Unsupervised Dialogue Act Classification with Optimum-Path Forest

Luiz Carlos Felix Ribeiro, João Paulo Papa
Department of Computing
São Paulo State University – UNESP
Bauru - SP, Brazil
lzcfelix@gmail.com
papa@fc.unesp.br

***Abstract*—Dialogue Act classification is a relevant problem for the Natural Language Processing field either as a standalone task or when used as input for downstream applications. Despite its importance, most of the existing approaches rely on supervised techniques, which depend on annotated samples, making it difficult to take advantage of the increasing amount of data available in different domains. In this paper, we briefly review the most commonly used datasets to evaluate Dialogue Act classification approaches and introduce the Optimum-Path Forest (OPF) classifier to this task. Instead of using its original strategy to determine the corresponding class for each cluster, we use a modified version based on majority voting, named M-OPF, which yields good results when compared to k-means and Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), according to accuracy and V-measure. We also show that M-OPF, and consequently OPF, are less sensitive to hyper-parameter tuning when compared to HDBSCAN.**

I. Introduction

Dialogue Acts (DAs) concisely identify each utterance in a dialogue according to a combination of pragmatic, semantic and syntactic criteria, characterizing the speaker's intention and goal [1]. Consequently, the task of DA classification consists in assigning each utterance of a conversation to a tag of a predefined set of labels that represent the domain of the problem. For example, the utterance *"Where is the closest shopping mall?"* can be assigned to the class *ask_directions*.

The DA classification task can also be seen as a sub-problem for other tasks, such as the development of chatbots where the DA of the utterance emitted by the user can be used to select the most appropriate response. Other applications involve learning conversational dynamics [2], categorization of sentences in posts from web forums to help the users telling informative from feedback messages [3], and prediction of thread structure [4].

The process of manually annotating data to train a supervised classifier is expensive, time-consuming and error-prone. This last characteristic is probably due either to not well-defined annotation rules or to sentence ambiguity. Regarding this last aspect, Stolcke et al. [1] noticed that in the annotation of the Switchboard corpus [5], the inter-labeler agreement was of 84% despite the existence of extensive annotation guidelines. Furthermore, due to its slowness, this phase becomes a bottleneck for both the adoption of new datasets and to test new ideas [6]. These aspects also make difficult to leverage data available from the increasing amount of sources, such as social networks and message-exchanging applications.

Differently from supervised learning, unsupervised techniques can cluster utterances according to their inter-similarity in a completely data-driven approach, avoiding the need to manually label the dataset using handcrafted rules, which is required to train a supervised classifier. It is also interesting to mention that unsupervised classification does not necessarily have to be an end goal, but it can also be used as an intermediate step for data exploration. After clustering the samples, the effort to inspect each data partition is presumably smaller than the cost to analyze samples individually. Furthermore, irregular clusters can be manually labeled, thus consuming less effort.

Despite these aspects, most of the work related to DA classification uses supervised approaches, as observed by different authors [3], [7]. Well-known algorithms such as k-means, k-medoids and hierarchical clustering have been previously used for DA classification as well [8], [9]. On the other hand, Jo et al. [10] model conversations using graphical models where each DA is a mixture of foreground and background topics, the latter being shared across multiple dialogues. Brychcín and Král [7] use Hidden Markov Models (HMM) with Multivariate Gaussian distributions to represent each utterance. Ritter et al. [2] also use HMMs not only to classify DAs but also to learn the dialogue structure.

Pattern recognition techniques have been around in the last decades, and there is always room for new ideas and developments. The Optimum-Path Forest (OPF) is a framework to the design of pattern classifiers based on graph partitions, where the idea is to rule a reward-based competition process in which some key samples compete among themselves to conquer the remaining samples of the dataset. This competition process ends up partitioning the graph into groups of samples, which can be labeled [11]–[13], unlabeled [14], and partially labeled [15].

This paper aims to introduce the unsupervised OPF classifier to the Natural Language Processing (NLP) field, more specifically for the DA classification task. OPF results are compared to other clustering algorithms: k-means and Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [16]. To the best of our knowledge, this is the first application of the unsupervised OPF classifier

in this context. The remainder of this paper is organized as follows: Section II presents the unsupervised Optimum-Path Forest classifier and Section III presents an overview of the most commonly used datasets and their corresponding relevant works for DA classification. On Section IV, the experiments are described and the results obtained are presented on Section V. Conclusions are drawn on Section VI.

II. Unsupervised Learning with Optimum-Path Forest

In this section, we briefly present the theoretical background related to unsupervised OPF. Let $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ be an unlabeled dataset such that $\mathbf{x}_i \in \mathbb{R}^n$ stands for a feature vector extracted from some sample related to the problem to be addressed. Additionally, let $\mathcal{G} = (\mathcal{X}, \mathcal{A}_k)$ be a graph derived from that dataset, which means $\mathcal{X}$ denotes the set of graph nodes (i.e., vertices) and $\mathcal{A}_k$ stands for a k-nearest neighbors adjacency relation.

In a nutshell, the OPF working mechanism is based on a reward-competition problem, where some samples called "prototypes" rule a competitive process among themselves to conquer the other samples from the dataset $\mathcal{X}$. Such competition ends up partitioning $\mathcal{X}$ into optimum-path trees (OPTs), which are rooted at each prototype node. It is worth mentioning that a sample that belongs to a given OPT is more "strongly connected" to the root and samples of that tree than to any other in the forest (i.e., a collection of all trees in the graph).

