An Unsupervised Particle Swarm Optimization Approach for Opinion Clustering

Ellen Souza
UAST - UFRPE
Serra Talhada, PE - Brazil
ellen.ramos@ufrpe.br

Adriano L. I. Oliveira, Alisson Silva, Gustavo Oliveira
CIn - UFPE
Recife, PE - Brazil
{alio, aps4, ghfmo}@cin.ufpe.br

Diego Santos
ECMOP - UPE
Recife, PE - Brazil
dgs2@ecomp.poli.br

***Abstract*—Supervised machine learning (ML) and lexicon-based are the most frequent approaches for opinion mining (OM), but they require considerable effort for preparing the training data and to build the opinion lexicon, respectively. This paper presents two unsupervised approaches for OM based on Particle Swarm Optimization (PSO). The PSO-based approaches were evaluated by eighteen experiments with different corpora types, domains, language, class balancing and pre-processing techniques. The proposed approaches achieved better accuracy on twelve experiments. Best results were obtained on corpora with a reduced number of dimensions and for specific domains. Best accuracy (0.79) was obtained by Discrete IDPSO on the OBCC corpus, outperforming supervised ML and lexicon-based approaches for this corpus.**

I. Introduction

The growth of social media and micro-blogs on the Internet provides a huge quantity of data that allows discovering the experiences, opinions, and feelings of users and customers [1]. Since it is a rich source of real-time information, there has been an increasing interest to create systems capable of extracting information from this kind of data [2].

According to [3], opinion mining (OM), also known as sentiment analysis, is the field of study that analyzes peoples sentiments, evaluations, attitudes, and emotions about different entities expressed in textual input. This is accomplished through the opinion classification of a document, sentence or feature into categories, such as: *positive*, *negative*, or *neutral*. This kind of classification is referred to sentiment polarity or polarity classification [4].

OM techniques can be divided into machine learning (ML) approach, lexicon-based approach, and hybrid approach which make use of both ML and lexicon [5], [4], [6]. The supervised ML applies classification algorithms to learn underlying patterns from example data to later attempt to classify new unlabeled data [2]. It has yielded high accuracy but needs a considerable amount of labeled data, commonly built manually and dependent on language and domain.

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* [5]. Three main approaches are used to build opinion lexicon: manual approach, which is very time consuming; dictionary-based in which an initial set (built manually) is grown 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.

As the most frequent approach for OM, presented above, are very time consuming, this paper proposes the use of unsupervised algorithms to analyze opinions by grouping a set of opinions (comments or reviews) into clusters of related opinions. The proposed approach involves two discrete versions of Particle Swarm Optimization (PSO) algorithm and natural language processing (NLP) tasks. The PSO has been successfully applied to clustering problems, including short texts [7], as it performs a global search process. Up to this point, there has been no evidence of the use of the PSO algorithm for opinion clustering. Preliminary results indicate the feasibility of the proposal.

This paper is organized as follows: section 2 briefly reviews the adopted techniques and presents the major related studies. Section 3 presents the two PSO approaches proposed for opinion clustering. Section 4 details the experimental setup and the obtained results. Section 5 brings the conclusion and highlights future works.

II. Background and Related Work

A. Text Clustering

Text clustering is an approach of 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.

The clustering-based opinion mining approach applies unsupervised learning algorithms which neither requires any human labeled training data, nor time for training [8]. However, it has some difficulties such as the one to catch subtle semantics that human beings use in speech and writing. This 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 a level acceptable with respect to analogy-making in human beings is an even more challenging issue [7].

The quality of the resulting clusters is commonly evaluated with respect to structural properties expressed in different internal clustering validity measures (ICVM), such as the global silhouette (GS) coefficient. These internal measures are very common in document and short-text clustering, but, as stated by [9], [10], the real effectiveness of the clustering algorithms can only be evaluated with external measures that incorporate the categorization criteria of the users. Common external measures are: *Accuracy*, *Precision*, *Recall*, and *F-score*.

