UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ELLEN POLLIANA RAMOS SOUZA

**SWARM OPTIMIZATION CLUSTERING METHODS FOR OPINION
MINING**

RECIFE

2017

ELLEN POLLIANA RAMOS SOUZA

**SWARM OPTIMIZATION CLUSTERING METHODS FOR OPINION
MINING**

*A Ph.D. Thesis presented to the Center for Informatics of
Federal University of Pernambuco in partial fulfillment of
the requirements for the degree of Philosophy Doctor in
Computer Science.*

Advisor: Prof. Dr. Adriano Lorena Inácio de Oliveira

RECIFE
2017

Catálogo na fonte
Bibliotecária Joana D'Arc Leão Salvador CRB 4-572

S729s Souza, Ellen Polliana Ramos.
Swarm optimization clustering methods for opinion mining / . – 2017.
109 f.: fig., tab.

Orientador: Adriano Lorena Inácio de Oliveira.
Tese (Doutorado) – Universidade Federal de Pernambuco. CIN. Ciência da Computação, Recife, 2017.
Inclui referências e apêndices.

1. Inteligência artificial. 2. Mineração de opinião. 3. Agrupamento de opinião. 4. Otimização de enxame. I. Oliveira, Adriano Lorena Inácio de (Orientador). II. Título.

006.3 CDD (22. ed.) UFPE-MEI 2017-129

Ellen Polliana Ramos Souza

Swarm Optimization Clustering Methods for Opinion Mining

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutora em Ciência da Computação

Aprovado em: 22/02/2017

Orientador: Prof. Dr. Adriano Lorena Inácio de Oliveira

BANCA EXAMINADORA

Profa. Dra. Flávia de Almeida Barros
Centro de Informática / UFPE

Prof. Dr. Cleber Zanchettin
Centro de Informática / UFPE

Prof. Dr. Renato Fernandes Correa
Departamento de Ciência da Informação/UFPE

Profa. Dra. Renata Vieira
Faculdade de Informática/PUC-RS

Prof. Dr. George Gomes Cabral
Departamento de Estatística e Informática/UFRPE

*I dedicate this Thesis to my parents, Antonio and Biatríz,
who have always supported me and taught me to never give
up my goals.*

Acknowledgements

I thank God for giving me health to carry out this work.

I thank my advisor, Prof. Adriano Lorena, for having accepted to guide a student, living and working 400 km away from the Center for Informatics, with small children, and little experience in the research area of the thesis.

I thank my parents, husband, and family for all support and encouragement for the realization of this dream. I thank also my friend, Richard McGill, for the support and review of papers related to this Thesis.

*"Obstacles are those frightful things you see when you take your eyes off
your goal."*

— HENRY FORD

Abstract

Opinion Mining (OM), also known as sentiment analysis, is the field of study that analyzes people's sentiments, evaluations, attitudes, and emotions about different entities expressed in textual input. This is accomplished through the classification of an opinion into categories, such as positive, negative, or neutral. Supervised machine learning (ML) and lexicon-based are the most frequent approaches for OM. However, these approaches require considerable effort for preparing training data and to build the opinion lexicon, respectively. In order to address the drawbacks of these approaches, this Thesis proposes the use of unsupervised clustering approach for the OM task which is able to produce accurate results for several domains without manually labeled data for the training step or tools which are language dependent. Three swarm algorithms based on Particle Swarm Optimization (PSO) and Cuckoo Search (CS) are proposed: the DPSOMUT which is based on a discrete PSO binary version, the IDPSOMUT that is based on an Improved Self-Adaptive PSO algorithm with detection function, and the IDPSOMUT/CS that is a hybrid version of IDPSOMUT and CS. Several experiments were conducted with different corpora types, domains, text language, class balancing, fitness function, and pre-processing techniques. The effectiveness of the clustering algorithms was evaluated with external measures such as accuracy, precision, recall, and F-score. From the statistical analysis, it was possible to observe that the swarm-based algorithms, especially the PSO ones, were able to find better solutions than conventional grouping techniques, such as K-means and Agglomerative. The PSO-based algorithms achieved better accuracy using a word bigram pre-processing and the Global Silhouette as fitness function. The OBCC corpus is also another contribution of this Thesis and contains a gold collection with 2,940 tweets in Brazilian Portuguese with opinions of consumers about products and services.

Keywords: Swarm Optimization. Text Clustering. Opinion Mining. Opinion Clustering. Twitter.

Resumo

A mineração de opinião, também conhecida como análise de sentimento, é um campo de estudo que analisa os sentimentos, opiniões, atitudes e emoções das pessoas sobre diferentes entidades, expressos de forma textual. Tal análise é obtida através da classificação das opiniões em categorias, tais como positiva, negativa ou neutra. As abordagens de aprendizado supervisionado e baseadas em léxico são mais comumente utilizadas na mineração de opinião. No entanto, tais abordagens requerem um esforço considerável para preparação da base de dados de treinamento e para construção dos léxicos de opinião, respectivamente. A fim de minimizar as desvantagens das abordagens apresentadas, esta Tese propõe o uso de uma abordagem de agrupamento não supervisionada para a tarefa de mineração de opinião, a qual é capaz de produzir resultados precisos para diversos domínios sem a necessidade de dados rotulados manualmente para a etapa treinamento e sem fazer uso de ferramentas dependentes de língua. Três algoritmos de agrupamento não-supervisionado baseados em otimização de partícula de enxame (Particle Swarm Optimization - PSO) são propostos: o DPSOMUT, que é baseado em versão discreta do PSO; o IDPSOMUT, que é baseado em uma versão melhorada e autoadaptativa do PSO com função de detecção; e o IDPSOMUT/CS, que é uma versão híbrida do IDPSOMUT com o Cuckoo Search (CS). Diversos experimentos foram conduzidos com diferentes tipos de corpora, domínios, idioma do texto, balanceamento de classes, função de otimização e técnicas de pré-processamento. A eficácia dos algoritmos de agrupamento foi avaliada com medidas externas como acurácia, precisão, revocação e f-medida. A partir das análises estatísticas, os algoritmos baseados em inteligência coletiva, especialmente os baseado em PSO, obtiveram melhores resultados que os algoritmos que utilizam técnicas convencionais de agrupamento como o K-means e o Agglomerative. Os algoritmos propostos obtiveram um melhor desempenho utilizando o pré-processamento baseado em n-grama e utilizando a Global Silhouette como função de otimização. O corpus OBCC é também uma contribuição desta Tese e contém uma coleção dourada com 2.940 *tweets* com opiniões de consumidores sobre produtos e serviços em Português brasileiro.

Palavras-chave: Otimização de Enxame. Agrupamento de Texto. Mineração de Opinião. Agrupamento de Opinião. Twitter.

List of Figures

2.1	Opinion mining approaches	20
2.2	Text mining process	22
3.1	Basic PSO topologies	36
3.2	Particle representation for opinion clustering	42
4.1	Topsy database collection process	48
5.1	Opinion clustering method	53
5.2	Tweets after tokenization	53
5.3	Tweets after POS tagging	54
5.4	Tweets after feature reduction	54
5.5	Tweets after word bigram pre-processing	55
5.6	Opinion versus word matrix	55
5.7	Opinion similarity matrix	56
6.1	Box-plots for the IMDb corpus with two balanced classes	78
6.2	Box-plots for the IMDb corpus with two unbalanced classes	79
6.3	Friedman test for the IMDb corpus	80
6.4	Box-plots for the OBCC corpus with two balanced classes	81
6.5	Box-plots for the OBCC corpus with two unbalanced classes	82
6.6	Box-plots for the OBCC corpus with three balanced classes	83
6.7	Box-plots for the OBCC corpus with three unbalanced classes	84
6.8	Friedman test for the OBCC corpus	85
6.9	Box-plots for the Sentiment140 corpus with two balanced classes	86
6.10	Box-plots for the Sentiment140 corpus with two unbalanced classes	87
6.11	Box-plots for the Sentiment140 corpus with three balanced classes	88
6.12	Box-plots for the Sentiment140 corpus with three unbalanced classes	89
6.13	Friedman test for the Sentiment140 corpus	90
6.14	Friedman test for all evaluated algorithms (accuracy)	90
A.1	Systematic mapping process	105
A.2	Generic search string based on terms extracted from the general research question	106
A.3	Automated search string used to retrieve primary studies from digital libraries	106

List of Tables

2.1	Document level opinion clustering related work	27
2.2	Domain, granularity level and pre-processing tasks of related OM applications for the Portuguese language	31
2.3	Other resources, tools and results of OM applications for the Portuguese language .	32
3.1	Swarm-based optimization algorithms	34
3.2	Studies which performed text clustering using PSO	39
3.2	Studies which performed text clustering using PSO	40
3.2	Studies which performed text clustering using PSO	41
4.1	OBCC corpus	48
4.2	OBCC annotators information	49
4.3	OBCC gold collection	50
4.4	OBCC, Sentiment140, and IMDb corpora details	51
5.1	Floresta Sinta(c)tica grammatical tags	54
5.2	Confusion matrix for positive and negative classes	58
6.1	Parameter values used in grid search to maximize accuracy	63
6.2	Average results for IMDb Movie Review corpus using a natural language pre-processing	65
6.3	Average results for OBCC corpus using a natural language pre-processing	66
6.4	Average results for Sentiment140 corpus using a natural language pre-processing .	67
6.5	Average results for IMDb Movie Review corpus using n-gram pre-processing . . .	68
6.6	Average results for OBCC corpus using n-gram pre-processing	69
6.7	Average results for Sentiment140 corpus using n-gram pre-processing	70
6.8	Results by class for IMDb Movie Review corpus using n-gram pre-processing . . .	71
6.9	Results by class for OBCC corpus using n-gram pre-processing	72
6.10	Results by class for Stanford Sentiment140 corpus using n-gram pre-processing . .	73
6.11	Difference between natural language and n-gram pre-processing techniques	74
6.12	Average accuracy obtained by all algorithms using n-gram pre-processing	78
6.13	Average accuracy for document level polarity classification related works	79
A.1	Systematic mapping manual (M) and automated (A) data sources.	107

List of Acronyms

OM	Opinion Mining	14
ML	Machine Learning	14
IR	Information Retrieval	14
NLP	Natural Language Processing	14
DM	Data Mining	14
PSO	Particle Swarm Optimization	17
CS	Cuckoo Search	17
GA	Genetic Algorithms	33
ABC	Artificial Bee Colony	37
SA	Simulated Annealing	33
KDT	Knowledge Discovery from Text	21
TF-IDF	Term Frequency-Inverse Document Frequency	25
VSM	Vector Space Model	22
POS	Part-of-speech	16
WSD	Word Sense Disambiguation	23
ICVM	Internal Clustering Validity Measures	17
GS	Global Silhouette	24
ADDC	Average Distance of Documents to the Cluster Centroid	42
MSE	Mean Square Error	24
ED	Expected Density	56
NB	Naïve Bayes	15
ME	Maximum Entropy	15
SVM	Support Vector Machine	15
IMDb	Internet Movie Database	26
OBCC	Opinion of Brazilian Consumers Corpus	17

Contents

1	INTRODUCTION	14
1.1	Motivation	15
1.2	Research Question and Objectives	16
1.3	Document Organization	18
2	OPINION MINING	19
2.1	Basic Concepts	19
2.2	Opinion Mining Process	21
2.2.1	<i>Text Pretreatment</i>	22
2.2.2	<i>Extraction of Knowledge Model</i>	23
2.2.3	<i>Evaluation of Knowledge Model</i>	24
2.3	Clustering of Opinions	24
2.4	Opinion of Brazilian Consumers	26
3	SWARM OPTIMIZATION METHODS	33
3.1	Swarm Intelligence	33
3.2	Particle Swarm Optimization	34
3.3	Text Clustering with Particle Swarm Optimization	36
3.4	Proposed PSO-based Methods	42
3.4.1	<i>Discrete Particle Swarm Optimization with Mutation (DPSOMUT)</i>	43
3.4.2	<i>Improved Self-Adaptive PSO with Mutation (IDPSOMUT)</i>	43
3.4.3	<i>Discrete Improved Self-Adaptive PSO with Cuckoo Search (IDPSOMUT/CS)</i>	45
4	OPINION MINING CORPORA	47
4.1	Opinion of Brazilian Consumer Corpus	47
4.1.1	<i>Tweets Extraction</i>	48
4.1.2	<i>Manual Annotation</i>	48
4.2	Selected Corpora	50
5	ADOPTED METHOD	52
5.1	General Overview	52
5.2	Pre-processing	53
5.2.1	<i>Natural Language Pre-processing</i>	53
5.2.2	<i>Word Bigram Pre-processing</i>	54
5.2.3	<i>VSM + TF-IDF Weighting</i>	55
5.2.4	<i>Cosine Similarity Measure</i>	56
5.3	Processing	56
5.3.1	<i>Global Silhouette (GS)</i>	56

5.3.2	<i>Average Distance Documents to the Cluster Centroid (ADDC)</i>	57
5.4	Evaluation	57
5.4.1	<i>External Measures</i>	58
5.4.2	<i>Comparison Algorithms</i>	59
6	RESULTS	62
6.1	Experimental Setup	62
6.2	Results for Natural Language Pre-processing	63
6.3	Results for Word Bigram Pre-processing	64
6.4	Comparison between Natural Language and Word Bigram Pre-processing	65
6.5	Comparison between Clustering and Swarm-based Algorithms	68
6.5.1	<i>IMDb Movie Review</i>	68
6.5.2	<i>Brazilian Portuguese OBCC</i>	71
6.5.3	<i>Stanford Sentiment140</i>	75
6.5.4	<i>General Performance Analysis</i>	75
6.6	Comparison between GS and ADDC Fitness Functions	76
6.7	Comparison with Similar Studies	76
6.8	Final Remarks	77
7	CONCLUSION	91
7.1	Conclusions	91
7.2	Future Works	93
7.3	Published Papers	93
7.3.1	<i>Published Papers</i>	93
7.3.2	<i>Submitted Papers</i>	94
	References	96
	Appendix	103
	Appendix A - Systematic Mapping Study	104

1

INTRODUCTION

The growth of user-generated text on social media, micro-blogs, and e-commerce websites provides a massive quantity of data that allows discovering the experiences, opinions, and feelings of electors, fans, customers, and others ([MARINE-ROIG; ANTON CLAVÉ, 2015](#)). These electronic *Word of Mouth* statements expressed on the web are prevalent in the business and service industries to enable customers to share their point of view ([RAVI; RAVI, 2015](#)).

In order to enhance the acquisition of a product or service and to improve the user satisfaction, most websites provide the opportunity for users to write reviews. On the other hand, customers identify online reviews as having a significant influence on their purchase in various economic sectors: 87% of customers consider those reviews on their purchase in the hotel sector, 84% in the travel sector, 79% for restaurants, 79% in the legal sector, 78% in the automotive sector, 76% in the medical sector, and 73% for home purchasing ([COMSCORE, 2016](#)).

Since OM is a rich source of real-time information, there has been an increasing interest in the scientific community to create systems capable of extracting information from this kind of data ([BALAZS; VELÁSQUEZ, 2016](#)). Opinion Mining (OM), also known as sentiment analysis, is the field of study that analyzes people's sentiments, evaluations, opinions, reviews, attitudes, and emotions about different entities ([LIU; ZHANG, 2012](#); [PANG; LEE, 2008](#)).

OM is considered as a text classification task, in which an opinion is classified into categories, such as *positive*, *negative*, or *neutral*, using various techniques from Natural Language Processing (NLP), Information Retrieval (IR), and Data Mining (DM). For [PANG; LEE \(2008\)](#), the ideal OM tool would process a set of search results for a given item, generating a list of product attributes and aggregating opinions about each of them. Much of the subsequent research self-identified as OM fits this description, but the term has also been interpreted more broadly to include many different types of text analysis ([PANG; LEE, 2008](#); [MEDHAT; HASSAN; KORASHY, 2014](#)).

Supervised Machine Learning (ML) and lexicon-based are the most frequent approaches for OM. There are also hybrid approaches which make use of both ML and lexicons ([MEDHAT; HASSAN; KORASHY, 2014](#); [PANG; LEE, 2008](#); [RAVI; RAVI, 2015](#)). The supervised ML methods apply classification algorithms to learn underlying patterns from example data to later

attempt to classify new unlabeled data (BALAZS; VELÁSQUEZ, 2016). For these methods, two sets of annotated data are needed: one for training and other for testing. A number of supervised ML algorithms have been used to classify opinions. Naïve Bayes (NB), Maximum Entropy (ME), and Support Vector Machine (SVM) have achieved great success for OM (RAVI; RAVI, 2015).

The lexicon-based approach, also known as semantic-based or symbolic-based approach, makes use of positive opinion words, used to express some desired states, and negative opinion words, used to express some undesired states. There are also opinion phrases and expressions which together are called opinion lexicon (MEDHAT; HASSAN; KORASHY, 2014). Three main approaches are used to build opinion lexicons: manual approach, which is very time consuming; dictionary-based, in which an initial set (built manually) is augmented by searching for their synonyms and antonyms in corpora such as WordNet and thesaurus; and corpus-based, which starts with a seed list of opinion words to find other opinion words in a large corpus with context specific orientations.

Although supervised ML approaches provide a high accuracy, its process is very time consuming and requires very often the human participation (KAISER; PIAZZA, 2011; RAVI; RAVI, 2015) to build the training dataset. Symbolic techniques, in turn, provide limited accuracy (LI; LIU, 2014). In order to address the drawbacks of these two traditional OM approaches, this Thesis investigated the use of clustering algorithms for the OM task.

1.1 Motivation

One of the main steps of the OM process is the application of learning algorithms to determine the polarity of an opinion. On the one hand, there are several studies based on the symbolic approach, which applies rules and lexicons that are commonly manually crafted. On the other hand, the use of supervised ML techniques is reported in many OM applications but it requires considerable effort and domain experts to perform manual annotation of a large corpus (KAISER; PIAZZA, 2011; RAVI; RAVI, 2015).

The accuracy of OM which uses supervised ML or lexicon can be influenced by the domain of the items to which it is applied. According to PANG; LEE (2008), one reason is that the same phrase can indicate different sentiment in different domains. The authors also state that differences in vocabularies across several domains also add difficulty when applying classifiers trained on labeled data in one domain to test data in another. The same occurs when building opinion lexicon as it needs to find opinion words with context specific orientations (MEDHAT; HASSAN; KORASHY, 2014).

Furthermore, much of the literature on OM has focused on texts written in English (MEDHAT; HASSAN; KORASHY, 2014; RAVI; RAVI, 2015). As a result, several resources, such as lexicons with sentiment labels and NLP tools, are only available for the English language. Adapting such resources to other languages is also related to domain adaptation (PANG; LEE,

2008). Specific for the Portuguese language, a number of gaps in the research area could be observed after performing a Systematic Mapping Study, detailed in Appendix A. The gaps include: the absence of benchmark datasets, minimal tools to support the Portuguese processing, and a reduced number of studies compared to other languages (RAVI; RAVI, 2015).

After conducting an extensive literature review, were found only the works of Li and Liu (LI; LIU, 2010, 2012, 2014) that deal with OM as a text clustering task. The authors introduce the clustering-based sentiment analysis approach at document level which is based on K-means clustering algorithm and natural language pre-processing techniques such as the Part-of-speech (POS) tagging. As the presented approach uses language-based resources, it needs a domain adaption. Also, although K-means is an extensively useful clustering algorithm, it suffers from many shortcomings, such as: the possibility of getting stuck at *local minimum*, as well as at *local maxima*, and being affected by *noise* and *outliers* (SARKAR; ROY; PURKAYASTHA, 2014a). So, these drawback have motivated a recent interest on clustering algorithms based on global optimization, such as genetic algorithms and swarm optimization techniques (FILHO et al., 2015).

The clustering of opinions is a task of unsupervised OM. Fast and high-quality text clustering algorithms play an important role in effectively assigning an opinion to its polarity class. Opinions are classified according to their polarity as: *positive*, *negative*, or *neutral* (that is, no opinion). The opinions come from several data sources, but mainly from social networks. This kind of text brings together a considerable challenge as the sampled texts are very short to fully understand the domain. Difficulties also include grammatical and spelling errors, abbreviations, ambiguity, and irony.

This Thesis presents an approach to the clustering of opinions problem by dividing a set of opinions, comments or reviews into clusters of related opinions. The approach involves swarm intelligence methods which aims to find clusters of opinions, thus, the opinions have to be similar to opinions in the same cluster and dissimilar to opinions from other clusters. As the topic of discussion may change influencing on people's opinions, phenomenon known as *Sentiment Drift*, the proposed unsupervised methods can be fairly competitive in streams analysis since they do not require labeled training data as the OM supervised methods.

1.2 Research Question and Objectives

Considering the drawbacks presented in current OM approaches, this Thesis aims to answer the following general research question:

Is it possible to develop a novel unsupervised clustering approach for the OM task which is able to produce accurate results for several domains without manually labeled data for the training step or tools which are language dependent?

In order to answer the presented research question, the following objectives are listed:

1. Develop an algorithm for clustering opinions according to their polarity;
2. Build and make publicly available a dataset containing opinions of Brazilian consumers about products and services, so that other researchers can assess the outcome of other approaches;
3. Build a golden collection to make it possible to compare existing OM approaches, which make use of external validity measures such as: accuracy, precision, recall and f-measure; and
4. Evaluate the results comparing different text clustering algorithms using the proposed dataset for the Brazilian language, as well as, with benchmark datasets for the English language.

Thus, as the main contribution of this Thesis, three clustering swarm methods based on Particle Swarm Optimization (PSO) and Cuckoo Search (CS) are proposed: the DPSOMUT, which is based on a discrete PSO binary version; the IDPSOMUT, that is based on an Improved Self-Adaptive PSO algorithm with detection function; and the IDPSOMUT/CS, that is a hybrid version of IDPSOMUT and CS. The reason for choosing the PSO was that it has been successfully applied to clustering problems, including short texts, and it also performs a global search process (CAGNINA; ERRECALDE; INGARAMO, 2008; CAGNINA et al., 2014; INGARAMO et al., 2009, 2011; CUI; POTOK; PALATHINGAL, 2005; CUI; POTOK, 2005).

The proposed swarm-based methods consider clustering as an optimization problem, where a given arbitrary objective function must be optimized. In this thesis, the performance of two Internal Clustering Validity Measures (ICVM) widely used as fitness function in document clustering is compared. Therefore, natural language and n-gram techniques are analyzed for pre-processing the clustering of opinions.

In order to evaluate the proposed methods, three corpora with different levels of complexity with respect to size, number of opinions, domains, languages, and class balancing were chosen. Also, well known clustering and swarm-based algorithms were selected for comparison. Results were expressed in terms of accuracy, precision, recall, and f-score.

Considering the absence of benchmark corpora and resources for other languages but English, another contribution of this Thesis is the construction of the Opinion of Brazilian Consumers Corpus (OBCC), which contains *tweets* with opinions of Brazilian consumers about products and services from telecommunication, telephony, and banking domains. It is worth highlighting that, up to this point, there has been no evidence of the PSO being used for clustering opinions, neither also studies dealing with opinion clustering for *tweets*.

1.3 Document Organization

Besides this introductory Chapter, which outlines the reasons for conducting this study, this Thesis is organized as follows:

Chapter 2 – Opinion Mining. This chapter introduces the concepts related to opinion mining and opinion clustering. It also presents a literature review and compares related works.

Chapter 3 – Swarm Optimization Methods. This chapter presents the main concepts related to swarm intelligence and details the three PSO methods proposed in this Thesis.

