Textual data summarization using the Self-Organized Co-Clustering model

Margot Selosse, Julien Jacques, Christophe Biernacki

Textual data summarization using the Self-Organized Co-Clustering model

Margot Selosse[a,*], Julien Jacques[a], Christophe Biernacki[b,c]

[a] *Université de Lyon, Lyon 2 & ERIC EA3083, 5 Avenue Pierre Mendès France, 69500 Bron, France*
[b] *Université de Lille - UFR de Mathématiques - Cité Scientifique - 59655 Villeneuve d'Ascq Cedex, France*
[c] *INRIA, 40, av. Halley - Bât A - Park Plaza 59650 Villeneuve d'Ascq*

Abstract

Recently, different studies have demonstrated the use of co-clustering, a data mining technique which simultaneously produces row-clusters of observations and column-clusters of features. The present work introduces a novel co-clustering model to easily summarize textual data in a document-term format. In addition to highlighting homogeneous co-clusters as other existing algorithms do we also distinguish noisy co-clusters from significant co-clusters, which is particularly useful for sparse document-term matrices. Furthermore, our model proposes a structure among the significant co-clusters, thus providing improved interpretability to users. The approach proposed contends with state-of-the-art methods for document and term clustering and offers user-friendly results. The model relies on the Poisson distribution and on a constrained version of the Latent Block Model, which is a probabilistic approach for co-clustering. A Stochastic Expectation-Maximization algorithm is proposed to run the model's inference as well as a model selection criterion to choose the number of co-clusters. Both simulated and real data sets illustrate the efficiency of this model by its ability to easily identify relevant co-clusters.

Keywords: coclustering, document-term matrix, Latent Block Model

*Corresponding author
Email addresses: margot.selosse@gmail.com (Margot Selosse), julien.jacques@univ-lyon2.fr (Julien Jacques), christophe.biernacki@inria.fr (Christophe Biernacki)

1. Introduction

While textual data has existed for centuries, its occurence, use and ease of access has exploded in recent years, thanks in particular to the Internet. Social networks have largely driven this phenomenon: in 2019, Twitter had almost 474,000 tweets per minute and Facebook reported 4.3 billion messages posted per day. Access to an infinite number of resources via forums, the digitisation of newspapers and the creation of websites are also other important factors.

However, since text is an unstructured type of data, its analysis is not trivial and requires the use of special methods. The representation of text alone is a challenge, as various recent papers have shown [1, 2]. Most problems related to the analysis of textual data are still open issues, and are challenged by strong technological obstacles. Therefore, when users deal with a large unknown corpus,they often need - as a first step - a global overview of their data set. In other words, users often need to summarize their data, for example by knowing which documents share the same topics and the main topics of each cluster. The most famous way to do this is probably the Latent Dirichlet Allocation model (LDA, [3]), which proposes a probabilistic modelling of the words appearing in the documents. Many extensions of LDA have been proposed over the years. For instance, recently, [4] combines LDA and clustering algorithms to highlight the main topics of their clusters. In [5], the authors analyse scientific literature related to the field of e-Health.In [6], the authors describe the Biterm Topic Model (BTM). It outperforms LDA on short texts (such as instant messages and tweets) for which LDA performs poorly, due to the sparsity of the data. In [7], the authors propose another version of the BTM: they represent the biterms (word-pairs) as graphs and use a deep convolutional network to encode word co-relationships.

This work presents the Self-Organised Co-Clustering model (SOCC). It aims at providing a tool to summarize large document-term matrices, whose rows correspond to documents and columns correspond to terms. The clustering ap-

proach, which forms homogeneous groups of observations (documents in this case), is a useful unsupervised technique with proven efficiency in several domains. However, in high-dimensional and sparse contexts, they are sometimes less adapted and difficult to interpret. When considering such data sets, co-clustering, which groups observations and features simultaneously, turns out to be more efficient. It exploits the dualism between rows and columns and the data set is summarized in blocks (the crossing of a row-cluster and a column-cluster). The clusters of documents help in finding similar documents while the clusters of terms tell us what the clusters of documents are about. In this context, our work helps in finding similar documents and their interaction with term clusters.

The co-clustering task can be done in several ways. For example, in [8], the authors describe an original approach that uses optimal transport theory to co-cluster continuous data. However, we mostly distinguish between two kinds of co-clustering approaches. Matrix factorization based methods, e.g. [9, 10], consist of factorizing the $N \times J$ data matrix $\boldsymbol{x}$ into three matrices $\boldsymbol{a}$ (of size $N \times G$), $\boldsymbol{b}$ (size $G \times H$) and $\boldsymbol{c}$ (size $H \times J$), with the condition that all three matrices are non-negative. More specifically, the approximation of $\boldsymbol{x}$ by $\boldsymbol{x} \approx \boldsymbol{abc}$ is achieved by minimizing the error function $\min_{(\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})} ||\boldsymbol{x} - \boldsymbol{abc}||$, with the constraints $(\boldsymbol{a} \geq 0, \boldsymbol{b} \geq 0, \boldsymbol{c} \geq 0)$, and $||.||$ denoting a suitable norm (such as the Frobenius norm, spectral norm etc.). The matrices $\boldsymbol{a}$ and $\boldsymbol{c}$ define the row and column cluster memberships respectively. Each value of the matrix $\boldsymbol{a}$ (or $\boldsymbol{c}$) corresponds to the degree in which a row (or a column) belongs to a row-cluster (or a column-cluster). The matrix $\boldsymbol{b}$ represents the *block* matrix: an element b_{gh} of $\boldsymbol{b}$ is a scalar that summarizes the observations belonging to row-cluster g and column-cluster h. This kind of method was successfully used to co-cluster textual data sets in [11] and [12]. However, it requires choosing the metric $||.||$ that best fits the structure of the underlying latent blocks based on available data, which can be difficult. Furthermore, to the best of our knowledge, these methods do not propose a way to select the correct number of blocks.

Probabilistic approaches, such as the Latent Block Model [13], take a differ-

ent approach. They usually assume that the data was generated from a mixture of probability distributions with each associated component corresponding to a block. The parameters of the related distributions and the posterior probabilities of the blocks from the data provided are then estimated. This approach models the elements of a block with a parametric distribution and provides more information than the previous methods, that model the blocks with a simple scalar. In addition, each block is interpretable from its distribution parameters. Moreover, criterion such as the Integrated Classification Likelihood (or ICL) [14] can be used for model selection purposes, including the choice of number of blocks.

However, when dealing with high-dimensional sparse data, several blocks may be mainly sparse (composed of zeros) and cause inference issues. In addition, highlighting homogeneous blocks is not always sufficient to obtain easy-to-interpret results. Indeed, despite being homogeneous, these sparse blocks are not relevant from an interpretation perspective, and we need a new step to select the pertinent blocks. In other words, it is left to the user to choose the most useful co-clusters and to determine which term clusters (column-clusters) are more specific to which document clusters (row-clusters). This task is not straightforward even with a reasonable number of row and column-clusters. Therefore, it is necessary to work on a co-clustering technique that offers ready-to-use results.

We can address this problem by imposing a pattern on the co-clustering structure. Such an approach directly produces the most meaningful co-clusters, and significantly simplifies the results and their analysis. In the present work, we propose a co-clustering approach based on the Latent Block Model [15], in which we impose a structure wherein column-clusters (clusters of terms) are separated into three parts. In the first part, each cluster of terms is specific to one cluster of documents. In the second part, each cluster of terms is specific to two clusters of documents. The third part contains only one column-cluster and gathers terms that are common to all clusters of documents. The main motivation of this paper is to provide a tool with high comprehensibility: having three sections offers explicable results, with a reasonable number of co-clusters.

The choice to constrain our model to pairwise interactions between clusters is essentially motivated by the classical ANOVA modelling, which is usually limited to the two-way analysis. Furthermore, pairwise interactions are more interpretable than higher order interactions, and interactions between more than three factors are expected to be infrequent. Figure 1 illustrates the proposed structure. On the left, we present a usual co-clustering with the Poisson Latent Block Model. On the right, we show a co-clustering with the SOCC structure: thin separations between the three parts of column-clusters were added, with the noisy blocks as the lighter ones.

Other works have introduced a structure in their related co-clustering. In [16] the authors propose a co-clustering using a double k-means, and impose that the meaningful blocks are on the diagonal. In [17] and [18], the authors propose block diagonal co-clustering techniques, with binary and counting data respectively. Firstly, this consists of constraining the co-clustering such that the number of row-clusters is equal to the number of column-clusters. Secondly, the blocks out of the diagonal are considered to be noisy and share the same parameter. In fact, these models are particular cases of the model we propose: they constrain the structure to only the first part of column-clusters mentioned above. While these methods proved their efficiency in the case of document-term matrices, they assume that a cluster of terms is specific to only one cluster of documents. However, a group of terms could be specific to several groups of documents. Let us assume for instance that the documents are research papers, with one cluster related to computer science and another one related to mathematics. Each cluster has its own specific terms but many terms (for instance those related to probability distributions) will appear in both communities. In this work, we address this issue by defining a more complete structure among blocks without losing interpretability.