As far as we know, the studies of Li and Liu [11], [8] are the only ones dealing with OM as a clustering problem. The authors applied term-frequency and inverse document frequency (TF-IDF) weighting method and voting mechanism, together with the k-means clustering algorithm. 88 % accuracy was obtained on a better quality dataset for the movie review corpus [4].

B. Text Clustering with Particle Swarm Optimization

PSO was first proposed in 1995 by Eberhart and Kennedy [12], [13] and it is inspired by the social behavior of a bird flock. 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.

PSO algorithm is a stochastic global optimization method to find the optimal or global optimum in the landscape of objective function. Compared with other evolutionary methods, PSO has an advantage of its simple implementation and the good trade-off between exploration and exploitation ability [14].

The first approach for text clustering using PSO was proposed in 2005 by [15], [16]. The authors tried PSO, K-means and hybrid PSO clustering algorithms on four different document corpora. Results illustrate that the hybrid PSO algorithm can generate more compact clustering results.

Other studies have also proposed PSO-based approaches for document clustering, but none has used PSO for opinion clustering. The CLUstering with a DIscrete PSO (CLUDIPSO) and its improved version (CLUDIPSO*) proposed in [9], [7] are the closest to our approach as they were developed for clustering short-text collections and made use of PSO solely. Experimental results show that PSO-based approaches can be highly competitive alternatives for clustering short-text corpora and can outperform the most effective clustering algorithms used in this area.

III. PSO-BASED CLUSTERING APPROACHES

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

Given (i) a set of opinions $O = o_1, o_2, \ldots, o_n$,

(ii) a desired number of clusters k, and

(iii) an objective function f that evaluates the quality of a clustering, we want to compute an assignment $\gamma : O \rightarrow 1, \ldots, K$ that minimizes (or, in some cases, maximizes) the objective function, which is often defined in term of similarity or distance measures.

Each valid cluster is represented as a particle (Fig.1), which is 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.

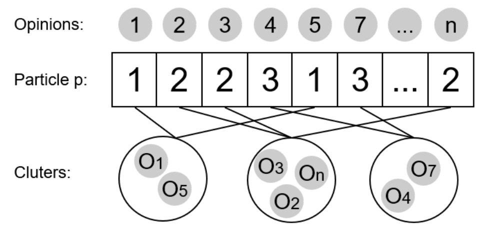

Fig. 1. Particle representation for the Clustering of Opinions.

Two discrete PSO-based algorithms are proposed in this paper: the first one is based on a discrete binary version of PSO, first proposed by [17], while the second one is based on an Improved Self-Adaptive PSO (IDPSO) algorithm with detection function [14]. Instead of operating in a continuous space, in the discrete version, trajectories are changes in 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 two algorithms are detailed in the following subsections.

A. Discrete PSO + Mutation (DPSOMUT)

The DPSOMUT pseudo-algorithm presented in this subsection, uses the ICVM GS coefficient (Eq. 1) as fitness function, once it has achieved good outcomes for short-text clustering [9]. 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 particles evolve at each iteration using two updated formulas: one for velocity (Eq. 2) and another for position (Eq. 3). 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 [9]. To avoid convergence to a local optimum, a mutation is applied by swapping particles randomly. 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.

$$s(i) = \frac{(b(i) - a(i))}{max(a(i), b(i))} \quad (1)$$

$$v_{id} = \omega(v_{id} + \gamma_1(pbest_{id} - x_{id}) + \gamma_2(gbest_d - x_{id})) \quad (2)$$

$$x_{id} = pbest_{id} \quad (3)$$

Algorithm 1 DPSOMUT Pseudo-algorithm

```
Input: opinion similarity matrix
Output: vector for each cluster
Initialize particles, cluster vector
while maximum iterations is not attained do
    for each particle do
        Calculate fitness value according to Eq. 1
        if fitness value better than the best fitness value (pbest) in history then
            Set current value as the new pbest
        end if
    end for
    Choose particle with the best fitness value of all particles as the gbest
    for each particle do
        if particle velocity greater than random number then
            Calculate particle velocity according to Eq. 2
            Update particle position according to Eq. 3
        end if
    end for
    Apply mutation by swapping particles randomly
end while
```

B. *Discrete Improved Self-Adaptive PSO (IDPSO)*

Researches around PSO showed that the values of the weights given to the inertia, and cognitive and social factors strongly influence the behavior of the particles of the algorithm 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 herself.