At a glance, the whole process can be summarized in the following steps:

1) To establish a proper neighborhood size and build up $\mathcal{A}_k$ (i.e., to find out "suitable" k values);
2) To elect the prototypes;
3) To start the competition process.

Concerning step 1), a number of different approaches to cope with the task could be considered. Rocha et al. [14] proposed to compute the best value of k (i.e., the neighborhood size), say that k^*, as the one that minimizes the normalized graph cut, which is a measure that considers both the dissimilarity between clusters as well as the similarity within the groups of samples [17].

Soon after computing k^*, the next move concerns finding the prototypes (i.e., step 2), also known as the "roots of the trees". Such essential samples are in charge of ruling the competition process that ends up partitioning the graph into OPTs (i.e., clusters).

The supervised OPF proposed by Papa et al. [11] elects the prototypes as the nearest samples from different classes, which can be accomplished by computing a Minimum Spanning Tree (MST) over the training graph. Then, the samples from different classes that are connected in the MST are marked as prototypes. However, unsupervised OPF does not make use of labeled datasets, which motivated Rocha et al. [14] to elect the prototypes as the samples that are located at the center of the clusters. Such samples can be computed by assigning a density score $\rho(\mathbf{x}_i)$ for each dataset sample $\mathbf{x}_i \in \mathcal{X}$. That score is computed using a probability density function (pdf) given by a Gaussian distribution considered in the neighborhood of each sample as follows:

$$\rho(\mathbf{x}_i) = \frac{1}{\sqrt{2\pi\sigma^2 k}} \sum_{\forall \mathbf{x}_j \in \mathcal{A}_k(\mathbf{x}_i)} \exp\left(\frac{-d(\mathbf{x}_i, \mathbf{x}_j)}{2\sigma^2}\right), \quad (1)$$

where $i \neq j$ and $\sigma = d_{max}/3$. In this case, d_{max} stands for the maximum arc-weight in $\mathcal{G}$. Using such formulation, $\rho(\mathbf{x}_i)$ considers all adjacent nodes for the probability computation purposes since a Gaussian function covers 99.7% of the samples within $d(\mathbf{x}_i, \mathbf{x}_j) \in [0, 3\sigma]$.

After computing Equation 1 for all nodes, the competition process among samples can take place. Each density value will be used to populate a priority queue, where the idea of the unsupervised OPF algorithm is to end up maximizing the cost of each sample, and thus partitioning the graph.

The definition of "cost" is based on paths on graphs, i.e., a sequence of adjacent samples with no cycles. Let $\pi_{\mathbf{x}_i}$ be a path with terminus at sample $\mathbf{x}_i$ and starting from some root $\mathcal{R}(\mathbf{x}_i)$, where $\mathcal{R}$ stands for the set of prototype samples. Additionally, let $\pi_{\mathbf{x}_i} = \langle \mathbf{x}_i \rangle$ be a trivial path (i.e., a path composed of a single sample) and $\pi_{\mathbf{x}_i} \cdot \langle \mathbf{x}_i, \mathbf{x}_j \rangle$ the concatenation of $\pi_{\mathbf{x}_i}$ and the arc $(\mathbf{x}_i, \mathbf{x}_j)$ such that $i \neq j$.

The OPF algorithm assigns to each path $\pi_{\mathbf{x}_i}$ a value $f(\pi_{\mathbf{x}_i})$ given by a connectivity function $f : \mathcal{X} \rightarrow \mathbb{R}$. In this context, a path $\pi_{\mathbf{x}_i}$ is considered optimum if $f(\pi_{\mathbf{x}_i}) \geq f(\tau_{\mathbf{x}_i})$ for any other path $\tau_{\mathbf{x}_i}$. Such sort of functions are known as "smooth functions", and they figure important constraints that ensure the theoretic correctness of the OPF algorithm [18]. A more comprehensive discussion regarding the conditions necessary to guarantee the proper behavior of the algorithm is presented in the work of Ciesielski et al. [19].

Among different path-cost functions that have been proposed in the literature, unsupervised OPF employs the following formulation for $\forall \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$ such that $i \neq j$:

$$f(\langle \mathbf{x}_i \rangle) = \begin{cases} \rho(\mathbf{x}_i) & \text{if } \mathbf{x}_i \in \mathcal{R} \\ \rho(\mathbf{x}_i) - \delta & \text{otherwise,} \end{cases} \quad (2)$$

and

$$f(\pi_{\mathbf{x}_i} \cdot \langle \mathbf{x}_i, \mathbf{x}_j \rangle) = \min\{f(\pi_{\mathbf{x}_i}), \rho(\mathbf{x}_j)\}, \quad (3)$$

where $\delta = \min_{\forall (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{A}_k | \rho(t) \neq \rho(s)} |\rho(t) - \rho(s)|$. In a nutshell, δ stands for the smallest quantity required to avoid plateaus in the regions nearby the prototypes (i.e., areas with the highest density).