The rest of this paper is organized as follows: Section 2 presents the Latent Block Model and its application in counting data with the Poisson distribution. Section 3 describes the novel method referred to as 'Self-Organized Co-Clustering' (SOCC). In Section 4, we assess the efficiency of our solution in

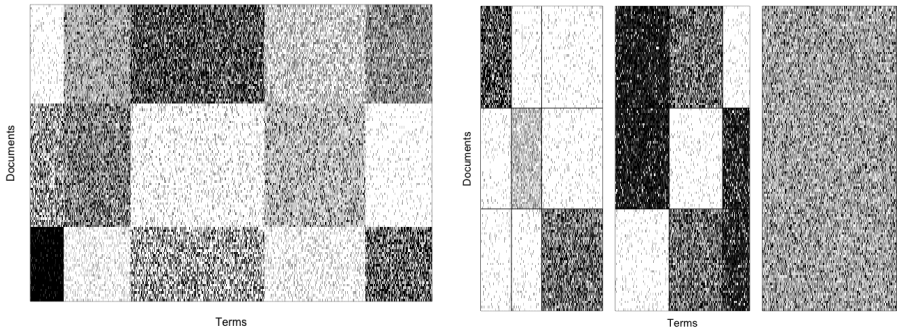

Figure 1: On the left, the usual Poisson Latent Block Model: we see that some blocks are not easily classifiable into noisy or significant blocks. On the right, the SOCC approach: we can easily distinguish betwenn the noisy blocks (shown in a lighter shade) and the significant blocks.

three ways. Firstly, we use simulated data, to evaluate the partition estimation of the SOCC model and state-of-the-art competing models. Secondly, we use real textual data sets to compare the proposed approach with these models, regarding both document clustering and term clustering. Thirdly, we describe a use case of the SOCC model on a real labelled data set. In Section 5, we detail an example for using the SOCC model in a truly unsupervised context. The last section concludes the paper and discusses topics for possible future research.

2. Background and notation

2.1. The Latent Block Model

The Latent Block Model (LBM) is a widely used model to carry out co-clustering [19]. It assumes that by knowing the row and column partitions, the elements of a block are independent and identically distributed. In this section, the hypotheses for the LBM are defined, and the mathematical details are given.

Let us consider the data matrix $\boldsymbol{x} = (x_{ij})_{i,j}$ with $1 \leq i \leq N$ and $1 \leq j \leq J$. It is assumed that G row-clusters and H column-clusters exist, and that they correspond to a partition $\boldsymbol{v} = (\boldsymbol{v}_i)_i$ of the rows and a partition $\boldsymbol{w} = (\boldsymbol{w}_j)_j$ of the columns. We have $\boldsymbol{v}_i = (v_{ig})_g$ with v_{ig} equal to 1 if row i belongs to cluster

g (where $1 \leq g \leq G$), and 0 otherwise. Similarly, we have $\boldsymbol{w}_j = (w_{jh})_h$ with w_{jh} equal to 1 when column j belongs to cluster h (where $1 \leq h \leq H$), and 0 otherwise. Thereafter, we no longer specify the ranges of i, j, g and h.

The first LBM hypothesis is that the univariate random variables x_{ij} (for all i and for all j) are conditionally independent given the row and column partitions $\boldsymbol{v}$ and $\boldsymbol{w}$. Therefore, the conditional probability density function (p.d.f) of $\boldsymbol{x}$ given $\boldsymbol{v}$ and $\boldsymbol{w}$ can be written:

$$p(\boldsymbol{x}|\boldsymbol{v}, \boldsymbol{w}; \boldsymbol{\alpha}) = \prod_{ijgh} f(x_{ij}; \alpha_{gh})^{v_{ig} w_{jh}},$$

where $\boldsymbol{\alpha} = (\alpha_{gh})_{g,h}$ is the distribution's parameters of block (g, h).

The second LBM hypothesis is that the latent variables $\boldsymbol{v}$ and $\boldsymbol{w}$ are independent, so $p(\boldsymbol{v}, \boldsymbol{w}; \boldsymbol{\gamma}, \boldsymbol{\rho}) = p(\boldsymbol{v}; \boldsymbol{\gamma}) p(\boldsymbol{w}; \boldsymbol{\rho})$ with:

$$p(\boldsymbol{v}; \boldsymbol{\gamma}) = \prod_{ig} \gamma_g^{v_{ig}} \text{ and } p(\boldsymbol{w}; \boldsymbol{\rho}) = \prod_{jh} \rho_h^{w_{jh}},$$

where $\gamma_g = p(v_{ig} = 1)$ and $\rho_h = p(w_{jh} = 1)$. This means that, for all i, the distribution of $\boldsymbol{v}_i$ is the multinomial distribution $\mathcal{M}(\gamma_1, \ldots, \gamma_G)$ and is not dependent on i. Similarly, for all j, the distribution of $\boldsymbol{w}_j$ is the Multinomial distribution $\mathcal{M}(\rho_1, \ldots, \rho_H)$ is not dependent on j.

Based on these considerations, the LBM parameter is defined as $\boldsymbol{\theta} = (\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\alpha})$, with $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_G)$ and $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_H)$ being the rows and columns mixing proportions. Therefore, if V and W are the sets of all possible labels $\boldsymbol{v}$ and $\boldsymbol{w}$, the probability density function of $\boldsymbol{x}$ is written:

$$p(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{\substack{(\boldsymbol{v}, \boldsymbol{w}) \\ \in V \times W}} \prod_{ig} \gamma_g^{v_{ig}} \prod_{jh} \rho_h^{w_{jh}} \prod_{ijgh} f(x_{ij}; \alpha_{gh})^{v_{ig} w_{jh}}. \quad (1)$$

2.2. The Poisson Latent Block Model (PLBM)

Counting data, such as those present in document-term matrices, can be modelled using the Poisson distribution. For a x_{ij} belonging to block (g, h) the Poisson distribution is parameterized with λ_{ij} such that $\lambda_{ij} = n_{i.} n_{.j} \delta_{gh}$. Here, the values $n_{i.}$, $n_{.j}$ correspond to a 'row effect' and a 'column effect' respectively,

and are computed as follows:

$$n_{i.} = \sum_j x_{ij} \text{ and } n_{.j} = \sum_i x_{ij}.$$

They are independent of the co-clustering and are computed from the document term matrix beforehand. Consequently, the LBM parameter α_{gh} of Section2.1 corresponds to δ_{gh}, and is referred to as 'the effect of block (g, h)' [13]. The probability density function is therefore given by:

$$f(x_{ij}; \delta_{gh}) = \mathcal{P}(n_{i.} n_{.j} \delta_{gh}) = \frac{1}{x_{ij}!} e^{-n_{i.} n_{.j} \delta_{gh}} (n_{i.} n_{.j} \delta_{gh})^{x_{ij}}. \tag{2}$$

2.3. Inference

The EM-algorithm [20] is a well-known method to perform parameter estimation with latent variables. It iterates two steps. The first step, referred to as the 'E-step', computes the expected complete log-likelihood conditionally to the observed data. The second step, referred to as the 'M-step' consists in maximizing the expected complete log-likelihood over the parameters $\boldsymbol{\theta}$. Given equations (1) and (2), the complete log-likelihood is written as:

$$\begin{aligned} L_c(\boldsymbol{\theta}; \boldsymbol{x}, \boldsymbol{v}, \boldsymbol{w}) &= \sum_{ig} v_{ig} \log \gamma_g + \sum_{jh} w_{jh} \log \rho_h \\ &+ \sum_{ijgh} v_{ig} w_{jh} \big(x_{ij} \log(n_{i.} n_{.j} \delta_{gh}) - n_{i.} n_{.j} \delta_{gh} - x_{ij}! \big). \end{aligned} \tag{3}$$

Thus, the E-step requires the computation of the probability $p(v_{ig} w_{jh} = 1 | \boldsymbol{x})$. In this case, it is not computationally tractable since the row and column partitions are not independent conditionally to $\boldsymbol{x}$. In such a situation, several alternatives to the EM algorithm exist, as the variational EM algorithm, the SEM-Gibbs algorithm or other algorithm linked to Bayesian inference [21]. In this work, we use the SEM-Gibbs version for its simplicity of implementation, its low sensitivity to initialization and its good performance. Instead of computing the probability $p(v_{ig} w_{jh} = 1 | \boldsymbol{x})$, we sample $(\boldsymbol{v}, \boldsymbol{w})$ through a Gibbs sampler. It requires the computation of the probabilities $p(v_{ig} = 1 | \boldsymbol{x}, \boldsymbol{w}; \boldsymbol{\theta})$ and $p(w_{jh} = 1 | \boldsymbol{x}, \boldsymbol{v}; \boldsymbol{\theta})$, which are tractable. Algorithm 1 presents the SEM-Gibbs algorithm for the PLBM inference.

Input: $\boldsymbol{x}$, G, H

Initialization: randomly choose $\boldsymbol{v}$ and $\boldsymbol{w}$, deduced $\gamma_g = \frac{1}{N}\sum_i v_{ig}$, $\rho_h = \frac{1}{J}\sum_j w_{jh}$

for *i in 1:nbSEM* **do**

Step 1: Sample $\boldsymbol{v}$ such that:

$p(v_{ig} = 1|\boldsymbol{x}, \boldsymbol{w}; \boldsymbol{\theta}) \propto \gamma_g \times \prod_{jh} f(x_{ij}; \delta_{gh})^{w_{jh}}$

Step 2: $\gamma_g = \frac{1}{N}\sum_i v_{ig}$,

$$\delta_{gh} = \frac{1}{n_{g.}n_{.h}} \sum_{ij} v_{ig} w_{jh} x_{ij},$$

with $n_{g.} = \sum_{ij} v_{ig} x_{ij}$ and $n_{.h} = \sum_{ij} w_{jh} x_{ij}$.