Chapter 4 – Opinion Mining Corpora. This chapter describes the characteristic of the Opinion of Brazilian Consumers (OBCC) corpus, which was built for this Thesis. It also details the other two corpora selected to evaluate the proposed methods: the IMDb movie review corpus and the Stanford Sentiment140 *tweet* corpus.

Chapter 5 – Adopted Method. In this chapter, the process adopted, the chosen pre-processing techniques, as well as the fitness function selected for the opinion clustering are detailed.

Chapter 6 – Results. This chapter presents the obtained results of the proposed methods. Results are compared with well known clustering and swarm-based algorithms and expressed in terms of internal and external validity measures.

Chapter 7 – Conclusion. This chapter discusses the work conducted for the Thesis as well the direction of future work. It also presents this thesis limitations and lists the published and submitted papers.

Appendix A – Systematic Mapping Study Process. This appendix presents a review which the goal of mapping current primary studies that relate to the application of Text Mining for the Portuguese language. It presents the research protocol, the mapping process, as well as, the general results.

2

Opinion Mining

This Chapter introduces the concepts related to the OM task. The basic concepts are detailed on Section 2.1. A general text mining process is presented on Section 2.2. Sec Section 2.3 surveys the clustering of opinions related works.

2.1 Basic Concepts

According to an extensive survey performed by [PANG; LEE \(2008\)](#), the term OM appeared for the first time in a paper by [KUSHAL; LAWRENCE; PENNOCK \(2003\)](#). OM is considered a text classification problem as it discriminates the orientation of a sentiment or opinion of given text in two or more classes. OM has been performed in various classes, such as binary, ternary, n -ary in the form of *stars*, and others ([RAVI; RAVI, 2015](#)). Some researchers have also considered various affect types, such as the six universal emotions: anger, disgust, fear, happiness, sadness, and surprise ([PANG; LEE, 2008](#)). The type of classification used in this Thesis is a *single-label* classification, wherein each opinion belongs to exactly one class or category ([FELDMAN, 2013](#)). Thus, opinions are classified into categories known as *positive*, *negative*, or *neutral*. This kind of classification is also referred to as sentiment polarity or polarity classification ([PANG; LEE, 2008](#)).

OM is observed to be carried out in three different levels: document, sentence, and aspect ([FELDMAN, 2013](#)). Document level classifies the whole document's opinion in positive, negative or neutral. Sentence level determines whether the sentence expresses any negative, positive or neutral opinion. Aspect or feature level focuses on all expressions of sentiment present within a given document and the aspect to which it refers. In addition to these three levels of analysis, other levels have been proposed, as shown in a recent survey ([RAVI; RAVI, 2015](#)) and are named as word, concept, phrase, link, clause, and sense level. Although *tweets* are short texts, they may contain more than one sentence. Thus, in this Thesis, a document level sentiment analysis has been the focus of analysis, since *tweets* did not have their sentences separated during the text pre-processing step presented in Chapter 5.

Literature reviews suggest that most OM applications might be classified into four distinct

categories: product reviews, movie reviews, political orientation extraction, and stock market predictions (RAVI; RAVI, 2015). Facebook® and Twitter® are important sources of opinions, however the former is less used in text mining as it often contains *non-text* data (e.g. graphics), and the analysis of the text by itself is not effective (EVANGELISTA; PADILHA, 2013; PAK; PAROUBEK, 2010). Other very common sources of opinions are the web pages of goods and services suppliers, such as booking.com, amazon.com, and others.

However, almost all online texts disregard spelling and grammar rules. This kind of texts have been classified as *noisy* as they still pose considerable problems both at the lexical and the syntactic levels (MOSTAFA, 2013). Jargon, contractions of existing words, abbreviations, the use of emoticons, and the creation of new words are the norm for online texts. So, to analyze such *noisy* texts, OM has investigated the use of several techniques, including word sense disambiguation, accurate detection of negation, and inferring semantic orientation (MOSTAFA, 2013).

The OM can be performed using ML and lexicon-based approaches, as shown in Figure 2.1. There are also hybrid approaches which make use of both ML and lexicon-based approaches (MEDHAT; HASSAN; KORASHY, 2014; PANG; LEE, 2008; RAVI; RAVI, 2015). ML can be further divided into supervised and unsupervised approaches. Supervised ML yields maximum accuracy, while semantic orientation provides better generality (RAVI; RAVI, 2015).

Figure 2.1: Opinion mining approaches

The supervised ML methods apply classification algorithms to learn underlying patterns from example data to later attempt to classify new unlabeled data (BALAZS; VELÁSQUEZ, 2016). For supervised approaches, two sets of annotated data are needed: one for training and other for testing. These methods have yielded high accuracy, however they need a considerable amount of labeled data, commonly built manually, which are dependent on idiom and domain. A number of supervised ML algorithms have been used to classify opinions. NB, ME, and SVM have achieved great success for OM (RAVI; RAVI, 2015).

The lexicon-based approach, also known as semantic-based or symbolic-based, makes use of positive opinion words, used to express some desired states, and negative opinion words, used to express some undesired states. There are also opinion phrases and idioms, which together are called *opinion lexicon* (MEDHAT; HASSAN; KORASHY, 2014). Three main approaches are used to build opinion lexicon:

- Manual approach, which is very time consuming;
- Dictionary-based, in which an initial set, manually built, is enlarged by searching for their synonyms and antonyms in corpora such as WordNet and thesaurus; and
- Corpus-based, which starts with a seed list of opinion words to find other opinion words in a large corpus with context specific orientations.

However, the accuracy of OM which uses supervised ML or lexicon can be influenced by the domain of the items to which it is applied. According to PANG; LEE (2008), one reason is that the same phrase can indicate a different sentiment in different domains. The authors also state that differences in vocabularies across several domains also add difficulty when applying classifiers trained on labeled data in one domain to test data in another. The same happens when building an opinion lexicon, as it needs to find opinion words with context specific orientations (MEDHAT; HASSAN; KORASHY, 2014). Although supervised ML approaches enjoy a high accuracy, its process requires human participation. Symbolic techniques, on the other hand, present limited accuracy (LI; LIU, 2014).

2.2 Opinion Mining Process

Text mining or Knowledge Discovery from Text (KDT) is the process of extracting knowledge from a large amount of unstructured data (DELEN; CROSSLAND, 2008). It was first mentioned in 1995 by Feldman and Dagan as *a machine supported analysis of text* (FELDMAN; DAGAN, 1995). As one of the text mining tasks, OM shares with it some concepts, process, and other definitions.

Although several text mining processes and methods exist, for mining large document collections it is necessary to pre-process the text documents and store the information in a data structure, which is more appropriate for further processing (HOTH0 et al., 2005). SHI; KONG (2009) make a synthesis of several studies and sum up the process of text mining presented in Figure 2.2. The authors divide text mining into three main steps: pre-treatment of the text, extraction of knowledge model, and evaluation of knowledge model. The collection of all original text data is known as corpus and the set of different words obtained from a corpus is called dictionary. This process was also adopted in this Thesis for the clustering of opinions.

Figure 2.2: Text mining process. The shaded boxes represent the three main steps (SHI; KONG, 2009)

2.2.1 Text Pretreatment

Text pre-processing or pre-treatment includes activities such as *tokenization*, in which a text document is split into a stream of words or tokens by removing all punctuation marks. In order to reduce the size of the dictionary, methods like filtering, *lemmatization*, and *stemming* can be applied during this step. The *lemmatization* methods try to map verb forms to the infinitive form of the tense and nouns to the singular form. *Stemming* methods try to build the basic forms of words, i.e. strip the plural 's' from nouns, the 'ing' from verbs, or other affixes. Another common filtering method is the *stop-word removal*. Therefore, local dictionaries may also be considered, including only words from the class that is being analyzed, reducing the size of the dictionary (WEISS et al., 2004).

In this step, the unstructured text is also converted to a structured form. Despite the simplicity of the data structure, without any explicit semantic information, the Vector Space Model (VSM) enables very efficient analysis of huge document collections (HOTHOTH et al., 2005). The VSM represents documents as vectors in *n-dimensional* wherein each element of the vector represents a word or a group of words of the document collection. Thus, the size of the vector is defined by the number of words, or groups of words, of the complete document collection (HOTHOTH et al., 2005). The simplest way of document encoding is to use binary term vectors, wherein a vector element is set to *one* if the corresponding word is used in the document and to *zero* if the word is not.

To improve the performance, term weighting schemes are used, where the weights reflect the importance of a word in a specific document of the corpus. One of the best known schemes is the *tf-idf* weighting that combine the definitions of term frequency and inverse document frequency, to produce a composite weight for each term in each document (WEISS et al. (2004).

Often, text mining methods may be applied without further pre-processing. However, researchers in the field of NLP, also began to apply additional linguistic pre-processing to improve text mining results (HOTHOTH et al., 2005; ENRÍQUEZ et al., 2013). More recently, ontologies, thesaurus, and dictionaries have also been applied. The NLP tasks frequently applied

- Topic Tracking: studies the problem of detecting new and upcoming topics in time ordered document collections;
- Concept Linking: connects related documents by identifying their commonly shared concepts and helps users to find information that are not easily obtained using traditional searching methods;
- Question Answering: find the best answer to a given question by knowledge-driven pattern matching;
- Information Visualization: organizes large textual sources in a visual hierarchy or map and provides browsing capabilities, in addition to simple searching.

Supervised and unsupervised learning are the most common approaches used in text mining. In the supervised learning, the *right answers* are given. In text domains, it is usually referred to as text learning, text classification, or text categorization. The unsupervised learning tries to group text by similarity and is also referred as text clustering. Semi-supervised and hybrid approaches are also found in literature. In order to identify the polarity of a given opinion, in this Thesis, the text clustering task is performed through the application of unsupervised algorithms.

2.2.3 Evaluation of Knowledge Model

Measuring the quality of a knowledge model is an important step in text mining application. The evaluation of knowledge model is usually assessed with respect to structural properties expressed in different external and internal measures. The main difference is whether or not external information is used for validation. External measures evaluate the performance by matching results to a prior knowledge ([KAROL; MANGAT, 2013](#)). They usually incorporate the categorization criteria of the humans ([CAGNINA; ERRECALDE; INGARAMO, 2008](#)) and are mostly used in supervised approaches. Common external measures are: accuracy, precision, recall, and f-score ([HOTHOTH et al., 2005](#)).

Unlike external validation measures, which use external information not present in the data, internal validation measures rely only on information in the data. They are commonly used in unsupervised approaches to evaluate the quality of a clustering structure by comparing different sets of clusters as a function of the given similarities. Well known ICVM are: the Purity comparative measure, Mean Square Error (MSE), and Global Silhouette (GS) ([HOTHOTH et al., 2005](#)).

2.3 Clustering of Opinions

Text clustering is an approach for automatically finding classes, concepts, or groups of patterns from unstructured data. It seeks to partition an unstructured set of objects into clusters

or groups. Thus, the objects have to be similar to objects in the same cluster and dissimilar to objects from other clusters. Clustering has been applied in many areas, including biology, medicine, anthropology, marketing, and economics (SARKAR; ROY; PURKAYASTHA, 2014a).

Much of the useful unstructured data to be processed is taken from web repositories whose documents are, frequently, short texts with a few tens words, such as scientific abstracts, news, opinions, and legal documents (CAGNINA et al., 2014). Due to the strong tendency of people using reduced language in text messages and, recently, with the popularity of *weblogs* and social networks, the short-text clustering task is becoming even more important because of the need to analyze comments and opinions from users in an unsupervised manner.

Studies indicate that the partitioning technique is well suited for clustering large datasets on account of its computational requirements being relatively low as shown in SARKAR; ROY; PURKAYASTHA (2014a). The best known partitioning clustering algorithm is the K-means and its variants. It seeks to find a partition that minimizes MSE. Although K-means is an extensively used clustering algorithm, it suffers from many shortcomings, such as: the possibility of getting stuck at *local minimum*, as well as at *local maxima*, and being affected by *noise* and *outliers* (SARKAR; ROY; PURKAYASTHA, 2014a). So, this has motivated a recent interest on clustering algorithms based on global optimization, such as genetic algorithms and swarm optimization techniques (FILHO et al., 2015).

Clustering-based opinion mining approaches explicitly consider clustering as an optimization problem, where a given arbitrary objective function must be optimized. These approaches apply unsupervised learning algorithms which neither requires any human labeled training data, nor time for training (LI; LIU, 2014). However, it has some difficulties such as the one to catch subtle semantics that human beings use in speech and writing. This problem gets worse when short-texts are analyzed. Without any contextual information and only a small number of words available in the document, achieving semantic comparisons at an acceptable level with respect to analogy-making in human beings is an even more challenging issue (CAGNINA et al., 2014).

The quality of the resulting clusters is commonly evaluated with respect to structural properties expressed in different ICVM, as shown in Section 2.2.3. These internal measures are very common in document and short-text clustering. However, as stated by CAGNINA; ERRECALDE; INGARAMO (2008); KAROL; MANGAT (2013), the real effectiveness of the clustering algorithms can only be evaluated with external measures that incorporate the categorization criteria of the users, such as the accuracy, precision, recall, and f-score measures.

After conducting an extensive literature review, there were found only the works of LI; LIU (2010, 2012, 2014) that deal with OM as a text clustering task. The work of COLETTA et al. (2014) was included as related work as it presents a hybrid approach which combines classifier and cluster ensembles for OM. Table 2.1 shows a summary of opinion clustering related work carried out at document level.

LI; LIU (2010, 2012) introduce the clustering-based sentiment analysis approach at document level. This approach is based on K-means clustering algorithm. Term Frequency-

Inverse Document Frequency (TF-IDF) weighting method, stemming, POS tagger, voting mechanism, and term scores extracted from WordNet were combined. 300 positive and 300 negative opinions were randomly chosen from Internet Movie Database (IMDb) (PANG; LEE; VAITHYANATHAN, 2002) to conduct an experiment with a balanced dataset of 600 reviews. Words which were not tagged as being either an adjective or adverb were eliminated from this data set. For each document vector, there were 20 clustering results which were positive or negative. Each of these results is regarded as a vote. One document will be considered as positive if it obtains more than 10 positive votes, otherwise it will be regarded as negative. The best result achieved accuracy ranging between 77.17% and 78.33% for two balanced classes.

LI; LIU (2014) enhanced their previous work (LI; LIU, 2012) by processing opposite and non-opinion contents (PANG; LEE, 2005), modifying their voting mechanism, and adopting a cosine distance measure approach. They also evaluate their method on three balanced and unbalanced classes. The best result for the binary classes (positive and negative) was obtained by processing non-opinion contents, achieving accuracy between 89.67% and 88.00%. For the three balanced classes, the best result was obtained using a cosine distance measure approach, achieving accuracy between 60.17% and 64.13%. For the three unbalanced classes, the best result was obtained using a voting mechanism approach, achieving accuracy between 49% and 53.76%.

COLETTA et al. (2014) presented a hybrid approach for opinion mining using Twitter' data. In this paper, the SVM classifier is combined with a cluster ensemble algorithm named Consensus between Classification and Clustering Ensembles (C3E) based on Squared Loss (SL). The C3E-SL, capable to combine classifier and cluster ensembles, can refine *tweets* classification from additional information provided by clusters, assuming that similar instances from the same clusters are more likely to share the same class label. Experimental results obtained on four *tweets* datasets showed that the combination of a SVM classifier with an ensemble cluster, as done by C3E-SL algorithm, can improve the *tweet* binary classification. In most cases, C3E-SL provided better results than the stand-alone SVM. For the HCR dataset, the obtained accuracy is the best one reported so far in the literature.

2.4 Opinion of Brazilian Consumers

Portuguese is one of the most used languages in the world, with almost 270 million people speaking some variant of the language in ten countries¹. In text mining, works dealing with Portuguese language are scarce, and most efforts are directed towards the English language (LIU; ZHANG, 2012; PANG; LEE, 2008; RAVI; RAVI, 2015; CALDERON et al., 2015).

After performing a systematic mapping study, detailed in Appendix A on several scientific

¹Brazil (202,656,788), Mozambique (24,692,144), Angola (24,300,000), Portugal (10,813,834), Guinea-Bissau (1,693,398), East Timor (1,201,542), Equatorial Guinea (722,254), Macau (587,914), Cabo Verde (538,535) and São Tomé e Príncipe (190,428). World Factbook

Table 2.1: Document level opinion clustering related work

Paper	Pre-processing	Algorithm	Domain	Corpus	Classes	Accuracy
Li and Liu (2010, 2012)	TF-IDF Stemming POS-tagger Filtering adjective and adverb Wordnet Term Score	Voting mechanism after 20 K-means runs	Movie Review	IMDb - English	Two balanced classes	77.17% - 78.33%
Li and Liu (2014)	Term Presence Stemming POS-tagger Filtering adjective and adverb Wordnet Term Score	Voting mechanism after 20 K-means runs Non-opinion contents processing	Movie Review	IMDb - English	Two balanced classes	89.67% - 88.00 %
		K-means Cosine distance	Movie Review	PANG; LEE (2005)	Three balanced classes	60.17 % - 64.13 %
		Voting mechanism after 20 K-means runs	Movie Review	PANG; LEE (2005)	Three unbalanced classes	49.00 % - 53.76 %
Coletta et al. (2014)	Removal of retweets, stop words, links, URLs, mentions, punctuation, and accentuation Stemming Opinion Lexicon	SVM classifier combined with a cluster ensemble using the C3E-SL algorithm	Health	Health Care Reform (HCR) Tweets in English	Two unbalanced classes	Average of 79.62%
			Politics	Obama-McCain Debate (OMD) Tweets in English	Two unbalanced classes	Average of 75.18%
			Information and Technology (4 companies)	Sanders Tweets in English	Two unbalanced classes	Average of 82.15%
			Several	Sentiment140 Tweets in English	Two balanced classes Trained with 500 tweets	Average of 77.69%
					Two balanced classes Trained with 1000 tweets	Average of 79.69%
					Two balanced classes Trained with 2000 tweets	Average of 77.80%
					Two balanced classes Trained with 3000 tweets	Average of 81.84%

databases, 6,075 primary studies were retrieved from automated and manual searches up to year 2014. Twenty-five papers relating the application of opinion mining for Portuguese (Brazilian and European variants) were identified, from which, eight papers have analyzed the consumers' sentiment or opinion and are detailed below. A number of gaps in the OM research area could be observed after this review and are presented below:

1. The absence of benchmark datasets. Even when good results are achieved, the used data-sets are rarely published. This makes it difficult to implement improvements, as well as comparisons;
2. There are minimal tools to support the Portuguese processing, forcing researchers, occasionally, to make use of resources available for other languages. Additionally, there was no evidence of a unique tool being employed during the whole OM process;
3. There is a reduced number of studies compared to other languages ([RAVI; RAVI, 2015](#)). The automated and manual search procedures retrieved 6075 papers published up to the year 2014, from which only 25 studies performed OM for the Portuguese language;
4. Lexicon-based, rule-based, and supervised learning are the only approaches applied to OM for the Portuguese language. No evidence was found about the use of clustering techniques, and
5. Finally, less than 50% of all 25 primary studies have fully answered the specific research questions. Important data for comparison like text domain and type, classes details, and language variants were not available.

The same research protocol from [SOUZA et al. \(2016\)](#) was applied to search and select primary studies from the year 2015 and three studies were found. The available studies adopted lexicon-based, supervised learning, rule-based, or hybrid approaches. No studies that make use of clustering approaches were found. A comparative analysis is presented in Table 2.2 and Table 2.3.

The paper presented by [CHAVES et al. \(2012\)](#) depicts the algorithm Polarity Recognizer in Portuguese (PIRPO) to classify sentiment in online reviews. PIRPO was constructed to identify polarity in Portuguese accommodation reviews. Each review is analyzed according to concepts from a domain ontology. The reviews were decomposed in sentences in order to assign a polarity to each concept of the ontology in the sentence. 180 comments in Portuguese were evaluated and preliminary results indicate an average F-score of 0.32 for polarity recognition.

[FREITAS; VIEIRA \(2013\)](#) propose and evaluate methods to identify polarity in Portuguese user generated reviews according to features described in domain ontologies. Experiments were performed using movie (Movie Ontology) and hotel (Hontology) ontologies for feature identification. For the polarity identification, a list of adjectives, verbs, nouns and adverbs,

and their polarities from Opinion Lexicon and SentiWord-Net were used. The 180 accommodation comments and the 150 movie comments were evaluated by a human. The algorithm was evaluated against the human evaluation.

The works of ROSA; RODRIGUEZ; BRESSAN (2013a,b) analyze the Brazilian Consumers' Sentiments on Twitter for the hair cosmetics domain using SentiMeter-Br. For the Portuguese dictionary performance validation, the results were compared with the SentiStrength algorithm and evaluated by three Specialists in the field of study. The polarity of the 500 texts extracted from Twitter was also tested through machine learning. Results achieved F-Measure of 0.87 for positive and 0.91 for negative phrases by Sequential Minimal Optimization (SMO) algorithm.

AVANCO; NUNES (2014) present results on lexicon-based classification of sentiment polarity in web reviews of products written in Brazilian Portuguese. They represent a first step towards a robust opinion miner from reviews of technology products. The results show that the better combination is the version of the algorithm that deals also with negation and intensification and uses the sentiment lexicon Sentilex. The average F-measure achieved 0.73.

The work of EVANGELISTA; PADILHA (2013) proposes a tool to classify web publications made on social networks such as Facebook and Twitter as positive, negative or neutral. For the classification task, the tool uses SentiWordNet lexical resource and the Naive Bayes algorithm. For the experiments, they collected publications from three Brazilian e-commerce companies. The results were very relevant for publications from Twitter using Naive Bayes algorithm for the e-commerce domain, with a correct classification rate of between 71% and 95%.

The work of SANTOS; LADEIRA (2014) describes an opinion mining application over a data-set extracted from the web and composed of reviews with several Internet slang, abbreviations, and typo errors. In this research, 759.176 Portuguese reviews were extracted from the Google Play application store and frameworks such as *Hadoop* and *Mahout* were adopted. Based on the conducted tests, the authors concluded that pre-processing has an insignificant role in opinion mining task for the specific domain of reviews of mobile apps. The work also contributed to the creation of a corpus consisting of 759 thousand reviews and a dictionary of slang and abbreviations commonly used in the Internet. The best performance for the classification was encountered using the SVM with F-measure of 0.832.

GRACIELA; BECKER; MOREIRA (2014) present a case study about the classification of emotions present in product reviews, evaluating an approach that combines lexicon-based emotion classification and automatic translation for multilingual opinion mining. The results on real data show that translating the reviews yields better results and that *lemmatization* does not bring significant changes. A corpus consisting of 150 reviews was collected from *amazon.com.br* and results are presented for each emotion type. The emotion that showed the best results was the 'happiness' with F-measure higher than 0.8.

ANCHIETA; ASSIS; NETO (2015) presented the feasibility study of *stylometric* features

in the sentiment classification. The authors created a corpus of product reviews from the most traditional on-line service in Portuguese, namely, Buscape, where users post their ‘pros’ and ‘cons’ comments about several products, services and companies. In order to analyze the *stylometric* features, three classifiers (SVM, Naive Bayes and J48) were used to compare and analyze with the baseline method of text feature extraction in the sentiment classification. The study compared the *stylometric* features with TF-IDF and Delta TF-IDF methods and measured the performance using the accuracy measure. Better results were achieved with the SVM classifier using TF-IDF (56,25%), Delta TF-IDF (72,62%), and stylometry (82,75%) of accuracy.

FREITAS; VIEIRA (2015) performed sentiment analysis in a data-set of the accommodation sector. The authors compared different POS-taggers and sentiment lexicons. They evaluate the impact of some linguistic rules regarding negation and the position of adjectives. A set of eleven different configurations were proposed. Best results (average f-score of 0.632) were achieved by configuration number 11, which makes use of TreeTagger, a synset with polarities from Onto.PT, and adjectives position rules.

