Cahiers du CLSL, n° 69, 2025, pp. 73-96

A framework for spatial clustering of textual objects: applications in topic clustering and text segmentation

Guillaume Guex
University of Lausanne
guillaume.guex@unil.ch

Abstract

We present a general, classical, framework of spatial clustering which can be applied to various textual objects (e.g. character n-grams, words, sentences). This framework proposes to cluster objects according to users defined linguistic similarity, while keeping a spatial coherence of objects among clusters. Two methods are derived from this formalism: *SpatialWord*, which applies to word-tokens, and *SpatialSent*, operating on sentences, which both balance between semantic similarities of objects and their position along the textual sequence. We show that these unsupervised methods, along with semi-supervised variants, can perform jointly two operations often achieved individually by methods in literature: (1) the extraction of a desirable number of topics from a document along with list of words to interpret them; and (2) the textual segmentation of the document reflecting these extracted topics. Case studies show that these methods perform competitively against state-of-the-art methods on baseline datasets.

1 Introduction

Automatically revealing topics in a document has been of great value for domains such as information retrieval, question answering or digital humanities, as it can effectively extract information from a document without actually reading it (distant reading). Historically, topic modeling approaches, such as *Latent Dirichlet Allocation (LDA)* (Blei et al., 2003) or *Non-negative Matrix Factorization (NMF)* (Arora et al., 2012), considered documents as bags-of-words, and supposed that similar

topics are found in documents having comparable vocabulary usages. These approaches give affinity weights (e.g. probabilities) to words depending on the topic, which allow its identification, but without taking into account one of the main arrangements made by authors in their text: topics are generally found in coherent, contiguous passages. Retrieving passages addressing a particular extracted topic is generally tedious with such approaches, as contiguous words can belong to very different topics. By contrast, text segmentation methods (Choi, 2000; Eisenstein & Barzilay, 2008; Glavaš et al., 2016; Koshorek et al., 2018; Riedl & Biemann, 2012), which are also used to automatically extract information from documents, mainly use the structure of the text, i.e. the relative position of textual elements (tokens, sentences, paragraphs) in the sequence. They generally use some detection of semantic shift to place breakpoints in documents, resulting in a segmentation that can reflect the topical structure intended by the author. However, unlike topic modeling approaches, these methods generally do not label the resulting segments, and are unable to quickly summarize the main topics found in a document with lists of most used words. While methods combining approaches, i.e. finding a textual segmentation and assigning labels to segments, are obviously valuable, they are rare to find. Some works use segmentation with a combined text classification (Agarwal & Yu, 2009; Arnold et al., 2019; Chen et al., 2009; Tepper et al., 2012), using shared knowledge over the whole corpus, but none, to our knowledge, use an unsupervised approach on a single file.

We propose here to use a very general and flexible framework based on *spatial autocorrelation*, originally used in Bavaud et al. (2015) and Ceré & Bavaud (2018) and previously inspired from Anselin (2010) and Cressie (1993), which allows *spatial clustering* methods of various textual objects, along with *semi-supervised classification* variants. Methods derived from this formalism are able to extract topics on a single document, and can be tuned to force a spatial coherence of these topics in the text, hence finding segments of text covering each topic. They can be applied to different textual objects: character n-grams, word-tokens, sentences, paragraphs; as long as two quantities are defined on them: (1) a *similarity* (or dissimilarity) between elements,

which can reflect semiotic, phonological, or semantic affinities between textual objects; (2) a *proximity structure*, defining how elements relate to each others in the textual sequence. The proposed framework applied on n objects with m groups given will result in a $(n \times m)$ fuzzy membership matrix, noted $\mathbf{Z} = (z_{ig})$, verifying $z_{ig} \geq 0$ and $\sum_g z_{ig} = 1$, where z_{ig} reflects the membership percentage of object i in group g. Along with the number of desired groups m, methods have two main hyperparameters, α and β, whose tuning can balance, on one hand, between the importance of the similarity of items vs their proximity, and, on the other hand, between the fuzziness vs the crispiness of group memberships. In case studies, we illustrate two methods derived from this formalism: a *semantic clustering of word-tokens* and a *semantic clustering of sentences* in a given document, along with their *semi-supervised classification variants*. We show that these methods can be used on *topic clustering* and *text segmentation* tasks, extracting interpretable topics along with text segments that cover them. For these tasks, our methods are compared with cutting-edge methods on gold standard datasets, showing that, while not state-of-the-art, they can compete against the best methods. Section 2 explains the framework used by our methods, section 3 explores case studies, and section 4 draws general conclusions. All datasets, Python scripts and results can be found in the Github repository of the article.[1]

2 Formalism

2.1 General framework

2.1.1 Dissimilarity and exchange matrices

While this article focuses on semantically clustering *word-tokens* or *sentences*, the framework used here can be defined in very general terms, as found in, e.g., Bavaud et al. (2015) and Ceré & Bavaud (2018). Consider n *objects*, indexed by $i \in \{1, \ldots, n\}$, with their *vector of relative weights* $\mathbf{f} = (f_i)$, where $f_i > 0$ and $\sum_i f_i = 1$, and the following matrices:

– An $n \times n$ symmetric *squared Euclidean dissimilarity matrix*

1 https://github.com/gguex/SemSim_AutoCor.