Step 3: Sample $\boldsymbol{w}$ such that:

$p(w_{jh} = 1|\boldsymbol{x}, \boldsymbol{v}; \boldsymbol{\theta}) \propto \rho_h \times \prod_{ig} f(x_{ij}; \delta_{gh})^{v_{ig}}$

Step 4: $\rho_h = \frac{1}{J}\sum_j w_{jh}$ and δ_{gh} as in Step 2.

end

Algorithm 1: Poisson SEM-Gibbs algorithm

3. Self-Organized Co-Clustering

3.1. An easy-to-read structure

In the latter section, all the δ_{gh} are unrelated, and consequently, each block should be interpreted separately from each other. In the model we propose, this independence does not hold true anymore: a structure is forced among the blocks so that the result is easier to read. Thus, for a given block (g, h), the corresponding block effect δ_{gh} will either be specific to column-cluster h with $\delta_{gh} = \delta_h$, or non-specific, with $\delta_{gh} = \delta$. In the case of non-specific block effect $\delta_{gh} = \delta$, the block (g, h) is considered as a noisy block. We refer to it as a 'non-meaningful' block, and it shares the same δ with all the other non-meaningful blocks. In the case of $\delta_{gh} = \delta_h$, the block (g, h) is 'meaningful', and shares the same δ_h with all the meaningful blocks of the same column-cluster h. In this case, the terms of the h^{th} column-cluster are considered to be specific to the

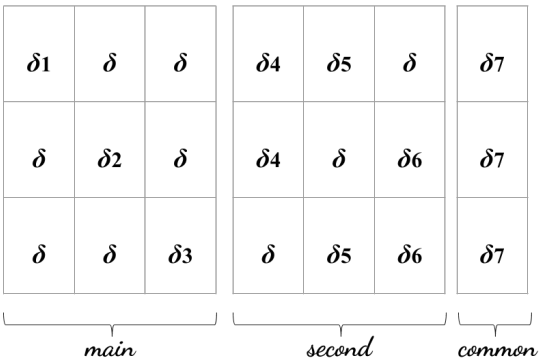

Figure 2: Co-clustering structure of the Self-Organized Co-Clustering model, with block effect parameters, in the case $G = 3$.

documents of one or several row-clusters.

To organize these meaningful and non-meaningful blocks, several rules are given. First of all, after choosing the number of row-clusters G, the co-clustering necessarily has $H = G + \binom{G}{2} + 1$ column-clusters. Moreover, the column-clusters are divided into three sections called *main*, *second* and *common*. The purpose of these sections is explained here.

The *main* section concerns the first G column-clusters, for $h \in \{1, ..., G\}$. In each column-cluster h of this section, only one block is meaningful and parameterized by δ_h. All the other blocks are non-meaningful and parameterized by δ. Consequently, for each cluster of documents (row-cluster), the meaningful block indicates the terms that are specific to these documents. As a result, in the *main* section, the meaningful blocks are located on the diagonal, and the other ones are the non-meaningful ones.

The *second* section concerns the following $\binom{G}{2}$ column-clusters ($h \in \{G + 1, ..., G + \binom{G}{2}\}$). In each column-cluster h of this section, two blocks are meaningful. Consequently, each column-cluster contains terms that are specific to two clusters of documents (row-clusters).

Finally, the *common* section consists of only one column-cluster and gathers the terms that are common to all documents.

This structure, as well as the corresponding block effect δ, are illustrated by Figure 2, in which we clearly see the meaningful blocks with $\delta_{gh} = \delta_h$ and non-meaningful blocks with $\delta_{gh} = \delta$. We also discern the organization among these blocks and the three different sections *main*, *second* and *common*. For instance, in the *main* section, the first column cluster is considered to be specific to first row-cluster, thus only the column cluster's first block has its own specific distribution with δ_1. On the other hand, the other blocks of this column-cluster are considered to be non-meaningful, and have a block effect parameter δ, which is common to all non-meaningful blocks. In the *second* section, we note, for example, that for $h = 4$ blocks $(1, 4)$ and $(2, 4)$ are meaningful, and share the same block effect δ_4. This means that terms from column-cluster 4 are specific to documents from row-clusters 1 and 2. Moreover, block $(4, 3)$ is non-meaningful and has the same effect δ as the other non-meaningful blocks. The *common* section is a bit particular insofar that it contains only one column-cluster, so $h = 7$. This column-cluster contains the terms that are specific to all groups of documents and its corresponding blocks all share the same δ_7.

3.2. The SOCC model and its inference

From Section 3.1, knowing the column-cluster h we can write: $g \in \mathcal{C}_h \cup \overline{\mathcal{C}}_h$, such that $\mathcal{C}_h$ are the meaningful blocks of column-cluster h and $\overline{\mathcal{C}}_h$ are the non-meaningful blocks of column-cluster h. In this case, the probability of the SOCC model is written as:

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{\substack{(\boldsymbol{v},\boldsymbol{w}) \\ \in V\times W}} \prod_{ig} \gamma_g^{v_{ig}} \prod_{jh} \rho_h^{w_{jh}} \prod_{ijh} \prod_{g\in\mathcal{C}_h} f(x_{ij};\delta_h)^{v_{ig}w_{jh}} \prod_{g\in\overline{\mathcal{C}}_h} f(x_{ij};\delta)^{v_{ig}w_{jh}}. \quad (4)$$

The complete log-likelihood is given by:

$$
\begin{aligned}
&L_c(\boldsymbol{\theta}; \boldsymbol{x}, \boldsymbol{v}, \boldsymbol{w}) = \\
&\sum_{ig} v_{ig} \log \gamma_g + \sum_{jh} w_{jh} \log \rho_h + \sum_{ijh} \Bigg(\\
&\sum_{g \in \mathcal{C}_h} v_{ig} w_{jh} \left[x_{ij} \log(n_{i.} n_{.j} \delta_h) - n_{i.} n_{.j} \delta_h - log(x_{ij}!) \right] + \\
&\sum_{g \in \overline{\mathcal{C}}_h} v_{ig} w_{jh} \left[x_{ij} \log(n_{i.} n_{.j} \delta) - n_{i.} n_{.j} \delta - log(x_{ij}!) \right] \Bigg).
\end{aligned}
\tag{5}
$$

As in Section 2.2, the SEM-Gibbs algorithm is chosen to estimate the partitions $(\boldsymbol{v}, \boldsymbol{w})$ and parameters $\boldsymbol{\theta} = (\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\delta})$ with $\boldsymbol{\delta} = (\delta, \delta_1, \ldots, \delta_H)$. In contrast with the Poisson LBM, the Poisson distribution $f(x_{ij}; \delta_{gh})$ of block (g, h) will depend on the meaningfulness of block (g, h). For all $h \in \mathcal{H}$ if $g \in \mathcal{C}_h$, then $f(x_{ij}; \delta_{gh}) = f(x_{ij}; \delta_h)$, and if $g \in \overline{\mathcal{C}}_h$, then $f(x_{ij}; \delta_{gh}) = f(x_{ij}; \delta)$, where f is the Poisson p.d.f. given by Equation (2).

The SEM-Gibbs algorithm proposed for the Self-Organized Co-Clustering model inference is summarized in Algorithm 2. It iterates the partitions sampling and the maximization of the parameters (steps 1 to 4) during a given number of iterations (nbSEM). The final parameter estimation, now denoted by $\hat{\boldsymbol{\theta}}$, is obtained by averaging the model parameters over the sample distribution (after a burn-in period). Lastly, the final partitions $\hat{\boldsymbol{v}}$ and $\hat{\boldsymbol{w}}$ are estimated with $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$, using another Gibbs sampler.

Choice of number of iterations. For the SEM-Gibbs algorithm, two numbers must be chosen: the total number of SEM-Gibbs iterations (nbSEM) and the number of iterations for the burn-in period. These numbers are graphically chosen by visualizing the values of the model's parameters along the SEM-Gibbs iterations. The parameters must reach their stationary state after the burn-in period, and the remaining number of iterations until the end must be sufficient to compute their respective means. Less subjective ways exist to assess the distribution's stationarity. In [22], the authors propose a general approach to monitor the convergence of MCMC outputs in which parallel chains are run with start-

Input: $\boldsymbol{x}$, G, H

Initialization: $\boldsymbol{v}$, $\boldsymbol{w}$, $\gamma_g = \frac{1}{N}\sum_i v_{ig}$, $\rho_h = \frac{1}{J}\sum_j w_{jh}$

for *i in 1:nbSEM* **do**

Step 1: Sample $\boldsymbol{v}$ such that:

$$p(v_{ig} = 1|\boldsymbol{x}, \boldsymbol{w}; \boldsymbol{\theta}) \propto \gamma_g \times \prod_{jh} f(x_{ij}; \delta_{gh})^{w_{jh}}$$

Step 2: $\gamma_g = \frac{1}{N}\sum_i v_{ig}$,

$$\delta = \frac{\sum_{ijhg\in\overline{C}_h} v_{ig} w_{jh} x_{ij}}{\sum_{ijhg\in\overline{C}_h} v_{ig} w_{jh} n_{i.} n_{.j}},$$

$$\delta_h = \frac{\sum_{ijg\in C_h} v_{ig} w_{jh} x_{ij}}{\sum_{ijg\in C_h} v_{ig} w_{jh} n_{i.} n_{.j}}.$$

Step 3: Sample $\boldsymbol{w}$ such that:

$$p(w_{jh} = 1|\boldsymbol{x}, \boldsymbol{v}; \boldsymbol{\theta}) \propto \rho_h \times \prod_{ig} f(x_{ij}; \delta_{gh})^{v_{ig}}$$

Step 4: $\rho_h = \frac{1}{J}\sum_j w_{jh}$, δ and δ_h as in Step 2.

end

Algorithm 2: SEM-Gibbs algorithm for the SOCC model.

ing values that are spread relative to the posterior distribution. Convergence is confirmed when the output from all chains is indistinguishable. Although this method is relevant here, we did not use it for two reasons. Firstly, we did not use it to avoid increasing the overall execution time of the algorithm. Secondly, this method is not necessarily useful in the SOCC model's case. Indeed, since this model is very constrained, the number of iterations required to reach convergence is limited. We can see in Figure 3, which represents the change in the SOCC parameters for simulated data that these parameters stabilize after very few iterations.