MARQUES-LUCENA; SARRAIPA (2015) present a framework for customers’ sentiment analysis, which has the ability to automatically acquire knowledge from web software. Thus, it addresses the target of customer’s sentiments with the purpose of supporting companies’ managers in their decision-making to improve their products and services. The framework focuses on the Portuguese hospitality industry and evaluates comments from Booking’s web software. The authors did not present results.

Table 2.2: Domain, granularity level and pre-processing tasks of related OM applications for the Portuguese language

Related Work	Domain	Granularity level	Pre-processing	Supervised ML
(Chaves et al. 2012)	Accommodation	Sentence Level	Sentence Splitter	—
(Freitas and Vieira 2013)	Accommodation	Feature	Tokenization, Sentence Splitter, Lemmatization, and, Part-Of-Speech	—
(Rosa, Rodriguez, and Bressan 2013a, 2013b)	Hair cosmetics	Sentence	Stopword Removal and word n-gram	SMO, C4.5, NB, and MNB
(Avanco and Nunes 2014)	Mobiles and Smart-phones	Document	—	—
(Evangelista and Padilha 2013)	E-commerce	Document	Stopword, Stemming, and Normalization	MNB
(dos Santos and Ladeira 2014)	Games and Mobile Apps	Document	Stopword, Stemming, Normalization, TF-IDF, word n-gram, and InfoGainAttributeE-valuator for feature selection	Neural Network, Bayesian network, NB, and SVM
(Graciela, Becker, and Moreira 2012)	E-books Kindle	Document	Document translation and Lemmatization	—
(Souza et al. 2016)	Telecommunication, telephony, and banking	Feature	Tokenization, Gazetteer Lookup, Sentence Splitter Part-Of-Speech, and Named Entity	SVM
(Anchieta et al. 2015)	Smartphone re-views	Sentence	Tokenization, Stopword removal, Stemming, TF-IDF, Delta TF-IDF, Stylometry, Part-Of-Speech	SMO, C4.5, NB,
(Freitas and Vieira 2015)	Accommodation	Feature	Tokenization, Lemmatization, Part-Of-Speech	—
(Marques-lucena et al. 2015)	Accommodation	Feature	Tokenization, Sentence splitter, Part-Of-Speech, Tree Parsing,	—

Table 2.3: Other resources, tools and results of OM applications for the Portuguese language

Related Work	Other resources	Tools	Data Source	Results
(Chaves et al. 2012)	List of polar adjectives from (M. Souza et al. 2011), Hontology, and Lexico-syntactic patterns	Phyton	Tripadvisor and Booking	F-score: 0.32
(Freitas and Vieira 2013)	List of polar adjectives from OpLexicon and SentiWord-Net, Movie Ontology, Hontology, and linguistic rules in windows of words	TreeTagger	Tripadvisor, Booking, and Omelete	Accommodation: Recall for positive: 0.58, Precision for positive: 0.10, Movie comments: F-measure for positive: 0.62
(Rosa, Rodriguez, and Bressan 2013a, 2013b)	Sentimeter-Br and Portuguese-Br Dictionary from Google Trends	Python and Weka	Twitter and Facebook book	F-Measure for positive: 0.87 and F-Measure for negative: 0.91 by SMO
(Avanco and Nunes 2014)	OpinionLexicon, SentiLex, and a subset of LIWC	—	Buscapé	F-measure: 0.73
(Evangalista and Padilha 2013)	OpenWordNet-PT	Java and Snowball Stemmer (PT)	Facebook, Twitter, e-bit, and Buscapé	Success rate: 95%
(dos Santos and Ladeira 2014)	Aspell Dictionary for Normalization and Map-reduce paradigm	PTStemmer for Stemming, Hadoop, Mahout, and Weka	Google Play	F-measure: 0.832
(Graciela, Becker, and Moreira 2012)	NRC emotion lexicon	Python and Google Traductor	Amazon	F-measure >0.8
(Souza et al. 2016)	Rule-based and Lexicon-based	GATE	Twitter	Precision: 0.66
(Anchieta et al. 2015)	TEP 2.0	TreeTagger, Weka	Buscapé	Accuracy with SVM: 56.25% (TF-IDF), 72.62% (Delta TF-IDF), and 82.75% (Stylometry)
(Freitas and Vieira 2015)	Linguistic rules, LIWC-PT, OpLexicon, SentiLex-PT, Onto.PT	TreeTagger, FreeLing, and CitiusTagger	TripAdvisor	Average F-score: 0.632
(Marques-lucena et al. 2015)	SentiLex-PT, Hontology, OpenThe-saurusPT	Spell Checker, LX-Tokenizer, LX-Parser	Booking	—

3

SWARM OPTIMIZATION METHODS

This Chapter details the three swarm-based methods proposed in this Thesis. Section 3.1 and Section 3.2 highlight the main concepts related to swarm intelligence and particle swarm optimization, respectively. Section 3.3 surveys the studies which have applied PSO to the text clustering task. Lastly, Section 3.4 presents the proposed swarm-based methods.

3.1 Swarm Intelligence

Nature-inspired meta-heuristic algorithms solve optimization problems by simulating biological or physical phenomena. [MIRJALILI; LEWIS \(2016\)](#) group these algorithms in three main categories: evolution-based, physics-based, and swarm-based methods. The evolution-based algorithms are inspired by the laws of natural evolution. The search process starts with a randomly generated population which is evolved over subsequent generations ([MIRJALILI; LEWIS, 2016](#)). One of the most well-known evolution-based algorithm is the Genetic Algorithms (GA). According to the same authors, the physics-based methods simulate the physical rules in the universe. The most popular physics-based algorithms is the Simulated Annealing (SA), whose inspiration comes from *annealing* in metallurgy, a technique involving heating and controlled cooling of a material. Finally, the swarm-based algorithms simulate social behavior of groups of animals. The PSO, proposed by [EBERHART; KENNEDY \(1995\)](#), is one of the most popular swarm-based algorithm and it incorporates swarming behaviors observed in flocks of birds.

The swarm intelligence is a biologically-inspired artificial intelligence paradigm for solving optimization problems which originally took its inspiration from examples of collective behavior of large numbers of individuals that collaborate with one another and their environment to achieve a common goal, and is seen in natural systems such as ant colonies, flocking birds, and fish schooling ([ABRAHAM; GUO; LIU, 2006](#)). The swarm based algorithms became attractive since PSO was proven to be very competitive ([MIRJALILI; LEWIS, 2016](#)). The main advantages of these algorithms is that they preserve search space information over subsequent iterations, usually include less operators compared to evolutionary approaches (selection, crossover, muta-

tion, elitism, etc.), and are easier to implement (ABRAHAM; GUO; LIU, 2006; MIRJALILI; LEWIS, 2016; EBERHART; KENNEDY, 1995). Table 3.1 presents a list of the swarm-based optimization algorithms from MIRJALILI; LEWIS (2016).

Table 3.1: Swarm-based optimization algorithms

Algorithm	Inspiration	Year of proposal
Particle Swarm Optimization (PSO)	Bird flock	1995
Marriage in Honey Bees Optimization Algorithm (MBO)	Honey Bees	2001
Artificial Fish-Swarm Algorithm (AFSA)	Fish swarm	2003
Termite Algorithm	Termite colony	2005
Ant Colony Optimization (ACO)	Ant colony	2006
Artificial Bee Colony (ABC)	Honey Bee	2006
Wasp Swarm Algorithm	Parasitic wasp	2007
Monkey Search	Monkey	2007
Wolf pack search algorithm	Wolf herd	2007
Bee Collecting Pollen Algorithm (BCPA)	Bees	2008
Cuckoo Search (CS)	Cuckoo	2009
Dolphin Partner Optimization (DPO)	Dolphin	2009
Bat-inspired Algorithm (BA)	Bat herd	2010
Firefly Algorithm (FA)	Firefly	2010
Hunting Search (HS)	Group of animals	2010
Bird Mating Optimizer (BMO)	Bird mating	2012
Krill Herd (KH)	Krill herd	2012
Fruit fly Optimization Algorithm (FOA)	Fruit fly	2012
Dolphin Echolocation (DE)	Dolphin	2013
Whale Optimization Algorithm (WOA)	Humpback whales	2016

3.2 Particle Swarm Optimization

Essentially, the PSO algorithm consists of an stochastic global optimization method to find the optimal or global optimum solution in the landscape of objective function. PSO is inspired by the social behavior of a bird flock. Compared to other evolutionary methods, PSO has the advantage of its simple implementation, the good trade-off between exploration and exploitation ability, and fast convergence (ZHANG; XIONG; ZHANG, 2013).

Considering a flock of birds searching for food in an area, there is only one piece of food in that area and all the birds are searching for it. In each iteration, the birds are only aware of how far the food is, so the best approach to get the food is to follow the bird which is nearest to it. In the PSO algorithm, the birds in a flock are symbolically represented as *particles*. These particles can be considered as simple agents *flying* through a problem space.

For applying PSO successfully, one of the key issues is finding how to map the problem solution into the PSO particle, which directly affects its feasibility and performance (ABRAHAM; GUO; LIU, 2006). The basic PSO algorithm has the following definitions:

1. The birds are the solutions, called as *particles*;
2. The collection of particles is called as *swarm*;
3. Each particle has a fitness value. The aim is to optimize this fitness value as evaluated by the *fitness function*;

4. Each particle has its own *velocity* and *position* calculated by the velocity and position functions, respectively;
5. The PSO is initialized with group of particles whose parameters are altered during each iteration;
6. In each iteration, every particle updates its fitness value and its *pbest*, which is the particle personal best value and
7. Meanwhile, during each iteration, the PSO reviews the *gbest*, which is the best value obtained by any of the particles to that point.

A particle location in the multi-dimensional problem space represents one solution for the problem. When a particle moves to a new location, a different problem solution is generated. This solution is evaluated by a fitness function, which provides a quantitative value of the solution's utility (KENNEDY; EBERHART, 1995). The velocity and direction of each particle moving along each dimension of the problem space will be altered with each generation of movement. In combination, the particle's personal experience and its neighbors' experience influence the movement of each particle through a problem space.

If $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$ is the current position of particle i , $V_i = (v_{i1}, v_{i2}, \dots, v_{in})$ is the current speed, $pbest_i = (p_{i1}, p_{i2}, \dots, p_{in})$ is the best position that particle i has visited, and $gbest = (p_{g1}, p_{g2}, \dots, p_{gn})$ is the best positions that all particles have visited, then the speed and position of each particle i ($1 \leq i \leq N$) for each generation are computed as equation (3.1) and equation (3.2), respectively:

$$v_i(t+1) = \omega \cdot v_i(t) + c_1 \cdot rand_1() \cdot (pbest_i - x_i(t)) + c_2 \cdot rand_2() \cdot (gbest - x_i(t)) \quad (3.1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (3.2)$$

Where ω is the inertia weight, c_1 and c_2 are positive acceleration factors, and $rand_1()$ and $rand_2()$ are random functions changing in the range of $[0, 1]$. In equation (3.1), the first part is the previous speed of particles, the second part is cognition, which stands for particle's own thinking, and the third part is social component, which represents the information sharing and cooperation of particles' neighborhood.

A neighborhood is a topology which connects each particle to some other particles. Neighborhoods have an inhibiting effect on information flow, since particles can, at each iteration of the algorithm, only receive information about local best positions that is already available to their direct neighbors (BOSMAN; ENGELBRECHT, 2014). Different topologies result in different variants of the PSO. Figure 3.1 presents two PSO basic topologies: the *gbest* PSO that uses a star topology, in which each particle is connected to every other particle and the *lbest*

PSO that uses a ring topology, in which each particle is connect to two other particles. The star topology is adopted in this Thesis. In this topology, the information flow is instant and it is not inhibited.

Figure 3.1: Basic PSO topologies

Besides the topology variants, many different variants of the PSO algorithm are found in literature. Some of these variants have been proposed to incorporate the capabilities of other evolutionary algorithms. A discrete binary version of PSO was first proposed by [KENNEDY; EBERHART \(1997\)](#). Instead of operating in a continuous space, in the binary version, trajectories are changes in the probability that a coordinate will take on a *zero* or *one* value. Many optimization problems are set in a space featuring discrete, qualitative distinctions between variables and between levels of variables. In this Thesis, three discrete versions of PSO are proposed, as well, as hybrid version with CS swarm-based algorithm.

3.3 Text Clustering with Particle Swarm Optimization

A significant amount of studies which successfully applied PSO for text or document clustering was found in literature. It is important to note that none of these studies applied PSO-based methods to opinion clustering. They are detailed bellow and a summary is present in Table 3.2.

In 2005, the two first approaches for document Clustering using PSO were proposed by [CUI; POTOK; PALATHINGAL \(2005\)](#); [CUI; POTOK \(2005\)](#). In the conducted experiments, the authors applied the PSO, K-means and hybrid PSO clustering algorithm on four different text document data-sets from Text REtrieval Conference (TREC) collections. The results illustrate that the hybrid PSO algorithm can generate more compact clustering results than the K-means algorithm.

The CLUstering with a DIcrete PSO (CLUDIPSO) was proposed and evaluated in [CAGNINA; ERRECALDE; INGARAMO \(2008\)](#); [INGARAMO et al. \(2009, 2011\)](#). CLUDIPSO was developed for clustering short-text collections based on a discrete PSO. In [CAGNINA et al. \(2014\)](#), an improved version of CLUDIPSO is proposed, the CLUDIPSO*, which includes a

different representation of particles, a more efficient evaluation of the function to be optimized and some modifications in the mutation operator. The experimental results show that PSO-based approaches can be highly competitive alternatives for clustering short-text corpora.

[J.HYMA; JHANSI; ANURUDHA \(2010\)](#) propose a hybridized approach of PSO and GA for document clustering. Two methods for integrating PSO and GA are presented. In the *Transitional* technique, the algorithm runs PSO for some time and then makes a transition to GA, then it runs in GA for some time and transits back to PSO. In the *Parallel* Technique, the population is divided into two parts and it is evolved with the two techniques respectively. The algorithm executes the two techniques simultaneously and selects a user specified number of best individuals from each system, exchanging after a user defined number of iterations. The individual with larger fitness value is more often selected.

Other approaches with discrete PSO and GA for document clustering are shown in [PREMALATHA; NATARAJAN \(2009, 2010\)](#). The algorithm adds reproduction by using crossover and mutation operators when the stagnation in movement of the particle is identified. Experiments results are examined with three document corpora and it was observed that the document clustering problem is effectively tackled with PSO methodology.

[LEE; ON \(2010\)](#) propose a role-based PSO (RoleBasedPSO) algorithm which assigns different roles to different groups of particles. One group of particles is responsible for searching the global space with a relatively large inertia weight, the second group of particles stabilizes the algorithm to converge with a relatively smaller inertia weight. The third group of particles change the inertia weight value linearly assuming that particles are reaching to global minimum with the number of iterations. Experimental results indicate that RoleBasedPSO shows 5% better performance in terms of accuracy and requires less number of iterations to converge to global minimum.

[THAKARE; CHAUDHARI \(2012\)](#) present a detailed overview of PSO, its variants, and a hybridization of PSO with Artificial Bee Colony (ABC). The authors propose an approach similar to the one presented by [J.HYMA; JHANSI; ANURUDHA \(2010\)](#), however making use of the ABC. A hybridized approach has been applied to address document clustering by [KHANNA; YADAV \(2013\)](#). The proposed clustering approaches, called KPSO, selected for implementation and comparison, are hybrids of traditional partitioning K-Means algorithm with PSO technique.

[KAROL; MANGAT \(2013\)](#) propose two techniques for document clustering involving the application of soft computing approach as an intelligent hybrid approach PSO algorithm. The proposed approach involves partitioning Fuzzy C-Means (FCPSO) and K-Means (KPSO) algorithms. The performance has been evaluated against traditional partitioning techniques K-Means and Fuzzy C Means on real text datasets. FCPSO provides even better results than KPSO as it deals well with the overlapping nature of documents.

[SARKAR; ROY; PURKAYASTHA \(2014a,b\)](#) present a comparative analysis of PSO and K-means algorithm for text clustering using Nepali Wordnet. In this study, texts are represented

in terms of *synsets* corresponding to a word. Bag of terms data representation of text is thus enriched with synonyms from WordNet. K-means, PSO and hybrid PSO+K-means algorithms are applied for clustering of text in Nepali language. Experimental evaluation demonstrates that hybrid PSO+K-means performs better.

An approach based on K-means, PSO and Sampling algorithms for document clustering is proposed by [KAMEL; OUCHEN; BAALI \(2016\)](#). The algorithm, named sampling-PSO-K-means, was evaluated on four data-sets from Banking and Finance, Programming Languages, Science, and Sport domains. The results are compared to those obtained by K-means, PSO, Sampling+K-means, and PSO+K-means and show that the proposed approach generates the most compact clusters.

Table 3.2: Studies which performed text clustering using PSO

Related Work	Proposed Algorithm(s)	Pre-processing tasks	Text Type	Best Results
(Cui, Potok, and Palathingal 2005)	PSO+K-means	VSM + TF-IDF, Euclidian distance and Cosine correlation as similarity measures	Documents from TREC	ADDC: 2.25961 (PSO + k-means and Euclidian distance)
(Cui and Potok 2005)	PSO+K-means, K-means+PSO, means+PSO+K-means	VSM + TF-IDF, Euclidian distance and Cosine correlation as similarity measures	Documents from TREC	ADVDC: 2.550±0.746 (PSO+K-means and Euclidian distance)
(Cagnina, recalde, and Ingaramo 2008)	CLU DIPSO (Discrete PSO)	VSM + SMART codification, Cosine correlation as similarity measure	Short-texts from CICLing-2002 and Micro4News	F-measure avg: 0.93 (CLU DIPSO and GS measure)
(Ingaramo et al. 2009)	CLU DIPSO (Discrete PSO), CLUCOPSO (Continuous PSO)	VSM + SMART codification, Cosine correlation as similarity measure	Short-texts from CICLing-2002, EasyAbstracts, and Micro4News	F-measure avg: 0.93 (CLU DIPSO and GS measure)
(Ingaramo et al. 2011)	CLU DIPSO MULTI (Discrete PSO for English and Spanish)	N/A	Short-texts from SEPLN-CICLingEN, SEPLN-CICLingES, JRC-AcquisEN, and JRC-AcquisES	F-measure avg: 0.75 (CLU DIPSOEN)
(L. Cagnina et al. 2014)	CLU DIPSO * (Discrete PSO + kNN)	VSM + TF-IDF, Cosine correlation as similarity measure	Short-texts from Micro4News, EasyAbstracts, SEPLN-CICLing, CICLing-2002, R4, R6, R8B, JRC6, R8-Test, JRC-Full, and R8-Train	F-measure avg: 0.93 (CLU DIPSO *)

Table 3.2: Studies which performed text clustering using PSO

Related Work	Proposed Algorithm(s)	Pre-processing tasks	Text Type	Best Results	
(J.Hyma, Y. Jhansi 2010)	PSO+GA (Transitional), PSO+GA (Parallel)	VSM + TF-IDF, Euclidian distance as similarity measure	10 documents	Intra Cluster Distance: 5.2884	PSO+GA (Transitional)
(Premalatha and Natarajan 2009)	DPSO Crossover + Mutation GA	VSM + TF-IDF, Cosine correlation as similarity measure	Documents from CISI, Cranfield, and ADI	ADDC: 0.4	(DPSO Crossover + Mutation GA)
(Premalatha and Natarajan 2010)	HPSO (Hybrid PSO), HPSO-Swap (HPSO with GA using Swap Operation), HPSO-CO (HPSO with GA using Crossover Operation), PSO-CO (PSO with Crossover), PSO-M (PSO with Mutation), PSO-CM (PSO with Crossover and Mutation)	VSM + TF-IDF, Cosine correlation as similarity measure	Documents from (Silvia et al 2003)	ADDC: 0.423458	(HPSO-CO)
(Lee and On 2010)	RoleBasedPSO	VSM + TF-IDF	Documents from name data set (Bekkerman 2009), Wikipedia and ECDL person name	RoleBasedPSO shows 2.5% improvement for Bekkerman, 3% for ECDL and 5% for Wikipedia name data set	
(Thakare and Chaudhari 2012)	PSO+Bee Algorithm (BA)	N/A	N/A	N/A	

Table 3.2: Studies which performed text clustering using PSO

Related Work	Proposed Algorithm(s)	Pre-processing tasks	Text Type	Best Results
(Khanna and Yadav 2013)	KPSO, PSO+K-means	N/A	N/A	N/A
(Karol and Mangat 2013)	FCPSO (Fuzzy Means+PSO) , KPSO (K-Means+PSO)	C- VSM + Word frequency, Euclidean distance as similarity measure	3000 Documents from Reuters-21587 and 20News-group	KPSO and FCPSO were 37% better for Entropy and 17% better for F-Measure
(Sarkar, Roy, and Purka-yastha 2014a)	PSO+K-means	Bag of terms of synsets corresponding to a word (wordnet) + TF-IDF, Cosine correlation as similarity measure	Documents from Technology Development for Indian Language website	Intra cluster: 7.9640, Inter cluster: 0.5171, (PSO+K-means)
(Sarkar, Roy, and Purka-yastha 2014b)	PSO+K-means+TF, PSO+K-means+TF-IDF, PSO+K-means+UNL	VSM+UNL+TF+TF-IDF, Cosine correlation as similarity measure	UNL data from UNL site	Accuracy: 0.9763, Intra cluster: 0.7442, Inter cluster: 0.7340 (PSO+K-means+UNL)
(Kamel, Ouchen, and Baali 2016)	Sampling+K-means, PSO+K-means, Sampling+PSO+K-means	VSM + TF-IDF, Euclidian distance as similarity measure	11000 Documents from TREC	ADDC: 2.911 (Sample+PSO+K-means)

3.4 Proposed PSO-based Methods

The proposed methods explicitly consider clustering as an optimization problem, where a given arbitrary objective function must be optimized and can be formally defined as follows:

1. Given a set of opinions $O = o_1, o_2, \dots, o_n$,
2. a desired number of clusters k , and
3. an objective function f that evaluates the quality of a clustering,
4. we want to compute an assignment $\gamma : O \rightarrow 1, \dots, K$ that minimizes (or, in some cases, maximizes) the objective function, which is often defined in term of similarity or distance measures.

Each particle contains a possible cluster solution (Figure 3.2), a n -dimensional integer vector, where n is the number of opinions in the corpus. Each position in a particle corresponds to an opinion of the collection and the integer value stored in this position identifies the group (cluster) to which it belongs. The best position currently found for the swarm ($gbest$) and the best position ($pbest$) reached by each particle are recorded at each iteration. In this Thesis, two objective function are evaluated: the GS and the Average Distance of Documents to the Cluster Centroid (ADDC).

Figure 3.2: Particle representation for opinion clustering

Three discrete PSO-based algorithms are proposed in this Thesis: the first one is based on a discrete binary version of PSO, first proposed by [KENNEDY; EBERHART \(1997\)](#) with a mutation mechanism; the second one is based on an Improved Self-Adaptive PSO (IDPSOMUT) algorithm with detection function ([ZHANG; XIONG; ZHANG, 2013](#)); and the last one is a hybrid version of PSO and CS, the IDPSOMUT/CS. Instead of operating in a continuous space, in the discrete version, trajectories change the probability that a coordinate will take on a discrete value. The swarm formula remains unchanged, except that velocity and position must be constrained to an interval. A logistic transformation can be used to accomplish this modification. The reason for selecting evolutionary computing and swarm intelligence algorithms is that they outperformed many data clustering classical methods ([SARKAR; ROY; PURKAYASTHA, 2013](#)), as detailed in Section 3.3. The proposed algorithms are detailed in the following subsections.