Among all possible paths $\pi_{\mathbf{x}_i}$ from the maxima of the pdf, the method assigns to sample $\mathbf{x}_i$ a final path whose minimum density value along it is maximum. Such final path value is represented by a cost map $\mathcal{C}$, as follows:

$$\mathcal{C}(\mathbf{x}_i) = \max_{\forall \pi_{\mathbf{x}_j} \in (\mathcal{X}, \mathcal{A}_k), i \neq j} \{f(\pi_{\mathbf{x}_j} \cdot \langle \mathbf{x}_j, \mathbf{x}_i \rangle)\}. \quad (4)$$

The OPF algorithm maximizes the connectivity map $\mathcal{C}(\mathbf{x}_i)$, $\forall \mathbf{x}_i \in \mathcal{X}$, by computing an optimum-path forest over the

dataset. Such forest is encoded as a predecessor map $\mathcal{P}$ with no cycles that assigns to each sample $\mathbf{x}_i \notin \mathcal{R}$ its predecessor $\mathcal{P}(\mathbf{x}_i)$ in the optimum path from $\mathcal{R}$, or a marker nil when $\mathbf{x}_i \in \mathcal{R}$.

The unsupervised OPF algorithm finds the number of the clusters on-the-fly, which means there is no need to have such information beforehand. The only parameter that needs to be set is the k_{max}, which constraints the search for suitable neighborhood sizes k^*.

III. Datasets

In this section, the most commonly used datasets for DA classification are described followed by the works considering them. Although there are different resources available for this task, most of the work based on unsupervised methods rely on domain-specific data, which are not publicly available, hindering the reproducibility of results and comparison among models. Furthermore, despite some datasets providing a specific split for train, evaluation and test sets, it is usually difficult to reproduce the exact same set of samples for each partition, as they commonly require further preprocessing such as grouping tags, removal of samples marked with error codes or removal of non-annotated utterances.

An interesting aspect of the available datasets is related to label frequency. In the International Computer Science Institute (ICSI) dataset [20], for instance, the most common class (*statement*) represents 59% of the data, being 4.2 times more frequent than the second most common label. This behavior can also be observed in the Switchboard dataset [5], where the most common class (*statement-non-opinion*) corresponds to about 36% of the samples, while 10 classes correspond to 1% of the data each, and other 25 classes together (more than half of the possible labels) represent only 5% of the entire dataset. It is important to mention that class imbalance is a widely known problem in machine learning, as the classifier can become biased towards the most frequent classes, making the training of effective models difficult.

A. HCRC Map Task dataset

The HCRC Map Task dataset [21] consists of the transcription of 128 dialogues between pairs of speakers: an instruction giver and a follower. By knowing that their maps are slightly different, the participants are asked to reproduce the route printed from the giver's map on the follower's map through verbal communication. The dataset is composed of 12 DAs and an additional *uncodable* class, however since there is no standard split of the dataset, different works use it in different ways, both in terms of splits and amount of classes to be considered.

Regarding supervised classification, Surendran and Levow [22] used Support Vector Machines (SVM) with Viterbi decoding and achieved 59.1% of accuracy considering just text features and using only the half of the dataset in which the participants cannot make eye contact. Tran et al. [23] obtained 63.3% of accuracy using hierarchical Long-Short Term Memory (LSTM) networks with attention mechanism, and Di Eugenio et al. [24] achieved 78.76% of classification accuracy with a k-Nearest Neighbors classifier using Feature Latent Semantic Analysis and considering as features not only the utterances but also the preceding DA, the current speaker, and the sub-dialogue type.

B. NPS Internet Chatroom Conversations dataset

The NPS Internet Chatroom Conversations dataset [25] consists of utterances extracted from 15 online chat rooms and exhibits domain-specific characteristics, such as the presence of slangs, emojis, and misspellings. The dataset does not provide standard splits and it is annotated with 15 classes, one of which corresponds to system notifications. Moldovan et al. [26] achieved 78.35% accuracy using a supervised Naïve Bayes classifier in this dataset. Concerning unsupervised approaches, Jo et al. [10] obtained a V-measure score of 0.33 using graphical models. Furthermore, these authors also applied the works from Brychcín and Král [7], Ezen-Can and Boyer [27], and from Lee et al. [28] in this dataset, achieving a V-measure of 0.28, 0.28 and 0.31, respectively.

C. Switchboard dataset

The Switchboard dataset [5] is formed by the transcription of 1,155 casual telephone conversations between pairs of participants about some random subject. This dataset contains 220 classes that can be clustered into 42 labels[1] which are commonly used in the literature for DA classification. Stolcke et al. [1] provide a split for training and testing models, while Lee and Dernoncourt [29] further sub-divide the first partition into train and development sets. Interestingly, it is possible to observe that three of the less frequent classes are not present in the standard test split.

In terms of supervised classification, Stolcke et al. [1] obtained 71% of accuracy using HMM; Kalchbrenner and Blunsom [30] achieved 73.9% accuracy using Recurrent Convolutional Neural Networks; Tran et al. [23] reached 74.5% using hierarchical LSTM with attention mechanism and Kumar et al. [31] obtained 79.2% accuracy using bidirectional LSTM with Conditional Random Fields (CRF).

Regarding unsupervised classification, Yang et al. [6] used a k-means classifier considering only the ten most frequent DAs and 50,000 sentences, achieving accuracy of 78.62%; Brychcín and Král [7] achieved a 65.7% of F1 score using HMM with Multivariate Gaussian distributions in the same split provided by Stolcke et al. [1], but since the model's accuracy is not provided, it is difficult to compare this result with other approaches.