3.3. Model selection

The definition of a model selection criterion has two purposes. Firstly, in the context of unsupervised methods, choosing the number of row-clusters G is

an issue. One of the great advantages of the SOCC model is that the number of column-clusters H is directly fixed by the number of row-clusters G. Indeed, as explained before, $H = G + \binom{G}{2} + 1$. However, the choice for the number of row-clusters G is still a problem. Secondly, as described in Algorithm 2, the SEM-Gibbs algorithm starts with a random initialization of partitions $(\boldsymbol{v}, \boldsymbol{w})$. However, this initialization has an impact on the convergence of the algorithm and on the resulting estimations. It is therefore recommended to execute the algorithm several times with different initializations and to have a criterion to choose the best solution.

The classical criteria, such as BIC [23], rely on penalizing the maximum log-likelihood value $L(\hat{\boldsymbol{\theta}}; \boldsymbol{x})$. However, due to the dependency structure on the row and column partitions conditionally to $\boldsymbol{x}$, the log-likelihood is not tractable.

Alternatively, an approximation of the ICL information criterion [14], referred to as 'ICL-BIC', can be used to overcome this problem. The key point is that this latter vanishes since ICL relies on the complete latent block information $(\boldsymbol{v}, \boldsymbol{w})$, instead of integrating on it as in the case with BIC. In particular, [21] detailed how to express ICL-BIC for the general case of categorical data. It is possible to use the ICL-BIC expression given by these authors by following their work in a stepwise manner. The resulting ICL-BIC is expressed by:

$$\begin{aligned} \text{ICL-BIC}(G) = &\log p(\hat{\boldsymbol{\theta}}; \boldsymbol{x}, \hat{\boldsymbol{v}}, \hat{\boldsymbol{w}}) \\ &- \frac{1}{2}(G-1)\log N - \frac{1}{2}(H-1)\log J - \frac{1}{2}(H+1)\log(NJ). \end{aligned} \tag{6}$$

The number G of row-clusters maximizing this criterion must be retained.

4. Numerical Experiments

In this section, we assess the quality of the SOCC model. First of all, we chose seven clustering, co-clustering and topic-modelling methods to compare the results: we list them in Section 4.1 and refer to them as 'baselines'. In Section 4.2, we simulate data through the SOCC model's process generation. We

run the baselines algorithms and compare their results with those of the SOCC model, in terms of partition estimation. We also evaluate the behaviour of the ICL criterion for choosing the number of row-clusters. In Section 4.3, we used real textual data sets whose documents are known to belong to some predefined classes and compared the row-clustering (or column-clustering) quality with the baseline methods. We conclude this section by illustrating with a use case how the SOCC model can be helpful for interpreting the co-clustering results.

4.1. Baselines

Seven clustering, co-clustering and topic-modelling methods were selected as baselines to compare our results. Two of them are based on the Latent Block Model. The Poisson Latent Block Model (PLBM,[13]), as detailed in Section 2, is a co-clustering algorithm that uses the direct application of the Latent Block Model. The Sparse Poisson Latent Block Model [18], referred to as 'SPLBM', is a constrained version of the Poisson Latent Block Model, which was also developed to co-cluster document-term matrices. This model, already described in the introduction, constrains its structure to the *main* structure of our model. Both models were implemented in `C++` from the pseudo-code of their respective papers. The Information Theory Co-Clustering method, referred to as 'ITCC' [24], is a co-clustering technique that uses information theory and the mutual information to discover the blocks. We used the `C++` implementation provided by their authors. The Orthogonal Non-negative Matrix Tri-Factorization method, referred to as 'ONMTF' [9], is a co-clustering algorithm based on matrix factorization. We implemented the pseudo-code provided in `R`. The Non-negative Matrix Factorization NMF [25] is a clustering algorithm based on matrix factorization. The R Package `NMF` [26] was used for the experiments. The Spherical Kmeans clustering method ('Skmeans') is the implementation of the kmeans algorithm, but with embedding of the Cosine similarity (and not the Euclidean distance). The R Package `skmeans` [27] was used for the experiments. Latent Dirichlet Allocation (LDA) [3] is a generative statistical model for topic modelling. The R package `textmineR` implementation was used to use it on the data

Table 1: Simulated parameters $\delta_{gh} \times 10^{-7}$. For each cell x_{ij} the Poisson parameter is equal to $n_{i.}n_{.j}\delta_{gh}$, with row margins $n_{i.}$ equal to 2455 on average, and columns margins $n_{.j}$ equal to 249 on average.

Cluster	1	2	3	4	5	6	7
1	8.6	2.9	2.9	49.8	47.8	2.9	34.0
2	2.9	9.0	2.9	49.8	2.9	52.9	34.0
3	2.9	2.9	9.4	2.9	47.8	52.9	34.0

sets. To assess the quality of the row-clusters, all of these seven methods were used. To assess the quality of the column-clusters, we obviously only selected the four co-clustering methods.

4.2. Simulated data set

4.2.1. Simulation setting

A data set with $N = 120$, $J = 1\ 200$, $G = 3$ and $H = 7$ was simulated. The parameters were chosen arbitrarily: the row mixing proportions $\boldsymbol{\gamma}$ are equal to $(.33, .33, .33)$ and the column mixing proportions $\boldsymbol{\rho}$ are equal to $(.08, .08, .17, .17, .17, .08, .25)$. The block effects are given in Table 1.

For the SOCC inference, the total number of iterations of the SEM-Gibbs algortihm was fixed to 50 with a burn-in period of size 35. In Figure 3, the evolution of parameters δ and δ_h for the *main* section is plotted. We see that the parameters stabilize in less than ten iterations. The numbers of fixed iterations are therefore enough to reach the stationary state.

4.2.2. Results

The SOCC model was run on 100 simulations, and the Adjusted Rand Index, referred to as 'ARI' [28] between the true partitions and the estimated partitions were computed. The ARI for row-clusters was always equal to 1. Regarding the column-clusters, the mean ARI was equal to .99. It shows that the inference algorithm for SOCC functions appropriately. It is worth noting that 25% of

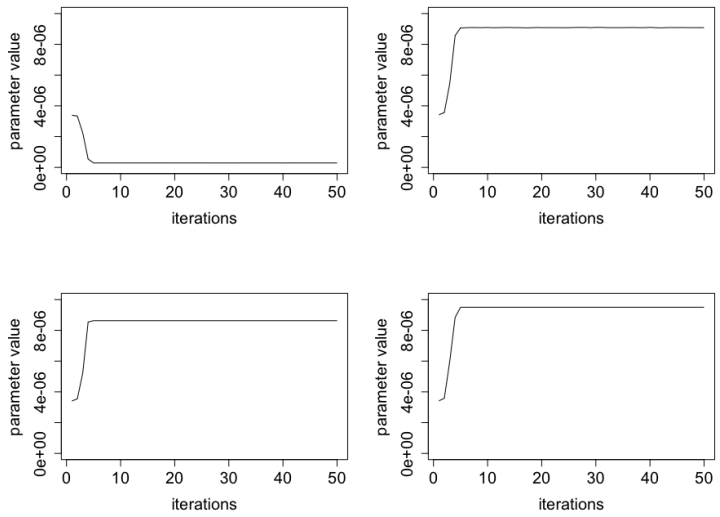

Figure 3: From left to right, and from top to bottom: change in parameters δ, δ_1, δ_2, δ_3 when executing the algorithm on the simulated data set. The parameters reach their stationary state in less than 10 iterations.

runs failed to reach a valid solution, systematically leading to empty clusters solutions. Such behaviour is a well-known drawback of co-clustering procedures [29, 30]. Nevertheless, this relative frequency of failures is not too high and not detrimental for the use of the SOCC model. When we obtain a solution with some empty clusters, we just have to restart the algorithm with another random initialization.

Furthermore, we executed the competitors' algorithms on this data set: the ARI boxplots for all methods are available in Figure 4. We see that on this simple data set, most of the methods perform well in terms of row clustering. This is the reason why we challenge the methods using real and more difficult data sets in Section 4.3.

4.2.3. Selection for the number of row clusters

For each of the 100 simulations, the co-clustering was run for $G = \{2, 3, 4, 5\}$ and the ICL criterion was computed. Table 2 presents the number of times each

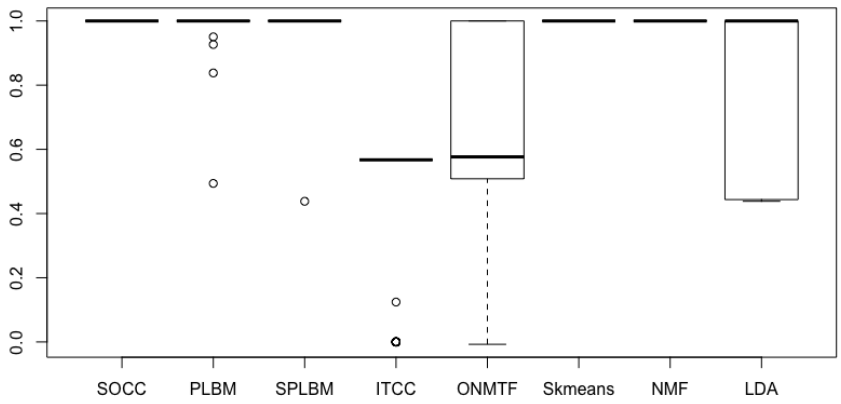

Figure 4: ARI for SOCC model and competitors models on simulated data set.

Table 2: Number of row and column-clusters (G, H) selected by ICL-BIC on the 100 simulated data sets, the right one being $(3, 7)$.