3.4.1 Discrete Particle Swarm Optimization with Mutation (DPSOMUT)

The DPSOMUT is a discrete version of the traditional PSO that also uses a mutation function to diversify the generation of possible solutions. Algorithm 1 presents a pseudo-code for DPSOMUT. Since the algorithm was modeled with a discrete approach, a new formula was developed for updating the positions. This modification was introduced to accelerate the convergence velocity of the algorithm as in [CAGNINA; ERRECALDE; INGARAMO \(2008\)](#). To avoid convergence to a local optimum, a mutation is applied by swapping particles randomly. The particle velocity and position are computed according to equation (3.3) and equation (3.4), respectively. x_{id} is the value of the particle i at the dimension d , v_{id} is the velocity of particle i at the dimension d , ω is the inertia factor, γ_1 and γ_2 are the personal and social learning factors, respectively.

$$v_{id} = \omega(v_{id} + \gamma_1(pb_{best_{id}} - x_{id}) + \gamma_2(g_{best_d} - x_{id})) \quad (3.3)$$

$$x_{id} = pb_{best_{id}} \quad (3.4)$$

Algorithm 1 DPSOMUT pseudo-algorithm

```

1: input: opinion similarity matrix
2: output: vector for each cluster
3: initialize particles, cluster vector
4: while maximum iterations is not attained do
5:   for each particle do
6:     calculate fitness value
7:     if fitness value better than the best fitness value (pbest) in history then
8:       set current value as the new pbest
9:     end if
10:  end for
11:  choose particle with the best fitness value of all particles as the gbest
12:  for each particle do
13:    if particle velocity greater than random number then
14:      calculate particle velocity according to equation (3.3)
15:      update particle position according to equation (3.4)
16:    end if
17:  end for
18:  apply mutation by swapping particles randomly
19: end while

```

3.4.2 Improved Self-Adaptive PSO with Mutation (IDPSOMUT)

Researches on PSO showed that the values of the weights given to the inertia, and cognitive and social factors strongly influence the behavior of the algorithm particles and can be

characterized as follows: inertia weight with high value promotes an exploratory search (global search), while an inertia with low weight promotes a refinement of the search space (local search). Likewise, cognitive and social factors that are correlated to the all swarm behavior is affected by the values of the weights of its parameters. A high value for the social factor favors the particle a search towards the best overall solution already found. In the same proportion, the cognitive factor reinforces a local search for each particle, favoring the best solution already found by itself.

The main characteristic of the improved self-adaptive PSO (IDPSOMUT) version is to make an exchange between a global to a local search operation, during the iterations (ZHANG; XIONG; ZHANG, 2013). This exchange is made from the dynamic change of inertia values, and cognitive and social factors. For these changes to take place, a detection function (equation (3.5)) needs to be computed. The expression $(gbest - x_i(t-1))$ is the Euclidean distance between the particle i and the best solution $gbest$ found by the swarm up to the iteration $(t-1)$. The expression $(pbest_i - x_i(t-1))$ is the Euclidean distance between the particle i and the best solution found by itself, $pbest_i$, up to the iteration $(t-1)$.

The $\varphi(t)$ parameter is used to update the values of inertia and cognitive and social factors according to equation (3.6), equation (3.7) and equation (3.8). The values of γ_1 and γ_2 are fixed and predefined, and t is the value of the iteration. The $\omega_{initial}$ and ω_{final} values are fixed, predefined, and describe the range in which the value of inertia will vary. K_{max} is the maximum number of iterations of the algorithm. $\varphi(t)$ is the detection function, and μ is an adjustment factor to ensure that ω , $\omega_{initial}$, and ω_{final} keep the reverse change.

$$\varphi(t) = |(gbest - x_i(t-1)) / (pbest_i - x_i(t-1))| \quad (3.5)$$

$$\gamma_1(t) = \gamma_1 \cdot \varphi^{-1}(t) \quad (3.6)$$

$$\gamma_2(t) = \gamma_2 \cdot \varphi(t) \quad (3.7)$$

$$\omega(t) = \frac{\omega_{initial} - \omega_{final}}{1 + e^{\varphi(t) \cdot (t - ((1 + \ln(\varphi(t))) \cdot K_{max}) / \mu)}} + \omega_{final} \quad (3.8)$$

In IDPSOMUT, the particle velocity and position are computed according to equation (3.9) and equation (3.10), respectively. Algorithm 2 presents a pseudo-code for IDPSOMUT. A mutation operator based on CAGNINA et al. (2014) was introduced, consisting of an exchange between two dimensions of a particle. This operator aims to reinforce the possibility that the algorithm does not fall into local minimums. According to CAGNINA et al. (2014), low mutation values have no effect on individuals and, on the other hand, high values make the algorithm slower, making the search process more difficult. Studies by ZHANG; XIONG; ZHANG (2013) indicate that IDPSO performs better than other PSO variations.

$$v_i(t+1) = \omega_i(t) * v_i(t) + \gamma_1(t) * rand * (pbest_i - x_i(t)) + \gamma_2(t) * rand * (gbest - x_i(t)) \quad (3.9)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (3.10)$$

Algorithm 2 IDPSOMUT pseudo-algorithm

```

1: input: opinion similarity matrix
2: output: vector for each cluster
3: initialize particles, cluster vector
4: while maximum iterations is not attained do
5:   for each particle do
6:     calculate fitness value
7:     if fitness value better than the best fitness value (pbest) in history then
8:       set current value as the new pbest
9:     end if
10:  end for
11:  choose particle with the best fitness value of all particles as the gbest
12:  for each particle do
13:    if particle position is equal to pbest then
14:      apply mutation by swapping particles randomly
15:    end if
16:    calculate particle velocity according to equation (3.9)
17:    update particle parameters according to equation (3.8)
18:    update particle position according to equation (3.10)
19:  end for
20: end while

```

3.4.3 Discrete Improved Self-Adaptive PSO with Cuckoo Search (IDPSOMUT/CS)

The IDPSOMUT/CS is a hybrid optimization technique that combines two algorithms: the IDPSO proposed by ZHANG; XIONG; ZHANG (2013) and MCSA proposed by FOULADGAR; LOTFI (2015). The objective is to combine their distinct characteristics to improve the search process. This combination was motivated due to a weakness in the IDPSO as, while switching from a global search to a local search, the algorithm can be stuck in a local minimum. Starting from this premise, a mechanism found in MCSA algorithm was investigated, called the *Egg Generation* method, which changes some random positions of each particle in order to disturb them to leave their location and try to discover a better one.

The IDPSOMUT procedure remains the same, only the *Egg Generation* method was inserted after the fitness function evaluation, according to the Algorithm 3. Some modifications were made to the *Egg Generation* method in relation to its original implementation. The algorithm was modeled with a discrete approach to update its positions, similar to DPSOMUT

and IDPSOMUT. Each *egg* represents a document, or a position of the search space, whereas the *nests* represents the complete solution to the problem. The method used in this Thesis follows the procedure below:

1. Generate Eggs: generate an amount x of random positions for each particle. The amount of x given by the range $[egg_min, egg_max]$.
2. Lay Eggs: for each defined position, a random number is generated by means of a Gaussian distribution and sums it with a current position.
3. Lastly, if a new part generated has a better shape than an old one, then an old part is updated, if not, a new part is discarded and an old part remains.

Algorithm 3 IDPSOMUT/CS pseudo-algorithm

```

1: input: opinion similarity matrix
2: output: vector for each cluster
3: initialize particles, cluster vector
4: while maximum iterations is not attained do
5:   for each particle do
6:     calculate fitness value
7:     for each particle do
8:       generate eggs
9:       lay eggs
10:      if new_fitness < old_fitness then
11:        update the old_fitness
12:      end if
13:    end for
14:    if fitness value better than the best fitness value (pbest) in history then
15:      set current value as the new pbest
16:    end if
17:  end for
18:  choose particle with the best fitness value of all particles as the gbest
19:  for each particle do
20:    calculate particle velocity according to equation (3.9)
21:    update particle position according to equation (3.10)
22:  end for
23: end while

```

4

OPINION MINING CORPORA

This chapter presents the three selected corpora for the experiments. It also details the OBCC corpus containing *tweets* in Brazilian Portuguese with opinions of consumers about products and services.

4.1 Opinion of Brazilian Consumer Corpus

Considering the absence of benchmark datasets for the Portuguese language and the initial stage of OM researches for the this language, this Thesis also presents and makes it publicly available¹ a corpus: The Opinion of Brazilian Consumers (OBCC) corpus contains a gold collection with 2,940 *tweets* in Brazilian Portuguese with opinions of consumers about products and services from Telecommunication, Telephony, and Banking.

Twitter is a micro-blogging service that was launched formally on July 13, 2006, but became popular in Brazil after the year 2010. The micro-blogging is one of the most used social media currently with 320 million monthly active users, generating about 1 billion messages per day ([TWITTER OFFICIAL WEBPAGE, 2016](#)). Since it is a rich source of real-time information, many entities (companies, politicians, government, etc.) have demonstrated interest in knowing the opinions of people about services and products.

Recent research shows that while Twitter's predominant language was English in the past, the global growth and adoption of the social media platform in recent years has increased the diversity in the use of languages. Portuguese is the fifth most used language on Twitter ([STATISTA, 2013](#)), with Brazil having approximately 20 times the population of Portugal and by choosing a random *tweet* in Portuguese, there is a 95% chance of it originating in Brazil ([LABOREIRO et al., 2013](#)). Studies also informed that 88% of Brazilian companies use at least one social network ([EVANGELISTA; PADILHA, 2013](#)).

¹www.miningbrgroup.com.br

4.1.1 Tweets Extraction

33,306 *tweets* were collected with opinions of consumers about products and services of ten Brazilian companies from telecommunication, telephony, and banking domains, as shown in Table 4.1. The companies were selected from the Brazilian Consumer Protection and Defense Program (PROCON) ranking of consumer complaints in the year 2013.

Table 4.1: OBCC corpus)

Business Segment	Companies	Collected ‘tweets’
Telecommunication/Telephony	Oi	3442
Telecommunication/Telephony	Claro	4652
Telecommunication/Telephony	TIM	4037
Telecommunication/Telephony	Vivo	3463
Banking	Itaú,	4411
Banking	Bradesco	3753
Banking	Banco de Brasil	1954
Banking	Caixa Econômica Federal	2993
Telephony	Samsung	1538
Telecommunication	Sky	3063
TOTAL		33306

Tweets are available publicly as a default and historical data from Twitter and other online sources are available on Topsy² database. Topsy database contains more than 475 billion *tweets*, videos, images, pins, and other social data elements included by Twitter users in their updates since Twitter’s launch, in 2006. As the only Twitter partner certified as both a data reseller and an analytics provider. Thus, a Java based web crawler was built to extract data from the Topsy database. As shown in Figure 4.1, the crawler receives the companies’ identification on Twitter as input and returned a list of *tweets* in JSON file format as output.

Figure 4.1: Topsy database collection process

4.1.2 Manual Annotation

The OBCC contains a gold collection with 2,940 *tweets* manually annotated between four classes. The *positive*, *negative*, and *neutral* classes were defined according to the definitions

²<http://topsy.com/>

presented in (PANG; LEE, 2008). The *Irrelevant* class was not used in the experiments as it contains noisy data.

The *tweets* were manually annotated as:

- Positive: *tweets* containing positive sentiments or opinions, i.e. subjective text.
- Negative: *tweets* containing negative sentiments or opinions, i.e. subjective text.
- Neutral: *tweets* which do not present sentiments or opinions, i.e. objective text.
- Irrelevant: *tweets* presenting advertisements or information not related to consumer' opinions

Five researchers participated in the process and a *tweet* was classified by two researchers, as shown in Table 4.2. The final polarity of each *tweet* was defined as the polarity assigned to it by the majority of the two annotators. In the cases where the annotators disagreed, i.e., each one classified the *tweet* as a different polarity, the first annotator decided the *tweet* polarity.

Table 4.2: OBCC annotators information

#	Age	Gender	Educational attainment	Tweets quantity
1	20	Male	Graduation student	4,000
2	36	Female	PhD candidate	2,000
3	21	Male	Graduation student	2,000
4	20	Female	Graduation student	2,000
5	21	Male	Graduation student	2,000

The Fleiss' Kappa coefficient FLEISS (1971) was computed in order to discover the agreement between the two annotators and a value of 73.11% was obtained. According to GOLDEN (2011) and WIEBE; WILSON; CARDIE (2005), the human capacity for correct evaluation of the subjectivity of a text is around 72% and 85%, thus, the corpus is appropriated for the evaluation.

In addition to the polarity classification, other information has been manually labeled by researchers and are available in the gold collection. They are:

- Company name,
- Service or product evaluated,
- User location (city, state and country), and
- Company location (city, state and country).

This collection was also partitioned into four subsets, according to balancing and number of classes (see Table 4.3). *U3* is an unbalanced dataset with three classes. This partition represents a *real feed* from Brazilian companies in Twitter micro-blogging, in which the majority of *tweets* contains negative opinions, i.e. claims. Dataset *U2* was built from *U3* without neutral opinions,

since most OM applications perform only binary classification between positive and negative opinions (binary classification). The balanced datasets *B2* and *B3* were also built from *U3* for class balancing comparison.

Table 4.3: OBBC gold collection

Dataset	Balancing	Classes	Quantity by Polarity	Total
B2	Balanced	2	Positive: 166, Negative: 166	332
B3	Balanced	3	Positive: 166, Negative: 166, Neutral: 166	498
U2	Unbalanced	2	Positive: 166, Negative: 1299	1465
U3	Unbalanced	3	Positive: 166, Negative: 1299, Neutral: 553	2018

4.2 Selected Corpora

For the experimental work, besides the OBCC corpus, two other corpora with different levels of complexity with respect to size, number of opinions, domains, language, POS tagging, and class balancing were selected. Table 4.4 presents the details about each corpora. The first column contains the corpus name, the second column presents the number of classes to be clustered, while the third column informs the class balancing type: *balanced* classes have the same number of opinion, while *unbalanced* classes have different numbers of opinions. Column *POS-Tagger* contains the tagger’s name used during the natural language pre-processing step. In the sequence, the number of opinions for each class (positive, negative, and neutral) and for all classes are presented. Then, the number of tokens from the whole corpus is presented, followed by the average number of tokens per opinion, the number of tokens after natural language pre-processing and, finally, the number of tokens after word bigram pre-processing. The main characteristics of each corpus are presented bellow:

- IMDb movie review corpus from [PANG; LEE; VAITHYANATHAN \(2002\)](#) contains opinions written in English about films. The document set consists of 1000 positive and 1000 negative movie reviews. We randomly selected a subset of 300 positive and 300 negative for the balanced corpus and a subset of 100 positive and 300 negative for the unbalanced dataset.
- Sentiment140 corpus, from Stanford University ([GO; BHAYANI; HUANG, 2010](#)), contains opinions written in English about brands, products, politicians, celebrities, and other topics from Twitter. The Sentiment140 gold collection contains 498 *tweets* from several domains distributed in three unbalanced classes. We built three other corpora from this collection: a balanced dataset with two and three classes, and an unbalanced dataset with two classes.

Table 4.4: OBCC, Sentiment140, and IMDb corpora details

Corpora	Lang.	Domain	Classes	Balancing	POS tagger	Pos	Neg	Neu	Total	Tokens total	Tokens Average	Tokens NLP	Tokens n-gram
IMDb	EN	Movie Review	2	Balanced	General	300	300	—	600	474,465	791	249,190	186,340
				Unbalanced	General	100	300	—	400	309,919	775	162,706	186,340
Sentiment140	EN	Several	2	Balanced	General	139	139	—	278	5,036	18	2,970	5,416
					Tweet	139	139	—	278	4,585	16	2,134	
				Unbalanced	General	182	177	—	359	6,651	19	3,896	5,416
					Tweet	182	177	—	359	6,058	17	2,790	
			3	Balanced	General	139	139	139	417	6,986	17	4,181	5,416
					Tweet	139	139	139	417	6,322	15	2,909	
				Unbalanced	General	182	177	139	498	8,601	17	5,107	5,416
					Tweet	182	177	139	498	7,795	16	3,565	
OBCC	PT	Telecommunication, Telephony and Banking	2	Balanced	General + Floresta	166	166	—	332	7,014	21	1,790	3,981
					General + Mac-Morpho	166	166	—	332	6,640	20	2,054	
				Unbalanced	General + Floresta	166	1,299	—	1,465	31,438	21	8,882	15,784
					General + Mac-Morpho	166	1,299	—	1,465	29,170	20	10,133	
			3	Balanced	General + Floresta	166	166	166	498	10,256	21	2,495	5,558
					General + Mac-Morpho	166	166	166	498	9,705	19	2,883	
				Unbalanced	General + Floresta	166	1,299	553	2,018	42,378	21	11,242	19,889
					General + Mac-Morpho	166	1,299	553	2,018	39,521	20	13,052	

5

ADOPTED METHOD

This chapter presents the method adopted for the opinion clustering task proposed in this Thesis. Section 5.1 provides a brief overview of the adopted process. Section 5.2 describes the two evaluated pre-processing approaches. In the sequence, the presented swarm-based methods and the chosen fitness functions are explained in Section 5.3. Finally, Section 5.4 presents the external measures and the comparison algorithms adopted to assess the results.

5.1 General Overview

Figure 5.1 presents the adopted process for the opinion clustering. It contains three steps which are detailed in the following subsections. Three different *corpora* were selected to assess the results: the IMDb ([PANG; LEE; VAITHYANATHAN, 2002](#)) corpus which contains movie reviews written in the English language; the Sentiment140 ([GO; BHAYANI; HUANG, 2010](#)) and the OBCC, presented in this Thesis, corpora contain *tweets* written in English and Brazilian Portuguese languages, respectively. The corpora are detailed in Chapter 4.

Two text *pre-processing* approaches are evaluated for the opinion clustering task: a natural language and a word bigram. In the sequence, opinions are represented using a VSM associated with the TF-IDF weighting scheme. After building the VSM model, the proposed approaches use the *cosine* measure to estimate the similarity between two opinions. The opinion similarity matrix is used as input for the next step.

In the *processing* step, the three swarm-based clustering algorithms proposed in this Thesis (Chapter 3) are executed. The GS and the ADDC ICVM are used as optimization function.

In the *evaluation* step, results are compared to popular clustering and swarm-based algorithms. Finally, the quality of the resulting clusters is evaluated through accuracy, precision, recall, and f-score measures.

Figure 5.1: Opinion clustering method

5.2 Pre-processing

Two text *pre-processing* approaches are evaluated in this Thesis: a natural language and a word bigram. All Pre-processing steps were performed using the Python NLTK and are detailed in the following subsections. For the n-gram model, meta-information, URLs and tags, words such as 'RT' or 'via', punctuation characters as @, full stops, commas, and others were removed from Twitter' corpus. For both *pre-processing* approaches, each opinion was split into words using the *word tokenize* implementation from the Python Natural Language Toolkit (NLTK). Figure 5.2 shows the result after this step. The words were also converted to lower case.

Figure 5.2: Tweets after tokenization.

5.2.1 Natural Language Pre-processing

A POS tagger determines the parts of speech, e.g. noun, verb, adjective, etc. for each term in a sentence. As example, Table 5.1 shows the Floresta Sinta(c)tica Grammatical tags. The Perceptron POS tagger was used for both English language corpora. For the Sentiment140 corpus, the Carnegie Mellon POS Tagger (OWOPUTI et al., 2012), specific for *tweets* written in English language was used. For the Brazilian Portuguese OBCC corpus, two POS tagger were selected: Perceptron and Unigram taggers. The former was trained using Floresta Sinta(c)tica corpus while the second was trained using MacMorpho corpus, both available at Python NLTK. The chosen POS taggers presented good outcomes for selected corpora. Figure 5.3 shows a *tweet* after POS tagging.

As adjectives, adverbs, nouns, and verbs are strong indicators of sentiment in an opinion (MARQUES-LUCENA; SARRAIPA, 2015; PANG; LEE, 2008), they were selected to

build a local dictionary. Words from other parts-of-speech were discarded during the *Feature Reduction* step as shown in Figure 5.4.

Table 5.1: Floresta Sinta(c)tica grammatical tags

Symbols	Category
n	noun
prop	proper noun
adj	adjective
n-adj	between noun and adjective
v	v-fin
	v-inf
	v-pcp
	v-ger
art	article
pron	pron-pers
	pron-det
	pron-indp
adv	adverb
num	numeral
prp	preposition
intj	interjection
conj	conj-s
	conj-c
	subordinating conjunction
	coordinating conjunction

Figure 5.3: Tweets after POS tagging.

Figure 5.4: Tweets after feature reduction.

5.2.2 Word Bigram Pre-processing

N-gram is a probabilistic language model widely used in text classification (RAVI; RAVI, 2015). In computational linguistic, an n-gram is a contiguous sequence of n items from a given sequence of text. These items can be phonemes, character, words, and others. A language model attempts to reflect the frequency with which each item occurs as a sentence in a text. Unigram refers to n-gram of size 1, bigram refers to n-gram of size 2, and so on. In this Thesis, in order to handle negation, a word bigrams feature model was used as shown on Figure 5.5.

Figure 5.5: Tweets after word bigram pre-processing.

Word n-gram have been widely used in polarity classification (TRIPATHY; AGRAWAL; RATH, 2016). Researchers have reported that unigrams outperform bigrams when classifying opinions from one specific corpus while others reported that bigrams and trigrams yield better results (PANG; LEE, 2008; RAVI; RAVI, 2015). TRIPATHY; AGRAWAL; RATH (2016) reported best results using a combination of unigrams, bigrams, and trigrams for three different corpora.

5.2.3 VSM + TF-IDF Weighting

For both, natural language and n-gram models, in the *Feature Transformation* step, opinions were represented using the VSM associated with the TF-IDF weighting scheme. The opinions (O) are represented as vectors $O_j = (w_{1j}, w_{2j}, w_{3j}, \dots, w_{tj})$ and each dimension corresponds to a separate word (w) or term for the opinion j . The opinion *versus* word matrix generated can be seen in Figure 5.6.

	word 1	word 2	...	word n
opinion 1	0	0.5		0.2
opinion 2	0.8	0		0
...				
opinion m	0.3	0		0.9

Figure 5.6: Opinion versus word matrix

The TF-IDF is the most popular term weighting scheme in text classification (HUANG, 2016). According to WU et al. (2004), TF-IDF can be described as follows: given a set of documents $D = d_1, d_2, \dots, d_n$ with $d_i \in D$, the frequency of the term $TF(d, t)$ is the number of times a term t occurs in a document, $IDF(t)$ is the number of documents in which the term occurs at least once. The inverse of the term frequency in the document collection (IDF) can be calculated by equation (5.1) and equation (5.2). As a result, TF assign a higher weight to a frequent term while the IDF decreases the weight if the term occurs in many documents.

$$IDF(t) = \log\left(\frac{|D|}{DF(t)}\right) \quad (5.1)$$

$$W(t) = TF(d, t) * IDF(t) \quad (5.2)$$

5.2.4 Cosine Similarity Measure

After building the VSM model, the proposed methods use the *cosine* measure to estimate the similarity between two opinions. This measure is widely used in text clustering literature (CUI; POTOK, 2005; CAGNINA; ERRECALDE; INGARAMO, 2008) and it computes the *cosine* of the angle between two documents as shown in equation (5.3), where $m_p^t m_j$ denotes the dot-product of two opinion vectors and $|\cdot|$ indicates the length of the vector. The result of this step is an opinion similarity matrix as shown on Figure 5.7, which is used as input for the processing step.

$$\cos(m_p, m_j) = \frac{(m_p^t m_j)}{|m_p| |m_j|} \quad (5.3)$$

	opinion 1	opinion 2	...	opinion n
opinion 1	1	0.5		0.2
opinion 2	0.8	1		0
...				
opinion m	0.3	0		1

Figure 5.7: Opinion similarity matrix

5.3 Processing

Three discrete PSO-based algorithms are proposed in this Thesis: the first one, DPSO-MUT, is based on a discrete binary version of PSO, first proposed by KENNEDY; EBERHART (1997). The second one is based on an Improved Self-Adaptive PSO (IDPSOMUT) algorithm with detection function (ZHANG; XIONG; ZHANG, 2013). The last one is an hybrid version of PSO and CS, the IDPSOMUT/CS. These algorithms are detailed in Chapter 3.