The main characteristic of the IDPSO [14] is to make an exchange between a global to a local search operation, during the iterations. 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 (Eq. 4) needs to be computed. $(gbest - x_i(t-1))$ is the Euclidean distance between the particle i and the best solution found by the swarm $gbest$ up to the iteration $(t-1)$. $(pbest_i - x_i(t-1))$ 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 Eq. 5, Eq. 6 and Eq. 7. 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. Also, to avoid convergence to a local optimum, a mutation is applied by swapping particles randomly [7].

$$\varphi(t) = |(gbest - x_i(t-1))/(pbest_i - x_i(t-1))| \quad (4)$$

$$\gamma_1(t) = \gamma_1.\varphi^{-1}(t) \quad (5)$$

$$\gamma_2(t) = \gamma_2.\varphi(t) \quad (6)$$

$$\omega(t) = \frac{\omega_{initial} - \omega_{final}}{1 + e^{\varphi(t).(t-((1+\ln(\varphi(t))).K_{max})/\mu)}} + \omega_{final} \quad (7)$$

IV. Experimental Setup and Results

A. *Corpora*

For the experimental work, three corpora with different levels of complexity with respect to size, number of opinions, domains, language, part-of-speech (POS) tagging, and class balancing were selected. Table I 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 pre-processing step. In the sequence, the number of opinions for each class, as well as the number of all opinions presented in the corpus are presented. *Tok* and *DTok* contains the number of tokens and different tokens, respectively. *Tok-POS* and *DTok-POS* contains the number of tokens and different tokens after POS tagging filtering.

The movie review corpus from [18] 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 corpus. The sentiment140 corpus, from Stanford University [19], contains opinions written in English about brand, product, or topic on 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. The OBCC corpus was proposed by [20] and contains a gold collection with 2940 *tweets* in Brazilian Portuguese with opinions of consumers about products and services. This collection was also partitioned into four subsets according to balancing and number of class.

TABLE I
CORPORA DETAILS

Corpora	Number of classes	Class Balancing	POS Tagger	Pos	Neg	Neu	Total	Tok	DTok	Tok POS	DTok POS
Movie Review	2	Balanced	General	300	300	——	600	474,465	23,869	249,190	23,410
		Unbalanced	General	100	300	——	400	309,919	19,549	162,706	19,150
Sentiment140	2	Balanced	General	139	139	——	278	5,036	1,497	2,970	1,345
			Tweet specific	139	139	——	278	4,585	1,538	2,134	1,030
		Unbalanced	General	182	177	——	359	6,651	1,786	3,896	1,614
			Tweet specific	182	177	——	359	6,058	1,835	2,790	1,225
	3	Balanced	General	139	139	139	417	6,986	2,049	4,181	1,872
			Tweet specific	139	139	139	417	6,322	2,108	2,909	1,364
		Unbalanced	General	182	177	139	498	8,601	2,314	5,107	2,118
			Tweet specific	182	177	139	498	7,795	2,375	3,565	1,540
OBCC	2	Balanced	General + Floresta	166	166	——	332	7,014	1,663	1,790	765
			General + Mac-Morpho	166	166	——	332	6,640	1,679	2,054	1,020
		Unbalanced	General + Floresta	166	1,299	——	1,465	31,438	4,356	8,882	1,854
			General + Mac-Morpho	166	1,299	——	1,465	29,170	4,379	10,133	2,802
	3	Balanced	General + Floresta	166	166	166	498	10,256	2,290	2,495	988
			General + Mac-Morpho	166	166	166	498	9,705	2,317	2,883	1,368
		Unbalanced	General + Floresta	166	1,299	553	2,018	42,378	5,512	11,242	2,145
			General + Mac-Morpho	166	1,299	553	2,018	39,521	5,594	13,052	3,446