(G, H)	(2,6)	**(3,7)**	(4,11)	(5,16)
# chosen	25	**75**	0	0

G was selected: the right number was selected in 75% of the cases. For the remaining 25% executions, $G = 2$ was selected.

4.3. Real data sets experiments

In this section, real labelled data sets are used to assess the quality of the proposed method. We describe the data sets that were used, the methods the SOCC was compared to and the results.

4.3.1. Real data sets

Eight data sets were retained for this Section. The **classic3** data set (dimensions $3,891 \times 5,236$) and the **classic4** data set[1] (dimensions $7,094 \times 5,896$) consist respectively of 3 different document collections (CISI, CRANFIELD, and MEDLINE) and 4 different document collections (CACM, CISI, CRANFIELD, and MEDLINE). **Pubmed5** ($12,648 \times 8,863$), **Pubmed4** ($11,131 \times 8,257$) and **Pubmed3** ($9,582 \times 7,454$) were built from the collection Pubmed10 [31], with approximately $15,500$ medical abstracts from the Medline database, partitioned across 10 different diseases and published between 2000 and 2008. Pubmed3 contains the three largest classes, while Pubmed4 (and Pubmed5) contains the four (and five) largest classes. The classes, ranked from the largest to the smallest, include documents about otitis, migraine, age-related macular degeneration, kidney calculi and hay fever. **Pubmed4min** ($2,121 \times 3,660$) was also extracted from the Pubmed10 data set. However, only the four smallest classes were extracted. The documents are about jaundice, Raynaud disease, chickenpox and gout. The **sports** ($8,580 \times 14,870$) and **yahoo** ($2,340 \times 10,431$) data sets were obtained from the CLUTO toolkit [32]. The yahoo data set contains 6 different document categories with each document corresponding to a web page listed in the subject hierarchy of *Yahoo!*. The sports data set contains documents regarding 7 different sports including baseball, basketball, bicycling, boxing, football, golfing and hockey.

Discussion on the number of clusters. The baselines data sets never have more than 7 known clusters in line, when other methods such as [18] execute their algorithm on data sets with up to 50 row-clusters. A limitation of the SOCC model is its difficulty in using it when the number of classes G is greater than 10. When $G = 10$, we have $G = G + \binom{G}{3} + 1 = 56$. With 56 column-clusters, the resulting co-clustering loses its interpretability, which is supposed to be a

[1] http://www.dataminingresearch.com/index.php/2010/09/classic3-classic4-datasets/

strength of the model. Therefore, it is recommended not to use the model when G is superior to 7.

4.3.2. Assessing the quality of row-clusters

To assess the document clustering quality, the ARI between the known partitions and those estimated were computed. For each data set, each method was executed 30 times. Figure 5 plots the ARIs boxplots for all data sets and methods. We can see on these boxplots that the SOCC approach obtains the highest median ARIs for the classic3, pubmed4min and sports data sets. On the classic3 data set, the SOCC model obtains a median ARI of 0.96, and so does the NMF method. The model with the second highest median ARI (0.95) is the SPLBM model. On the pubmed4min data set, the median ARI for the SOCC model is equal to 0.55. The PLBM method yields the second highest ARI value with 0.46. Finally, on the sports data set, the SOCC obtains the highest median ARI value (0.44), and the NMF methods ranks second with an ARI value equal to 0.43.

On the other data sets, the SOCC model obtains satisfactory results and ranks as the second-best method in terms of ARI after Skmeans. This latter clustering method yields better results on data sets pubmed3, pubmed4, and pubmed5 but it presents one of the worst performances for classic4, pubmed4min and sports. Therefore, even if it obtains good results on some data sets, its inconsistency on the other data sets makes it an unreliable method. For this reason, SOCC seems to be the best method from a document clustering standpoint. The reason for this success is probably due to the model's parsimony.

4.3.3. Assessing the quality of column-clusters

In most studies, the evaluation of co-clustering algorithms is only based on resulting row-clusters. This is due to the lack of public data sets providing the true partitions for both observations and features. In document clustering, for example, popular benchmarks provide the true document labels, while the term clusters remain unknown. To overcome this problem and improve over

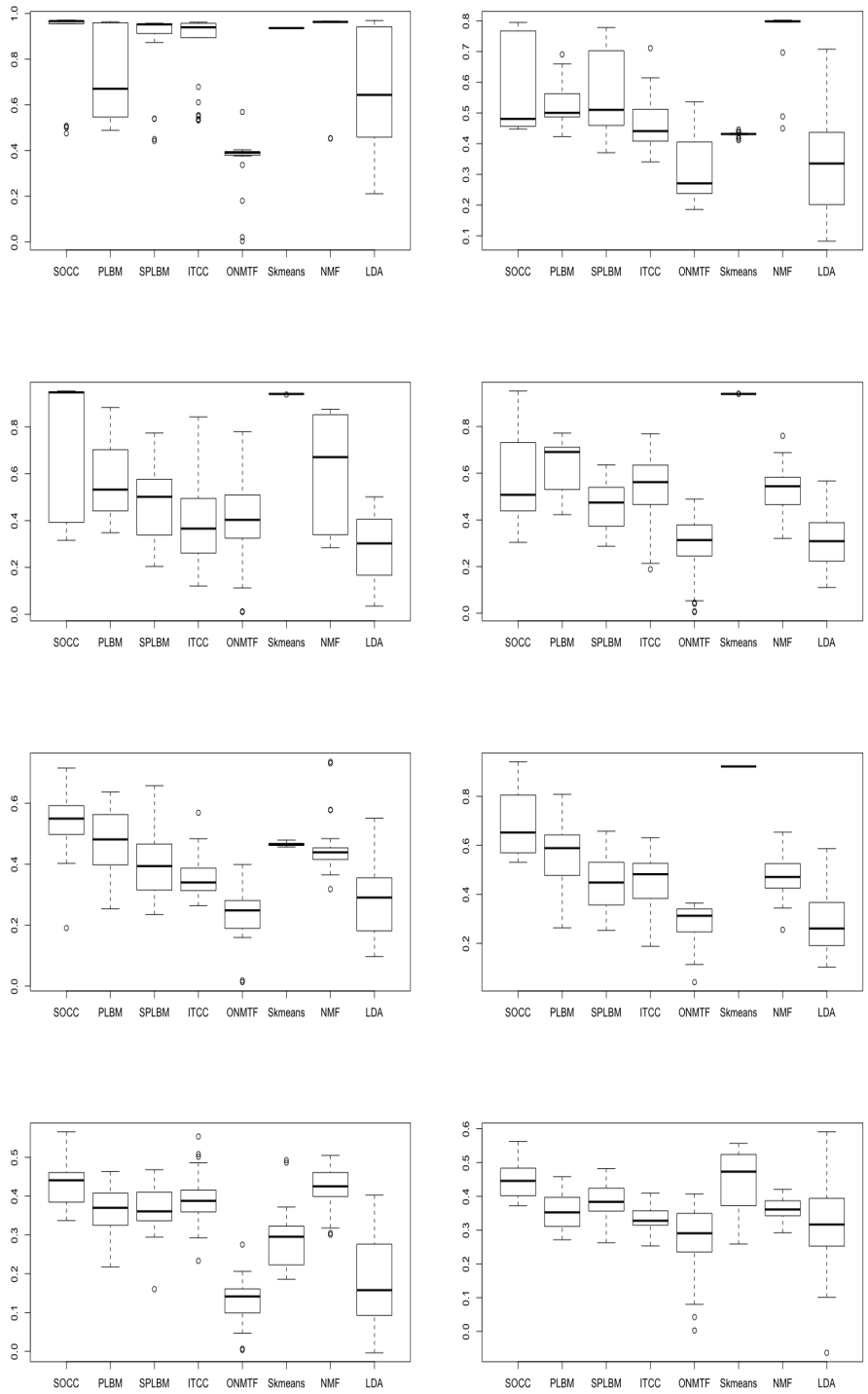

Figure 5: ARIs for document clustering. From left to right and top to bottom: classic3, classic4, pubmed3, pubmed4, pubmed4min, pubmed5, sports, yahoo.

currently used evaluation methods, we propose the following strategy. For a given column-cluster, the ten most frequents terms are extracted. We compute the average Jaccard similarity between these terms on all the documents: this value is considered as a proximity measure between terms of the column-cluster. We average this proximity measure over all the column-clusters. In terms of interpretation, this criterion based on Jaccard similarities is used to assess how a co-clustering gathers terms that often occur in the same document. We report the scores obtained by the methods on the data sets in Table 3. From these results, it can be seen that for the classic4, pubmed3, pubmed4, pubmed4min, pubmed5, sports and yahoo data sets, all algorithms perform equally well but the SOCC model has the highest averaged score. Regarding the classic3 data set, ONMTF yields a better result (.89), but is closely followed by the SOCC model (.88).

Table 3: Average similarity measurements between the top 10 terms of each column-cluster.