Clustering algorithms explore documents by searching consistent patterns among them and forming groups. The quality of the resulting groups is normally evaluated with respect to structural properties expressed in different ICVM, such as: GS, ADDC, Expected Density (ED) measures, and others. These unsupervised measures of cluster validity can also be used as objective functions by clustering algorithm during the grouping process. In this sense, the proposed approach considers clustering as an optimization problem and compares two ICVM widely used in document clustering.

5.3.1 Global Silhouette (GS)

The GS measure combines two key aspects to determine the quality of a given clustering: *cohesion* and *separation*. Cohesion measures how closely related the objects in a same cluster are, whereas separation quantifies how distinct (well-separated) a cluster is from other

clusters (CAGNINA et al., 2014). The GS coefficient of a clustering is the average cluster silhouette of all the obtained groups. GS has been successfully applied, as PSO-based optimization function, to short-text clustering (CAGNINA; ERRECALDE; INGARAMO, 2008; CAGNINA et al., 2014; INGARAMO et al., 2009, 2011).

The GS is computed by equation (5.4), where $a(i)$ is the average dissimilarity of i with all other data within the same cluster and $b(i)$ is the average dissimilarity of i to any other cluster, of which i is not a member. The silhouette ranges from -1 to 1 , where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.

$$s(i) = \frac{(b(i) - a(i))}{\max(a(i), b(i))} \quad (5.4)$$

5.3.2 Average Distance Documents to the Cluster Centroid (ADDC)

ADDC indicates the value of the average distance between documents and the cluster centroid to which they belong. The ADDC is used as the fitness value to evaluate the solution represented by each PSO particle. The fitness value is measured by equation (5.5), where m_{ij} denotes the j_{th} document vector, which belongs to cluster i , O_i is the centroid vector of i_{th} cluster, $d(O_i, m_{ij})$ is the distance between document m_{ij} and the cluster centroid O_i , p_i stands for the document number, which belongs to cluster C_i , and N_c is the cluster number (CUI; POTOK; PALATHINGAL, 2005).

The smaller the ADDC value, the more compact the clustering solution is. ADDC has been successfully applied, as PSO-based optimization function, to several document clustering approaches (CUI; POTOK; PALATHINGAL, 2005; CUI; POTOK, 2005; PREMALATHA; NATARAJAN, 2009, 2010; SARKAR; ROY; PURKAYASTHA, 2014a,b; KAROL; MANGAT, 2013; KAMEL; OUCHEN; BAALI, 2016).

$$f = \frac{\sum_{i=1}^{N_c} \left\{ \frac{\sum_{j=1}^{p_i} d(O_i, m_{ij})}{p_i} \right\}}{N_c} \quad (5.5)$$

5.4 Evaluation

The quality of the resulting clusters is commonly evaluated with respect to structural properties expressed in different ICVM such as the GS. These internal measures are very common in document and short-text clustering, but, as stated by CAGNINA; ERRECALDE; INGARAMO (2008); KAROL; MANGAT (2013), the real effectiveness of the clustering algorithms can only be evaluated with external measures that incorporate the categorization criteria of the users. The following external measures are used in this Thesis: *accuracy*, *precision*, *recall*, and *f-score*.

5.4.1 External Measures

The external metrics are calculated using terms such *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), and *False Negative* (FN) (TRIPATHY; AGRAWAL; RATH, 2016). As shown on Table 5.2, considering a grouping of two classes (positive and negative), TP corresponds to opinions that belong to positive class and were also labeled as positive by the classifier, while FP represents the opinions that were classified as positive but belong to the negative class. In the same way, TN indicates negative opinions that were correctly labeled as negative and FN corresponds to positive opinions that were classified as negative.

Table 5.2: Confusion matrix for positive and negative classes

		Correct labels	
		Positive	Negative
Predicted labels	Positive	TP	FP
	Negative	FN	TN

■ Accuracy

It is measured by the ratio between the number of all correctly classified opinions and the total number of opinions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.6)$$

■ Precision

Calculated by the ratio of the number of opinions correctly classified in the first class to the total number of opinions classified in the first class. It is a metric to measure the exactness of the classifier result.

$$Precision = \frac{TP}{TP + FP} \quad (5.7)$$

■ Recall

Measured by the ratio of the number of opinions correctly classified in the first class to the total number of opinions that actually belong to the first class. It indicates the completeness of the classifier result.

$$Recall = \frac{TP}{TP + FN} \quad (5.8)$$

■ F-measure

Also called F-score or F1, it is the harmonic mean of precision and recall measures.

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (5.9)$$

5.4.2 Comparison Algorithms

Five algorithms were selected for comparison: the K-means and Agglomerative clustering algorithms and the PSO, Artificial Bee Colony and Cuckoo Search swarm-based algorithms.

■ K-means clustering

The K-means method (MACQUEEN et al., 1967) is currently used to divide a dataset into k groups. It proceeds by selecting k initial cluster centers and, iteratively, refining them as follows:

1. Each instance d_i is assigned to its closest cluster center;
2. Each cluster center C_j is updated to be the mean of its constituent instances.

The number of required iterations can vary in a wide range from a few to several thousand depending on the number of patterns, number of clusters, and the input data distribution. The algorithm runs several iterations until there is no further change in assignment of instances to clusters. In this Thesis, the k number of cluster on each dataset is previously know and the Euclidean distance metric is used as similarity measure.

■ Hierarchical Agglomerative clustering

Hierarchical Agglomerative algorithms find the clusters by initially assigning each object to its own cluster and then repeatedly merging pairs of clusters until a certain stopping criterion is met. The various clustering criterion functions that are considered can be used to determine the pairs of clusters to be merged at each step in the following way:

1. Consider an n -document dataset and the clustering solution that has been computed after performing l merging steps, this solution will contains exactly $n - l$ clusters, as each merging step reduces the number of clusters by one.
2. Now, given this $(n - l)$ -way clustering solution, the pair of clusters that is selected to be merged next, is the one that leads to an $(n - l - 1)$ -way solution that optimizes the particular criterion function.

That is, each one of the $(n - l)x(n - l - 1)/2$ pairs of possible merges is evaluated, and the one that leads to a clustering solution that has the maximum (or minimum) value of the particular criterion function is selected. Thus, the criterion function is locally optimized within the particular algorithm stage. Depending on the desired solution, this process continues until either there are only k cluster left, or when the entire Agglomerative tree is obtained.

■ Cuckoo Search (CS)

The CS used in this Thesis is a search algorithm proposed by [YANG; DEB \(2009\)](#), based on population, similar to GA and PSO. It explores both mechanism of elitism and a targeted search. Three basic steps characterize this algorithm:

1. Each cuckoo lays one egg at a time in a randomly chosen host nest;
2. The best nest with the best egg quality will be transferred to next generation and
3. Finally, a number of host nodes is defined according to a probability $p_a \in [0,1]$ to be discarded.

The host bird can either lay a parasitic egg in another nest, or also lose its old nest by a new one. The new nests defined according to the probability p_a , are made up random solutions. For this Thesis, the algorithm was modeled with a discrete approach to update its positions, similar to DPSOMUT and IDPSOMUT. Each *egg* represents a document, or a position of the search space, whereas the *nests* represents the complete solution to the problem.

The goal is to explore new solutions to replace bad solutions. When using random walk, new solutions tend to be exploited instead of local search. For this same reason, a portion of the population is randomized so that the new solutions stay away from the best current solution.

■ Artificial Bee Colony (ABC)

ABC ([KARABOGA; BASTURK, 2007](#)) contains three types of bees: *employed bees*, *onlookers* and *scouts*. A bee waiting on the dance area for making decision to choose a food source, is called an onlooker and a bee going to the food source visited by itself previously is named an employed bee. A bee carrying out random search is called a scout. For every food source, there is only one employed bee. In other words, the number of employed bees is equal to the number of food sources. The employed bee of an abandoned food source becomes a scout. The search carried out by the artificial bees can be summarized as follows:

- Employed bees determine a food source based on neighboring food sources stored in memory;
- Employed bees share their information with onlookers within the hive and then the onlookers select one of the food sources;
- Onlookers select a food source between all food sources in a neighbor of sources chosen randomly;

- An employed bee of which the source has been abandoned becomes a scout and starts to search a new food source randomly.

After finding the food source, the bee utilizes its own capability to memorize the location and then immediately starts exploiting it. It is important to note that not all bees start foraging simultaneously. The experiments confirmed that new bees begin foraging at a rate proportional to the difference between the eventual total number of bees and the number of bees presently foraging.

6

RESULTS

This Chapter presents and discusses the obtained results. Section 6.1 presents the configuration parameters used in the experiments. Section 6.2 and Section 6.3 present the results obtained for natural language and word bigram pre-processing, respectively. Section 6.4 discusses the difference between the two used pre-processing techniques. Section 6.5 compares the swarm-based and clustering algorithms for the word bigram pre-processing. Section 6.6 evaluates the GS and ADDC fitness functions for the non-hybrid PSO-based algorithms. Finally, Section 6.7 discuss the overall results with similar works.

6.1 Experimental Setup

The PSO-based algorithms proposed in this study were developed using Java language, except the DPSOMUT which was implemented using Python. The ABC was also developed using Python language. The K-means and Agglomerative versions available on Python scikit-learn toolkit were selected. Default setup parameters were adopted for each algorithm. For the K-means, CS, IDPSOMUT, and IDPSOMUT/CS, we performed 25 runs with 1000 iterations per run. For the Agglomerative, DPSOMUT, and ABC we performed 25 runs with 50 iterations each. The following parameters were defined:

- DPSOMUT, PSO: swarm size = 20 particles, dimensions of each particle = number of opinions, $\omega = [0.9 - 0.4]$, γ_1 and $\gamma_2 = 1$.
- IDPSOMUT: swarm size = 30 particles, dimensions of each particle = number of opinions, $\omega_{initial}$ and $\omega_{final} = [0.1 - 0.8]$, γ_1 and $\gamma_2 = [2.4 - 1.3]$, $\mu = 100$, and max_{pm} and $min_{pm} = [2 - 0]$.
- IDPSOMUT/CS: swarm size = 30 particles, dimensions of each particle = number of opinions, $\omega_{initial}$ and $\omega_{final} = [0.3 - 0.8]$, γ_1 and $\gamma_2 = 2$, $egg_{min} = 5$, and egg_{max} = number of dimensions.
- CS: $p_a = 0.7$, $\alpha = 2$, $\lambda = 3$.

- ABC: number of bees = 20, onlooker bees = 50%, employed bees = 50%, scout bees = 100%.

A grid search was performed to identify a better parameter setting for each compared algorithm (see Table 6.1). The best parameters for an algorithm were those that maximize its clustering accuracy with low computational cost.

Table 6.1: Parameter values used in grid search to maximize accuracy

Parameters	Methods	Values
Particles	PSO	[5, 10, 15, 20, 25, 30]
	DPSOMUT	
	DPSOMUT	
	IDPSOMUT/CS	
ω	CS and Discrete ABC	[1.3, 1.1, 0.9]
	DPSOMUT	
$\omega_{initial}$	IDPSOMUT	[1, 0.9, 0.8, 0.7]
	IDPSOMUT/CS	
max_{pm}	IDPSOMUT	[0.4, 0.3, 0.2, 0.1]
	IDPSOMUT/CS	
	PSO	
	DPSOMUT	
min_{pm}	IDPSOMUT	[4, 3, 2, 1]
	PSO	
	DPSOMUT	
γ_1	IDPSOMUT	[2, 1, 0]
	IDPSOMUT	
γ_2	IDPSOMUT	[2.4, 2, 1.8, 1.6]
	IDPSOMUT/CS	
γ_3	IDPSOMUT	[2, 1.7, 1.5, 1.3]
	IDPSOMUT/CS	
egg_{min}	IDPSOMUT/CS	[20, 15, 10, 5]
p_a /onlooker bees	CS	[0.8, 0.7, 0.6, 0.5]
	Discrete ABC	
scout bees	CS	[1, 0.7, 0.5, 0.2]

6.2 Results for Natural Language Pre-processing

Table 6.2, Table 6.3, and Table 6.4 presents the results using the natural language pre-processing methods. Eighteen experiments were performed with different corpora types, class balancing, POS taggers, and algorithms. In the *Class Balancing* column, 'B' refers to balanced classes, while 'U' refers to unbalanced classes. The number '2' refers to binary classification (positive or negative), while the number '3' refers to ternary classification (positive, negative, or neutral).

The two proposed PSO-based algorithms uses the GS coefficient (equation (5.4)), as fitness function, and are named as: DPSOMUT (GS) and IDPSOMUT (GS). The PSO-based approaches achieved better accuracy on twelve experiments (tagged with asterisks). The best accuracy (79%) was obtained by the IDPSOMUT algorithm on OBCC (Unigram POS Tagger + Floresta Sinta(c)tica) corpus for two balanced classes.

The PSO-based approaches achieved better results in corpora with a reduced number of terms (dimensions) and for specific domains, such as the OBCC corpus. The worst results were achieved in corpora with different domains, such as the Sentiment140 corpus. It was observed a significant improvement in the results of the Twitter' corpus which used a *tweet* specific POS tagger. For the Brazilian Portuguese language, a *tweet* specific POS tagger was not found. As observed for the English language, this specific tagger added an improvement in the results. There was no significant difference in the accuracy for the Brazilian Portuguese corpus tagged with Floresta Sinta(c)tica or Mac-Morpho corpora.

6.3 Results for Word Bigram Pre-processing

Table 6.5, Table 6.6, and Table 6.7 present the results using a word bigram pre-processing for IMDb, OBCC, and Sentiment140 corpora, respectively. In this experiment, the same algorithms presented in previous section were selected. It was included the hybrid PSO-based algorithm, the IDPSOMUT/CS. Results were compared to two swarm-based algorithms (ABC and CS). Also, another fitness function widely used in document clustering, the ADDC, was evaluated.

For the OBCC corpus, the IDPSOMUT (GS) algorithm obtained the best result in all four experiments. The best accuracy (79%) was reached in a balanced dataset with two classes. For the IMDb movie review corpus, the IDPSOMUT (ADDC) obtained the best accuracy (56%) for the balanced dataset, while the IDPSOMUT/CS obtained the best accuracy (63%) for the unbalanced dataset. For the Stanford Sentiment140 corpus, again, the IDPSOMUT (GS), algorithm obtained the best result in all four experiments. The best accuracy (61%) was reached in datasets with two classes.

Table 6.8, Table 6.9, and Table 6.10 present the results by class using a word bigram pre-processing for IMDb, OBCC, and Sentiment140 corpora, respectively. For the IMDb movie review corpus and for the OBCC corpus, the algorithms performed better in identifying negative opinions, while for the Sentiment140, they performed better in identifying positive opinions.

Table 6.2: Average results for IMDb Movie Review corpus using a natural language pre-processing

Class Balancing	Algorithm	Precision	Recall	F-Score	Accuracy
B2	K-means	0.502	0.502	0.500	0.502
	Agglomerative	0.483	0.483	0.483	0.483
	PSO (GS)	0.496	0.497	0.457	0.497
	DPSOMUT (GS)	0.497	0.498	0.451	0.498
	IDPSOMUT (GS)	0.369	0.452	0.396	0.359
U2	K-means	0.510	0.512	0.482	0.522
	Agglomerative	0.489	0.491	0.486	0.574
	PSO (GS)	0.504	0.504	0.503	0.621
	DPSOMUT (GS)	0.496	0.496	0.496	0.629
	IDPSOMUT (GS)	0.155	0.370	0.211	0.374

6.4 Comparison between Natural Language and Word Bi-gram Pre-processing

In this Thesis, the natural language and the word bigram pre-processing techniques were evaluated in order to determine which technique produces better results for the opinion clustering task. For this experiment, the following algorithms were selected: K-means, Agglomerative, PSO (GS), DPSOMUT (GS), and IDPSOMUT (GS). The algorithms were tested with the same parameters for the two types of pre-processing techniques.

Table 6.11 compares the accuracy obtained between the two pre-processing techniques. It was observed that the n-gram pre-processing improved the accuracy of fifteen experiments from a total of thirty for datasets with two classes and improved thirteen experiments from a total of twenty for datasets with three classes. In two experiments there was no change in accuracy. In summary, the n-gram pre-processing performs better in twenty-nine experiments out of forty-eight experiments (58%).

When analyzing only the proposed PSO-based algorithms DPSOMUT (GS) and IDPSOMUT (GS), the n-gram pre-processing improved the accuracy of fourteen out of twenty experiments (70%). In two experiments there was no change in accuracy. The best improvements were observed on Sentiment140 corpus.

To confirm this assumption, the *paired T-test* was performed. The null hypothesis states that there is no change between the two pre-processing techniques, that is, *difference* = 0, while the alternative hypothesis indicates that, *difference* > 0, that is, the changes between natural language and n-gram pre-processing brings significant improvements.

Based on these hypotheses, the R software was used to calculate the statistical test with a 95% confidence level. The *P*-value obtained was equal to 0.1827, indicating that this value is in the critical region, which confirms that the null hypothesis was rejected with a 0.05 significance level. Therefore, it is statistically proven that, for these corpora, the n-gram pre-processing produces better results than the natural language one.

Table 6.3: Average results for OBCC corpus using a natural language pre-processing

Class Balancing	Corpus	Algorithm	Precision	Recall	F-Score	Accuracy
B2	OBCC Unigram POS Tagger Floresta Corpus	K-means	0.550	0.523	0.454	0.523
		Agglomerative	0.527	0.552	0.463	0.552
		PSO (GS)	0.518	0.512	0.472	0.512
		DPSOMUT (GS)	0.516	0.509	0.454	0.509
		IDPSOMUT (GS)	0.779	0.784	0.788	0.790
	OBCC Perceptron POS Tagger Mac-Morpho Corpus	K-means	0.530	0.514	0.774	0.779
		Agglomerative	0.440	0.364	0.349	0.440
		PSO (GS)	0.525	0.519	0.489	0.519
		DPSOMUT (GS)	0.510	0.507	0.467	0.507
		IDPSOMUT (GS)	0.782	0.756	0.764	0.773
U2	OBCC Unigram POS Tagger Floresta Corpus	K-means	0.505	0.506	0.353	0.467
		Agglomerative	0.356	0.318	0.147	0.156
		PSO (GS)	0.506	0.511	0.491	0.694
		DPSOMUT (GS)	0.505	0.509	0.491	0.701
		IDPSOMUT (GS)	0.710	0.599	0.597	0.785
	OBCC Perceptron POS Tagger Mac-Morpho Corpus	K-means	0.507	0.511	0.353	0.463
		Agglomerative	0.599	0.564	0.298	0.302
		PSO (GS)	0.507	0.513	0.492	0.696
		DPSOMUT (GS)	0.506	0.511	0.494	0.706
		IDPSOMUT (GS)	0.246	0.662	0.358	0.732
B3	OBCC Unigram POS Tagger Floresta Corpus	K-means	0.330	0.330	0.244	0.330
		Agglomerative	0.275	0.126	0.162	0.275
		PSO (GS)	0.334	0.334	0.311	0.334
		DPSOMUT (GS)	0.339	0.337	0.309	0.337
		IDPSOMUT (GS)	0.324	0.180	0.229	0.324
	OBCC Perceptron POS Tagger Mac-Morpho Corpus	K-means	0.320	0.323	0.248	0.323
		Agglomerative	0.365	0.358	0.313	0.365
		PSO (GS)	0.324	0.326	0.306	0.326
		DPSOMUT (GS)	0.335	0.338	0.311	0.338
		IDPSOMUT (GS)	0.404	0.249	0.302	0.371
U3	OBCC Unigram POS Tagger Floresta Corpus	K-means	0.306	0.310	0.230	0.357
		Agglomerative	0.574	0.561	0.253	0.254
		PSO (GS)	0.333	0.333	0.273	0.315
		DPSOMUT (GS)	0.335	0.334	0.272	0.314
		IDPSOMUT (GS)	0.329	0.267	0.273	0.423
	OBCC Perceptron POS Tagger Mac-Morpho Corpus	K-means	0.401	0.337	0.247	0.347
		Agglomerative	0.291	0.079	0.094	0.095
		PSO (GS)	0.331	0.331	0.270	0.311
		DPSOMUT (GS)	0.330	0.332	0.270	0.310
		IDPSOMUT (GS)	0.376	0.334	0.342	0.527

Table 6.4: Average results for Sentiment140 corpus using a natural language pre-processing

Class Balancing	Corpus	Algorithm	Precision	Recall	F-Score	Accuracy
B2	Sentiment140 General POS tagger	K-means	0.499	0.499	0.480	0.499
		Agglomerative	0.514	0.519	0.483	0.514
		PSO (GS)	0.479	0.484	0.454	0.484
		DPSOMUT (GS)	0.487	0.491	0.444	0.491
		IDPSOMUT (GS)	0.251	0.261	0.249	0.269
	Sentiment140 Twitter POS tagger	K-means	0.521	0.508	0.470	0.508
		Agglomerative	0.518	0.523	0.489	0.518
		PSO (GS)	0.481	0.487	0.443	0.487
		DPSOMUT (GS)	0.477	0.486	0.428	0.486
		IDPSOMUT (GS)	0.239	0.255	0.244	0.234
U2	Sentiment140 General POS tagger	K-means	0.495	0.495	0.480	0.496
		Agglomerative	0.246	0.343	0.283	0.370
		PSO (GS)	0.491	0.492	0.459	0.489
		DPSOMUT (GS)	0.489	0.493	0.451	0.489
		IDPSOMUT (GS)	0.245	0.206	0.220	0.277
	Sentiment140 Twitter POS tagger	K-means	0.521	0.510	0.452	0.510
		Agglomerative	0.500	0.500	0.460	0.504
		PSO (GS)	0.482	0.487	0.441	0.483
		DPSOMUT (GS)	0.485	0.491	0.435	0.486
		IDPSOMUT (GS)	0.248	0.264	0.255	0.222
B3	Sentiment140 General POS tagger	K-means	0.304	0.312	0.293	0.312
		Agglomerative	0.293	0.302	0.281	0.293
		PSO (GS)	0.341	0.341	0.321	0.341
		DPSOMUT (GS)	0.331	0.333	0.307	0.333
		IDPSOMUT (GS)	0.331	0.223	0.256	0.283
	Sentiment140 Twitter POS tagger	K-means	0.368	0.347	0.281	0.347
		Agglomerative	0.345	0.396	0.299	0.345
		PSO (GS)	0.337	0.335	0.313	0.335
		DPSOMUT (GS)	0.336	0.333	0.306	0.333
		IDPSOMUT (GS)	0.474	0.301	0.354	0.421
U3	Sentiment140 General POS tagger	K-means	0.330	0.332	0.317	0.332
		Agglomerative	0.305	0.313	0.301	0.319
		PSO (GS)	0.340	0.337	0.309	0.318
		DPSOMUT (GS)	0.337	0.335	0.300	0.313
		IDPSOMUT (GS)	0.383	0.239	0.286	0.316
	Sentiment140 Twitter POS tagger	K-means	0.355	0.334	0.270	0.338
		Agglomerative	0.345	0.397	0.304	0.369
		PSO (GS)	0.335	0.335	0.303	0.314
		DPSOMUT (GS)	0.322	0.326	0.287	0.303
		IDPSOMUT (GS)	0.498	0.302	0.365	0.425

Table 6.5: Average results for IMDb Movie Review corpus using n-gram pre-processing

Balancing	Algorithm	Precision	Recall	F-Score	Accuracy
B2	K-means	0.504	0.460	{0.502}	0.504
	Agglomerative	0.528	0.565	0.451	0.528
	PSO (GS)	0.493	0.494	0.462	0.494
	DPSOMUT (GS)	0.496	0.473	0.461	0.497
	IDPSOMUT (GS)	0.497	0.387	0.459	0.497
	Cuckoo Search	0.500	0.451	0.500	0.500
	Discrete ABC	0.498	0.457	0.498	0.498
	IDPSOMUT/CS	0.475	0.367	0.452	0.475
	IDPSOMUT (ADDC)	0.564	0.463	0.535	0.564
	DPSOMUT (ADDC)	0.485	0.465	0.453	0.489
U2	K-means	0.501	0.501	0.464	0.500
	Agglomerative	0.497	0.497	0.487	0.555
	PSO (GS)	0.495	0.495	0.495	0.617
	DPSOMUT (GS)	0.497	0.497	0.497	0.632
	IDPSOMUT (GS)	0.553	0.551	0.523	0.616
	Cuckoo Search	0.497	0.498	0.469	0.507
	Discrete ABC	0.502	0.503	0.470	0.505
	IDPSOMUT/CS	0.539	0.536	0.534	0.633
	IDPSOMUT (ADDC)	0.456	0.446	0.443	0.584
	DPSOMUT (ADDC)	0.496	0.497	0.496	0.629

Although the difference was not very large between the two pre-processing techniques, it is a relevant result as the n-gram allows one to perform a simple text pre-processing which is fully language-independent, unlike the natural language pre-processing, in which the components need to be selected according to the text language and also to the text type (e.g. POS tagger specific for *tweets*).