B. Pre-Processing

Fig. 2 presents the pre-processing steps executed for each corpora. All steps were performed using the Python NLTK. The Perceptron POS tagger was used for both English language corpora. For the Sentiment140 corpus, we also used the Carnegie Mellon POS Tagger [21] specific for *tweets* written in English language. For the Brazilian Portuguese OBCC corpus, two POS tagger were selected: Perceptron and Unigram taggers. The first tagger was trained using Floresta Sinta(c)tica corpus while the second was trained using MacMorpho corpus, both available at Python NLTK. The chosen PSO taggers presented good outcomes for selected corpora.

As adjectives, adverbs, nouns, and verbs are strong indicators of sentiment in an opinion [22], [4], they were selected to build a local dictionary. Words from other parts of speech were discarded during the *Feature Reduction* step. In the *Feature Transformation* step, opinions were represented using the vector space model (VSM) associated with the TF-IDF weighting scheme. The opinions (O) are represented as vectors $O_j = (w_{1j}, w_{2j}, w_{3j}, \ldots, w_{tj})$ and each dimension corresponds to a separate word (w) or term for the opinion j. After building the VSM model, the proposed approaches use the *cosine* measure to estimate the similarity between two opinions. The measure is widely used in text clustering literature [16], [9] and it computes the cosine of the angle between two documents. The result of this step is an *Opinion Similarity Matrix* used as input for the algorithms.

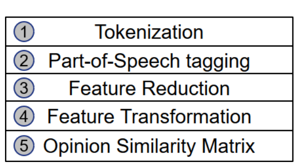

Fig. 2. Text Pre-processing.

C. Experimental Setup

Except the Discrete IDPSO, that was implemented using Java language, all the other algorithms were developed on Python, using the NLTK and Scikit-learn toolkits. The PSO-based and K-means algorithms have the same computational complexity $(O(n^2))$, while the Agglomerative has a complexity of $O(n^2 * \log(n))$. Default setup parameters were adopted for each algorithm. For the K-means, we performed 25 runs with 1000 iteration per run. For the Agglomerative, we performed 25 runs with 10 iteration per run. For the Discrete IDPSO, we performed 50 runs with 1000 iteration per run, using the following parameters: swarm size = 20 particles, dimensions of each particle = number of opinions, $\omega_{initial}$ and ω_{final} = [0.8 - 0.1], γ_1 and γ_2 = [2.4 - 1.3], μ = 100, and max_{pm} and min_{pm} = [2 - 0]. For the DPSOMUT, 25 runs were performed with 50 iteration each, using the following parameters: swarm size = 10 particles, dimensions of each particle = number of opinions, ω = [0.9 - 0.4], $\gamma 1$ and $\gamma 2$ = 1.

D. Results and Discussion

As shown on Table II, eighteen experiments were performed with different corpora types, class balancing and pre-processing techniques. The PSO-based approaches achieved better accuracy on twelve experiments (tagged with asterisks). The best accuracy (0.79) was obtained by the Discrete IDPSO algorithm on OBCC (Unigram POS Tagger + Floresta Sinta(c)tica) corpus.

For the experiments with two classes (positive and negative), an accuracy above 0.7 was reached by the PSO-based approaches for all classes and 0.8 for identifying negative class. However, for the experiments with three classes (positive, negative and neutral), the best result obtained by the PSO-based approaches reaches the accuracy of 0.5 for all classes and 0.6 for neutral class. The reason is that those corpora (with three classes) has very overlapping 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. We could observe a significant improvement in the results of the *tweet*-based corpus which used a *tweet* specific POS tagger. For the Brazilian Portuguese language, we did not found a *tweet* specific POS tagger. As observed for the English language, this specific tagger added an improvement in the results. We could not observe significant difference in the accuracy for the Brazilian Portuguese corpus tagged with Floresta Sinta(c)tica or Mac-Morpho corpora.

