OvNMTF Algorithm: an Overlapping Non-Negative Matrix Tri-Factorization for Coclustering

Waldyr L. de Freitas Junior, Sarajane M. Peres, Valdinei Freire
Escola de Artes, Ciências e Humanidades
Universidade de São Paulo
São Paulo, Brasil
waldyrjunior@usp.br,sarajane@usp.br,valdinei.freire@usp.br

Lucas Fernandes Brunialti
Cobli
São Paulo, Brasil
lucas@cobli.co

***Abstract*—Coclustering algorithms are an alternative to classic one-sided clustering algorithms. Because of its ability to simultaneously cluster rows and columns of a dyadic data matrix, coclustering offers a higher value-added information: it offers column clusters besides row clusters, and the relationship between them in terms of coclusters. Different structures of coclusters are possible, and those that overlap in terms of rows or columns still represent an open question with room for improvements. In addition, while most related literature cites coclustering as a means of producing better results from one-side clustering, few initiatives study it as a tool capable of providing higher quality descriptive information about this clustering. In this paper, we present a new coclustering algorithm - OvNMTF, based on triple matrix factorization, which properly handle overlapped coclusters, by adding degrees of freedom for matrix factorization that enable the discovery of specialized column clusters for each row cluster. As a proof of concept, we modeled text analysis as a coclustering problem with column overlaps, assuming that given words (data matrix columns) are associated with over one document cluster (row cluster) because they can assume different semantic relationships in each association. Experiments on synthetic data sets show the OvNMTF algorithm reasonableness; experiments on real-world text data show its power for extracting high quality information.**

***Index Terms*—coclustering, matrix factorization**

I. Introduction

In clustering analysis, we use similarity between data to discover patterns that characterize them and their relationships [1]. This process organizes the data points into clusters to maximize the similarity between those in the same cluster and minimize similarity between those organized into distinct clusters [2]. One of the possible strategies to implement such a process is to partition the rows of a data matrix [3] where the rows represent the data under analysis and the columns represent the data descriptive attributes. In principle, the similarity analysis performed in clustering process considers all attributes, resulting in a holistic analysis [4].

Alternatively, in a coclustering problems, we implement pattern discovery through similarity analysis applied simultaneously to data and attributes [5], i.e., data clustering is based on the distributions of attributes and attributes clustering is based on distribution of data [6]. Coclustering gives greater flexibility in defining clusters because it can perform partial similarity analysis and offer more precise data clustering. Besides, this process results in a cocluster structure that embed more detailed data clusters descriptions. This way of formulating descriptive data analysis has been promising for real problems characterized by subjective patterns interpretations, as in image and text data analysis [4], [5], [7]–[9].

In [7], the authors apply coclustering in text data to explore the structure of coclusters and easily[1] identify polysemic words and their context. Considering that coclusters represent relationships between row clusters (document clusters) and column clusters (word clusters) and recognizing that a word can take on different meanings depending on the context (document clusters), the usefulness of overlapping coclusters becomes noticeable. In this paper, we present a new algorithm OvNMTF (Overlapping Non-negative Matrix Tri-Factorization) capable of finding a coclustering solution that adequately addresses the coclusters overlap problem. This algorithm represents an evolution of our previous one, the BinOvNMTF (Overlapping Binary Non-negative Matrix Tri-Factorization) algorithm [10], which we proposed earlier and was restricted to making binary associations between row clusters and column clusters.

Matrix factorization methods have been widely applied in dyadic data analysis [5], [7], [11], [12], mainly for text data analysis. Non-Negative Matrix Factorization (NMF) is the basis of clustering and coclustering algorithms, such as: NMF for partial similarities-based data analysis [4], NMF for clustering [13], NMF for coclustering [14], Low Rank NMF [15], Semi-NMF (SNMF) [16], Orthogonal NMTF (ONMTF) [17], Graph regularized NMF (GNMF) [18], Fast NMTF (FNMTF) [8], BinOvNMTF [10], Word co-occurrence regularized NMTF (WC-NMTF) [19]. All of these algorithms were presented accompanied by experiments involving text data. The duality between rows and columns explored by coclustering algorithms has been shown to be effective in high dimensionality and sparse spaces [5], characteristic of the vector representation used for text data.

The research related to this class of algorithms shows that there is still room for improvement regarding the treatment of overlapping coclusters. Figure 1 illustrates this with three data matrices (I, II and III) with positive real values: the darker the

[1] In this case, "easily" means "with no further post-processing efforts", since in text coclustering results there is information about word clusters besides information about document clusters.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

blue color the higher the value in the data matrix cell; and the reconstruction of the original datasets by combining the matrices resulting from matrix factorization carried out by: the widely-known clustering algorithm *k-Means* [20] and the coclustering algorithms ONMTF, FNMTF, BinOvNMTF and OvNMTF. For this discussion, we have assumed that the algorithms' parameters should be established according to *a priori* knowledge: the three datasets comprise data distributions with three row clusters and three column clusters[2].

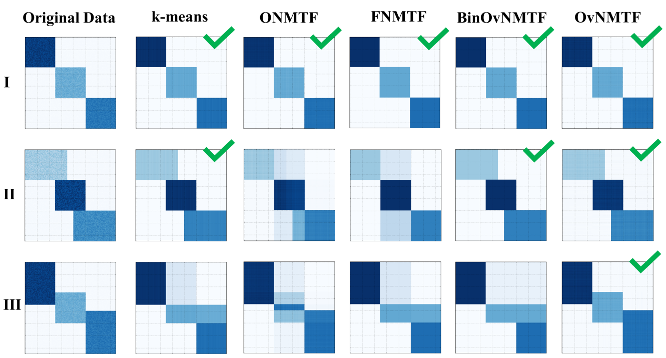


Fig. 1. Capacity to reconstruct the original data matrices of clustering and coclustering algorithms. The check symbol shows the proper reconstructions.