$\mathbf{D} = (d_{ij})$, verifying $d_{ij} \geq 0$, containing pairwise dissimilarities between these objects.

– An $n \times n$ symmetric matrix of joint probabilities $\mathbf{E} = (e_{ij})$, called *exchange matrix*, verifying $e_{ij} \geq 0$ and $e_{i\bullet} = e_{\bullet i} = f_i$ ("$\bullet$" refers to a sum over the replaced index), containing spatial relationships between objects. Sometimes, we use the associated *Markov chain transition matrix* $\mathbf{W} = (w_{ij})$, defined with $w_{ij} := e_{ij}/f_i$. The margins of this matrix must contain object weights in order for the functionals to be formally defined.

2.1.2 Membership matrix and functionals

A *fuzzy clustering* of these n objects into m groups can be defined by a $n \times m$ *membership matrix* $\mathbf{Z} = (z_{ig})$, with $z_{ig} \geq 0$ and $z_{i\bullet} = 1$, whose components represent the membership of object i to group g. The membership matrix defines the *relative group weights vector* $\boldsymbol{\rho} = (\rho_g)$, with $\rho_g := \sum_i f_i z_{ig}$ and m vectors of *within-group distribution* $\mathbf{f}^g = (f_i^g)$ with $f_i^g := f_i z_{ig}/\rho_g$. Different functionals can be computed from a membership matrix $\mathbf{Z}$.

The *within-group inertia* is defined as

$$\Delta_W[\mathbf{Z}] := \sum_g \rho_g \Delta_g \quad \text{where} \quad \Delta_g = \frac{1}{2} \sum_{ij} f_i^g f_j^g d_{ij} \ . \tag{1}$$

A low within-group inertia reflects homogeneity between objects of the same group, as defined by the dissimilarity matrix $\mathbf{D} = (d_{ij})$.

The *generalized cut* reads

$$\mathcal{C}^\kappa[\mathbf{Z}] := \sum_g \frac{\rho_g^2 - e(g,g)}{\rho_g^\kappa} \quad \text{where} \quad e(g,g) := \sum_{ij} e_{ij} z_{ig} z_{jg} \ . \tag{2}$$

A low generalized cut functional indicates strong neighborhood relationships between tokens of the same group, as defined by the exchange matrix $\mathbf{E} = (e_{ij})$. The hyperparameter $\kappa \in [0, 1]$ allows

us to interpolate objects the *N-cut objective* (Shi & Malik, 2000) when $\kappa = 1$ and the *modularity criterium* (Newman, 2006) when $\kappa = 0$.

The token-group dependency can be expressed by the *mutual information*

$$\mathcal{K}[\mathbf{Z}] := \sum_{ig} \rho_g f_i^g \log\left(\frac{f_i^g}{f_i}\right) = \sum_{ig} f_i z_{ig} \log\left(\frac{z_{ig}}{\rho_g}\right) \tag{3}$$

which is low if distributions f_i^g correspond to f^i, i.e. f_i^g are independent of group g. Therefore, low mutual information indicates fuzziness in group memberships.

Finally, by combining all previous functionals, we can define the *free energy* with

$$\mathcal{F}[\mathbf{Z}] := \beta\Delta_W[\mathbf{Z}] + \frac{\alpha}{2}\mathcal{C}^\kappa[\mathbf{Z}] + \mathcal{K}[\mathbf{Z}] \ . \tag{4}$$

Searching the membership matrix $\mathbf{Z}$ minimizing this functional results in a fuzzy clustering of objects depending on hyperparameters α, β and κ. An interpretation of hyperparameter effects in the case of textual data can be found in section 2.2.4.