The studies of [11], [8] are the only ones dealing with opinion mining as a clustering problem. An accuracy of 0.8 was obtained on a better quality dataset for the movie review corpus [4]. Our average accuracy for this corpora was 0.62, reached by the DPSOMUT algorithm. The studies [11], [8] filtered only adjectives and adverbs after PSO tagging and used a voting mechanism after 10 K-means runs to determine which class the opinion belongs to. Reported accuracy with supervised (ML) and lexicon-based approaches for [4] corpus vary from 0.76 to 0.92.

No opinion clustering approach using Twitter data was found in literature. The best precision (0.66) and f-score (0.40) for the OBCC corpus using SVM and opinion lexicon was obtained by [20]. Our PSO-based approaches outperformed this results, achieving best precision and f-score of 0.85 and 0.86, respectively. Due to space limit, those results are not shown on Table II. For the Stanford Sentiment140, our PSO-based clustering obtained very poor results when compared with existing supervised (ML) and lexicon-based approaches. Reported accuracy with supervised (ML) for Sentiment140 corpus vary from 0.65 to 0.83.

V. Conclusion

This paper presented an unsupervised way to analyze people's opinions on social media and micro-blogs. Two PSO-based approaches were proposed and evaluated with eighteen experiments with different corpora types, domains, language, class balancing and pre-processing techniques. The PSO-based approaches achieved better accuracy on twelve experiments. Best results were obtained on corpora with a reduced number of terms (dimensions) and for specific domains. The proposed approaches also outperformed the ML and lexicon-based approaches for the OBCC corpus. Although the PSO-based approaches obtained poor results for the corpora with different domains, they still competitive as no labeled data, nether opinion lexicons, both very time consuming, are required for the analysis of opinions.

Due to lack of space, we report the overall results analyzing only the accuracy measure and for all classes together. Further analysis of data using other measures and statistical methods need to be performed. As future work, we intend to improve results of the proposed approaches by using hybrid and semi-supervised techniques.

Acknowledgment

Ellen Souza is supported by FACEPE (IBPG-0765-1-0311).

References

[1] E. Marine-Roig and S. Anton Clavé, "Tourism analytics with massive user-generated content: A case study of Barcelona," *Journal of Destination Marketing & Management*, pp. 1–11, 2015.

[2] J. A. Balazs and J. D. Velásquez, "Opinion Mining and Information Fusion: A survey," *Information Fusion*, vol. 27, pp. 95–110, 2016.

[3] B. Liu and L. Zhang, "A SURVEY OF OPINION MINING AND SENTIMENT ANALYSIS," *Mining Text Data*, vol. Chapter 1, pp. 415–463, 2012.

[4] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 12, pp. 1–135, 2008.

[5] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014.

[6] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications," *Knowledge-Based Systems*, vol. 89, 2015.

[7] L. Cagnina, M. Errecalde, D. Ingaramo, and P. Rosso, "An efficient Particle Swarm Optimization approach to cluster short texts," *Information Sciences*, vol. 265, pp. 36–49, 2014.

[8] G. Li and F. Liu, "Sentiment analysis based on clustering : a framework in improving accuracy and recognizing neutral opinions," pp. 441–452, 2014.

[9] L. C. Cagnina, M. L. Errecalde, and D. A. Ingaramo, "A DISCRETE PARTICLE SWARM OPTIMIZER FOR CLUSTERING SHORT-TEXT CORPORA," *Proceedings of International Conference on Bioinspired Optimization Methods and their Applications, BIOMA 2008*, pp. 1–10, 2008.

[10] S. Karol and V. Mangat, "Evaluation of text document clustering approach based on particle swarm optimization," *Central European Journal of Computer Science*, vol. 3, no. 2, pp. 69–90, 2013.