Figure 1 shows that both clustering algorithms have good reconstruction capability if the hypothesis of row clusters does not assume overlap (I and II). For these algorithms, the hypothesis of overlapping column clusters (III) does not matter because the problem is one-side (row) clustering. Under such test conditions, the ONMTF and FNMTF coclustering algorithms fail in reconstructing for both overlapping cases (in rows (II) and columns (III)). The BinOvNMTF coclustering algorithm can handle overlap in one dimension only (II). The OvNMTF coclustering algorithm overcomes the difficulties and presents good reconstruction in both overlapping situations. Therefore, the contributions of this paper are: (i) formalization for the non-negative triple matrix factorization problem with row or column overlaps (the OvNMTF problem); (ii) an algorithm, based on multiplicative update rules, for solving the OvNMTF problem (the OvNMTF algorithm); and (iii) a proof of concept for illustrating the effects of applying the OvNMTF problem on real-world text datasets.

This paper is organized as follows: Section II presents the theoretical background that supports the introduction of the OvNMTF problem, whereas Section III presents the OvNMTF algorithm; the experiments are discussed in Section IV; Section V presents the conclusions.

II. Theoretical Background

Let a data matrix $X \in \mathbb{R}^{n \times m}$, with n rows (datapoints) and m columns (attributes), in which the set of rows (or the set of datapoints) can be interpreted as a set of vectors $N = \{x_{1\cdot}, \ldots, x_{n\cdot}\}$. In a clustering problem, we expect to find k parts of N, denoted by subsets $\mathcal{K} \subseteq N$, being $p \in \{1, \ldots, k\}$. The set $\mathcal{K} = \{K_1, \ldots K_k\}$ is said to be the resulting clusters of rows that solve the clustering problem. From the standpoint of coclustering problems, X comprises a set of row vectors $N = \{x_{1\cdot}, \ldots, x_{n\cdot}\}$ and a set of columns vectors $M = \{x_{\cdot 1}, \ldots, x_{\cdot m}\}$, and we expect to find $k \times l$ coclusters represented by submatrices in X denoted by $X_{K_p L_q}$, being k subsets $\mathcal{K} \subseteq N$, l subsets $\mathcal{L} \subseteq M$, $p \in \{1, \ldots, k\}$ and $q \in \{1, \ldots, l\}$. In a solution to coclustering problems, the cocluster $X_{K_p L_q}$ is a cluster of datapoints in K_p, in view of the attributes in L_q. In this section, we present the theoretical background concerning clustering and coclustering problems. Such problems are detailed for one of the following reasons: the problem is the basis for developing OvNMTF; the algorithm that solves the problem was used in the experiments.

[2]For all algorithms, the parameters referring to the number of rows/columns clusters sought were set to 3.

A. K-Means

The *k-means* clustering problem is one of the most studied problems in the clustering field. This problem is classically solved by applying the *Lloyd* algorithm, also called the *k-means* algorithm [21]. The goal is to find k prototype vectors that quantize a dataset vector space, regarding a minimal vector quantization error. Here, as in [22], we elaborate the *k-means* clustering problem as the factorization of the data matrix X into two matrices, U as a cluster indicator matrix and C as a prototype vector matrix, so that $X \approx UC$; $\|X - UC\|_F^2$ gives a reconstruction error of the original data matrix (see $\mathcal{F}_1$).

$$\mathcal{F}_1(U, C) = \min_{U,C} \sum_{i=1}^{n} \sum_{p=1}^{k} u_{ip} \|x_{i\cdot} - c_{p\cdot}\|^2 = \min_{U,C} \|X - UC\|_F^2$$
$$\text{subj. to } U \in \Psi^{n \times k},\ C \in \mathbb{R}^{k \times m},\ \textstyle\sum_{p=1}^{k} u_{ip} = 1\ \forall i,$$

in which $\Psi = \{0, 1\}$ and $\| \cdot \|_F$ is the *Frobenius* norm for matrices.

B. NMF

Two reasons motivate the use of Non-Negative Matrix Factorization (NMF) for clustering problems resolution: the possibility of applying it as a data analysis method capable of extracting knowledge about an object from the study of its parts [4], therefore implementing partial similarity-based analysis; the adequacy of data representation used in various clustering contexts to the factorization method requirements, since such representations cover the relationship between pairs of elements coming from two distinct finite sets (dyadic data) [5]. For example, in text data clustering, two finite sets are used to represent texts: documents and words. A positive data matrix organizes information regarding the occurrence or absence of a word in a document (a dyadic relationship), allowing the use of the NMF method. NMF-based algorithms have as input a data matrix $X \in \mathbb{R}_+^{n \times m}$, with n rows that constitutes a set of row vectors $N = \{x_{1\cdot}, \ldots, x_{n\cdot}\}$, and m columns that constitutes a set of columns vectors $M = \{x_{\cdot 1}, \ldots, x_{\cdot m}\}$. The relation between each row $x_{i\cdot}$ and each column $x_{\cdot j}$ is represented by x_{ij}, with $i \in \{1, \ldots, n\}$

and $j \in \{1, \ldots, m\}$ [4]. NMF can be seen as a double factor decomposition, in the form of the problem $\mathcal{F}_2$:

$$\mathcal{F}_2(U,V) = \min_{U,V} \|X - UV^T\|_F^2, \\ \text{subj. to } U \geq 0, V \geq 0,$$

in which $U \in \mathbb{R}_+^{n\times k}$, $V \in \mathbb{R}_+^{m\times k}$, $\|\cdot\|_F$ is the *Frobenius* norm for matrices and $\|X - UV^T\|_F^2$ gives the reconstruction error. The *Frobenius* norm and the reconstruction error are also adopted in the other problems formulated in this and in the following sections.

According to [7], the columns in the factor matrix V correspond to basis vectors for the original data matrix reconstruction, while each row in the factor matrix U represents an encoding that gives the extent to which each basis vector will be used in the reconstruction process. Thus, the columns in V can be seen as the prototype vectors for row clusters extracted from the original data matrix.

C. BVD

Block Value Decomposition (BVD) searches for hidden block structures in a data matrix and can be used for analysis of dyadic data [5]. It is suitable for implementing coclustering solutions because BVD considers both data dimensions (rows and columns) simultaneously and explores their relationship by decomposing the data matrix $X \in \mathbb{R}^{n\times m}$ into three matrices ($\mathcal{F}_3$): U as a row coefficient matrix, S as a block-structured matrix and V as a column coefficient matrix.

$$\mathcal{F}_3(U,S,V) = \min_{U,S,V} \|X - USV^T\|_F^2, \\ \text{subj. to } U \geq 0, V \geq 0$$

in which $U \in \mathbb{R}_+^{n\times k}, S \in \mathbb{R}^{k\times l}$ and $V \in \mathbb{R}_+^{m\times l}$.