Data set	SOCC	PLBM	SPLBM	ITCC	ONTMF
Classic3	.88 (.07)	.86 (.08)	.86 (.08)	.86 (.08)	**.89** (.07)
Classic4	**.91** (.06)	.88 (.07)	.88 (.07)	.87 (.07)	.87 (.07)
Pubmed3	**.85** (.13)	.77 (.13)	.79 (.12)	.76 (.13)	.80 (.08)
Pubmed4	**.88** (.12)	.80 (.15)	.80 (.13)	.80 (.14)	.81 (.09)
Pubmed4min	**.87** (.11)	.79 (.13)	.81 (.09)	.80 (.13)	.84 (.08)
Pubmed5	**.90** (.12)	.78 (.13)	.81 (.13)	.83 (.13)	.85 (.08)
Sports	**.88** (.11)	.79 (.11)	.79 (.11)	.77 (.11)	.78 (.10)
YahooKB1	**.85** (.20)	.67 (.31)	.70 (.33)	.69 (.31)	.69 (.31)

4.3.4. pubmed4min use case

In this section, we demonstrate using the Pubmed4min data set that the SOCC results are easy-to-interpret. Regarding the *main* section, when we seek the 10 most frequent terms of the first column-cluster, we get 'varicella', 'vaccin', 'ag', 'children', 'year', 'immun', 'zoster', 'hospit', 'chickenpox' and 'adult'. These terms are closely related to chickenpox (or varicella), so we can easily guess that the first row-cluster's documents are about chickenpox. When we seek the 10 most frequent terms of the second column-cluster, we get 'jaundic', 'obstruct', 'liver', 'bile', 'biliari', 'hepat', 'duct', 'rat', 'stent' and 'bilirubin'. Again, we can easily assert that the second row-cluster's documents are about jaundice. Regarding the *second* section, if we look at column-cluster 5, which corresponds to the terms specific to row-clusters 1 and 2, we get: 'rate', 'complic', 'neg', 'mortal', 'morbid', 'infant', 'neonat', 'bacteri', 'safe', 'inva'. These terms are mostly related to children, which seems consistent since jaundice and chickenpox are very common in toddlers and newborns. Furthermore, jaundice can occur as a complication of chickenpox, justifying the presence of 'complic' in the list.

5. Harry Potter use case

In this section, we use the SOCC model on the **Harry Potter** data set. For each stage of performing a co-clustering, we show the difficulties encountered by the classical co-clustering methods and how the SOCC model overcomes them. The Harry Potter data set contains the first three volumes of the famous series ([33, 34, 35]), entitled 'Harry Potter and the Philosopher's Stone', 'Harry Potter and the Chamber of Secrets' and 'Harry Potter and the Prisoner of Azkaban'. In the resulting Document-Term matrix, each line represents a chapter, and each column represents a term.

5.1. Co-clustering set up

Data set pre-processing. The original text was changed. Firstly, the punctuation

and numbers were removed. Secondly, the terms that appeared only once were removed because they do not often add useful information. The whole was then transformed to a classic Document-Term frequency matrix. The resulting matrix is of dimensions $N = 57$ and $J = 6,884$.

Setting the number of iterations. When dealing with a new data set, the user must choose the total number of iterations and the number of burn-in iterations. For this, they must execute the SEM-Gibbs algorithm with the different numbers of clusters they want to test (see paragraph 'Finding the right numbers of clusters' below) with an arbitrary number of iterations. Then, they must check that the parameters reached their stationary state before the number of burn-in iterations. For the Harry Potter data set, and with $G = 7$, we see in Figure 6 that the parameters reached their stationary state before the 75th iteration. The total number of iterations can then be fixed to 100 and the number of burn-in iterations to 75.

Finding the right number of clusters. For the baselines PLBM, ONMTF and ITCC, the user has to define two numbers of clusters G and H at this stage. Furthermore, the ONMTF and ITCC methods have no criteria to define these numbers. The SOCC model induces H from G so the user only has to choose G. Furthermore, the ICL criterion defines the best number of clusters once the algorithm is run on the different possibilities. On the Harry Potter data set, we ran the SEM-Gibbs algorithm for $G = \{2, 3, 4, 5, 6, 7, 8\}$, and got the corresponding ICL values. The largest ICL value was obtained with $G = 7$. Table 4 presents the maximum ICL values for each number of row-clusters tested. Figure 7 plots the Document-Term matrix sorted by row-clusters and column-clusters.

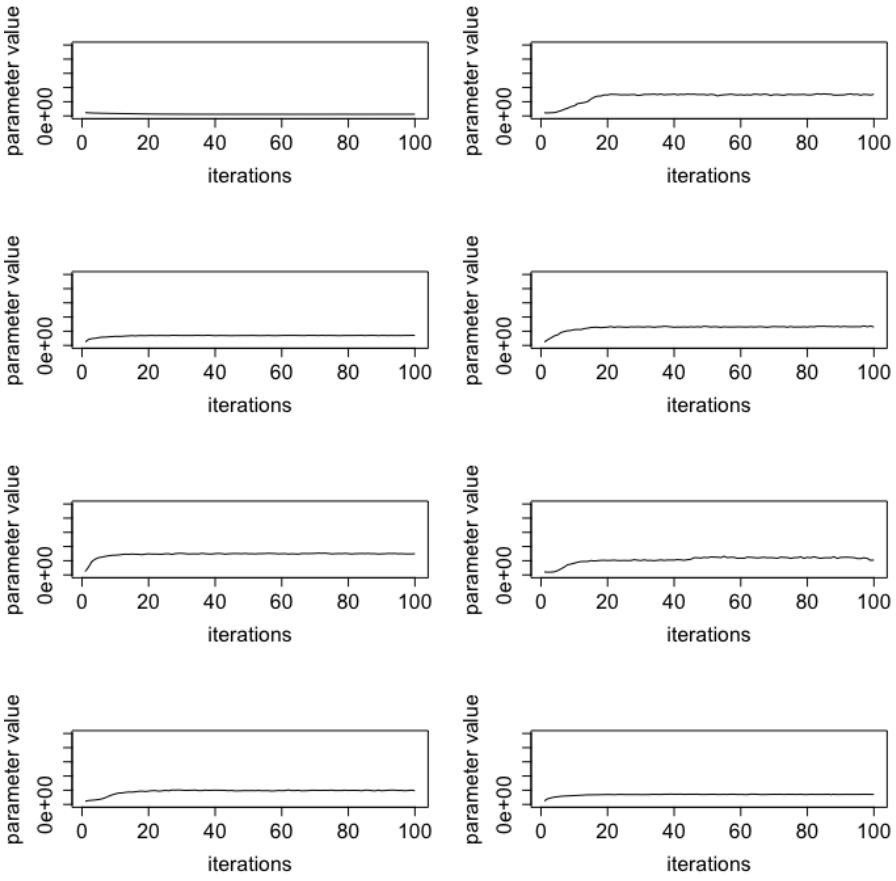

Figure 6: Changes in parameters for the Harry Potter data set for δ, δ_1, δ_2, δ_3, δ_4, δ_5, δ_6, δ_7.

Table 4: Maximum ICL values for each G tested.

number of row clusters G	2	3	4	5	6	7	8
max ICL value	-231774.9	-228133.4	-226650.7	-225895.4	-226709.2	**-225072.6**	-226035.7

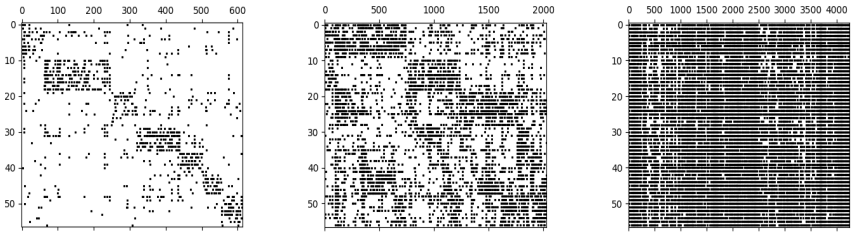

Figure 7: Co-clustering of the Harry Potter data set with the SOCC method. From left to right: the *main*, the *second* and the *common* sections. The graphic was produced using the `Python` function *spy*() with argument *markersize* set to 1.2.

5.2. Interpretation of the results

At this stage, the user has a co-clustered Document-Term matrix. Using the methods ONMTF, ITCC and PLBM, they are able to obtain the chapters of the books that are gathered into the same group. However, they cannot easily know the main topic of each group. For example, for the PLBM method, they should find the highest block effect and observe the corresponding row-cluster and column-cluster to obtain the relevant chapters and terms. With the SOCC model, the user can directly know which blocks are of interest. In this section, we studied the terms belonging to column-clusters and found the main underlying topic. We do not list every term but chose the ones that are most related to the topic concerned. Here, we develop an interpretation of the column-clusters that are related to the first row-cluster. The entire interpretation of the results is available as an appendix.

Interpretation of column-clusters for the main section. Seven clusters in line were detected by the SOCC model. There are, therefore, also seven column-clusters in the main section. The first contains the terms specific to the chapters of the first row-cluster, the second contains the terms specific to the chapters of the second row-cluster, and so on. We highlight below that this specific co-clustering structure is easily readable by users. Some terms specific to the chapters of the first row-cluster are ‘agony’, ‘hewhomustnotbenamed’, ‘pain’,

'quirrell' and 'serpent'. These terms refer to Harry Potter's enemy, called Lord Voldemort. People are so afraid of him that they never say his name aloud and thus refer to him as 'he-who-must-not-be-named'. He loves serpents and torturing his opponents. Quirrell is his servant in Volume 1. We proposed the label Voldemort for this cluster. We did the same work on the six other row-clusters (see the appendix for the full interpretation). The topics of the other clusters of chapters are related to: animagus, Quidditch, the Dursleys, the Weasleys, classmates and magical creatures .

A note on the main section compared to the SPLBM model. Until now, most of the other co-clustering techniques have shown weaknesses in the overall process: ONMTF and ITCC do not have a criterion to choose the number of blocks. For PLBM, the two numbers G and H have to be chosen and interpretation is difficult once the co-clustering is performed. The SPLBM model does not have these problems. In fact, the SPLBM is similar to the *main* section in the sense that it considers the meaningful blocks as being on the diagonal of the matrix. However, the *main* section is more selective and interpretable. Indeed, when running the SPLBM on the Harry Potter data set with $G = 7$, there will be 983 terms per column-clusters on average. It is therefore difficult to read them all and grasp what each row-cluster is about. In our case, the *second* and *common* sections get a large majority of the terms. In the same example, on the Harry Potter data set, the *main* section has 78 terms on average. Therefore, it is easier to read them quickly and get the topic of each row-cluster, as we just demonstrated above.