[11] G. Li and F. Liu, "A Clustering-based Approach on Sentiment Analysis," *International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, pp. 331–337, 2010.

[12] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *Proceedings of IEEE International Conference on Neural Networks*, pp. 1942–1948, 1995.

[13] R. Eberhart and J. Kennedy, "A New Optimizer Using Particle Swarm Theory," *Sixth International Symposium on Micro Machine and Human Science*, pp. 39–43, 1995.

[14] Y. Zhang, X. Xiong, and Q. Zhang, "An Improved Self-Adaptive PSO Algorithm with Detection Function for Multimodal Function Optimization Problems," *Mathematical Problems in Engineering*, vol. 2013, no. iii, 2013.

[15] X. Cui, T. E. Potok, and P. Palathingal, "Document clustering using particle swarm optimization," *Proceedings 2005 IEEE Swarm Intelligence Symposium*, pp. 185–191, 2005.

[16] X. Cui and T. E. Potok, "Document clustering analysis based on hybrid pso+k-means algorithm," *Special Issue*, pp. 27–33, 2005.

[17] J. Kennedy and R. C. Eberhart, "A DISCRETE BINARY VERSION OF THE PARTICLE SWARM ALGORITHM," *IEEE International Conference On Systems, Man, And Cybernetics*, pp. 4–8, 1997.

[18] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up ? Sentiment Classification using Machine Learning Techniques," *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, no. July, pp. 79–86, 2002.

[19] A. Go, R. Bhayani, and L. Huang, "Twitter Sentiment Classification using Distant Supervision," Tech. Rep., 2010.

[20] E. Souza, D. Castro, D. Vitório, I. Teles, A. L. I. Oliveira, and C. Gusmão, "Characterizing User-Generated Text Content Mining: A Systematic Mapping Study of the Portuguese Language," *New Advances in Information Systems and Technologies*, pp. 1015–1024, 2016.

[21] O. Owoputi, B. O. Connor, C. Dyer, K. Gimpel, and N. Schneider, "Part-of-Speech Tagging for Twitter: Word Clusters and Other Advances," Carnegie Mellon University, Tech. Rep., 2012.

[22] C. Marques-lucena and J. Sarraipa, "Framework for Customers Sentiment Analysis," *Advances in Intelligent Systems and Computing*, pp. 849–860, 2015.