The problem $\mathcal{F}_3$ is an alternative to the problem $\mathcal{F}_2$ because it uses triple factorization of matrices and can give us a coclustering structure. The authors in [5] provide the following interpretation: S is a compact representation of X, the matrix US contains basis vectors for the columns in X, the matrix SV^T contains basis vectors for rows in X, and the factor matrices U and V denote the extent to which rows and columns are associated with their respective row/column clusters. Thus, prototype vectors can be extracted for both row and column clusters, and the notion of coclusters can be explored by examining the information contained in the factor matrix S, as in [7]. The BVD problem restricted to positive data matrix, i. e., $X \in \mathbb{R}_+^{n\times m}$, results in the NBVD (Non-negative Block Value Decomposition) problem [5].

D. ONMTF

The problem $\mathcal{F}_4$ was proposed in [17]. In this problem, in addition to the nonnegativity constraints used in the $\mathcal{F}_2$ and the triple factorization used in the $\mathcal{F}_3$, two orthogonality constraints were added for row clusters and column clusters matrices, respectively: $U^TU = I$ and $V^TV = I$, in which I is an identity matrix. These constraints restrict the problem of factoring $X \approx USV^T$ to a smaller number of possible solutions with more rigorous interpretation.

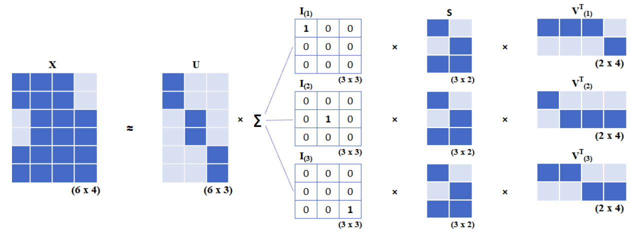


Fig. 2. OvNMTF factorization process with five factor matrices

$$\mathcal{F}_4(U,S,V) = \min_{U,S,V} \|X - USV^T\|_F^2 \\ \text{subj. to } U \geq 0, S \geq 0,\ V \geq 0, U^TU = I,\ V^TV = I,$$

in which $U \in \mathbb{R}_+^{n\times k}, S \in \mathbb{R}_+^{k\times l}$ and $V \in \mathbb{R}_+^{m\times l}$.

III. OvNMTF

In this section, we formalize the Overlapping Non-negative Matrix Tri-Factorization (OvNMTF) problem and introduce an algorithm based on multiplicative update rules for solving it. The OvNMTF problem (Problem $\mathcal{F}_5$) is based on the assumptions established in NMF and BVD problems. We formulate ($\mathcal{F}_5$) as:

$$\mathcal{F}_5(U,S,V_{(1)},\ldots,V_{(k)}) = \\ \min_{U,S,V_{(1)},\ldots,V_{(k)}} \|X - U\sum_{p=1}^{k} I_{(p)}SV_{(p)}^T\|_F^2 \\ \text{subj. to } U \geq 0, S \geq 0, V_{(p)} \geq 0, \quad \forall p$$

in which $U \in \mathbb{R}_+^{n\times k}$, $S \in \mathbb{R}_+^{k\times l}$, $V_{(p)} \in \mathbb{R}_+^{m\times l}$, $p \in \{1,\ldots,k\}$ as the index for the set of matrices $\{V_{(1)},\ldots,V_{(k)}\}$, $I_{(p)} \in \{0,1\}^{k\times k}$ are constant selector matrices with zero in all cells except the unique cell $(i_{(p)})_{pp}$ that assumes the value 1.

Each matrix $SV_{(p)}^T$ contains basis vectors for row clusters in X. The set of selector matrices I_p organizes the basis vectors by associating each one to a specific row cluster. Thus, in the minimization process, each row cluster is optimized with respect to one specific matrix $SV_{(p)}$. Similarly, the optimization of basis vectors for columns is oriented to specific column clusters. The set of matrices $V_{(p)}$ adds degrees of freedom in the factorization process. On the one hand, the association between columns and rows becomes more accurate, as illustrated in the experiments (Section IV); on the other hand, the time complexity of the algorithm used for factorizing X increases when compared, for example, to ONMTF Figure 2 shows a graphical visualization of the matrix factorization proposed in $\mathcal{F}_5$.

The derivation of the multiplicative update rules to implement the minimization process for $\mathcal{F}_5$ followed a gradient-based approach. The $\mathcal{F}_5$ gradient calculation is as in [7], thus $\nabla\mathcal{F}_5 = [\nabla\mathcal{F}_5]^+ - [\nabla\mathcal{F}_5]^-$. Expanding $\mathcal{F}_5$ using matrix trace properties [23]:

$$\mathcal{F}_5 = tr\Big[\Big(X - U\sum_{p=1}^{k} I_{(p)} S V_{(p)}^T\Big)^T \Big(X - U\sum_{p=1}^{k} I_{(p)} S V_{(p)}^T\Big)\Big]$$
$$= tr\big[X^TX\big] - 2tr\Big[X^TU\sum_{p=1}^{k} I_{(p)} S V_{(p)}^T\Big] + tr\Big[\sum_{p=1}^{k} V_{(p)} S^T I_{(p)} U^T U \sum_{p'=1}^{k} I_{(p')} S V_{(p')}^T\Big].$$

To calculate $\nabla_U \mathcal{F}_5$, consider $[\nabla_U \mathcal{F}_5]^-$ and $[\nabla_U \mathcal{F}_5]^+$:

$$[\nabla_U \mathcal{F}_5]^- = -2\nabla_U\Big(tr\big[X^TU\sum_{p=1}^{k} I_{(p)} S V_{(p)}^T\big]\Big) = -2X\sum_{p=1}^{k} V_{(p)} S^T I_{(p)},$$
$$[\nabla_U \mathcal{F}_5]^+ = \nabla_U\Big(tr(X^TX) + tr\big[\sum_{p=1}^{k} V_{(p)} S^T I_{(p)} U^T U \sum_{p'=1}^{k} I_{(p')} S V_{(p')}^T\big]\Big)$$
$$= U\sum_{p=1}^{k}\sum_{p'=1}^{k} I_{(p')} S V_{(p')}^T V_{(p)} S^T I_{(p)} + U\sum_{p=1}^{k}\sum_{p'=1}^{k} I_{(p)} S V_{(p)}^T V_{(p')} S^T I_{(p')} = 2U\sum_{p=1}^{k}\sum_{p'=1}^{k} I_{(p)} S V_{(p)}^T V_{(p')} S^T I_{(p')}.$$