D. International Computer Science Institute Meeting dataset

The ICSI Meeting dataset [20] contains the transcription of 75 meetings of ICSI teams where each utterance is annotated with one of 11 general tags and a combination of 39 specific tags, producing a total of 2,083 different labels. However, it is possible to map these tags to only 5 classes using the mapping

[1] A clustering procedure is provided by Christopher Potts at http://compprag.christopherpotts.net/swda.html.

introduced by Ang et al. [32]. Additionally, a standard split is provided for train, evaluation and test partitions.

In supervised classification, Ortega et al. [33] achieved 84.3% accuracy using LSTM with attention mechanism; Lee and Dernoncourt [29] obtained 84.6% using Convolutional Neural Networks (CNN) with context information, and Kumar et al. [31] attained 90.9% of accuracy using bidirectional LSTM with CRF.

E. Dialog System Technology Challenges datasets

The Dialog System Technology Challenges, previously Dialog State Tracking Challenge (DSTC), stands for series of competitions held yearly since 2013 aiming to foster the development of models to understand and extract relevant information from utterances in a conversation [34]. In all the competitions up to 2016, the provided datasets consisted of conversation transcriptions only. In its first three editions [34] the released dataset was formed of human-machine conversations annotated with, apart from other information, DA labels. Notwithstanding, the participants were not required to perform this type of classification. On DSTC4 [35] and DSTC5 [36] an optional multi-label DA classification task was created based on the TourSG dataset [37]. Despite being formed only by conversations between humans, such dataset is not publicly available.

IV. Materials and Methods

In order to run the experiments, the ICSI, NPS and Map Task datasets were considered, since they are formed by conversations between humans, are publicly available and contain a reasonable amount of classes. This section is organized as follows: initially, the preprocessing steps for each dataset are described, followed by the feature extraction procedure to represent utterances as vectors, and then the evaluation procedure is described. The code used to preprocess and analyze each dataset is made available online[2].

A. Preprocessing

Since each dataset stems from different domains, specific preprocessing techniques were applied to them individually. Words were segmented from sentences using the Stanford Tokenizer [38] and were further lowercased. Special symbols, including numbers, were removed, unless explicitly noted. Additionally, stop words and words with a frequency smaller than f_{min}, which was determined empirically, were disregarded.

The authors of the NPS dataset have anonymized the identity of the chatroom participants by replacing their usernames with unique identifiers following a convention. These patterns were all replaced by the word *"user"*, aiming to map the names to a word that has a representation under the vectorial model, as described in Section IV-B. Following, links were removed and emojis, which are characteristic of this domain and are commonly used to express some kind of sentiment, were replaced by a unique emoji, which is also present in the vectorial model used. The original fifteen classes, including system notifications, were considered. For the Map Task dataset, the only additional preprocessing step consisted in removing all samples from the *"uncodable"* class.

Finally, concerning the ICSI dataset, the initial labels were mapped to the 5 classes introduced by Ang et al. [32] and an additional class that indicates samples that were not labeled. This latter type of utterance was then removed, along with two entire transcripts and all samples marked with error codes, as described in the dataset manual. Considering that the utterances were transcribed from records, the punctuation was kept as it consists solely of dots, which signalize end of an utterance, question marks and a special symbol indicating incomplete utterances. The characteristics of each dataset after these steps are summarized in Table I according to the number of samples, the size of the folds used to train, evaluate and test the models, the amount of classes $|C|$ and the vocabulary size $|V|$, which represents the number of unique words in each dataset.

B. Feature extraction

Word vector models, such as Glove [39], word2vec [40] and fastText [41], have become the cornerstone of modern NLP approaches. This can be attributed to the quality of the obtained results and the availability of pre-trained vectors in different languages. Furthermore, since these are unsupervised techniques, it is possible to train new models on unseen data without incurring the cost of labeling.

The main idea behind these methods consists in learning a fixed-length vector representation $\mathbf{v}_w$ for each word w in a vocabulary $\mathcal{V}$. This procedure aims to map words with similar syntactic or semantic meaning to similar vectors, which is measured through the cosine similarity function. An example showing that this approach is able to capture word sense consists in converting words to their corresponding vectors, performing some basic arithmetic operation in this space and then mapping the resulting vector back to its closest word. The most popular example of this operation is the transformation $king - man + woman = queen$ due to Mikolov et al. [42].

From the trained vectors it is possible to compute sentence representations in a number of ways, as briefly reviewed by Arora et al. [43], being the simplest of them to consider the average of the word vectors that form the utterance of interest. These authors also propose a simple approach that, in some scenarios, outperforms more complex alternatives based in Recurrent Neural Networks. According to this technique, the initial representation $\mathbf{x}'_i$ for each utterance s_i from a dataset of utterances $\mathcal{S}$ is computed using the following equation:

$$\mathbf{x}'_i = \frac{1}{|s_i|} \sum_{w \in s_i} \frac{a}{a + p(w)} \mathbf{v}_w, \tag{5}$$

where $p(w)$ is the probability of occurrence of the word w in the dataset used to train the word vector model, $|s_i|$ is the utterance length, and a is a smoothing parameter. The obtained vectors $\mathbf{x}'_i$ are grouped, forming a matrix $E \in \mathbb{R}^{|\mathcal{S}| \times d}$, where $|\mathcal{S}|$ is the amount of utterances in the dataset and d is the word vectors dimensionality. Next, let $\mathbf{r}$ be the first singular

[2]https://github.com/lzfelix/sibgrapi2018_opf

TABLE I
CHARACTERISTICS OF THE DATASETS USED IN THE WORK.