Interpretation of column-clusters for the second section. With regard to the *second* section, as mentioned before, its corresponding column-clusters have terms that are related to two row-clusters. Now that we know what each row-cluster is about individually, due to the *main* section, we can see the terms that link them. The SOCC model looks for common words for every row-cluster pair.

This can be a limitation: for example, the chapters related to the Dursleys and the chapters related to Quidditch do not have a lot in common and the column-cluster related to these two groups of chapters only contains the word 'card', which is unrelated to both. However, most of the column-clusters that relates to two clusters of chapters are of interest to the user. Here are some examples for the column-clusters related to row-cluster 1 (see the appendix for the full interpretation):

- Row clusters 1 and 4, which are about Voldemort and the Dursleys share meaningful blocks in column-cluster 10. The corresponding terms include 'mother', 'nephew', 'petunias' and 'scar'. Petunia Dursley is Harry's aunt. She is connected to Voldemort because he killed her sister. He also attempted to kill Harry as a young boy but he survived, and he was left with a scar on his forehead. Petunia then adopted her nephew.

- Row-clusters 1 and 5, which are about Voldemort and the Weasleys share meaningful blocks in column-cluster 11. This column-cluster has terms such as 'basilisks', 'tom', 'riddle' and 'ginny'. This makes sense because Ginny is Mr. and Ms. Weasley's daughter. She is closely connected to Voldemort in Volume 2. The wizard finds a way to bring Tom Riddle to life. Tom is the past version of himself, when he was a normal teenager in the school. Tom casts a spell on Ginny so that she wakes the giant basilisk serpent up in the Chamber of Secrets. The snake then attacks the school's students.

- Row-clusters 1 and 6, which are about Voldemort and Harry's classmates share meaningful blocks in column-cluster 12. The corresponding terms include 'ernie', 'petrified' and 'serpents'. In Volume 2, Ernie is Harry's classmate. In duelling class, Harry speaks to a serpent, an ability both he and Voldemort hold. Ernie thinks that he is ordering the snake to attack Justin Finch-Fletchey. His suspicions grow when Justin is found petrified in the corridor. He spreads the rumour that Harry's destiny was

to become a powerful dark wizard and that is why Voldemort wanted to kill him.

A note on the common section. The *common* section is composed by a unique column-cluster. However, this cluster contains the majority of the terms, with $\rho_{29} = 0.63$ (thus, 63% of terms). The corresponding terms include 'harry', 'potter', 'ron', 'hermiones', 'granger' and 'hogwarts'. These terms are very important for the Harry Potter story, and at first, it seems odd that they are not in the *main* section. However, this phenomenon is explained by considering that the *common* section includes the terms that are frequent to all row-clusters. Furthermore, if the term 'harry' appeared in a column-cluster of the *main* section, it would not bring any valuable information about the chapters of this row-cluster, since Harry is present in all chapters.

5.3. Conclusions on the study of the Harry Potter data set

This section brought an insight on how to use the SOCC model on a completely unsupervised data set. Furthermore, for each stage of the process of co-clustering, we indicated how the tasks left to the user are easier with the SOCC model in comparison with the other co-clustering methods.

6. Conclusion and future work

In this paper, we propose the SOCC model, a novel approach to easily co-cluster textual data sets. The model offers easy-to-read results, and quickly shows the terms that are specific to one group of documents, the terms that are specific to two groups of documents, the terms that are common to all documents. The resulting algorithm is not only more accurate than other state-of-the-art methods but it is also able to detect the number of co-clusters, as a result of the ICL-BIC criterion. An R package `SOCC` is available upon request to perform these functionalities.

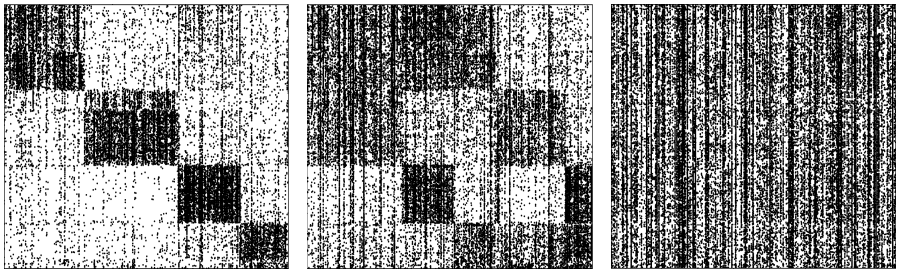

Figure 8: Co-clustering of pubmed4min data set with the SOCC method. From left to right: the *main*, the *second* and the *common* sections. The graphic was produced using the `Python` function *spy*() with the argument *markersize* set to 1.3.

In future work, we could define other structures, for example with clusters of terms specific to 3 or more groups of documents. The first concern here is the increasing number of column-clusters (which would require at least $\binom{G}{3}$ more column-clusters). Also, it would be interesting to investigate a more developed model selection: we can allow the structure to not have all $G + \binom{G}{2} + 1$ column clusters. For example, in Figure 8, we see the pubmed4min SOCC co-clustering with $G = 4$. We know that the *second* part comprises $\binom{4}{2} = 6$ column-clusters. We can easily notice five of them, but the sixth one is very small: is this column-cluster necessary? We could use the ICL criterion to dispose of the irrelevant column-clusters. However, loosening the strict structure assumption would result in other issues arising: testing all solutions could significantly increase the overall execution time.

References

[1] L. Wu, I. E.-H. Yen, K. Xu, F. Xu, A. Balakrishnan, P.-Y. Chen, P. Ravikumar, M. J. Witbrock, Word mover's embedding: From Word2Vec to document embedding, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 4524–4534.

[2] T. Thongtan, T. Phienthrakul, Sentiment classification using document

embeddings trained with cosine similarity, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, Association for Computational Linguistics, Florence, Italy, 2019, pp. 407–414.

[3] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, Journal of Machine Learning Research 3 (2003) 993–1022.

[4] M. V. Mantyla, M. Claes, U. Farooq, Measuring lda topic stability from clusters of replicated runs, in: Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM 18, Association for Computing Machinery, New York, NY, USA, 2018.

[5] G. Drosatos, E. Kaldoudi, A probabilistic semantic analysis of ehealth scientific literature, Journal of Telemedicine and Telecare.

[6] X. Yan, J. Guo, Y. Lan, X. Cheng, A biterm topic model for short texts, in: Proceedings of the 22nd International Conference on World Wide Web, WWW 13, Association for Computing Machinery, New York, NY, USA, 2013, pp. 1445–1456.

[7] Q. Zhu, Z. Feng, X. Li, GraphBTM: Graph enhanced autoencoded variational inference for biterm topic model, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 4663–4672.

[8] C. Laclau, I. Redko, B. Matei, Y. Bennani, V. Brault, Co-clustering through optimal transport, in: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 2017, pp. 1955–1964.

[9] C. Ding, T. Li, W. Peng, H. Park, Orthogonal nonnegative matrix t-factorizations for clustering, in: Proceedings of the 12th ACM SIGKDD

International Conference on Knowledge Discovery and Data Mining, KDD '06, ACM, New York, NY, USA, 2006, pp. 126–135.

[10] H. Wang, F. Nie, H. Huang, C. Ding, Nonnegative matrix tri-factorization based high-order co-clustering and its fast implementation, in: 2011 IEEE 11th International Conference on Data Mining, 2011, pp. 774–783.

[11] N. D. Buono, G. Pio, Non-negative matrix tri-factorization for co-clustering: An analysis of the block matrix, Information Sciences 301 (2015) 13 – 26.

[12] A. Salah, M. Ailem, M. Nadif, Word co-occurrence regularized non-negative matrix tri-factorization for text data co-clustering, in: Proceedings of the Thirty-Second International Conference on Artificial Intelligence (AAAI'18), 2018.

[13] G. Govaert, M. Nadif, Latent block model for contingency table, Communications in Statistics - Theory and Methods 39 (3) (2010) 416–425.

[14] C. Biernacki, G. Celeux, G. Govaert, Assessing a mixture model for clustering with the integrated completed likelihood, IEEE Transactions on Pattern Analysis and Machine Intelligence. 22 (7) (2000) 719–725.

[15] G. Govaert, M. Nadif, Co-Clustering, Computing Engineering series, ISTE-Wiley, 2014.

[16] C. Laclau, M. Nadif, Hard and fuzzy diagonal co-clustering for document-term partitioning, Neurocomput. 193 (C) (2016) 133–147.

[17] C. Laclau, M. Nadif, Diagonal latent block model for binary data, Statistics and Computing 27 (5) (2017) 1145–1163.

[18] M. Ailem, F. Role, M. Nadif, Sparse poisson latent block model for document clustering, IEEE Trans. Knowl. Data Eng. 29 (7) (2017) 1563–1576.

[19] G. Govaert, M. Nadif, Clustering with block mixture models, Pattern Recognition 36 (2003) 463–473.

[20] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the em algorithm, Journal of he Royal Statistical Society, series B 39 (1) (1977) 1–38.

[21] C. Keribin, V. Brault, G. Celeux, G. Govaert, Estimation and Selection for the Latent Block Model on Categorical Data, Research Report RR-8264, INRIA (Nov. 2013).

[22] A. Gelman, D. Rubin, Inference from iterative simulation using multiple sequences, Statistical Science 7 (4) (1992) 457–472.

[23] G. Schwarz, Estimating the dimension of a model, The Annals of Statistics 6 (1978) 461–464.

[24] I. S. Dhillon, S. Mallela, D. S. Modha, Information-theoretic co-clustering, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03, ACM, New York, NY, USA, 2003, pp. 89–98.