To calculate $\nabla_S \mathcal{F}_5$, consider $[\nabla_S \mathcal{F}_5]^-$ and $[\nabla_S \mathcal{F}_5]^+$:

$$[\nabla_S \mathcal{F}_5]^- = -2\nabla_S\Big(tr\big[X^TU\sum_{p=1}^{k} I_{(p)} S V_{(p)}^T\big]\Big) = -2\sum_{p=1}^{k} I_{(p)} U^T X V_{(p)},$$

$$[\nabla_S \mathcal{F}_5]^+ = \nabla_S\Big(tr(X^TX) + tr\big[\sum_{p=1}^{k} V_{(p)} S^T I_{(p)} U^T U \sum_{p'=1}^{k} I_{(p')} S V_{(p')}^T\big]\Big)$$
$$= \sum_{p=1}^{k}\sum_{p'=1}^{k} I_{(p')} U^T U I_{(p)} S V_{(p')}^T V_{(p)} + \sum_{p=1}^{k}\sum_{p'=1}^{k} I_{(p)} U^T U I_{(p')} S V_{(p')}^T V_{(p)} = 2\sum_{p=1}^{k}\sum_{p'=1}^{k} I_{(p)} U^T U I_{(p')} S V_{(p')}^T V_{(p)}.$$

To calculate $\nabla_{V_{(i)}} \mathcal{F}_5$, consider $[\nabla_{V_{(i)}} \mathcal{F}_5]^-$ and $[\nabla_{V_{(i)}} \mathcal{F}_5]^+$:

$$[\nabla_{V_{(i)}} \mathcal{F}_5]^- = -2\nabla_{V_{(i)}}\Big(tr\big[X^TU\sum_{p=1}^{k} I_{(p)} S V_{(p)}^T\big]\Big) = -2\nabla_{V_{(i)}}\Big(tr\big[X^TU I_{(i)} S V_{(i)}^T\big]\Big) = -2X^TUI_{(i)}S.$$

$$[\nabla_{V_{(i)}} \mathcal{F}_5]^+ = \nabla_{V_{(i)}}\Big(tr(X^TX) + tr\big[\sum_{p=1}^{k} V_{(p)} S^T I_{(p)} U^T U \sum_{p'=1}^{k} I_{(p')} S V_{(p')}^T\big]\Big)$$
$$= \nabla_{V_{(i)}}\Big(tr\big[\sum_{p=1}^{k}\sum_{p'=1}^{k} V_{(p)} S^T I_{(p)} U^T U I_{(p')} S V_{(p')}^T\big]\Big)$$
$$= \nabla_{V_{(i)}}\Big(tr\big[V_{(i)} S^T I_{(i)} U^T U I_{(i)} S V_{(i)}^T\big]\Big) + \nabla_{V_{(i)}}\Big(tr\big[\sum_{p\neq i\in\{1,\dots,k\}} V_{(p)} S^T I_{(p)} U^T U I_{(i)} S V_{(i)}^T\big]\Big) + \nabla_{V_{(i)}}\Big(tr\big[\sum_{p'\neq i\in\{1,\dots,k\}} V_{(i)} S^T I_{(i)} U^T U I_{(p')} S V_{(p')}^T\big]\Big)$$
$$= 2V_{(i)} S^T I_{(i)} U^T U I_{(i)} S + \sum_{p\neq i\in\{1,\dots,k\}} V_{(p)} S^T I_{(p)} U^T U I_{(i)} S + \sum_{p'\neq i\in\{1,\dots,k\}} V_{(p')} S^T I_{(p')} U^T U I_{(i)} S$$
$$= 2\sum_{p=1}^{k} V_{(p)} S^T I_{(p)} U^T U I_{(i)} S$$

The final gradients for $U, S, V_{(p)}, \forall p \in \{1,\dots,k\}$ are:

$$\nabla_U \mathcal{F}_5 = 2\big(-X\sum_{p=1}^{k} V_{(p)} S^T I_{(p)} + \sum_{p=1}^{k} U \sum_{p'=1}^{k} I_{(p)} S V_{(p)}^T V_{(p')} S^T I_{(p')}\big)$$
$$\nabla_S \mathcal{F}_5 = 2\big(-\sum_{p=1}^{k} I_{(p)} U^T X V_{(p)} + \sum_{p=1}^{k}\sum_{p'=1}^{k} I_{(p)} U^T U I_{(p')} S V_{(p')}^T V_{(p)}\big)$$
$$\nabla_{V_{(p)}} \mathcal{F}_5 = 2\big(-X^TUI_{(p)}S + \sum_{p'=1}^{k} V_{(p')} S^T I_{(p')} U^T U I_{(p)} S\big)$$

Algorithm 1 implements the minimization process for $\mathcal{F}_5$ by updating U, S and V using the multiplicative rule:

$$X^{t+1} \leftarrow X^t \odot \frac{-[\nabla_X \mathcal{F}]^-}{[\nabla_X \mathcal{F}]^+}.$$

In this algorithm, t is an iteration counter, $U^{(t)}$, $S^{(t)}$ and $V_{(p)}^{(t)}$ are respectively the U, S e $V_{(p)}$ matrices in the th-iteration, $\mathcal{U}(0,1) \in\ ]0,1]$ is an uniformly distributed number generator, $\odot$ is the Hadamard product and stop conditions as a maximum number of iterations t_{max} or the reconstruction error convergence according to the limit ϵ (free parameter).