Dataset	$\|C\|$	$\|V\|$	f_{min}	# Samples	Fold size
NPS	15	3,885	1	10,568	697
Map Task	12	1,052	2	26,158	1,738
ICSI	5	5,004	3	106,047	7,068

vector from E, then the final sentence vectors are computed as follows:

$$\mathbf{x}_i = \mathbf{x}'_i - \mathbf{r}\mathbf{r}^T\mathbf{x}'_i, \tag{6}$$

which are further normalized to have unitary length. In our experiments, words that are present in the utterances but not in $\mathcal{V}$ are completely ignored. We empirically chose to use the 300-dimensional pre-trained GloVe word vectors[3] and $a = 10^{-3}$, observing the analysis developed by Arora et al. [43].

C. Evaluation procedure

To train and evaluate the results produced by k-means[4], HDBSCAN[5] and OPF[6] classifiers, a 15-fold cross-validation procedure was used. Each dataset was split into 15 partitions, of which 14 were used to project the classifiers and one for testing. From the 14 parts, 13 were used for training and one for fine-tuning. This process was repeated 15 times, ensuring that each part was used once for validation and once for testing. The validation set was used to fine-tune the hyper-parameters, namely: k for the k-means, β_1 and β_2 for HDBSCAN, and the ranges $[k_{min}, k_{max}]$ for the OPF classifier. Regarding the latter, despite the possibility of fixing $k_{min} = 1$ and setting k_{max} in a way to create a large search interval for k^*, we used multiple small ranges instead to study how this value influences the quality of the generated clusters.

It is worth mentioning that the HDBSCAN hyper-parameters β_1 and β_2 control, respectively, the minimum amount of data necessary to form a cluster and how rigorously the algorithm marks samples as noise, where higher values correspond to more samples being assigned to this class.

After the partitioning procedure, it is necessary to determine to which classes from the dataset each cluster corresponds to. For the k-means and HDBSCAN classifiers, we label the partitions with the most frequent class among its elements. On the other hand, in its original conception, OPF propagates the ground-truth label from the prototype to all conquered samples. In our experiments, we propose an alternative approach to label the clusters with the majority class as well, hereinafter named M-OPF. Notice that the clusters formed by OPF and M-OPF are identical, being just the label propagation strategy different.

[3]Available at http://nlp.stanford.edu/data/glove.840B.300d.zip

[4]We used the implementation from `scikit-learn`.

[5]We used the implementation from `scikit-learn-contrib`.

[6]We used the modified implementation of `LibOPF` from https://github.com/lzfelix/LibOPF/tree/unsupervised.

To evaluate the results we consider the accuracy and the V-measure [44], which consists in the harmonic mean between homogeneity (h), in Equation 7, and completeness (c), in Equation 8, with $h \in [0, 1]$ and $c \in [0, 1]$:

$$h = \begin{cases} 1 & \text{if } |C| = 1, \\ 1 - \frac{H(C|K)}{H(C)} & \text{otherwise.} \end{cases} \tag{7}$$

$$c = \begin{cases} 1, & \text{if } |K| = 1, \\ 1 - \frac{H(K|C)}{H(K)} & \text{otherwise,} \end{cases} \tag{8}$$

where C and K are, respectively, the sets of ground truth classes and clusters generated after the mapping procedure is applied, $|\cdot|$ represents set size, N is the number of samples clustered, α_{ck} is the number of samples from the c-th ground truth class allocated into the k-th cluster, and

$$\begin{aligned} H(C|K) &= -\sum_{k=1}^{|K|}\sum_{c=1}^{|C|}\frac{\alpha_{ck}}{N}\log\frac{\alpha_{ck}}{\sum_{d=1}^{|C|}\alpha_{dk}}, \\ H(C) &= -\sum_{c=1}^{|C|}\frac{\sum_{k=1}^{|K|}\alpha_{ck}}{|C|}\log\frac{\sum_{k=1}^{|K|}\alpha_{ck}}{|C|}, \\ H(K|C) &= -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{\alpha_{ck}}{N}\log\frac{\alpha_{ck}}{\sum_{d=1}^{|K|}\alpha_{cd}}, \\ H(K) &= -\sum_{k=1}^{|K|}\frac{\sum_{c=1}^{|C|}\alpha_{ck}}{|C|}\log\frac{\sum_{c=1}^{|C|}\alpha_{ck}}{|C|}. \end{aligned} \tag{9}$$

Homogeneity measures to what extent each cluster contains samples from a single ground truth class, while completeness evaluates the ability of the algorithm to allocate all samples of the same ground truth class to a single data partition. In the extreme case where every cluster is formed by a single sample, $h = 1$ as each partition is as homogeneous as possible, but $c = 0$, since the algorithm fails to group all samples of the same ground truth class under a single cluster. Contrariwise, $h = 0$ and $c = 1$ if all samples are allocated to the same cluster, as the latter criterion is satisfied at the cost of having a single cluster as heterogeneous as possible. In both cases the V-measure is zero, considering that it is the harmonic mean of such metrics. Finally, the obtained results are analyzed under the Wilcoxon signed-rank test [45] with $p = 0.05$.