6.5 Comparison between Clustering and Swarm-based Algorithms

In this section, the performance of the algorithms is evaluated for the word bigram pre-processing. Initially, the algorithms are evaluated separately for each corpus. Following, a general analysis is performed for all corpora in order to identify which algorithm is more efficient for the opinion clustering task.

6.5.1 IMDb Movie Review

According to Figure 6.1, it can be observed that, for the balanced dataset with two classes, the algorithms performs similarly. The *means* are close to 0.5 value, excepting the IDPSOMUT (GS) and the IDPSOMUT (ADDC) algorithms. These two algorithms present the same type of search, differing only in the optimization function used. This shows that the objective function does not present a change in performance, but rather the search type used.

Table 6.6: Average results for OBCC corpus using n-gram pre-processing

Balancing	Algorithm	Precision	Recall	F-Score	Accuracy
B2	K-means	0.513	0.500	0.361	0.500
	Agglomerative	0.512	{0.753}	0.360	0.512
	PSO (GS)	0.509	0.507	0.478	0.507
	DPSOMUT (GS)	0.516	0.510	0.467	0.510
	IDPSOMUT (GS)	0.622	0.741	0.567	0.622
	Cuckoo Search	0.508	0.508	0.508	0.508
	Discrete ABC	0.495	0.495	0.495	0.495
	IDPSOMUT/CS	0.498	0.497	0.486	0.498
	IDPSOMUT (ADDC)	0.516	0.515	0.513	0.516
	DPSOMUT (ADDC)	0.517	0.512	0.468	0.512
U2	K-means	0.498	0.499	0.310	0.512
	Agglomerative	0.502	0.557	0.106	0.117
	PSO (GS)	0.505	0.509	0.490	0.694
	DPSOMUT (GS)	0.503	0.506	0.490	0.703
	IDPSOMUT (GS)	0.717	0.689	0.700	0.870
	Cuckoo Search	0.506	0.503	0.420	0.512
	Discrete ABC	0.499	0.499	0.412	0.501
	IDPSOMUT/CS	0.499	0.500	0.456	0.607
	IDPSOMUT (ADDC)	0.560	0.524	0.456	0.559
	DPSOMUT (ADDC)	0.505	0.508	0.492	0.704
B3	K-means	0.348	0.334	0.193	0.334
	Agglomerative	0.337	0.446	0.180	0.337
	PSO (GS)	0.332	0.334	0.315	0.334
	DPSOMUT (GS)	0.344	0.341	0.322	0.341
	IDPSOMUT (GS)	0.345	0.251	0.191	0.347
	Cuckoo Search	0.339	0.204	0.254	0.339
	Discrete ABC	0.325	0.325	0.325	0.325
	IDPSOMUT/CS	0.361	0.213	0.266	0.362
	IDPSOMUT (ADDC)	0.335	0.201	0.252	0.335
	DPSOMUT (ADDC)	0.334	0.335	0.314	0.335
U3	K-means	0.251	0.326	0.159	0.279
	Agglomerative	0.334	0.306	0.058	0.084
	PSO (GS)	0.332	0.333	0.267	0.307
	DPSOMUT (GS)	0.333	0.334	0.268	0.308
	IDPSOMUT (GS)	0.321	0.287	0.255	0.600
	Cuckoo Search	0.329	0.227	0.230	0.339
	Discrete ABC	0.329	0.329	0.268	0.309
	IDPSOMUT/CS	0.354	0.265	0.282	0.449
	IDPSOMUT (ADDC)	0.334	0.251	0.261	0.410
	DPSOMUT (ADDC)	0.329	0.329	0.268	0.309

Table 6.7: Average results for Sentiment140 corpus using n-gram pre-processing

Balancing	Algorithm	Precision	Recall	F-Score	Accuracy
B2	K-means	0.464	0.494	0.370	0.494
	Agglomerative	0.475	0.359	0.339	0.475
	PSO (GS)	0.486	0.489	0.459	0.489
	DPSOMUT (GS)	0.495	0.497	0.454	0.497
	IDPSOMUT (GS)	0.610	0.730	0.554	0.615
	Cuckoo Search	0.495	0.495	0.494	0.495
	Discrete ABC	0.502	0.502	0.501	0.502
	IDPSOMUT/CS	0.521	0.522	0.509	0.521
	IDPSOMUT (ADDC)	0.523	0.524	0.520	0.523
	DPSOMUT (ADDC)	0.483	0.489	0.443	0.489
U2	K-means	0.407	0.485	0.353	0.488
	Agglomerative	0.481	0.364	0.342	0.487
	PSO (GS)	0.497	0.498	0.464	0.494
	DPSOMUT (GS)	0.498	0.499	0.454	0.495
	IDPSOMUT (GS)	0.610	0.730	0.554	0.615
	Cuckoo Search	0.496	0.496	0.496	0.496
	Discrete ABC	0.505	0.505	0.504	0.505
	IDPSOMUT/CS	0.493	0.492	0.482	0.495
	IDPSOMUT (ADDC)	0.485	0.490	0.445	0.487
	DPSOMUT (ADDC)	0.485	0.490	0.445	0.487
B3	K-means	0.344	0.334	0.213	0.334
	Agglomerative	0.326	0.273	0.191	0.326
	PSO (GS)	0.339	0.338	0.316	0.338
	DPSOMUT (GS)	0.333	0.333	0.304	0.333
	IDPSOMUT (GS)	0.358	0.213	0.194	0.357
	Cuckoo Search	0.324	0.193	0.242	0.324
	Discrete ABC	0.331	0.331	0.330	0.331
	IDPSOMUT/CS	0.358	0.211	0.264	0.357
	IDPSOMUT (ADDC)	0.308	0.175	0.222	0.308
	DPSOMUT (ADDC)	0.330	0.331	0.307	0.331
U3	K-means	0.326	0.334	0.209	0.346
	Agglomerative	0.319	0.261	0.193	0.347
	PSO (GS)	0.334	0.333	0.300	0.312
	DPSOMUT (GS)	0.338	0.336	0.297	0.313
	IDPSOMUT (GS)	0.402	0.331	0.258	0.438
	Cuckoo Search	0.331	0.200	0.248	0.333
	Discrete ABC	0.332	0.332	0.331	0.332
	IDPSOMUT/CS	0.316	0.185	0.232	0.328
	IDPSOMUT (ADDC)	0.329	0.197	0.246	0.340
	DPSOMUT (ADDC)	0.336	0.339	0.305	0.317

Table 6.8: Results by class for IMDb Movie Review corpus using n-gram pre-processing

Class Balancing	Algorithm	Positive			Negative		
		Precision	Recall	F-score	Precision	Recall	F-score
B2	K-means	0,536	0,603	0,567	0,546	0,477	0,509
	Agglomerative	0,903	0,516	0,657	0,153	0,613	0,245
	PSO (GS)	0,489	0,248	0,329	0,496	0,741	0,594
	DPSOMUT (GS)	0,494	0,238	0,321	0,498	0,757	0,601
	IDPSOMUT (GS)	0,722	0,489	0,577	0,273	0,598	0,341
	Cuckoo Search	0,516	0,500	0,508	0,484	0,500	0,492
	Discrete ABC	0,493	0,537	0,519	0,524	0,533	0,545
	IDPSOMUT/CS	0,717	0,489	0,581	0,250	0,469	0,326
	IDPSOMUT (ADDC)	0,795	0,544	0,644	0,333	0,632	0,427
U3	DPSOMUT (ADDC)	0,577	0,259	0,350	0,512	0,842	0,622
	K-means	0,235	0,390	0,293	0,739	0,577	0,648
	Agglomerative	0,380	0,247	0,299	0,613	0,748	0,674
	PSO (GS)	0,242	0,251	0,246	0,747	0,739	0,743
	DPSOMUT (GS)	0,246	0,227	0,236	0,749	0,767	0,757
	IDPSOMUT (GS)	0,427	0,315	0,323	0,679	0,787	0,722
	Cuckoo Search	0,465	0,242	0,318	0,517	0,745	0,610
	Discrete ABC	0,284	0,600	0,392	0,791	0,547	0,643
	IDPSOMUT/CS	0,323	0,284	0,303	0,731	0,766	0,748
	IDPSOMUT (ADDC)	0,200	0,163	0,172	0,711	0,730	0,715
	DPSOMUT (ADDC)	0,559	0,300	0,364	0,522	0,763	0,606

For the unbalanced dataset with two classes (Figure 6.2), most of the algorithms remained unstable, that is, they did not obtain good performance due to their variation. The best *means* were achieved by the DPSOMUT (GS and ADDC), followed by the IDPSOMUT (GS and ADDC).

The *P*-value obtained from the Friedman test was equal to 0.988 (Figure 6.3), indicating that the algorithms are statistically equal for this corpus. However, the IDPSOMUT (GS) and DPSOMUT (GS) presented best overall performance for the IMDb corpus, since they reported the best *means* within the compared algorithms.

With this analysis, there is evidence that PSO-based algorithms, specifically those that use a mutation mechanism, are more suitable for this corpus. This also indicates that the main criteria to obtain good results for the IMDb corpus is the type of search, and the mechanism used to avoid local minimums.

6.5.2 Brazilian Portuguese OBCC

Analyzing the box-plots presented in Figure 6.4, Figure 6.5, Figure 6.6, and Figure 6.7, it is observed that, for the unbalanced datasets, the algorithms presented quite different *means*, which may indicate that the data distribution is complex, that is, the search space has many local minimums. Because of this, different strategies generate different results.

It can also be observed that the accuracy variance is fairly low, except for the K-means algorithm. One explanation for this is that different types of reboots can trap K-means in several

Table 6.9: Results by class for OBCC corpus using n-gram pre-processing

Class Balancing	Algorithm	Positive			Negative			Neutral		
		Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
B2	K-means	0,714	0,030	0,058	0,505	0,988	0,668			
	Agglomerative	1,000	0,506	0,672	0,024	1,000	0,047			
	PSO (GS)	0,513	0,272	0,355	0,505	0,743	0,601			
	DPSOMUT (GS)	0,525	0,227	0,316	0,506	0,794	0,618			
	IDPSOMUT (GS)	0,270	0,911	0,415	0,973	0,572	0,720			
	Cuckoo Search	0,474	0,494	0,484	0,513	0,494	0,503			
	Discrete ABC	0,530	0,530	0,549	0,530	0,530	0,530			
	IDPSOMUT/CS	0,321	0,477	0,383	0,648	0,488	0,557			
	IDPSOMUT (ADDC)	0,509	0,587	0,545	0,642	0,567	0,602			
	DPSOMUT (ADDC)	0,621	0,319	0,356	0,535	0,867	0,655			
U2	K-means	0,091	0,481	0,113	0,906	0,516	0,508			
	Agglomerative	1,000	0,114	0,204	0,004	1,000	0,008			
	PSO (GS)	0,121	0,270	0,167	0,889	0,748	0,812			
	DPSOMUT (GS)	0,118	0,252	0,161	0,888	0,760	0,819			
	IDPSOMUT (GS)	0,520	0,440	0,475	0,914	0,938	0,926			
	Cuckoo Search	0,488	0,112	0,182	0,508	0,886	0,646			
	Discrete ABC	0,133	0,572	0,216	0,906	0,528	0,664			
	IDPSOMUT/CS	0,371	0,112	0,171	0,623	0,886	0,732			
	IDPSOMUT (ADDC)	0,364	0,100	0,157	0,584	0,878	0,702			
	DPSOMUT (ADDC)	0,167	0,355	0,227	0,904	0,780	0,836			
B3	K-means	0,328	0,952	0,488	0,111	0,006	0,011	0,250	0,012	0,023
	Agglomerative	0,994	0,337	0,504	0,000	0,000	0,000	0,018	1,000	0,036
	PSO (GS)	0,343	0,228	0,273	0,315	0,207	0,249	0,338	0,567	0,423
	DPSOMUT (GS)	0,352	0,226	0,275	0,342	0,214	0,263	0,337	0,584	0,428
	IDPSOMUT (GS)	0,075	0,061	0,067	0,932	0,327	0,482	0,030	0,365	0,023
	Cuckoo Search	0,298	0,185	0,228	0,360	0,208	0,264	0,329	0,197	0,246
	Discrete ABC	0,362	0,572	0,435	0,366	0,560	0,437	0,000	0,000	0,000
	IDPSOMUT/CS	0,191	0,144	0,164	0,575	0,276	0,373	0,289	0,190	0,229
	IDPSOMUT (ADDC)	0,121	0,085	0,100	0,434	0,214	0,286	0,273	0,158	0,200
	DPSOMUT (ADDC)	0,366	0,266	0,267	0,345	0,252	0,281	0,353	0,576	0,442
U3	K-means	0,086	0,970	0,159	0,653	0,931	0,789	0,833	0,873	0,400
	Agglomerative	0,994	0,084	0,156	0,000	0,000	0,000	0,009	0,833	0,018
	PSO (GS)	0,082	0,224	0,120	0,641	0,207	0,313	0,273	0,567	0,369
	DPSOMUT (GS)	0,083	0,221	0,120	0,641	0,206	0,312	0,275	0,575	0,372
	IDPSOMUT (GS)	0,083	0,221	0,120	0,641	0,206	0,312	0,275	0,575	0,372
	Cuckoo Search	0,336	0,043	0,077	0,344	0,484	0,402	0,335	0,161	0,218
	Discrete ABC	0,089	0,367	0,146	0,663	0,357	0,447	0,308	0,382	0,341
	IDPSOMUT/CS	0,212	0,035	0,060	0,574	0,581	0,577	0,259	0,173	0,208
	IDPSOMUT (ADDC)	0,400	0,051	0,091	0,347	0,488	0,406	0,322	0,158	0,212
	DPSOMUT (ADDC)	0,110	0,265	0,139	0,660	0,227	0,337	0,282	0,593	0,390

Table 6.10: Results by class for Stanford Sentiment140 corpus using n-gram pre-processing

Class	Algorithm	Precision	Positive Recall	F-score	Precision	Negative Recall	F-score	Precision	Neutral Recall	F-score
Balancing										
B2	K-means	0,516	0,964	0,663	0,654	0,835	0,624			
	Agglomerative	0,928	0,487	0,639	0,022	0,231	0,039			
	PSO (GS)	0,479	0,255	0,332	0,493	0,723	0,586			
	DPSOMUT (GS)	0,493	0,217	0,301	0,498	0,776	0,606			
	IDPSOMUT (GS)	0,970	0,571	0,719	0,250	0,890	0,389			
	Cuckoo Search	0,530	0,501	0,515	0,473	0,502	0,487			
	Discrete ABC	0,500	0,576	0,565	0,500	0,568	0,444			
	IDPSOMUT/CS	0,708	0,531	0,607	0,374	0,561	0,449			
	IDPSOMUT (ADDC)	0,586	0,521	0,551	0,460	0,527	0,490			
	DPSOMUT (ADDC)	0,272	0,310	0,296	0,755	0,783	0,779			
U2	K-means	0,556	0,896	0,640	0,508	0,864	0,635			
	Agglomerative	0,945	0,497	0,652	0,017	0,231	0,032			
	PSO (GS)	0,502	0,253	0,336	0,492	0,743	0,591			
	DPSOMUT (GS)	0,505	0,221	0,307	0,492	0,776	0,602			
	IDPSOMUT (GS)	0,970	0,571	0,719	0,250	0,890	0,389			
	Cuckoo Search	0,531	0,513	0,522	0,482	0,500	0,491			
	Discrete ABC	0,500	0,555	0,544	0,524	0,599	0,514			
	IDPSOMUT/CS	0,643	0,509	0,568	0,360	0,493	0,416			
	IDPSOMUT (ADDC)	0,620	0,556	0,585	0,490	0,558	0,519			
	DPSOMUT (ADDC)	0,523	0,302	0,330	0,517	0,802	0,603			
B3	K-means	0,444	0,928	0,493	0,409	0,842	0,459	0,444	0,863	0,480
	Agglomerative	0,935	0,335	0,493	0,022	0,150	0,038	0,022	0,333	0,041
	PSO (GS)	0,339	0,205	0,255	0,344	0,216	0,265	0,335	0,593	0,428
	DPSOMUT (GS)	0,319	0,182	0,231	0,346	0,194	0,248	0,333	0,623	0,434
	IDPSOMUT (GS)	0,998	0,341	0,508	0,069	0,064	0,066	0,007	0,234	0,007
	Cuckoo Search	0,355	0,205	0,260	0,299	0,185	0,229	0,330	0,198	0,247
	Discrete ABC	0,393	0,432	0,401	0,355	0,388	0,383	0,388	0,453	0,437
	IDPSOMUT/CS	0,524	0,261	0,349	0,242	0,168	0,198	0,278	0,185	0,222
	IDPSOMUT (ADDC)	0,428	0,217	0,288	0,183	0,124	0,148	0,312	0,184	0,231
	DPSOMUT (ADDC)	0,409	0,266	0,309	0,370	0,223	0,282	0,340	0,655	0,455
U3	K-means	0,667	0,874	0,497	0,371	0,949	0,509	0,385	0,928	0,416
	Agglomerative	0,923	0,360	0,518	0,011	0,091	0,020	0,022	0,333	0,041
	PSO (GS)	0,354	0,196	0,252	0,369	0,212	0,269	0,278	0,591	0,378
	DPSOMUT (GS)	0,360	0,187	0,246	0,375	0,200	0,260	0,280	0,622	0,386
	IDPSOMUT (GS)	0,997	0,393	0,564	0,205	0,205	0,205	0,004	0,394	0,004
	Cuckoo Search	0,351	0,229	0,277	0,335	0,216	0,263	0,305	0,153	0,204
	Discrete ABC	0,416	0,401	0,398	0,413	0,401	0,399	0,350	0,446	0,392
	IDPSOMUT/CS	0,584	0,299	0,395	0,290	0,206	0,240	0,149	0,091	0,112
	IDPSOMUT (ADDC)	0,243	0,152	0,187	0,256	0,152	0,190	0,173	0,082	0,111
	DPSOMUT (ADDC)	0,092	0,271	0,139	0,671	0,233	0,344	0,288	0,582	0,390

Table 6.11: Difference between natural language and n-gram pre-processing techniques

Class Balancing	Corpus	Algorithm	Natural Language	N-gram	Difference
B2	OBCC	K-means	0.523	0.494	-0.028
		Agglomerative	0.552	0.476	-0.076
		PSO (GS)	0.519	0.507	-0.012
		DPSOMUT (GS)	0.509	0.510	0.001
		IDPSOMUT (GS)	0.790	0.654	-0.136
	IMDb Movie Review	K-means	0.502	0.514	0.012
		Agglomerative	0.483	0.528	0.045
		PSO (GS)	0.497	0.494	-0.003
		DPSOMUT (GS)	0.498	0.495	-0.003
		IDPSOMUT (GS)	0.359	0.542	0.183
	Sentiment140	K-means	0.499	0.488	-0.011
		Agglomerative	0.514	0.471	-0.043
		PSO (GS)	0.487	0.489	0.002
		DPSOMUT (GS)	0.491	0.482	-0.009
		IDPSOMUT (GS)	0.269	0.658	0.390
U2	OBCC	K-means	0.467	0.561	0.093
		Agglomerative	0.156	0.229	0.074
		PSO (GS)	0.696	0.694	-0.002
		DPSOMUT (GS)	0.701	0.694	-0.007
		IDPSOMUT (GS)	0.785	0.637	-0.149
	IMDb Movie Review	K-means	0.522	0.508	-0.014
		Agglomerative	0.574	0.333	-0.241
		PSO (GS)	0.621	0.617	-0.004
		DPSOMUT (GS)	0.629	0.650	0.021
		IDPSOMUT (GS)	0.374	0.558	0.184
	Sentiment140	K-means	0.496	0.497	0.001
		Agglomerative	0.370	0.524	0.153
		PSO (GS)	0.489	0.494	0.005
		DPSOMUT (GS)	0.489	0.485	-0.004
		IDPSOMUT (GS)	0.277	0.655	0.378
B3	OBCC	K-means	0.330	0.343	0.012
		Agglomerative	0.275	0.325	0.050
		PSO (GS)	0.334	0.334	0
		DPSOMUT (GS)	0.337	0.338	0.001
		IDPSOMUT (GS)	0.324	0.368	0.044
	Sentiment140	K-means	0.312	0.314	0.002
		Agglomerative	0.293	0.422	0.129
		PSO (GS)	0.335	0.338	0.003
		DPSOMUT (GS)	0.333	0.335	0.002
		IDPSOMUT (GS)	0.283	0.331	0.048
U3	OBCC	K-means	0.357	0.307	-0.050
		Agglomerative	0.254	0.092	-0.162
		PSO (GS)	0.334	0.307	-0.027
		DPSOMUT (GS)	0.314	0.314	0.000
		IDPSOMUT (GS)	0.423	0.504	0.082
	Sentiment140	K-means	0.332	0.355	0.023
		Agglomerative	0.319	0.239	-0.080
		PSO (GS)	0.335	0.312	-0.023
		DPSOMUT (GS)	0.313	0.343	0.030
		IDPSOMUT (GS)	0.316	0.472	0.156

local minimums.

The swarm-based algorithms did not have this difficulty, as they presented low variance for all datasets and similar performance, excepting the IDPSOMUT (GS), which reported the best *means* in three of the four datasets, as well as the smallest variances. One possible explanation for its best result compared to other swarm-based techniques is that this algorithm has the ability to perform local searches, refining the best found solutions.