IV. Experiments and Results Analysis

We carried out two types of experiments:

- Experiment #1 (Section IV-A): We carried out this experiment on synthetic datasets to verify the reconstruction capacity provided by the OvNMTF algorithm in the presence of row or column overlapping cocluster structures. Here, the *K-means* algorithm is a reference for cluster capacity analysis; the ONMTF algorithm, based on multiplicative update rules [7], was chosen for illustration because of its similarity to OvNMTF algorithm.
- Experiment #2 (Section IV-B): We carried out this experiment on real-world text datasets to test the cluster discovery capacity of OvNMTF, and its power to produce information about topics (clusters of words) and how these topics describe clusters of documents. The cluster discovery evaluation was performed based on the Rand Index (RI) [24]. The evaluation of information production capacity was performed through the analysis of the words that made up each cocluster. Since the evaluation comprises cocluster analysis, only the results produced by ONMTF and by OvNMTF were analyzed.

A. Experiment #1

a) Datasets: Synthetic datasets were built on three of the eight data structure types [25][3]. These datasets are shown in the first column of Figure 1 and are relate to: (I) coclusters with exclusive rows and columns; (II) coclusters with exclusive rows and overlapping columns; (III) coclusters with exclusive columns and overlapping rows. Structure I was chosen to show the effectiveness of the algorithm in solving the classic clustering problem. The structures II and III were chosen to show the OvNMTF contribution. Each dataset has 150 datapoints (rows) and 150 attributes (columns).

[3] In [25], the biclustering problem is covered. Biclustering and coclustering are similar problems. Although each has its own definitions, the latter can be seen as an extension to the former [26], [27].

Algorithm 1 OvNMTF algorithm

input: data X, number of rows clusters k, number of columns clusters l, max iterations t_{max}

initialize: $U^{(0)} \leftarrow \mathcal{U}(0,1), S^{(0)} \leftarrow \mathcal{U}(0,1), V_{(p)}^{(0)} \leftarrow \mathcal{U}(0,1), \forall p$ e $t \leftarrow 0$

while (no convergence) **and** ($t \leq t_{max}$) **do**

$$U^{(t+1)} \leftarrow U^{(t)} \odot \frac{\sum_{p=1}^{k} X V_{(p)}^{(t)} S^{(t)^T} I_{(p)}}{\sum_{p=1}^{k} \sum_{p'=1}^{k} U^{(t)} I_{(p)} S^{(t)} V_{(p)}^{(t)^T} V_{(p')}^{(t)} S^{(t)^T} I_{(p')}}$$

for $p \leftarrow 1 : k$ **do**

$$V_{(p)}^{(t+1)} \leftarrow V_{(p)}^{(t)} \odot \frac{X^T U^{(t+1)} I_{(p)} S^{(t)}}{\sum_{p'=1}^{k} V_{(p')} S^T I_{(p')} U^T U I_{(p)} S}$$

end for

$$S^{(t+1)} \leftarrow S^{(t)} \odot \frac{\sum_{p=1}^{k} I_{(p)} U^{(t+1)^T} X V_{(p)}^{(t+1)}}{\sum_{p=1}^{k} \sum_{p'=1}^{k} I_{(p)} U^{(t+1)^T} U^{(t+1)} I_{(p')} S^{(t)} V_{(p')}^{(t+1)^T} V_{(p)}^{(t+1)}}$$

$t \leftarrow t + 1$

end while

return $U^{(t)}, S^{(t)}, V_{(1)}^{(t)}, \ldots, V_{(k)}^{(t)}$

TABLE I

RECONSTRUCTION CAPACITY: *ok* - GOOD RECONSTRUCTION, CORRECT INFORMATION ON OVERLAPPING ROWS/COLUMNS; $\times$ - GOOD RECONSTRUCTION, NO INFORMATION ON OVERLAPPING COLUMNS; $+$ - POOR RECONSTRUCTION, PARTIAL INFORMATION ON OVERLAPPING ROWS/COLUMNS; $\circ$ - BEYOND THE SCOPE OF THE ALGORITHM

	k-means	ONMTF	OvNMTF
base (I)	*ok*	*ok*	*ok*
base (II)	*ok*, $\times$	$+$	*ok*
base (III)	$\circ$	$+$	*ok*

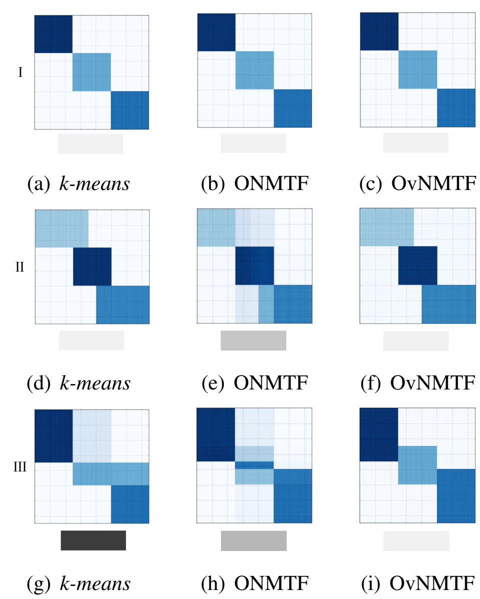


Fig. 3. Reconstruction capacity details with $k = l = 3$. The darker the grayscale bar the bigger the reconstruction error

b) Parameters setup: The following parameters setup was set: $k = 3$ for *k-means*, according to the actual number of clusters in the dataset, and $k = l = 3$ for ONMTF and OvNMTF, according to the actual number of row clusters and column clusters in the dataset; random initialization for C in *k-means*, U, S, V in ONMTF, and U, S and V_k in OvNMTF; stop conditions based on the maximum number of iterations (300 for *k-means*, 1000 for ONMTF and OvNMTF) and $\epsilon = 1\text{e}{-04}$ for reconstruction error convergence in ONMTF and OvNMTF; 10 runs of each algorithm in each dataset.

c) Reconstruction capacity: The reconstructions analyzed in this section concern the best result obtained for each algorithm in each dataset. Table I shows a summary of the results, and Figure 3 allows us a visual analysis.