V. EXPERIMENTAL RESULTS

This section describes the results obtained in the considered datasets. The best hyper-parameters were searched in the intervals described in Table II. For the k-means classifier, we use the interval $[2, 15]$ when training it in ICSI dataset, as it is known beforehand that there are only five classes in this scenario. Regarding the HDBSCAN, as every sample belongs to one of the defined classes, the samples identified as noise are aggregated in a single cluster, which is also labeled according to the most common class among its elements. Besides, during preliminary tests, we observed that increasing β_2 only harms

TABLE II
HYPER-PARAMETERS SEARCH VALUES.

Models	Hyper-parameters
k-means	$k \in [2, 15]$ or $k \in [2, 20]$
OPF, M-OPF	$[k_{min}, k_{max}] \in \{[1, 5], [5, 10], [10, 20], [20, 50]$ $[50, 100], [100, 150]\}$
HDBSCAN	$\beta_1 \in \{5, 10, 15, 20, 25, 30, 35, 40\}$; $\beta_2 = 1$

TABLE III
EXPERIMENTAL RESULTS FOR EACH CLASSIFIER.

Models	ICSI	Map Task	NPS
Homogeneity			
k-Means	0.49 ± 0.02	0.24 ± 0.01	0.41 ± 0.02
OPF	0.47 ± 0.03	0.21 ± 0.02	**0.51 ± 0.02**
M-OPF	**0.55 ± 0.01**	0.29 ± 0.01	**0.50 ± 0.02**
HDBSCAN	0.42 ± 0.01	**0.34 ± 0.01**	0.44 ± 0.02
Completeness			
k-Means	**0.57 ± 0.02**	0.32 ± 0.01	0.62 ± 0.04
OPF	0.47 ± 0.02	0.21 ± 0.01	0.48 ± 0.02
M-OPF	0.55 ± 0.01	0.33 ± 0.02	0.48 ± 0.02
HDBSCAN	0.50 ± 0.01	**0.36 ± 0.01**	**0.66 ± 0.03**
V-Measure			
k-means	0.53 ± 0.02	0.28 ± 0.01	0.49 ± 0.02
OPF	0.47 ± 0.02	0.21 ± 0.01	0.50 ± 0.02
M-OPF	**0.55 ± 0.01**	0.28 ± 0.01	**0.54 ± 0.02**
HDBSCAN	0.46 ± 0.01	**0.30 ± 0.01**	0.53 ± 0.02
Accuracy			
k-means	80.18 ± 0.63	41.20 ± 1.30	68.72 ± 1.63
OPF	78.81 ± 1.57	34.96 ± 1.65	64.25 ± 1.64
M-OPF	**83.09 ± 0.35**	**43.55 ± 1.17**	**71.31 ± 1.59**
HDBSCAN	77.64 ± 0.39	**43.77 ± 1.00**	69.93 ± 2.07

the performance of this classifier, therefore we fixed its value to 1.

The performance of each classifier is shown in Table III, where the best results, according to the Wilcoxon signed-rank test, are displayed in bold. The hyper-parameters found during the fine-tuning process for each classifier were $k = 20$ for k-means in all datasets, except for ICSI, where $k = 15$. For OPF and M-OPF, we used $k^* = 5$, and regarding HDBSCAN we employed $\beta_1 = 5$ and $\beta_2 = 1$.

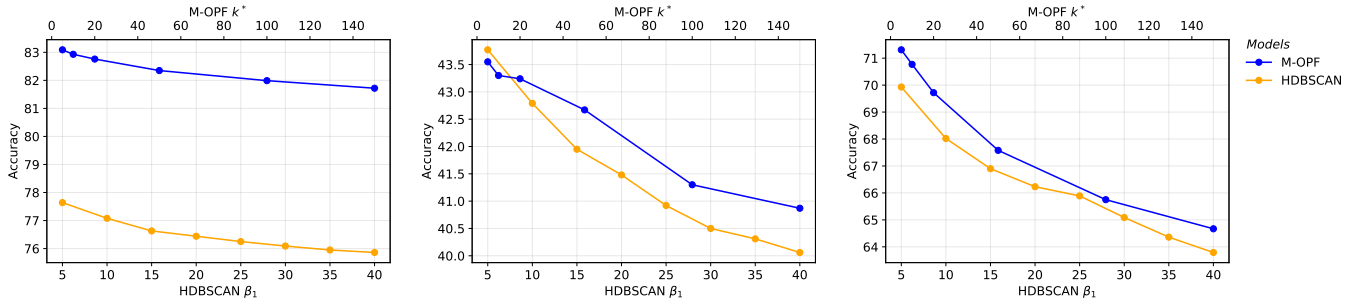
By analyzing the homogeneity measurements, we can infer that M-OPF favors the creation of clusters that better represent each class when compared to other approaches. This gain is due to the fact that M-OPF makes the most common ground truth label in each cluster to conquer all of its samples, differently from the original approach proposed by OPF, where the ground truth label of the prototype, which might not be too common in that cluster, ends up ruling it. Again, it is important to highlight that the difference in results between OPF and M-OPF is due only to the strategy used to determine to which class each cluster corresponds to. When completeness is considered, we can observe that HDBSCAN is best suited for fitting more samples of the same class into a single cluster. This aspect may be due to the fact that such a classifier does not assume that the underlying data follow a Gaussian distribution, hence being able to generate clusters of arbitrary shapes. Notwithstanding, grouping more samples of the same class within a single cluster through this method comes at the expense of creating data partitions with a greater diversity of ground truth labels, which is reflected in its homogeneity and V-measure. Finally, when homogeneity and completeness are considered through this latter metric, we can observe that M-OPF offers the best compromise between both aspects in two datasets, being slightly worse than HDBSCAN in the Map Task scenario only.

