ta todo Mundo com o demo 0.007911392405063306 [null -> 0.007911392405063306]

RT @rogercustodio13: ME SIGA QUE EU SIGO DE VOLTA GALERA ø/ #TIMBETALAB 03 -0.03052363169185398 [[Eu, http://dbpedia.org/resource/European_Union, http://dbpedia.org/ontology/Place] -> -0.01526537233117662, [Siga, http://dbpedia.org/resource/Shiga_Prefecture, http://dbpedia.org/ontology/Place] -> -0.01526537233117662]

RT @KatyPerryBR: Sara Bareilles falou sobre as comparações entre Brave e Roar: "Katy é uma amiga de longa data, ela me manda SMS sempre. Nã... 0.25085203528624017 [[Brasil, http://pt.dbpedia.org/resource/Brasil, http://dbpedia.org/ontology/Place] -> 0.12746381725081246, [Sara_Bareilles, http://pt.dbpedia.org/resource/Sara_Bareilles, http://xmlns.com/foaf/0.1/Person] -> 0.12746381725081246]

a tarde vo tira pra durmi heheee -0.045620178884103926 [null -> -0.045620178884103926]

A.2.2.2 Neutral

vou levantar mais não é pq meu pai tá saindo e pá, mais pq tô com fome haushuas -0.21404047988998745 [null -> -0.21404047988998745]

Faz cota que não vejo Invento na hora :S -0.39219589886522 [null -> -0.39219589886522]

Vou sair mais cedo, vai da pra ver o maloka no encontro, uhuuu 0.4093443785165159 [null -> 0.4093443785165159]

A Mel on aaawnt' &lt; 3 ;') 0.5454545454545454 [null -> 0.5454545454545454]

@nahpeixinha ta reclamando de quêe??!! eu e a Cami q ia faze junto hj... u.u ia fica melhor ainda.... hahahaha' 0.5088666152659985 [null -> 0.5088666152659985]

Curtem please... Team Paul Wesley - Brasil http://t.co/p2Ff79KBoA 0.643678587016916 [[Paul_Wesley, http://pt.dbpedia.org/resource/Paul_Wesley, http://xmlns.com/foaf/0.1/Person] -> 0.3646292795631169, [Brasil, http://pt.dbpedia.org/resource/Brasil, http://dbpedia.org/ontology/Place] -> 0.3646292795631169]

A.2.2.3 Negative

As pessoas que mais amamos são as mais dificies de mater por perto... 0.5185185191994445 [null -> 0.5185185191994445]

@diogoandregomes CONHEÇO A BANDA MAS NUNCA CURTI MUITO AHAH 0.1501110682455395 [null -> 0.1501110682455395]

Mano to comendo mt pao ultimamente 0.030075187969924814 [null -> 0.030075187969924814]

iPhone tem um caso amoroso com a tomada elétrica 0.1988349455496745 [null -> 0.1988349455496745]

To na aula de matemática –' 0.0 [null -> 0.0]

A.2.3 No Agreement

A.2.3.1 Neutral

Levei uma bolada na cabeça, fiz engolir a bola também :) 0.4785343444388182 [null -> 0.4785343444388182]

@negah_do_AxL sim...a bebida entra e a verdade, sobre a mentira q essa barbara eh , saiiii kkkk bebadavadiaEvans #AFazenda 0.46772753602417344 [[Do, http://pt.dbpedia.org/resource/Do, http://xmlns.com/foaf/0.1/Person] -> 0.46772753602417344]

@mdsmanoela Não Tive Aula Hj u-u -0.625 [null -> -0.625]