In Figure 3, the first dataset (I) represents a classical clustering problem, without overlapping rows/columns. All algorithms tested could produce good reconstructions of the original dataset. The dataset II was properly reconstructed by *k-means* because if we analyze the problem represented in this dataset from the row clusters standpoint, it is equivalent to the classical clustering problem. In the datasets II and III, the ONMTF algorithm could not correctly associate rows/columns with over one row/column cluster with the chosen set of parameters. The OvNMTF algorithm properly reconstructed the datasets II and III, since it could properly organize the degrees of freedom conferred by the multiple matrices V_k.

The reconstruction errors for the dataset I are similar. For the dataset II, although the algorithm *k-means* offers a good reconstruction, with error similar to that produced by OvNMTF, it cannot produce information about column clusters, i.e. having a good reconstruction capacity does not guarantee good descriptions for data clusters, the reconstruction error

TABLE II
QUANTITATIVE INFORMATION ON TEXT REAL-WORLD DATASETS

	PNC	PNC toy	NIPS
# unique terms	6,710	36,342	6,881
# terms	69,301	1,187,334	746,826
# documents	300	4,575	555
# row clusters	3	13	9
% zeros in the data matrix	0.997	0.993	0.804

TABLE III
AVERAGE RI FOR TEXT DATASETS, WITH $k = 3$ FOR PNC TOY, $k = 13$ FOR PNC AND $k = 9$ FOR NIPS AND THE BEST l VALUES FOR EACH COMBINATION (DATASET $\times$ ALGORITHM $\times$ VECTOR REPRESENTATION)

		k-means	**ONMTF**	**OvNMTF**
PNC toy	*tf*	0.7017	$0.3372 : l = 5$	**0.7466** $: l = 4$
	tf_{norm}	0.7086	$0.6479 : l = 5$	**0.7487** $: l = 3$
	tfidf	0.3869	$0.1758 : l = 3$	**0.6674** $: l = 6$
	tfidf_{norm}	0.4701	$0.5717 : l = 3$	**0.6755** $: l = 6$
PNC	*tf*	0.3137	$0.1437 : l = 16$	**0.3384** $: l = 10$
	tf_{norm}	0.3049	$0.1802 : l = 19$	**0.3455** $: l = 16$
	tfidf	0.2784	$0.1279 : l = 7$	**0.3534** $: l = 7$
	tfidf_{norm}	0.2750	$0.1184 : l = 7$	**0.3554** $: l = 16$
NIPS	*tf*	0.1573	$0.1579 : l = 6$	**0.1672** $: l = 6$
	tf_{norm}	0.1527	$0.1352 : l = 15$	**0.1641** $: l = 9$
	tfidf	0.1368	$0.1442 : l = 18$	**0.1711** $: l = 9$
	tfidf_{norm}	0.1519	$0.1318 : l = 9$	**0.1742** $: l = 12$

produced by ONMTF are about seven times bigger than that produced by OvNMTF. In the dataset III, the reconstruction error produced by ONMTF are about ten times bigger than that produced by OvNMTF, and the *k-means* algorithm produced a very high reconstruction error. The ONMTF reconstruction errors can be improved if the parameter l is set to higher values. However, in such a case, the discovered knowledge on column clusters will differ from the *a priori* knowledge.

B. Experiments #2

a) Datasets: The text data analysis has been chosen to illustrate the accuracy and added value of the information that OvNMTF can extract. This experiment was carried out on three text datasets, as in [10]; Table II lists quantitative information about these datasets:

1) Portuguese news items collection (PNC): A collection of Portuguese language news items. Each news item consists of an url, title, subtitle, body and topic in which the item was manually classified. The news items are distributed on 13 unevenly classes.
2) Portuguese news items collection (PNC toy): A subset of the PNC collection. It comprises 300 news items distributed in a balanced way in three topics (sports, games and activities for young people).
3) NIPS14-17 (NIPS): A dataset related to scientific papers published in the Neural Information Processing Systems Congress, 2001-2003 - volumes 14-17. The complete dataset comprises scientific papers published in 18 volumes, however, only the papers in the volumes 14 to 17 are labeled. Such documents are organized on topics that cover 13 technical areas and are unevenly distributed; documents from nine most voluminous areas were used.

b) Parameters setup: The following parameters setup was established: k was set according to the actual number of classes/topics in each dataset and l assumes a list of values, since there is no *a priori* knowledge about the actual number of word clusters, thus:

- PNC TOY: $k = 3$ e $l \in \{2, 3, 4, 5, 6\}$;
- PNC: $k = 13$ e $l \in \{7, 10, 13, 16, 19\}$;
- NIPS: $k = 9$ e $l \in \{6, 9, 12, 15, 18\}$;

random initialization for factor matrices; stop conditions based on the maximum number of iterations (1,000 for PDN TOY, 10,000 for PNC and NIPS) and $\epsilon = 1\mathrm{e}{-04}$ for reconstruction error convergence; 10 runs of each combination: algorithm *versus* dataset *versus* k, l combination values *versus* vector representations for text data.

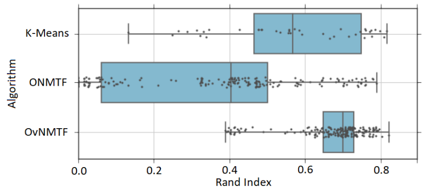
c) Vector space model for text data: We chose a count-based distributional semantics model to build the vector space model for text data [28], [29]. It uses the text data in each document (news item) to produce a document-word matrix. In the preprocessing phase, stopwords were dropped, documents' token were stemmed [30] and then, *tf* and *tf-idf* scores [31] and their respective normalized versions (tf_{norm}, tfidf_{norm}) were computed for each document.

d) Clustering results: We evaluate clustering quality by using the Rand Index (RI). The results are presented in terms of: average RI for each combination of vector representations and algorithms; distribution of the RI values for all runs of each algorithm. Table III presents the best average RI values for each algorithm in the different vector representations; Figure 4 shows the distributions of the RI values.

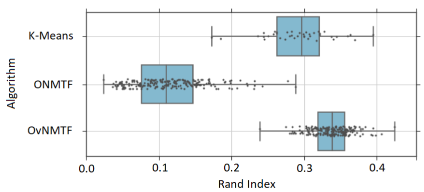
OvNMTF presented the best average RI values in all cases shown on Table III. As the complexity of the problem in terms of the desired number of row clusters increased, all algorithm's performance declined and the superiority of the algorithm OvNMTF over the others decreased. However, considering another aspect of the complexity of the problem - the sparsity of the data matrix (see Table II), OvNMTF has a positive highlight. These results reveal the good clustering capability presented by the algorithm introduced in this paper.