Regarding accuracy, M-OPF outperforms the other models, except in the Map Task dataset, although the difference among values is not statistically significant. Furthermore, in the other cases, M-OPF has the smallest standard deviation for this metric, indicating better stability. Following, we analyze the accuracy results for M-OPF and HDBSCAN in terms of their hyper-parameters in order to gain a better understanding of the role played by these values. Figure 1 displays the classifiers performance as k^* and β_1 vary. We chose to compare only two techniques since the hyper-parameters for each algorithm have different intervals. Furthermore, HDBSCAN also shows good results under the V-measure.

Although HDBSCAN has two hyper-parameters (β_2 was initially set to 1), M-OPF provides higher accuracy results than the best version of the first classifier, regardless the choice of its unique hyper-parameter in the ICSI dataset, as shown in Figure 1a. Regarding the Map Task scenario, both classifiers yield statistically similar accuracy results for the best hyper-parameters found, however, when $k^* > 5$ and $\beta_1 > 5$, M-OPF becomes better than HDBSCAN with statistical significance, as displayed in Figure 1b. According to Figure 1c, M-OPF starts getting worse than HDBSCAN in the NPS dataset when $k^* = 20$, however, this difference is not statistically significant when compared to the best version of HDBSCAN. Consequently, the M-OPF only becomes worse than the latter with $k^* = 50$. Therefore, it is reasonable to conclude that M-OPF, and consequently OPF, are less sensitive to hyper-parameter choice when compared to the other considered models, presumably requiring less domain knowledge in order to provide good results.

VI. CONCLUSIONS

In this paper, we introduced the unsupervised OPF classifier to the NLP field, more specifically to the DA classification task. Additionally, we proposed a small modification in the strategy used to determine to which class each of the clusters formed by OPF correspond to. Instead of propagating the ground-truth label from the cluster prototype to all of its conquered samples, we use a majority voting procedure. Albeit simple and useful only when the ground-truth is available, such procedure helps to evidence the clustering power of the OPF algorithm, as the data partitions formed by OPF and M-OPF are identical. Under this regime, M-OPF showed good results in the three considered datasets concerning accuracy and V-measure. Furthermore, M-OPF showed to be less sensitive to



(a) (b) (c)

Fig. 1. Comparison of accuracy results under different values of k^* for M-OPF and β_1 for HDBSCAN considering (a) ICSI (b) Map Task and (c) NPS datasets.

hyper-parameter fine-tuning results than HDBSCAN. Finally, despite the existence of different datasets publicly available for DA classification, most of the work in this direction rely on supervised techniques, leaving opportunities yet to be explored by unsupervised approaches.

Acknowledgments

The authors are grateful to FAPESP grants #2013/07375-0, #2014/16250-9, #2014/12236-1, and #2016/19403-6, Capes, and also CNPq grant #307066/2017-7.

References

[1] A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. V. Ess-Dykema, and M. Meteer, "Dialogue act modeling for automatic tagging and recognition of conversational speech," *Computational Linguistics*, vol. 26, no. 3, pp. 339–373, 2000.

[2] A. Ritter, C. Cherry, and B. Dolan, "Unsupervised modeling of twitter conversations," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 172–180.

[3] K. Perumal and G. Hirst, "Semi-supervised and unsupervised categorization of posts in web discussion forums using part-of-speech information and minimal features," in *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2016, pp. 100–108.

[4] L. Wang, M. Lui, S. N. Kim, J. Nivre, and T. Baldwin, "Predicting thread discourse structure over technical web forums," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 13–25.

[5] D. Jurafsky, "Switchboard swbd-damsl shallow-discourse-function annotation coders manual," *Institute of Cognitive Science Technical Report*, 1997.

[6] X. Yang, J. Liu, Z. Chen, and W. Wu, "Semi-supervised learning of dialogue acts using sentence similarity based on word embeddings," in *Audio, Language and Image Processing (ICALIP), 2014 International Conference on*. IEEE, 2014, pp. 882–886.

[7] T. Brychcín and P. Král, "Unsupervised dialogue act induction using gaussian mixtures," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2017, pp. 485–490.

[8] R. Higashinaka, N. Kawamae, K. Sadamitsu, Y. Minami, T. Meguro, K. Dohsaka, and H. Inagaki, "Unsupervised clustering of utterances using non-parametric bayesian methods," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[9] A. Ezen-Can, J. F. Grafsgaard, J. C. Lester, and K. E. Boyer, "Classifying student dialogue acts with multimodal learning analytics," in *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*. ACM, 2015, pp. 280–289.

[10] Y. Jo, M. M. Yoder, H. Jang, and C. P. Rosé, "Modeling dialogue acts with content word filtering and speaker preferences," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 2169–2179.

[11] J. P. Papa, A. X. Falcão, and C. T. N. Suzuki, "Supervised pattern classification based on optimum-path forest," *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 120–131, 2009.

[12] J. P. Papa, A. X. Falcão, V. H. C. Albuquerque, and J. M. R. S. Tavares, "Efficient supervised optimum-path forest classification for large datasets," *Pattern Recognition*, vol. 45, no. 1, pp. 512–520, 2012.