2.1.3 Finding a local minima

Canceling the derivative of the free energy with respect to z_{ig} under the constraints $z_{i\bullet}$ yields the minimization condition

$$z_{ig} = \frac{\rho_g \exp(-h_{ig})}{\sum_h \rho_h \exp(-h_{ih})} \tag{5}$$

where

$$\rho_g[\mathbf{Z}] := \sum_i f_i z_{ig} \tag{6}$$

$$h_{ig}[\mathbf{Z}] := \beta d_i^g + \alpha\rho_g^{-\kappa}(\rho_g - \sum_j w_{ij} z_{jg}) - \frac{\alpha\kappa}{2}\rho_g^{-\kappa-1}\left(\rho_g^2 - e(g,g)\right) \tag{7}$$

with $d_i^g = \sum_j f_j^g d_{ij} - \Delta_g$ the squared Euclidean dissimilarity from i to the centroid of group g. These Equations define an iterative procedure converging to a local minimum for $\mathcal{F}[\mathbf{Z}]$: a random membership matrix is taken as $\mathbf{Z}^0$, ρ_g^t and h_{ig}^t are computed with (6) and (7) respectively, and $\mathbf{Z}^{t+1}$ is given by (5).[2] Pseudo-code for the algorithm can be found in appendix A.1.

2.1.4 Semi-supervised framework

Working with the membership matrix $\mathbf{Z}$ for objects allows us to easily adapt the algorithm in a semi-supervised framework. Let $\mathcal{T}$ be the group of *tagged objects*, which consists in m disjoint subgroups $\mathcal{T} = \cup_{g=1}^m \mathcal{T}_g$, where $\mathcal{T}_g$ contains objects which should be in group g. The initial membership values z_{ig}^0 for tagged object $i \in \mathcal{T}$ are then set to :

$$z_{ig}^0 = \begin{cases} 1 & \text{if } i \in \mathcal{T}_g \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Moreover, at the end of each iteration, the tagged objects are reset to their initial values z_{ig}^0, forcing them to be in their respective group. If the generalized cut functional is high enough (i.e. α in (4) is large), these tagged objects should act as anchor points, "spreading" their labels to their neighbors.

2.1.5 Complexity and scaling

The algorithm complexity is $\mathcal{O}(n^2 m)$, which can be problematic in case of large datasets. However, this issue can be alleviated by decomposing the dataset into small overlapping blocks, and running the algorithm on each of them independently, while transferring the labels from one block to another by fixing membership of objects at their intersection. Formally, our dataset $\mathcal{D}$ can be decomposed into p blocks $\mathcal{B}_k$ of size $n_b < n$, with $\mathcal{D} = \cup_{k=1}^p \mathcal{B}_k$ and $\mathcal{B}_k \cap \mathcal{B}_{k-1} \neq \emptyset,\ \forall k \in \{2, \ldots, p\}$. We first find the memberships $z_{ig}^{\mathcal{B}_1}$ of the objects in the group $\mathcal{B}_1$, and

2 During applications, $\mathbf{Z}^0$ is picked with $\mathbf{f}^g$ close the uniform distribution for every g, and the new membership matrix is computed with $\lambda \mathbf{Z}^{t+1} + (1-\lambda)\mathbf{Z}^t$ with $\lambda \in (0, 1]$ a decreasing adaptative learning parameter which allows the algorithm to reach the bottom of "valleys" of high gradients.

proceed sequentially by fixing $z_{ig}^{\mathcal{B}_k} = z_{ig}^{\mathcal{B}_{k-1}},\ \forall i \in \mathcal{B}_k \cap \mathcal{B}_{k-1}$. The algorithm is then $\mathcal{O}(n_b^2 pm)$, but some performance is lost in the process. We examine the speed/performance trade-off of this procedure in Section 3.4.3.

2.2 Textual data

The framework of section 2.1 can be adapted to textual data, where objects can be, e.g., character n-grams, word-tokens, sentences, or paragraphs, as long as we can define: (1) dissimilarities reflecting linguistic differences between items; and (2) a spatial proximity structure, indicating how objects interact with each other in the textual sequence. The relative weights f_i for textual objects can be defined as uniform or proportional to their length (e.g. the number of words in a sentence).

2.2.1 Semantic (dis)similarities

When working with textual objects, the matrix $\mathbf{D}$ should represent *linguistic dissimilarities* between objects, which can be defined notably at the semiotic, phonological, or semantic level, depending on objects and applications. We can generally construct these dissimilarities from external sources, giving either dissimilarities or similarities. Dissimilarities d_{ij} can be obtained from similarities s_{ij} with, e.g., $d_{ij} = \max_{kl} s_{kl} - s_{ij}$. In this article, we exclusively work with word-tokens and sentences, and with dissimilarities defined at the semantic level. For tokens, semantic similarities can be constructed with *WordNet* (Fellbaum, 1998), *type-based Word Embeddings* (Bojanowski et al., 2017; Mikolov et al., 2013a; Pennington et al., 2014), or *Transformers* (Devlin et al., 2019). Note that when using type-based Word Embeddings, these dissimilarities are defined between *types*, and every pair of tokens having the same signature of types will yield the same dissimilarity. If we consider sentences, there are also many models permitting to build semantic similarities, such as *Sentence Embeddings* (Mikolov et al., 2013b; Reimers & Gurevych, 2019).