Considering all algorithms runs, OvNMTF stands out for its stability. The RI distributions graphs shown in Figure 4 illustrate that the variability in the results presented by the OvNMTF algorithm is smaller than the variability presented by the other algorithms. Specifically, for PNC and PNC toy datasets, the algorithm OvNMTF also achieves the best maximum RI values and concentrate the results on high RI values. For the NIPS dataset, the algorithm ONMTF has better maximum RI values, but concentrates most of its RI values around the lowest values presented by the algorithm OvNMTF.

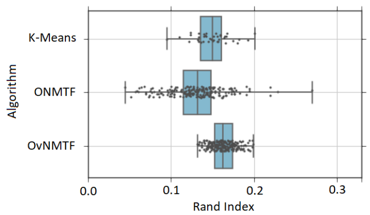
The vector representation used meant a sensitivity issue for cluster quality in the case of the PNC toy dataset (the dataset with the most sparsity and least complexity in the number of row clusters). For this dataset, *k-means* and OvNMTF performed better with the vector representation purely based on



(a) PNC toy



(b) PNC



(c) NIPS

Fig. 4. Rand index distribution for all runs of each algorithm and dataset

frequencies (*tf*), although the differences among the OvNMTF results were smaller; ONMTF performed better for normalized representations. Considering the RI values distribution, OvNMTF suggests greater stability, which puts it at an advantage in terms of vector representation independence.

e) Semantic features extraction: To extract the semantic features, we follow the ideas presented in [4] and [7]. In [4], the authors argue that it makes more sense for a document to be associated with a small set of topics than one or all possible topics; [7] argues that words have different meanings depending on the context in which they are used, and factor matrices allow us to identify word clusters with words in common but associated with different contexts.

Due to the addition of multiple matrices $V_{(p)}$ in OvNMTF, and the use of each one as an independent basis for document clusters, each one specializes in generating topics for one document cluster. The association of this characteristic with the ideas mentioned above motivate our hypothesis that the feature extraction supported by OvNMTF will be more accurate than that offered by ONMTF and will trustingly express more reliable descriptions for document clusters. To test our hypothesis, we compared the best runs (in terms of RI) of the ONMTF and OvNMTF algorithms, for the PNC toy dataset.

Table IV shows the normalized factor matrix S resultant from the ONMTF factorization. From this matrix, we identified coclusters of interest, i.e., document and word clusters whose relations in matrix S have the highest values. Thus, to extract semantic features to each cocluster of interest, we

TABLE IV

MATRIX S (NORMALIZED) GOT FROM A ONMTF RUN ON THE PCN TOY DATASET, USING $k = 3$, $l = 5$ AND VECTOR REPRESENTATION $tfidf_{norm}$

	WC#1	WC#2	WC#3	WC#4	WC#5
DC#1	0,0	**0,5**	0,1	0,0	**0,4**
DC#2	0,0	0,05	0,05	**0,9**	0,0
DC#3	**0,4**	0,1	**0,5**	0,0	0,0

(a) WC#1 (b) WC#2 (c) WC#3 (d) WC#4 (e) WC#5

Fig. 5. Top-20 words in each word cluster discovered by ONMTF

analyze them in terms of their 20 most relevant words. The relevance of the words is given by the values in the matrix V that associate them with the word clusters. Figure 5 shows the words relevance through word clouds[4]. In the word cloud, the bigger the word, the bigger its relevance.

The word GAMES is the only one that appears in more than one cluster (WC#1 and WC#4). An informal interpretation of the words organization in the clusters allows us to infer that: WC#2 and WC#5 describe the news items in DC#1 as "soccer news"; WC#4 describes the news items in DC#2 as "e-sport news"; WC#1 describe the news items in DC#3 as "extreme sports news"; and WC#3 does not bring easily interpretable semantic information, imposing a degree of uncertainty about the definition of the topic referent to DC#3.

A normalized factor matrix S, resultant from the OvNMTF factorization, has each row associated with one matrix $V_{(p)}$, that determines the subset of word clusters optimized for one document cluster associated with that row in S. Thus, in the run under analysis (PCN toy dataset, $k = 3$, $l = 2$ and tf_{norm}), there are two word clusters for each one of the three document clusters. For DC#1, WC#1 = 0.38 and WC#2 = 0.62; for DC#2, WC#3 = 0.46 and WC#4 = 0.54; and for DC#3, WC#5 = 0.94 e WC#6 = 0.06. The semantic features extraction was carried out and the wordclouds are shown in Figure 6.

(a) WC#1 (b) WC#2 (c) WC#3 (d) WC#4 (e) WC#5 (f) WC#6

Fig. 6. Top-20 words in each word cluster discovered by OvNMTF

From wordclouds, we can give the following description for document clusters: WC#1 and WC#2 describe the news items in DC#1 as "e-sports news"; WC#3 and WC#4 describe the news items in DC#2 as "extreme sports news"; and WC#5 and WC#6 describe the news items in DC#3 as "soccer news". As

[4]Words used in word clouds were translated from English to Portuguese. GAME means *JOGO*; GAMES refers to the use of the English language word within the Portuguese language texts (commonly in the context of e-sports).

expected, the description of each document cluster is similar to that obtained from the ONMTF but, we declare two advantages arising from the OvNMTF coclustering framework:

- The arrangement of a subset of word clusters (for each document clusters) is completely independent of another subset, thus, OvNMTF can use the same word in different word clusters subsets more often. This makes it possible to better identify polysemic words.
- Each subset of word cluster concerns only one, document cluster. Thus, the word clusters analysis give us more accurate descriptions for document clusters. Moreover, the actual role of a polysemic word can be more easily identified in a context of more accurate interpretation.