The *P*-value obtained from the Friedman test was equal to 0.018, indicating that the algorithms are statistically different for this corpus. Also, with the Nemenyi post-test, it demonstrates that two groups of algorithms are statistically equal. These are represented by the blue and green lines to the right. According to Figure 6.8, it is highlighted that only one algorithm, the IDPSOMUT (GS), presented statistical difference from the others (as represented by the green line).

6.5.3 *Stanford Sentiment140*

According to the box-plots presented in Figure 6.9, Figure 6.10, Figure 6.11, and Figure 6.12, two points are easily observable: the first is that the algorithms have similar performance. For the datasets with two classes, accuracy is closer to 0.5, and, for the ones with three classes, the accuracy are closer to 0.3. This leads us to believe that the search space for this corpus is very uniform, causing all the algorithms to land in the same region.

The second point is related to the IDPSOMUT (GS) accuracy. It presents the best *means* for all datasets, distancing itself enough from the accuracy of the other algorithms, despite the *P*-value of 0.086 (Figure 6.13) obtained from Friedman. This value indicates that the algorithms are statistically equal for this corpus.

Similar to what was obtained for the OBCC corpus, it is believed that the significant performance of the IDPSOMUT (GS) is related to its various search engines, such as its ability to switch between exploration and exploitation, as well as with the mutation mechanism.

6.5.4 *General Performance Analysis*

From the presented box-plots, it was possible to observe that the swarm-based algorithms, particularly the PSO-based ones, were able to find better solutions than conventional clustering techniques, such as K-means and Agglomerative.

This statement can be confirmed in Table 6.12, which shows the accuracy for all evaluated algorithms. The IDPSOMUT (CS) reported the best accuracy in seven out of ten experiments, followed by the IDPSOMUT/CS, the DPSOMUT (GS) and IDPSOMUT (ADDC) algorithms.

In order to provide a more confident analysis, assuming that data do not follow a *normal distribution*, Friedman's non-parametric test was used together with the Nemenyi *post-hoc* to evaluate if the group of compared algorithms has equal performance. The test was performed using the Matlab tool with a 95% confidence level.

Figure 6.14 shows the results obtained from the statistical tests. The central line presents the Friedman rank. According to the rank, the IDPSOMUT (GS) showed better performance than the other algorithms. The P -value obtained from Friedman test was equal to 0.012, indicating that this value is in the critical region, which confirms that the null hypothesis, that the algorithms are statistically equal, was rejected.

Finally, with the Nemenyi post-test, it was observed that two groups of algorithms are statistically equal. These are represented by the blue and green lines to the right. According to Figure 6.14, it is highlighted that only one algorithm, the IDPSOMUT, presented statistical difference from the others (from the green line), thus confirming its efficiency and competitiveness in opinion clustering.

6.6 Comparison between GS and ADDC Fitness Functions

For the non-hybrid PSO-based algorithms, two ICVMs widely used in short-text and document clustering were also compared: The GS, which combines cohesion and separation concepts to determine the quality of a given clustering and ADDC that computes the value of the average distance between documents and the cluster centroid to which they belong.

According to the box-plots presented previous section, the GS fitness function achieved better accuracy than ADDC in almost all datasets. The GS also presented smaller variances. This results confirm the finds of [CAGNINA et al. \(2014\)](#) that the GS is more suitable for short-texts such as papers' abstracts and opinions. One possible explanations for the ADDC' poor performance is the complex data distribution, which made it difficult also for the algorithms based on centroid metrics such as the K-means.

6.7 Comparison with Similar Studies

Table 6.13 presents a performance analysis of the three opinion mining approaches presented in Figure 2.1 for the corpora used in this Thesis. This table does not present a comparative analysis. Instead, it shows the achieved results from other approaches which have used the same corpora as ours. Column one presents the average accuracy obtained by supervised ML and lexicon-based approaches, column two presents the results for an hybrid approach that uses supervised ML and clustering, while the third column shows the average accuracy for the unsupervised approaches detailed in Section 2.3 and Table 2.1.

Only the studies of [LI; LIU \(2010, 2012, 2014\)](#) make use of clustering techniques for OM, as the methods proposed in this Thesis. The authors evaluated their results using the IMDb corpus. The best accuracy for this corpora was 63.3%, reached by the IDPSOMUT/CS algorithm, using n-gram pre-processing. The mentioned studies used language-dependent pre-processing, unlike the one proposed in this Thesis.

For the OBCC corpus, best precision of 66% and f-score of 40% using ML and opinion lexicon was obtained by [SOUZA et al. \(2016\)](#). The proposed PSO-based approaches outperformed these results, achieving best precision and f-score of 72% and 70%, respectively. [TELES; SANTOS; SOUZA \(2016\)](#) also evaluated their result with the OBCC corpus using a supervised ML smoothed technique and obtaining best accuracy of 94%, outperforming the proposed clustering approach (accuracy of 87%).

For the Sentiment140 corpus, the best accuracy (61.50%) was achieved using a word bigram pre-processing. The average accuracy obtained for this corpus using supervised ML achieved accuracy ranged between 61.77% and 71.79% ([ABBASI; HASSAN; DHAR, 2014](#)). This corpus was also evaluated using an approach that combines supervised ML, opinion lexicon and clustering algorithm as shown in Table 6.13.

6.8 Final Remarks

As detailed in Section 6.4, it could be observed that the word n-gram pre-processing seems to be more suitable for the opinion clustering task, especially when dealing with very short text such as *tweets*. This is a significant result towards the construction of domain-independent opinion mining applications, since the text language is also considered a domain adaptation problem ([PANG; LEE, 2008](#)).

According to the explanations in Section 6.5, the proposed swarm-based algorithms outperformed the clustering algorithms in all experiments with n-gram pre-processing, indicating that the proposed algorithms are more suitable for this search space. It also confirms the findings of [CAGNINA; ERRECALDE; INGARAMO \(2008\)](#); [CAGNINA et al. \(2014\)](#); [INGARAMO et al. \(2009, 2011\)](#), which have successfully applied PSO-based algorithms for the short-text clustering task.

As observed in Section 6.6, similar to the algorithms that uses centroid metrics such as the K-means, the ADDC ICVM perform worse than the GS in almost all experiments with n-gram pre-processing, confirming also the findings of [CAGNINA; ERRECALDE; INGARAMO \(2008\)](#); [CAGNINA et al. \(2014\)](#); [INGARAMO et al. \(2009, 2011\)](#) which have successfully applied the GS measure as optimization function for the short-text clustering task.

Although the proposed swarm-based methods did not outperform current supervised OM methods and even some unsupervised methods, particularly for the datasets with three classes, they still very competitive as no labeled data, nether opinion lexicons, both very time consuming, are required for the analysis of opinions. Also, the proposed methods based on n-gram pre-processing has the advantage of being domain-independent.

Figure 6.1: Box-plots for the IMDb corpus with two balanced classes

Table 6.12: Average accuracy obtained by all algorithms using n-gram pre-processing

Corpus	Class	K-means	Agglo.	PSO (GS)	DPSOMUT (GS)	IDPSOMUT (GS)	CS	ABC	IDPSOMUT/CS	IDPSOMUT (ADDC)	DPSOMUT (ADDC)
IMDb	B2	0.504	0.528	0.494	0.497	0.497	0.500	0.498	0.475	0.564	0.489
	U2	0.500	0.555	0.617	0.632	0.616	0.507	0.505	0.633	0.584	0.629
OBCC	B2	0.500	0.512	0.507	0.510	0.622	0.508	0.495	0.498	0.516	0.512
	U2	0.512	0.117	0.694	0.703	0.870	0.512	0.501	0.607	0.559	0.704
	B3	0.334	0.337	0.334	0.341	0.347	0.339	0.325	0.362	0.335	0.335
	U3	0.279	0.084	0.307	0.308	0.600	0.339	0.309	0.449	0.410	0.309
Sentiment	B2	0.494	0.475	0.489	0.497	0.615	0.495	0.502	0.521	0.523	0.489
	U2	0.488	0.487	0.494	0.495	0.615	0.496	0.505	0.495	0.487	0.487
	B3	0.334	0.326	0.338	0.333	0.357	0.324	0.331	0.357	0.308	0.331
	U3	0.346	0.347	0.312	0.313	0.438	0.333	0.332	0.328	0.340	0.317

Figure 6.2: Box-plots for the IMDb corpus with two unbalanced classes

Table 6.13: Average accuracy for document level polarity classification related works

Corpus	Supervised ML and Lexicon-based	Supervised and unsupervised ML	Class balancing	Unsupervised (related work)	Proposed methods (nlp)	Proposed methods (w-bigram)
OBCC	66.00% - 94.87% (TELES; SANTOS; SOUZA, 2016)	n/a	B2	n/a	79.00%	62.20%
			U2	n/a	73.20%	87.00%
			B3	n/a	37.10%	34.70%
			U3	n/a	52.70%	60.00%
IMDb	65.90% - 88.88% (TRIPATHY; AGRAWAL; RATH, 2016)	n/a	B2	88.00% - 89.67% (LI; LIU, 2014)	49.10%	56.40%
			U2	n/a	62.90%	63.30%
Sentiment140	61.77% - 71.79% (ABBASI; HASSAN; DHAR, 2014)	77.69% - 81.84% (COLETTA et al., 2014)	B2	n/a	49.10%	61.50%
			U2	n/a	48.90%	61.50%
			B3	n/a	42.20%	35.70%
			U3	n/a	42.50%	43.80%

Figure 6.3: Friedman test for the IMDb corpus

Figure 6.4: Box-plots for the OBCC corpus with two balanced classes

Figure 6.5: Box-plots for the OBCC corpus with two unbalanced classes

Figure 6.6: Box-plots for the OBCC corpus with three balanced classes

Figure 6.7: Box-plots for the OBCC corpus with three unbalanced classes

Figure 6.8: Friedman test for the OBCC corpus

Figure 6.9: Box-plots for the Sentiment140 corpus with two balanced classes

Figure 6.10: Box-plots for the Sentiment140 corpus with two unbalanced classes

Figure 6.11: Box-plots for the Sentiment140 corpus with three balanced classes

Figure 6.12: Box-plots for the Sentiment140 corpus with three unbalanced classes

Figure 6.13: Friedman test for the Sentiment140 corpus

Figure 6.14: Friedman test for all evaluated algorithms (accuracy)

7

CONCLUSION

This Chapter presents the conclusions of the Thesis (Section 7.1), highlights future work (Section 7.2), and lists the published papers resulting from the research work (Section 7.3).

7.1 Conclusion

This Thesis presented and evaluated three swarm-based algorithms for opinion clustering: the DPSOMUT which is based on a discrete PSO binary version, the IDPSOMUT that is based on an Improved Self-Adaptive PSO algorithm with detection function, and the IDPSOMUT/CS that is a hybrid version of IDPSOMUT and Cuckoo Search (CS).

The PSO-based algorithms consider clustering as an optimization problem, where a given arbitrary objective function must be optimized. In this Thesis, the performance of two internal clustering validity measures widely used in document clustering as fitness function were evaluated: Global Silhouette (GS) coefficient and the Average Distance Documents to the Cluster Centroid (ADDC).

Three different corpora with different levels of complexity with respect to size, number of opinions, domains, languages, and class balancing were selected to assess the results: the IMDb ([PANG; LEE; VAITHYANATHAN, 2002](#)) corpus which contains movie reviews written in the English language; the Sentiment140 ([GO; BHAYANI; HUANG, 2010](#)) and the OBCC corpora contain *tweets* written in English and Brazilian Portuguese languages, respectively, with opinions about brands, products, politicians, celebrities, and other topics from Twitter.

Considering the absence of benchmark data-sets for the Portuguese language and the initial stage of OM research in Portuguese, this Thesis also presents and makes it publicly available the OBCC corpus, containing *tweets* in Brazilian Portuguese with opinions of consumers about products and services.

Two text pre-processing approaches were evaluated: a natural language pre-processing which applied a part-of-speech (POS) tagger for feature selection, and a word bigram feature model. K-means and Agglomerative clustering algorithms, as well as, PSO, ABC, and CS

swarm-based algorithms were selected for comparison. Results were expressed in terms of accuracy, precision, recall, and F-score.

From the statistical analysis, it was possible to observe that the swarm-based algorithms, especially the PSO ones, were able to find better solutions than conventional grouping techniques, such as K-means and Agglomerative. The PSO-based algorithms achieved better accuracy using a word bigram pre-processing and the GS as fitness function.

Best results were obtained on datasets with two classes (positive and negative). Best accuracy (87%) was obtained by IDPSOMUT on the OBCC corpus. Although the PSO-based algorithms obtained lower result for datasets with three classes compared to the others related works, they are still competitive considering that in this case neither labeled data, nor opinion lexicons are required for the opinion clustering approach.

It is worth highlighting that this study presents limitations related to Twitter' corpora. Symbols and images from *tweets* which may help to identify its polarity (TRIPATHY; AGRAWAL; RATH, 2016) were not considered. Sarcasm and irony detection was not also handled.

Thus, the contributions of this Thesis are as follows:

1. A clustering based approach for opinion mining based on swarm intelligence;
2. Natural language and word n-gram techniques evaluation for pre-processing the clustering of opinions;
3. Different clustering and swarm-based algorithms such as K-means, Agglomerative, PSO, ABC, and CS used for comparison;
4. For the PSO proposed algorithms, two different internal clustering validity measures (ICVM), such as the Global Silhouette (GS) coefficient and the Average Distance Documents to the Cluster Centroid (ADDC), selected for comparison as fitness function,
5. Three corpora with different languages, class balancing, text size, data source, and several domains used for clustering purpose,
6. A corpus, the OBCC, containing a gold collection with 2,940 *tweets* in Brazilian Portuguese with opinions of consumers about products and services made publicly available, and
7. A systematic mapping study that plotted the landscape of current studies relating to the application of text mining in the Portuguese language up to year 2014.

7.2 Future Works

As there is no consensus whether higher-order n-grams are useful features for sentiment polarity (PANG; LEE, 2008), it is intended to evaluate other word n-gram feature models and their combinations. (TRIPATHY; AGRAWAL; RATH, 2016) reported accuracy of 88.94% for the IMDb corpus, using an approach which combines word unigram, bigram, and trigram feature models for binary (positive and negative) sentiment classification.

In order to improve the results for datasets with three classes (positive, negative and neutral), it is considered to apply a two-step clustering approach (PANG; LEE, 2008; LI; LIU, 2014). In the first step, subjective and objective (neutral) opinions are identified, then, in the second step, a binary clustering is performed only with the subjective opinions.

As future work, it is intended to improve results of the proposed algorithms by using semi-supervised techniques. The semi-supervised learning needs a small number of seeds for each group and this may improve the resulting clusters (CHENG; SHI; QIN, 2012; GU; LU, 2012). This type of machine learning uses unlabeled data to complement the information provided by the labeled data in the training process, which is particularly useful in applications including *tweet* sentiment analysis, where a huge quantity of unlabeled data is accessible (SILVA; COLETTA; HRUSCHKA, 2016). Another future work is to investigate clustering algorithms for high-dimensional search space.

It is also possible that a different number of classes may provide a better clustering solution. In this Thesis, it was only investigated two (positive, negative) and three classes (positive, negative and neutral). As future work it is intended to investigate different levels of positivity and negativity by increasing the number of classes.

Finally, it is intended to investigate and apply *concept drift* techniques in order to evaluate the sentiment or opinion variation over time, activity known as *Sentiment Drift* (SILVA et al., 2011; GUERRA; MEIRA JR.; CARDIE, 2014). The temporal information is very relevant as the topic of discussion may change, influencing on people's opinions. Concept drift deals with the evolving nature of the stream, which causes the target concept to change and requires learning models to be constantly updated (GUERRA; MEIRA JR.; CARDIE, 2014; LI et al., 2016; SONG et al., 2016). The proposed methods can be fairly competitive in streams analysis since they do not require training as the supervised methods.

7.3 Published Papers

7.3.1 Published Papers

1. Souza, E., Castro, D., Vitória, D., Teles, I., Oliveira, A.L.I., Gusmão, C.: *Characterizing User-Generated Text Content Mining: A Systematic Mapping Study of the Portuguese Language*. Proceedings of 4th World Conference on Information Systems and Tech-

nologies (WorldCIST'16). New Advances in Information Systems and Technologies, Springer International Publishing, pp. 1015–1024, 2016.

The results of a Systematic Mapping Study focused on mapping the use of text mining techniques to process user-generated text content (UGTC) in the Portuguese language were published in this paper.

2. Souza, E., Alves, T., Teles, I., Oliveira, A.L.I., Gusmão, C.: *TOPIE: an Open-source Opinion Mining Pipeline to Analyze Consumers' Sentiment in Brazilian Portuguese*. In Proceedings of 12h Computational Processing of the Portuguese Language (PROPOR'2016). Lecture Notes in Computer Science, Springer International Publishing, pp. 307-312, 2016.

The OBCC corpus and a hybrid OM open-source pipeline based on supervised ML and opinion lexicon were presented in this paper.

3. Souza, E., Vitório, D., Castro, D., Oliveira, A.L.I., Gusmão, C.: *Characterizing Opinion Mining: a Systematic Mapping Study of the Portuguese Language*. In Proceedings of 12h Computational Processing of the Portuguese Language (PROPOR'2016). Lecture Notes in Computer Science, Springer International Publishing, pp. 122–127, 2016.

The results of a Systematic Mapping Study focused on mapping a specific text classification sub-task, Opinion Mining, for the Portuguese language were published in this paper.

4. Souza, E., Oliveira, A. L. I., Silva, A., Oliveira, G., Santos, D.: *An Unsupervised Particle Swarm Optimization Approach for Opinion Clustering*. In Proceedings of 5th Brazilian Conference on Intelligent Systems (BRACIS'16), pp. 307 -312, 2016.

The results for the first two swarm-based methods (DPSOMUT and IDPSOMUT) applied to the clustering of opinion task using only natural language pre-processing were published in this paper.

7.3.2 Submitted Papers

1. Souza, E., Costa, D., Castro, D. W., Vitório, D., Teles, I., Almeida, R., Alves, T., Oliveira, A.L.I., Gusmão, C.: *Characterising Text Mining: a Systematic Mapping Review of the Portuguese Language*. IET Software Journal, 2016.

We received an invitation to submit an extended version of *Characterizing User-Generated Text Content Mining: A Systematic Mapping Study of the Portuguese Language* paper to the IET Software Journal in April, 2016. The extended paper was submitted in August, 2016. The paper was accepted with corrections in October and a new version with the requested corrections was submitted in December.

The extended paper differs from published previous paper as it maps not only the use of text mining techniques for user-generated text content in the Portuguese language, but for all kind of texts

2. Souza, E., Oliveira, A. L. I., Silva, A., Oliveira, G., Santos, D.: *Swarm Optimization Clustering Methods for Opinion Mining*. Natural Computing Journal, 2017.

We received an invitation to submit an extended version of the paper *An Unsupervised Particle Swarm Optimization Approach for Opinion Clustering* to the Natural Computing Journal in November, 2016. The extended version was submitted in January, 2017.

All the results presented in this Thesis involving the three swarm-based approaches, the two pre-processing techniques, and the GS and ADDC fitness functions evaluation are detailed in this paper.

References

- ABBASI, A.; HASSAN, A.; DHAR, M. Benchmarking Twitter Sentiment Analysis Tools. In: LREC-2014, THE NINTH INTERNATIONAL CONFERENCE ON LANGUAGE RESOURCES AND EVALUATION. **Proceedings...** [S.l.: s.n.], 2014. n.March, p.823–829.
- NEDJAH, N.; MOURELLE, L. d. M. (Ed.). **Swarm Intelligence: foundations, perspectives and applications**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. 3–25p.
- ANCHIETA, R. T.; ASSIS, F.; NETO, R. Using Stylometric Features for Sentiment Classification. **Lecture Notes in Computer Science**, [S.l.], p.189–200, 2015.
- AVANCO, L. V.; NUNES, M. D. G. V. Lexicon-Based Sentiment Analysis for Reviews of Products in Brazilian Portuguese. **Brazilian Conference on Intelligent Systems**, [S.l.], p.277–281, 2014.
- BALAZS, J. A.; VELÁSQUEZ, J. D. Opinion Mining and Information Fusion: a survey. **Information Fusion**, [S.l.], v.27, p.95–110, 2016.
- BOSMAN, P.; ENGELBRECHT, A. P. Diversity Rate of Change Measurement for Particle Swarm Optimisers. **Swarm Intelligence, Lecture Notes in Computer Science**, [S.l.], p.86–97, 2014.
- BUDGEN, D. et al. Using Mapping Studies in Software Engineering. **Proceedings of Psychology of Programming Interest (PPIG) Group**, [S.l.], p.195–204, 2008.
- CAGNINA, L. C.; ERRECALDE, M. L.; INGARAMO, D. A. A Discrete Particle Swarm Optimizer for clustering short-text corpora. **Proceedings of International Conference on Bioinspired Optimization Methods and their Applications, BIOMA 2008**, [S.l.], p.1–10, 2008.
- CAGNINA, L. et al. An efficient Particle Swarm Optimization approach to cluster short texts. **Information Sciences**, [S.l.], v.265, p.36–49, 2014.
- CALDERON, N. A. et al. Mixed-Initiative Social Media Analytics at the World. **IEEE International Conference on Big Data**, [S.l.], p.1678–1687, 2015.
- CHAVES, M. S. et al. PIRPO: an algorithm to deal with polarity in portuguese online reviews from the accommodation sector. In: INTERNATIONAL CONFERENCE ON APPLICATIONS OF NATURAL LANGUAGE PROCESSING AND INFORMATION SYSTEMS, 17., Berlin, Heidelberg. **Proceedings...** Springer-Verlag, 2012. p.296–301. (NLDB'12).
- CHENG, S.; SHI, Y.; QIN, Q. Particle swarm optimization based semi-supervised learning on Chinese text categorization. In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION, 2012. **Anais...** [S.l.: s.n.], 2012. p.1–8.
- COLETTA, L. F. S. et al. Combining Classification and Clustering for Tweet Sentiment Analysis. In: BRAZILIAN CONFERENCE ON INTELLIGENT SYSTEMS. **Anais...** [S.l.: s.n.], 2014. p.210–215.
- COMSCORE. **ComScore**: cross-platform measurement company. 2016.