[25] P. Paatero, U. Tapper, Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values, Environmetrics 5 (1994) 111–126.

[26] R. Gaujoux, C. Seoighe, A flexible r package for nonnegative matrix factorization, BMC Bioinformatics 11 (1) (2010) 367.

[27] K. Hornik, I. Feinerer, M. Kober, C. Buchta, Spherical k-means clustering, Journal of Statistical Software 50 (10) (2012) 1–22.

[28] L. Hubert, P. Arabie, Comparing partitions, Journal of Classification 2 (1) (1985) 193–218.

[29] V. Brault, Estimation et sélection de modèle pour le modèle des blocs latents, Ph.D. thesis, Université Paris Sud-Paris XI (2014).

[30] M. Selosse, J. Jacques, C. Biernacki, Model-based co-clustering for mixed type data, Computational Statistics & Data Analysis 144 (2020) 106866.

[31] Y. Chen, L. Wang, M. Dong, J. Hua, Exemplar-based visualization of large document corpus (infovis2009-1115), IEEE Transactions on Visualization and Computer Graphics 15 (6) (2009) 1161–1168.

[32] G. Karypis, CLUTO a clustering toolkit, Tech. Rep. 02-017, Dept. of Computer Science, University of Minnesota (2002).

[33] J. K. Rowling, Harry Potter and the Philosopher's Stone, 1st Edition, Vol. 1, Bloomsbury Publishing, London, 1997.

[34] J. K. Rowling, Harry Potter and the Chamber of Secrets, 1st Edition, Vol. 1, Bloomsbury Publishing, London, 1998.

[35] J. K. Rowling, Harry Potter and the Prisoner of Azkaban, 1st Edition, Vol. 1, Bloomsbury Publishing, London, 1999.

Appendix

We detail the interpretation of the co-clustering performed by the SOCC model on the Harry Potter data set.

Interpretation of column-clusters for the main section. Seven clusters in line were detected by the SOCC model. Therefore, there are also seven column-clusters in the main section. The first contains the terms specific to the chapters of the first row-cluster, the second contains the terms specific to the chapters of the second row-cluster, and so on. We highlight below that this specific co-clustering structure is easily readable for to users.

- Cluster 1: Some terms specific to the chapters of this row-cluster are 'agony', 'hewhomustnotbenamed', 'pain', 'quirrell' and 'serpent'. These terms refer to Harry Potter's enemy, called Lord Voldemort. People are so afraid of him that they never say his name aloud and refer to him as 'he-who-must-not-be-named'. He loves serpents and torturing his opponents. Quirrell is his servant in Volume 1. We propose for this cluster the label 'Voldemort' for this cluster.

- Cluster 2: Some terms specific to the chapters of this row-cluster are 'animagus', 'black', 'dementors', 'godfather','james','lupin', 'murderer', 'peter', 'pettigrew', 'remus', 'scabbers','sirius', 'transform' and 'werewolf'. These terms relate to friendships of Harry's father. James Potter, Sirius Black, Remus Lupin and Peter Pettigrew were friends in Hogwart. Remus was a werewolf so his friends learnt how to transform into animals to be able to handle his strength when he turned into a a werewolf. Wizards with this capacity are called animagus. Finally, Pettigrew betrayed their friends and delivered James to Voldemort. Proposed label: Animagus.

- Cluster 3: Specific related terms here are 'alicia', 'angelina', 'beater', 'broom', 'captain', 'championship', 'chaser', 'cheers', 'commentary', 'game', 'goalposts', 'johnson', 'jordan', 'katie', 'lee', 'locker', 'match', 'quaffle',

'refereeing', 'scores', 'spinnet', 'teams' and 'win'. These terms relate to Quidditch, a sport where wizard must score points while flying on magic brooms. Alicia Spinnet, Angelina Johnson and Katie Bell are players on Harry's team. Lee Jordan is the match commentator of the school. Proposed label: Quidditch.

- Cluster 4: Here, specific related terms are 'birthday', 'cousin', 'drive', 'dudley', 'dursley', 'figg', 'moustache', 'petunia', 'privet', 'relative', 'television', 'uncle', 'vernon'. These terms refer to Harry's family. When his parents died, his aunt and uncle (Petunia and Vernon Dursley) adopted him. They have a child named Dudley, and the family lives in the Privet Drive street. Proposed label: the Dursleys.

- Cluster 5: Some terms specific to the chapters of this row-cluster are 'arthur', 'booklist', 'bookshop', 'burrow', 'molly', 'mum', 'supplies', 'shop' and 'weasley'. These terms relate to the Weasleys. They are members of the family of Ron Weasley, Harry's best friend. They live in a house called the Burrow. Arthur and Molly Weasley are Ron's parents. Every summer, Harry spends a part of summer with them, and they go to shop for the supplies for the following year. Proposed label: the Weasleys.

- Cluster 6: Some terms specific to the chapters of this row-cluster are 'bulstrode', 'crabbes', 'dueling', 'finchfletchey', 'goyles', 'greenhouse', 'justin', 'longbottoms', 'mandrakes', 'millicent' and 'sprout'. These terms are related to Harry's courses, and in particular his classmates. Crabbes, Goyles, Justin Finch-Fletchey, Milicent Bulstrode and Longbottom are all Harry's classmates. Ms. Sprout is the botany teacher, and the mandrakes are a special kind of magical plants. Proposed label: classmates.

- Cluster 7: Some terms specific to the chapters of this row-cluster are 'aragog', 'bane', 'centaurs', 'dragon', 'firenze', 'fluffy', 'forest', 'giant', 'goblins', 'hagrid', 'norbert', 'spider' and 'unicorn'. These terms refer to magical creatures that live in Harry' world. His friend Hagrid (a half giant

wizard) has a passion about them. He owns a three-headed dog called Fluffy. In his childhood, he also raised Aragog, a giant spider. Firenze and Bane are centaurs living in the forest near Harry's school. Proposed label: magic creatures.

Therefore, the *main* section highlights seven main clusters of chapters that are related to: Voldemort, animagus, Quidditch, the Dursleys, the Weasleys, classmates and magical creatures.

Interpretation of column-clusters for the second section. With regard to the *second* section, as mentioned before, its corresponding column-clusters have terms that are related to two row-clusters. Since we now know what each row-cluster is about individually, from the *main* section, we can see the terms that link them. The SOCC model looks for common words for every row-cluster pair. This can be a limitation: for example, the chapters related to the Dursleys and the chapters related to Quidditch do not have a lot in common and the column-cluster related to these two groups of chapters contains only the word 'card', which is unrelated to both. However, most of the column-clusters that relates to two clusters of chapters are of interest to users. Here are some examples:

- Row clusters 1 and 4, which are about Voldemort and the Dursleys, share meaningful blocks in column-cluster 10. The corresponding terms include 'mother', 'nephew', 'petunias' and 'scar'. Petunia Dursley is Harry's aunt. She is connected to Voldemort because he killed her sister. He also attempted to kill Harry as a young boy, but he survived, and he was left with a scar on his forehead. Petunia then adopted her nephew.

- Row-clusters 1 and 5, which are about Voldemort and the Weasleys share meaningful blocks in column-cluster 11. This column-cluster has terms such as 'basilisks', 'tom', 'riddle' and 'ginny'. This makes sense because Ginny is Mr. and Ms. Weasley's daughter. She is closely connected to Voldemort in Volume 2. The wizard finds a way to bring Tom Riddle to

life. Tom is the past version of himself, when he was a normal teenager in the school. Tom casts a spell on Ginny so that she wakes the giant basilisk serpent up in the Chamber of Secrets. Then, this snake attacks the school's students.

- Row-clusters 1 and 6, which are about Voldemort and Harry's classmates share meaningful blocks in column-cluster 12. The correspond terms include 'ernie', 'petrified' and 'serpents'. In Volume 2, Ernie is Harry's classmate. In duelling class, Harry speaks to a serpent, an ability both he and Voldemort hold. Ernie thinks that he is ordering the snake to attack Justin Finch-Fletchey. His suspicions grow when Justin is found petrified in the corridor. He spreads the rumour that Harry's destiny was to become a powerful dark wizard and that is why Voldemort wanted to kill him.

- Row-clusters 3 and 5, which are about quidditch and the Weasleys share meaningful blocks in column-cluster 20. It contains only three words, for which the two most frequent are 'fred' and 'george'. Fred and George are twins and they are also members of the Weasley family. Both of them are 'beaters' on Harry's Quidditch team.

- Row-clusters 3 and 6, which are about Quidditch and the Harry's classmates share meaningful blocks in column-cluster 21. The column-cluster contains the terms 'crabbe', 'goyle', 'malefoy' and 'slytherins'. Crabbe, Goyle and Malefoy belong to the Slytherin house at the school. They are Harry's classmates and hate him. In Volume 3, Harry and his classmates discover that he faints in the presence of dementors (a creature that can absorb your soul). Later on in the year, Harry fainted while playing in a Quidditch match, when Crabbe, Goyle and Malefoy arrived on the field disguised as dementors.

- Row-clusters 4 and 5, which are about the Dursleys and the Weasleys share meaningful blocks in column-cluster 23. The corresponding terms

include ‘auntie’, ‘bedroom’, ‘brothers’, ‘errol’, ‘ink’, ‘letters’, ‘september’, ‘sons’, ‘summer’ and ‘written’. The vocabulary related to a family context connects the two row-clusters because both of them relate to families. The terms ‘summer’ and ‘september’ relate to the fact that Harry spends part of his summer vacations at his aunt’s place and the other part at the Weasley’s. The terms ‘errol’, ‘ink’ and ‘letters’ refers to Errol, Ron Weasley’s owl, which he uses to write to Harry when he is at his aunt’s.