V. Conclusion

In this paper, we formalize the overlapping non-negative matrix triple factorization problem (the OvNMTF problem), as an alternative to NMF-based problems. OvNMTF was designed to naturally deal with overlapping row and columns in coclustering analysis. To implement the OvNMTF problem minimization process, we derived an algorithm based on multiplicative update rules (the OvNMTF algorithm). The reasonableness of the algorithm was tested on synthetic data sets; its usefulness and its power to extract information were attested in real-world text datasets. OvNMTF produced better clustering results than *k-means*, and it was superior to ONMTF considering the experimentation model and the limits imposed for the variation of the parameters k and l. In terms of semantic features extraction, the OvNMTF algorithm has also brought benefits as it allows us to infer more accurate descriptions for each document cluster. The results show that the proposed algorithm implements an optimization process that allows the use of degrees of freedom to efficiently compose coclusters. However, the OvNMTF algorithm is more complex in terms of runtime than ONMTF due to the larger number of matrices involved in the factorization process. Therefore, its use in real-world problems needs to consider this cost. This drawback is the point of attention for the next steps of this research.

References

[1] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, *Advances in Knowledge Discovery and Data Mining*. American Association for Artificial Intelligence AAAI/MIT Press, 1996.

[2] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Survey*, vol. 31, no. 3, pp. 264–323, September 1999.

[3] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques, 3th Edition*. Morgan Kaufmann, 2011.

[4] D. D. Lee and S. H. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp. 788–791, October 1999.

[5] B. Long, Z. M. Zhang, and P. S. Yu, "Co-clustering by block value decomposition," in *Proc. of the 11th ACM SIGKDD Int. Conf. on Knowl. Disc. Data Min., Illinois, USA*, August 2005, pp. 635–640.

[6] J. Wang, Z. Zhao, J. Zhou, H. Wang, B. Cui, and G. Qi, "Recommending flickr groups with social topic model," *Information Retrieval*, vol. 15, no. 3-4, pp. 278–295, June 2012.

[7] J. Yoo and S. Choi, "Orthogonal nonnegative matrix tri-factorization for co-clustering: Multiplicative updates on stiefel manifolds," *Inf. Process. and Manag.: an Int. J.*, vol. 46, pp. 559–570, September 2010.

[8] H. Wang, F. Nie, H. Huang, and F. Makedon, "Fast nonnegative matrix tri-factorization for large-scale data co-clustering," in *Proc. of the 22nd Int. Joint Conf. on Artif. Intell., Volume Two, Barcelona, Spain*. AAAI Press, July 2011, pp. 1553–1558.

[9] S. Huang, Z. Xu, and J. Lv, "Adaptive local structure learning for document co-clustering," *Knowledge-Based Systems*, vol. 148, pp. 74–84, May 2018.

[10] L. F. Brunialti, S. M. Peres, V. F. da Silva, and C. A. de Moraes Lima, "The BinOvNMTF algorithm: Overlapping columns co-clustering based on non-negative matrix tri-factorization," in *Brazilian Conf. on Intelligent Systems, BRACIS, Uberlândia, Brazil*, October 2017, pp. 330–335.

[11] T. Hofmann, J. Puzicha, and M. I. Jordan, "Learning from dyadic data," in *Advances in Neural Information Processing Systems 11, NIPS Conf., Denver, Colorado, USA*, November-December 1998, pp. 466–472.

[12] N. D. Buono and G. Pio, "Non-negative matrix tri-factorization for co-clustering: An analysis of the block matrix," *Inf. Sci.*, vol. 301, pp. 13–26, April 2015.

[13] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proc. of the 26th Annual Int. ACM SIGIR Conf. on R&D in Information Retrieval, Toronto, Canada*, July 2003, pp. 267–273.

[14] T. Li and C. H. Q. Ding, "The relationships among various nonnegative matrix factorization methods for clustering," in *Proc. of the 6th IEEE Int. Conf. on Data Min. (ICDM), China*, December 2006, pp. 362–371.

[15] F. Shahnaz, M. W. Berry, V. P. Pauca, and R. J. Plemmons, "Document clustering using nonnegative matrix factorization," *Inf. Process. and Manag.: an Int. J.*, vol. 42, no. 2, pp. 373–386, March 2006.

[16] C. H. Q. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 45–55, January 2010.

[17] C. H. Q. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix tri-factorizations for clustering," in *Proc. of the 12th ACM SIGKDD Int. Conf. on Knowl. Disc. Data Min., Philadelphia, PA, USA*, August 2006, pp. 126–135.

[18] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. on Pattern Analysis and Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, August 2011.

[19] A. Salah, M. Ailem, and M. Nadif, "Word co-occurrence regularized non-negative matrix tri-factorization for text data co-clustering," in *Proc. of the 32nd AAAI Conf. on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA*, February 2018, pp. 3992–3999.

[20] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of the 5th Berkeley Symp. on Mathematical Statistics and Probability, Volume 1: Statistics, Berkeley, California, USA*. University of California Press, 1967, pp. 281–297.

[21] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–136, March 1982.

[22] C. H. Q. Ding and X. He, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Proc. of the 2005 Int. Conf. on Data Mining, Newport Beach, CA, USA*, April 2005, pp. 606–610.

[23] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics, 2th Edition*, ser. Wiley Series in Probability and Statistics: Texts and References Section. John Wiley, 1999.

[24] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *J. of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, December 1971.

[25] S. C. Madeira and A. L. Oliveira, "Biclustering algorithms for biological data analysis: A survey," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 1, no. 1, pp. 24–45, August 2004.

[26] R. G. Pensa, J. Boulicaut, F. Cordero, and M. Atzori, "Co-clustering numerical data under user-defined constraints," *Statistical Analysis and Data Mining, NY, USA*, vol. 3, no. 1, pp. 38–55, February 2010.

[27] Y. Cheng and G. M. Church, "Biclustering of expression data," in *Proc. of the 8th Int. Conf. on Intelligent Systems for Molecular Biology, La Jolla / San Diego, CA, USA*, August 2000, pp. 93–103.

[28] G. Salton, A. Wong, and C. Yang, "A vector space model for automatic indexing," *Comm. of the ACM*, vol. 18, pp. 613–620, November 1975.

[29] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys (CSUR)*, vol. 34, no. 1, pp. 1–47, March 2002.

[30] M. F. Porter, "Snowball: A language for stemming algorithms," Published online, October 2001, accessed 11.10.2019, 15.30h. [Online]. Available: http://snowball.tartarus.org/texts/introduction.html

[31] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge University Press, 2008.