- CUI, X.; POTOK, T. E. Document Clustering Analysis Based on Hybrid PSO+K-means Algorithm. **Special Issue**, [S.l.], p.27–33, 2005.
- CUI, X.; POTOK, T. E.; PALATHINGAL, P. Document clustering using particle swarm optimization. **Proceedings 2005 IEEE Swarm Intelligence Symposium**, [S.l.], p.185–191, 2005.
- DELEN, D.; CROSSLAND, M. D. Seeding the survey and analysis of research literature with text mining. **Expert Systems with Applications**, [S.l.], v.34, p.1707–1720, 2008.
- EBERHART, R.; KENNEDY, J. A New Optimizer Using Particle Swarm Theory. **Sixth International Symposium on Micro Machine and Human Science**, [S.l.], p.39–43, 1995.
- ENRÍQUEZ, F. et al. A Comparative Study of Classifier Combination Applied to NLP Tasks. **Inf. Fusion**, [S.l.], v.14, n.3, p.255–267, 2013.
- EVANGELISTA, T. R.; PADILHA, T. P. P. Monitoramento de Posts Sobre Empresas de E-Commerce em Redes Sociais Utilizando Análise de Sentimentos. **Brazilian Workshop on Social Network Analysis and Mining (BraSNAM)**, [S.l.], 2013.
- FELDMAN, R. Techniques and Applications for Sentiment Analysis. **Commun. ACM**, New York, NY, USA, v.56, n.4, p.82–89, Apr. 2013.
- FELDMAN, R.; DAGAN, I. Knowledge Discovery in Textual Databases (KDT). **International Conference on Knowledge Discovery and Data Mining (KDD)**, [S.l.], p.112–117, 1995.
- FILHO, T. M. S. et al. Hybrid methods for fuzzy clustering based on fuzzy c-means and improved particle swarm optimization. **Expert Systems with Applications**, [S.l.], v.42, p.6315 – 6328, 2015.
- FLEISS, J. L. Measuring nominal scale agreement among many raters. **Psychological Bulletin**, [S.l.], v.76, n.5, p.378—382, 1971.
- FOULADGAR, N.; LOTFI, S. A novel approach for optimization in dynamic environments based on modified cuckoo search algorithm. **Soft Computing**, [S.l.], p.1–15, 2015.
- FREITAS, L. A. de; VIEIRA, R. Ontology-based feature level opinion mining for portuguese reviews. In: INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 22. **Proceedings...** [S.l.: s.n.], 2013. p.367–370.
- FREITAS, L. de; VIEIRA, R. Exploring Resources for Sentiment Analysis in Portuguese Language. In: BRAZILIAN CONFERENCE ON INTELLIGENT SYSTEMS. **Anais...** [S.l.: s.n.], 2015. p.152–156.
- GO, A.; BHAYANI, R.; HUANG, L. **Twitter Sentiment Classification using Distant Supervision**. [S.l.: s.n.], 2010.
- GOLDEN, P. **Write here, write now**. 2011.
- GRACIELA, A.; BECKER, K.; MOREIRA, V. Um estudo de caso de mineração de emoções em textos multilíngues. In: BRAZILIAN WORKSHOP ON SOCIAL NETWORK ANALYSIS AND MINING (BRASNAM). **Anais...** [S.l.: s.n.], 2014.

- GU, L.; LU, X. Semi-supervised subtractive clustering by seeding. In: INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS AND KNOWLEDGE DISCOVERY, 2012. **Anais...** [S.l.: s.n.], 2012. p.738–741.
- GUERRA, P. C.; MEIRA JR., W.; CARDIE, C. Sentiment Analysis on Evolving Social Streams: how self-report imbalances can help. In: ACM INTERNATIONAL CONFERENCE ON WEB SEARCH AND DATA MINING, 7., New York, NY, USA. **Proceedings...** ACM, 2014. p.443–452. (WSDM '14).
- GUPTA, V.; LEHAL, G. S. A Survey of Text Mining Techniques and Applications. **Journal of Emerging Technologies in WEB Intelligence**, [S.l.], v.1, n.1, p.60–76, 2009.
- HOTHOTH, A. et al. A Brief Survey of Text Mining. **LDV Forum - GLDV Journal for Computational Linguistics and Language Technology**, [S.l.], p.1–37, 2005.
- HUANG, Y. Conceptually categorizing geographic features from text based on latent semantic analysis and ontologies. **Annals of GIS**, [S.l.], v.22, n.2, p.113–127, 2016.
- INGARAMO, D. et al. Particle Swarm Optimization for clustering short-text corpora. **Frontiers in Artificial Intelligence and Applications**, [S.l.], v.196, n.1, p.3–19, 2009.
- INGARAMO, D. et al. A particle swarm optimizer to cluster parallel spanish-english short-text corpora. **CEUR Workshop Proceedings**, [S.l.], v.824, n.1cl, p.43–48, 2011.
- J.HYMA; JHANSI, Y.; ANURUDHA, S. A new hybridized approach of PSO & GA for document clustering. **International Journal of Engineering Science and Technology**, [S.l.], v.2, n.5, p.1221–1226, 2010.
- KAISER, C.; PIAZZA, A. Are Humans Like Ants? Analyzing Collective Opinion Formation in Online Discussions. **Privacy, security, risk ...**, [S.l.], 2011.
- KAMEL, N.; OUCHEN, I.; BAALI, K. A Sampling-PSO-K-means Algorithm for Document Clustering. **Advances in Intelligent Systems and Computing**, [S.l.], v.388, p.405–411, 2016.
- KARABOGA, D.; BASTURK, B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. **Journal of global optimization**, [S.l.], v.39, n.3, p.459–471, 2007.
- KAROL, S.; MANGAT, V. Evaluation of text document clustering approach based on particle swarm optimization. **Central European Journal of Computer Science**, [S.l.], v.3, n.2, p.69–90, 2013.
- KENNEDY, J.; EBERHART, R. Particle Swarm Optimization. **Proceedings of IEEE International Conference on Neural Networks**, [S.l.], p.1942–1948, 1995.
- KENNEDY, J.; EBERHART, R. C. A Discrete Binary Version of The Particle Swarm Algorithm. **IEEE International Conference On Systems, Man, And Cybernetics**, [S.l.], p.4–8, 1997.
- KHANNA, K.; YADAV, M. L. A Swarm Based Approach to Improve Traditional Document Clustering Approach. **International Journal on Recent and Innovation Trends in Computing and Communication**, [S.l.], v.1, n.May, p.433–437, 2013.

- KITCHENHAM, B. A.; DYBA, T.; JORGENSEN, M. Evidence-based Software Engineering. **Proceedings of the 26th International Conference on Software Engineering**, [S.l.], p.273–281, 2004.
- KITCHENHAM, B.; CHARTERS, S. Guidelines for performing Systematic Literature Reviews in Software Engineering. **Technical Report EBSE-2007-01, School of Computer Science and Mathematics, Keele University**, [S.l.], 2007.
- KUSHAL, D.; LAWRENCE, S.; PENNOCK, D. M. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In: WWW. **Anais...** [S.l.: s.n.], 2003. p.519–528.
- LABOREIRO, G. et al. Determining Language Variant in Microblog Messages. **Proceedings of the 28th Annual ACM Symposium on Applied Computing**, [S.l.], p.902–907, 2013.
- LEE, I.; ON, B.-W. Role Based Particle Swarm Optimization Algorithm To Cluster Web Documents. **Proceedings of International Conference on Artificial Intelligence and Pattern Recognition (AIPR)**, [S.l.], 2010.
- LI, C.-T. et al. Exploiting concept drift to predict popularity of social multimedia in microblogs. **Information Sciences**, [S.l.], v.339, p.310 – 331, 2016.
- LI, G.; LIU, F. A Clustering-based Approach on Sentiment Analysis. **International Conference on Intelligent Systems and Knowledge Engineering (ISKE)**, [S.l.], p.331–337, 2010.
- LI, G.; LIU, F. Application of a clustering method on sentiment analysis. **Journal of Information Science**, [S.l.], v.38, n.2, p.127–139, 2012.
- LI, G.; LIU, F. Sentiment analysis based on clustering : a framework in improving accuracy and recognizing neutral opinions. **Applied Intelligence**, [S.l.], p.441–452, 2014.
- LIU, B.; ZHANG, L. A Survey Of Opinion Mining And Sentiment Analysis. **Mining Text Data**, [S.l.], v.Chapter 1, p.415–463, 2012.
- MACQUEEN, J. et al. Some methods for classification and analysis of multivariate observations. In: BERKELEY SYMPOSIUM ON MATHEMATICAL STATISTICS AND PROBABILITY. **Proceedings...** [S.l.: s.n.], 1967. v.1, n.14, p.281–297.
- MARINE-ROIG, E.; ANTON CLAVÉ, S. Tourism analytics with massive user-generated content: a case study of barcelona. **Journal of Destination Marketing & Management**, [S.l.], p.1–11, 2015.
- MARQUES-LUCENA, C.; SARRAIPA, J. Framework for Customers ' Sentiment Analysis. **Advances in Intelligent Systems and Computing**, [S.l.], p.849–860, 2015.
- MEDHAT, W.; HASSAN, A.; KORASHY, H. Sentiment analysis algorithms and applications: a survey. **Ain Shams Engineering Journal**, [S.l.], v.5, n.4, p.1093–1113, 2014.
- MIRJALILI, S.; LEWIS, A. The Whale Optimization Algorithm. **Advances in Engineering Software**, [S.l.], v.95, p.51 – 67, 2016.
- MOSTAFA, M. M. More than words : social networks ' text mining for consumer brand sentiments. **Expert Systems With Applications**, [S.l.], v.40, n.10, p.4241–4251, 2013.

OWOPUTI, O. et al. **Part-of-Speech Tagging for Twitter: word clusters and other advances**. [S.l.]: Carnegie Mellon University, 2012.

PAK, A.; PAROUBEK, P. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In **Proceedings of LREC**, [S.l.], p.1320–1326, 2010.

PANG, B.; LEE, L. Seeing Stars: exploiting class relationships for sentiment categorization with respect to rating scales. In: ANNUAL MEETING ON ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, 43. **Proceedings...** [S.l.: s.n.], 2005. p.115–124. (ACL '05).

PANG, B.; LEE, L. Opinion Mining and Sentiment Analysis. **Foundations and Trends® in Information Retrieval**, [S.l.], v.2, n.1–2, p.1–135, 2008.

PANG, B.; LEE, L.; VAITHYANATHAN, S. Thumbs up ? Sentiment Classification using Machine Learning Techniques. **Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)**, [S.l.], n.July, p.79–86, 2002.

PETERSEN, K. et al. Systematic Mapping Studies in Software Engineering. **Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering**, [S.l.], p.68–77, 2008.

PREMALATHA, K.; NATARAJAN, A. Discrete PSO with GA Operators for Document Clustering. **International Journal of Recent Trends in Engineering**, [S.l.], v.1, n.1, p.20–24, 2009.

PREMALATHA, K.; NATARAJAN, A. M. Hybrid PSO and GA models for document clustering. **International Journal of Advances in Soft Computing and its Applications**, [S.l.], v.2, n.3, p.302–320, 2010.

RAVI, K.; RAVI, V. A survey on opinion mining and sentiment analysis: tasks, approaches and applications. **Knowledge-Based Systems**, [S.l.], v.89, 2015.

ROSA, R. L.; RODRIGUEZ, D. Z.; BRESSAN, G. SentiMeter-Br: a social web analysis tool to discover consumers' sentiment. **Proceedings of IEEE International Conference on Mobile Data Management**, [S.l.], v.2, p.122–124, 2013.

ROSA, R. L.; RODRIGUEZ, D. Z.; BRESSAN, G. SentiMeter-Br: a new social web analysis metric to discover consumers' sentiment. **Proceedings of the International Symposium on Consumer Electronics, ISCE**, [S.l.], p.153–154, 2013.

SANTOS, F. L. dos; LADEIRA, M. The role of text pre-processing in opinion mining on a social media language dataset. In: BRAZILIAN CONFERENCE ON INTELLIGENT SYSTEMS (BRACIS). **Anais...** [S.l.: s.n.], 2014.

SARKAR, S.; ROY, A.; PURKAYASTHA, B. Application of Particle Swarm Optimization in Data Clustering: a survey. **International Journal of Computer Applications**, [S.l.], v.65, n.25, p.38–46, 2013.

SARKAR, S.; ROY, A.; PURKAYASTHA, B. Clustering of Documents using Particle Swarm Optimization and Semantics Information. **International Journal of Computer Science and Information Technologies**, [S.l.], v.5, n.3, p.4175–4180, 2014.

- SARKAR, S.; ROY, A.; PURKAYASTHA, B. S. A Comparative Analysis of Particle Swarm Optimization and K-means Algorithm For Text Clustering Using Nepali Wordnet. **International Journal on Natural Language Computing (IJNLC)**, [S.l.], v.3, n.3, p.83–92, 2014.
- SHI, G.; KONG, Y. Advances in Theories and Applications of Text Mining. **International Conference on Information Science and Engineering (ICISE2009)**, [S.l.], p.4167–4170, 2009.
- SILVA, I. S. et al. Effective Sentiment Stream Analysis with Self-augmenting Training and Demand-driven Projection. In: INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, 34., New York, NY, USA. **Proceedings...** ACM, 2011. p.475–484. (SIGIR '11).
- SILVA, N. F. F. D.; COLETTA, L. F. S.; HRUSCHKA, E. R. A Survey and Comparative Study of Tweet Sentiment Analysis via Semi-Supervised Learning. **ACM Comput. Surv.**, New York, NY, USA, v.49, n.1, p.15:1–15:26, June 2016.
- SONG, G. et al. Dynamic Clustering Forest: an ensemble framework to efficiently classify textual data stream with concept drift. **Information Sciences**, [S.l.], v.357, p.125 – 143, 2016.
- SOUZA, E. et al. Characterizing Opinion Mining: a systematic mapping study of the portuguese language. In: COMPUTATIONAL PROCESSING OF THE PORTUGUESE LANGUAGE (PROPOR'2016). LECTURE NOTES IN COMPUTER SCIENCE, SPRINGER INTERNATIONAL PUBLISHING, 12. **Proceedings...** Springer International Publishing, 2016. p.122–127.
- SOUZA, E. et al. TOPIE: an open-source opinion mining pipeline to analyze consumers sentiment in brazilian portuguese. In: COMPUTATIONAL PROCESSING OF THE PORTUGUESE LANGUAGE: 12TH INTERNATIONAL CONFERENCE, PROPOR 2016, TOMAR, PORTUGAL, JULY 13-15, 2016, PROCEEDINGS. **Anais...** Springer International Publishing, 2016. p.95–105.
- STATISTA. **Most-used languages on Twitter as of September 2013**. 2013.
- TAN, A.-h. Text Mining : the state of the art and the challenges concept-based. **In Proceedings of the PAKDD 1999 Workshop on Knowledge Disocvery from Advanced Databases**, [S.l.], p.65—70, 1999.
- TELES, V.; SANTOS, D.; SOUZA, E. Uma Análise Comparativa de Técnicas Supervisionadas para Mineração de Opinião de Consumidores Brasileiros no Twitter. In: XIII ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL E COMPUTACIONAL (ENIAC-2016). **Anais...** [S.l.: s.n.], 2016. p.217–228.
- THAKARE, P. A. D.; CHAUDHARI, S. M. Introducing a Hybrid Swarm Intelligence Based Technique for Document Clustering. **International Journal of Engineering Research and Applications (IJERA)**, [S.l.], v.2, n.6, p.1455–1459, 2012.
- TRIPATHY, A.; AGRAWAL, A.; RATH, S. K. Classification of sentiment reviews using n-gram machine learning approach. **Expert Systems with Applications**, [S.l.], v.57, p.117 – 126, 2016.
- TWITTER Official Webpage. 2016.

- WEISS, S. et al. **Text Mining: predictive methods for analyzing unstructured information**. [S.l.]: SpringerVerlag, 2004.
- WIEBE, J.; WILSON, T.; CARDIE, C. Annotating Expressions of Opinions and Emotions in Language. **Language Resources and Evaluation**, [S.l.], v.39, n.2, p.165–210, 2005.
- WU, S.-T. et al. Automatic pattern-taxonomy extraction for web mining. In: WEB INTELLIGENCE, 2004. WI 2004. PROCEEDINGS. IEEE/WIC/ACM INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2004. p.242–248.
- YANG, X.-S.; DEB, S. Cuckoo search via Lévy flights. In: NATURE & BIOLOGICALLY INSPIRED COMPUTING, 2009. NABIC 2009. WORLD CONGRESS ON. **Anais...** [S.l.: s.n.], 2009. p.210–214.
- ZHANG, Y.; XIONG, X.; ZHANG, Q. An Improved Self-Adaptive PSO Algorithm with Detection Function for Multimodal Function Optimization Problems. **Mathematical Problems in Engineering**, [S.l.], v.2013, n.iii, 2013.

Appendix

SYSTEMATIC MAPPING STUDY

Systematic reviews allow the integration and synthesis of evidence from several studies in an objective and unbiased manner ([KITCHENHAM; CHARTERS, 2007](#)). Individual studies contributing to a systematic review are called *primary studies*, while a systematic review is a type of a *secondary study*. Two common types of systematic reviews are:

- The systematic literature review (SLR), often referred to as a systematic review, is a means of identifying, interpreting, and evaluating all relevant primary studies to a particular research question, topic area, or phenomenon of interest ([KITCHENHAM; CHARTERS, 2007](#)). It is related to specific research questions such as: *"Is intervention I on population P more effective for obtaining outcome O in context C than comparison treatment T?"*
- Systematic mapping study (SMS) offers a structure of the type of research reports and results that have been published by categorization. It is also known as scope studying, since it gives a visual summary or map of its results ([PETERSEN et al., 2008](#)). The systematic mapping often requires less effort than systematic reviews while providing focuses on exploratory and broader questions such as: *"What do we know about topic T?"*

While an SLR is a mean of identifying, evaluating and interpreting all available research relevant to a particular question ([KITCHENHAM; CHARTERS, 2007](#)), a SMS intends to "map out" the research undertaken rather than to answer detailed research question ([PETERSEN et al., 2008](#); [BUDGEN et al., 2008](#)) and that is the reason, SMSs are usually termed a "Mapping Study" or a "Scoping Review".

In this SMS, the definitions from [PETERSEN et al. \(2008\)](#) were merged with good practices presented in the guideline proposed by [KITCHENHAM; CHARTERS \(2007\)](#); [KITCHENHAM; DYBA; JORGENSEN \(2004\)](#), such as the protocol definition. Therefore, a process for an SMS was applied (Figure A.1), making best use of both techniques.

Data from the 203 included primary studies were systematically extracted, analyzed and synthesized to answer the descriptive, classificatory, and historical research questions (RQ)

presented next:

- RQ1: What is the evolution in the number of publications up to year 2014?
- RQ2: Which individuals, organizations, and countries are the main contributors in the research area?
- RQ3: What are the text mining tasks and sub-tasks applied for the Portuguese language?
- RQ4: What are the mining techniques, algorithms, methods, and tools available to the Portuguese language?
- RQ5: What are the characteristics of the corpora used?
- RQ6: How were the text mining approaches evaluated?

The search for publications was done in five steps, manually performed by humans, i.e., the researchers, with the goal of mapping current primary studies that relate the application of Text Mining for Portuguese up to year 2014. The five steps are detailed in the following subsections.

Figure A.1: Systematic mapping process

A.1 Research Protocol

The research protocol was built in the first step. It includes, among other things, the research questions, the inclusion and exclusion criteria, and data sources selection. The outcome of this step is the review scope.

Primary studies that reported text mining applications for Portuguese were searched in the literature. *Full* and *short* papers that met at least one of the exclusion criteria presented below were removed from this SMS:

- Published in a language other than English or Portuguese;
- Unavailable on online scientific libraries;
- Keynote speeches, workshop reports, books, thesis, and dissertations;

A.2 Conduct Search

In this step, two different types of searches were performed: automated and manual. In the former, a search string was used to retrieve papers from digital libraries, but, according to [KITCHENHAM; DYBA; JORGENSEN \(2004\)](#), in most cases this, is not sufficient and other sources must also be searched. Therefore, a manual search on several conference proceedings was performed, with the goal of ensuring that a maximum number of relevant primary studies was included in the mapping study. The intersection of proceedings that are indexed by more the one digital library is intentional.

The search string was constructed from two terms extracted from the general research question: *Text Mining* and *Portuguese language*, as shown in Figure A.2. The synonyms and translation for both terms were included in the final query as shown in Figure A.3.

Figure A.2: Generic search string based on terms extracted from the general research question

(("mineração de texto" OR "text mining" OR "KDT" OR "knowledge discovery in text" OR "descoberta de conhecimento em texto" OR "text classification" OR "classificação de texto" OR "extração de informação" OR "information extraction" OR "sentiment analysis" OR "análise de sentimento" OR "opinion mining" OR "mineração de opinião" OR "text cluster" OR "agrupamento de texto") AND ("português" OR "portuguese" OR "língua portuguesa"))

Figure A.3: Automated search string used to retrieve primary studies from digital libraries

As shown in Table A.1, the automated search was performed on six digital libraries using every available field. Due to a limited number of words (forty) in the query, the search string for the IEEE engine had to be split into two queries and duplicated papers had to be removed. A total of 6,075 primary studies was retrieved from this step.

PROPOR is the main conference focusing on the Portuguese language, jointly promoted by Brazil and Portugal. Papers published up to the year 2000 were not included as they were

Table A.1: Systematic mapping manual (M) and automated (A) data sources.

Data Source	Type	Retrieved Studies	Included Studies
International Conference on the Computational Processing of Portuguese (PROPOR)	M	217	22
Text Mining and Applications (TEMA) track of Portuguese Conference on Artificial Intelligence	M	34	6
Brazilian Workshop of Social Network Analysis and Mining (BRAS-NAM)	M	99	11
Brazilian Symposium on Information and Human Language Technology (STIL)	M	251	44
ACM symposium on Document engineering (DocEng)	M	273	1
Linguatca Database (www.linguatca.pt)	M	1312	30
Message Understanding Conferences (MUC)	M	159	-
Text Analysis Conference (TAC)	M	322	-
Text REtrieval Conference (TREC)	M	1715	-
Document Understanding Conference (DUC)	M	167	-
IEEE Xplore Digital Library	A	306	19
ACM Digital Library	A	277	29
Science Direct	A	159	4
Scopus	A	552	21
Portal de Periódicos Capes	A	229	15
SciELO Scientific Electronic Library Online	A	2	1
TOTAL		6075	203

unavailable online. TEMA is a forum for researchers working with text mining and related areas conjoined with the Portuguese Conference on Artificial Intelligence. BRASNAM and STIL are computational linguistics forums in Brazil which deal not only with Portuguese processing, but also with English and Spanish. Until the year 2008, STIL was called “Information Technology and Human Language Workshop (TIL)”. DocEng is an annual international forum for presentations and discussions on principles, tools and processes that improve our ability to create, manage, and maintain documents. *Linguateca* database is a distributed language resource centre for the Portuguese language, including a large web catalog of resources, researchers, tools, and publications related to the computational processing of Portuguese. MUC, TAC, TREC and DUC are annual competitions to encourage the development of new and better methods.

A.3 Pre-selection

A pre-selection is accomplished in accordance with the inclusion and exclusion criteria established in the review protocol. Primary studies were analysed using the same procedure for both automated and manual search strategies. Three pairs of researchers applied the inclusion and exclusion criteria on all 6,075 retrieved primary studies after reading the *title*, *abstract*, and *keywords*, resulting in a list of 661 potentially relevant studies (see Table A.1).

At the end of this step, a consensus meeting was held to solve possible discrepancies. When agreement was not possible, a researcher who was not involved in the original group worked to solve the discrepancies. Electronic spreadsheets and Mendeley[®] software were used to support the search and selection process.

A.4 Selection

The process was similar to the previous step. The difference is that researchers applied the inclusion and exclusion criteria on 661 potentially relevant primary studies after reading the *full* paper, resulting in a list of 203 included studies (see Table A.1).

A.5 Data Extraction

In this fifth step, data from included primary studies were extracted and synthesised to answer the general and specific research questions. The results were packed (<http://bit.ly/1NF6Xmv>) for further replication.

Four researchers worked independently to extract data from the included papers, guided by an extraction form, implemented in an electronic spreadsheet and with the support of Mendeley[®] software. Another researcher inspected the disagreements in the extracted data and *ad hoc* consensus meetings were held as required. The extraction form comprises the following data:

- Publication title: the publication title and subtitle (when available);
- Authors: the publication author(s), and its affiliation;
- Year: the publication year;
- Venue: publication conference/journal, proceedings;
- Algorithm/Tool: algorithms or tools adopted to develop the text mining application;
- Approach: supervised, unsupervised or hybrid approaches;
- Task/Sub-tasks: text mining tasks according to Section 2;
- Language Variant: Portuguese language variant. Brazilian and European, for example;
- Text type: the text characteristics, such as: short, long, formal or informal text;
- Corpus: data-set characteristics to train and validate the text mining application;
- Evaluation metrics: the set of used metrics to evaluate the text mining application;
- Domain: the domain area wherein the study was applied;