[13] J. P. Papa, S. E. N. Fernandes, and A. X. Falcão, "Optimum-path forest based on k-connectivity: Theory and applications," *Pattern Recognition Letters*, vol. 87, pp. 117–126, 2017.

[14] L. M. Rocha, F. A. M. Cappabianco, and A. X. Falcão, "Data clustering as an optimum-path forest problem with applications in image analysis," *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 50–68, 2009.

[15] W. P. Amorim, A. X. Falcão, J. P. Papa, and M. H. Carvalho, "Improving semi-supervised learning through optimum connectivity," *Pattern Recognition*, vol. 60, no. Supplement C, pp. 72–85, 2016.

[16] R. J. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 2013, pp. 160–172.

[17] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[18] A. X. Falcão, J. Stolfi, and R. A. Lotufo, "The image foresting transform: theory, algorithms, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 19–29, 2004.

[19] K. C. Ciesielski, A. X. Falcão, and P. A. V. Miranda, "Path-value functions for which dijkstra's algorithm returns optimal mapping," *Journal of Mathematical Imaging and Vision*, vol. 60, no. 7, pp. 1025–1036, Sep 2018. [Online]. Available: https://doi.org/10.1007/s10851-018-0793-1

[20] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters, "The icsi meeting corpus," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, vol. 1, April 2003, pp. I–364–I–367 vol.1.

[21] J. Carletta, S. Isard, G. Doherty-Sneddon, A. Isard, J. C. Kowtko, and A. H. Anderson, "The reliability of a dialogue structure coding scheme," *Computational linguistics*, vol. 23, no. 1, pp. 13–31, 1997.

[22] D. Surendran and G.-A. Levow, "Dialog act tagging with support vector machines and hidden markov models," in *Ninth International Conference on Spoken Language Processing*, 2006.

[23] Q. H. Tran, I. Zukerman, and G. Haffari, "A hierarchical neural model for learning sequences of dialogue acts," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, vol. 1, 2017, pp. 428–437.

[24] B. Di Eugenio, Z. Xie, and R. Serafin, "Dialogue act classification, higher order dialogue structure, and instance-based learning," *Dialogue & Discourse*, vol. 1, no. 2, pp. 1–24, 2010.

[25] E. N. Forsythand and C. H. Martell, "Lexical and discourse analysis

of online chat dialog," in *Semantic Computing, 2007. ICSC 2007. International Conference on*. IEEE, 2007, pp. 19–26.

[26] C. Moldovan, V. Rus, and A. C. Graesser, "Automated speech act classification for online chat." *The 22nd Midwest Artificial Intelligence and Cognitive Science Conference*, vol. 710, pp. 23–29, 2011.

[27] A. Ezen-Can and K. E. Boyer, "Understanding student language: An unsupervised dialogue act classification approach," *Journal of Educational Data Mining (JEDM)*, vol. 7, no. 1, pp. 51–78, 2015.

[28] D. Lee, M. Jeong, K. Kim, S. Ryu, and G. G. Lee, "Unsupervised spoken language understanding for a multi-domain dialog system," *IEEE Transactions On Audio, Speech, and Language Processing*, vol. 21, no. 11, pp. 2451–2464, 2013.

[29] J. Y. Lee and F. Dernoncourt, "Sequential short-text classification with recurrent and convolutional neural networks," *arXiv preprint arXiv:1603.03827*, 2016.

[30] N. Kalchbrenner and P. Blunsom, "Recurrent convolutional neural networks for discourse compositionality," *arXiv preprint arXiv:1306.3584*, 2013.

[31] H. Kumar, A. Agarwal, R. Dasgupta, S. Joshi, and A. Kumar, "Dialogue act sequence labeling using hierarchical encoder with crf," *arXiv preprint arXiv:1709.04250*, 2017.

[32] J. Ang, Y. Liu, and E. Shriberg, "Automatic dialog act segmentation and classification in multiparty meetings," in *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 1, March 2005, pp. 1061–1064.

[33] D. Ortega and N. T. Vu, "Neural-based context representation learning for dialog act classification," *arXiv preprint arXiv:1708.02561*, 2017.

[34] J. Williams, A. Raux, and M. Henderson, "The dialog state tracking challenge series: A review," *Dialogue & Discourse*, vol. 7, no. 3, pp. 4–33, 2016.

[35] S. Kim, L. F. D'Haro, R. E. Banchs, J. D. Williams, and M. Henderson, "The fourth dialog state tracking challenge," in *Proceedings of the 7th International Workshop on Spoken Dialogue Systems (IWSDS)*. Springer, 2016.

[36] S. Kim, L. F. D'Haro, R. E. Banchs, J. D. Williams, M. Henderson, and K. Yoshino, "The fifth dialog state tracking challenge," in *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 511–517.

[37] J. Williams, A. Raux, D. Ramachandran, and A. Black, "The dialog state tracking challenge," in *Proceedings of the Special Interest Group on Discourse and Dialogue 2013 Conference*, 2013, pp. 404–413.

[38] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Association for Computational Linguistics (ACL) System Demonstrations*, 2014, pp. 55–60. [Online]. Available: http://www.aclweb.org/anthology/P/P14/P14-5010

[39] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1532–1543.

[40] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[41] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv preprint arXiv:1607.04606*, 2016.

[42] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[43] S. Arora, Y. Liang, and T. Ma, "A simple but tough-to-beat baseline for sentence embeddings," *International Conference on Learning Representations*, 2017.

[44] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 2007.

[45] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.