TABLE II
AVERAGE PRECISION, RECALL, F-SCORE AND ACCURACY FOR EACH CORPUS

Number of Classes	Class Balancing	Corpus	Algorithm	Precision	Recall	F-score	Accuracy
2	Balanced	OBCC (Unigram POS Tagger + Floresta)*	k-means	0.550	0.523	0.454	0.523
			Agglomerative	0.527	0.552	0.463	0.552
			DPSOMUT	0.516	0.509	0.454	0.509
			Discrete IDPSO	0.798	0.788	0.788	0.790
		OBCC (Perceptron POS Tagger + MacMorpho)*	k-means	0.530	0.514	0.443	0.514
			Agglomerative	0.440	0.364	0.349	0.440
			DPSOMUT	0.510	0.507	0.467	0.507
			Discrete IDPSO	0.782	0.756	0.764	0.773
		Movie Review	k-means	0.502	0.502	0.500	0.502
			Agglomerative	0.483	0.483	0.483	0.483
			DPSOMUT	0.497	0.498	0.451	0.498
			Discrete IDPSO	0.369	0.452	0.396	0.359
		Sentiment140	k-means	0.499	0.499	0.480	0.499
			Agglomerative	0.514	0.519	0.483	0.514
			DPSOMUT	0.487	0.491	0.444	0.491
			Discrete IDPSO	0.251	0.261	0.249	0.269
		Sentiment140 (Twitter specific POS tagger)	k-means	0.521	0.508	0.470	0.508
			Agglomerative	0.518	0.523	0.489	0.518
			DPSOMUT	0.477	0.486	0.428	0.486
			Discrete IDPSO	0.239	0.255	0.244	0.234
2	Unbalanced	OBCC (Unigram POS Tagger + Floresta)*	k-means	0.505	0.506	0.353	0.467
			Agglomerative	0.356	0.318	0.147	0.156
			DPSOMUT	0.505	0.509	0.491	0.701
			Discrete IDPSO	0.241	0.706	0.358	0.713
		OBCC (Perceptron POS Tagger + MacMorpho)*	k-means	0.507	0.511	0.353	0.463
			Agglomerative	0.599	0.564	0.298	0.302
			DPSOMUT	0.506	0.511	0.494	0.706
			Discrete IDPSO	0.246	0.662	0.358	0.732
		Movie Review*	k-means	0.510	0.512	0.482	0.522
			Agglomerative	0.489	0.491	0.486	0.574
			DPSOMUT	0.496	0.496	0.496	0.629
			Discrete IDPSO	0.155	0.370	0.211	0.374
		Sentiment140	k-means	0.495	0.495	0.480	0.496
			Agglomerative	0.246	0.343	0.283	0.370
			DPSOMUT	0.489	0.493	0.451	0.489
			Discrete IDPSO	0.245	0.206	0.220	0.277
		Sentiment140 (Twitter specific POS tagger)	k-means	0.521	0.510	0.452	0.510
			Agglomerative	0.500	0.500	0.460	0.504
			DPSOMUT	0.485	0.491	0.435	0.486
			Discrete IDPSO	0.248	0.264	0.255	0.222
3	Balanced	OBCC (Unigram POS Tagger + Floresta)*	k-means	0.330	0.330	0.244	0.330
			Agglomerative	0.275	0.126	0.162	0.275
			DPSOMUT	0.339	0.337	0.309	0.337
			Discrete IDPSO	0.400	0.249	0.299	0.365
		OBCC (Perceptron POS Tagger + MacMorpho)*	k-means	0.320	0.323	0.248	0.323
			Agglomerative	0.365	0.358	0.313	0.365
			DPSOMUT	0.335	0.338	0.311	0.338
			Discrete IDPSO	0.404	0.249	0.302	0.371
		Sentiment140*	k-means	0.304	0.312	0.293	0.312
			Agglomerative	0.293	0.302	0.281	0.293
			DPSOMUT	0.331	0.333	0.307	0.333
			Discrete IDPSO	0.331	0.223	0.256	0.283
		Sentiment140 (Twitter specific POS tagger)*	k-means	0.368	0.347	0.281	0.347
			Agglomerative	0.345	0.396	0.299	0.345
			DPSOMUT	0.336	0.333	0.306	0.333
			Discrete IDPSO	0.474	0.301	0.354	0.421
3	Unbalanced	OBCC (Unigram POS Tagger + Floresta)*	k-means	0.306	0.310	0.230	0.357
			Agglomerative	0.574	0.561	0.253	0.254
			DPSOMUT	0.335	0.334	0.272	0.314
			Discrete IDPSO	0.356	0.316	0.322	0.499
		OBCC (Perceptron POS Tagger + MacMorpho)*	k-means	0.401	0.337	0.247	0.347
			Agglomerative	0.291	0.079	0.094	0.095
			DPSOMUT	0.330	0.332	0.270	0.310
			Discrete IDPSO	0.376	0.334	0.342	0.527
		Sentiment140	k-means	0.330	0.332	0.317	0.332
			Agglomerative	0.305	0.313	0.301	0.319
			DPSOMUT	0.337	0.335	0.300	0.313
			Discrete IDPSO	0.383	0.239	0.286	0.316
		Sentiment140 (Twitter specific POS tagger)*	k-means	0.355	0.334	0.270	0.338
			Agglomerative	0.345	0.397	0.304	0.369
			DPSOMUT	0.322	0.326	0.287	0.303
			Discrete IDPSO	0.498	0.302	0.365	0.425