2.2.2 Spatial structure

Note that the structure of textual objects is a sequence that can be indexed by $i \in \{1, \ldots, n\}$. The spatial structure, encoded by the ex-

change matrix $\mathbf{E}$, should reflect how these objects relate to each other in the textual sequence. It could be crafted carefully, by using, e.g. a syntactic dependency parser in order to weight token relationships accordingly. However, we chose to construct it uniquely on the neighborhood relationships as it already shows satisfying results. Note that because of the constraint $e_{i\bullet} = e_{\bullet i} = f_i$ and because of boundary effects, the computation of these matrices is not trivial, even for uniform weights. Our suggestion is to compute textual exchange matrices as proposed in Ceré & Bavaud (2018), also seen in A.2.

1. The *uniform* exchange matrix $\mathbf{E}^{\mathrm{U}(r)}$ with range $r \in \mathbb{N}$, where $e_{ij}^{\mathrm{U}(r)}$ is constant for $j \in [i-r, i+r]$ and 0 otherwise. This matrix can be computed with the Metropolis-Hastings algorithm.

2. The *diffusive* exchange matrix $\mathbf{E}^{\mathrm{D}(t)}$, closely related to the diffusion map kernel (Nadler et al., 2005), with diffusive time factor $t > 0$. For a large enough t, the strength of links between i and its neighbors j will be normally shaped, thus exponentially decreasing with $|i-j|$.

The choice between $\mathbf{E}^{\mathrm{U}(r)}$ and $\mathbf{E}^{\mathrm{D}(t)}$, as well as their ranges r or t, can be considered as hyperparameters to be tuned. Computations for these matrices are given in Appendix A.2.

2.2.3 Topic interpretation

When semantic dissimilarities are used, the resulting clusters can be interpreted as *topics*. When using word-tokens as objects, the membership matrix $\mathbf{Z}$ has the following interpretation :

$$z_{ig} = \mathrm{P}(T_i = g) \tag{9}$$

where T_i is the variable containing the topic of token i. Note that with our algorithm, a token can be a part of multiple topics (fuzziness) and also depends on its neighborhood (spatial component), reflecting the fact that generally, a particular topic covers several contiguous textual objects.

When using a topic modeling approach, in order to interpret topics, we

are interested in the probability of each word-type constituting topics or, conversely, the probability of being in a certain topic when using a specific word-type. In other words, we need to compute:

$$\mathrm{P}(T = g|W = w) = \text{“Probability to be in topic } g \text{ when the type is } w\text{”} \tag{10}$$

$$\mathrm{P}(W = w|T = g) = \text{“Probability to draw type } w \text{ when the topic is } g\text{”} \tag{11}$$

where T is the variable containing the topic and W the variable containing the word-type for a randomly drawn token in the text. It is possible to express these quantities with the components of the matrix $\mathbf{Z}$ and using the variable W_i, containing the word-type at position i:

$$\mathrm{P}(T = g|W = w) = \frac{1}{n_w} \sum_{i|W_i=w} \mathrm{P}(T_i = g) = \frac{1}{n_w} \sum_{i|W_i=w} z_{ig} \tag{12}$$

$$\begin{aligned} \mathrm{P}(W = w|T = g) &= \frac{\mathrm{P}(T = g|W = w) \cdot \mathrm{P}(W = w)}{\mathrm{P}(T = g)} \\ &= \frac{\frac{1}{n_w} \sum_{i|W_i=w} \mathrm{P}(T_i = g) \cdot \frac{n_w}{n}}{\frac{1}{n} \sum_i \mathrm{P}(T_i = g)} = \frac{\sum_{i|W_i=w} z_{ig}}{z_{\bullet g}} \end{aligned} \tag{13}$$

where n_w designates the number of occurrences of word-type w. In case studies, section 3.4.1, we see that Equation (12) helps to reflect the vocabulary specificity of each topic, whereas Equation (13) gives clues about the writing style of each topic, which can remain similar among topics in some cases.

2.2.4 Hyperparameters effects

The result of the clustering depends on hyperparameters α, β and κ in (4). Increasing α relatively to β favors groups containing long sequences of textual objects, while increasing β for a fixed α strengthens linguistic homogeneity inside clusters. The limit $\beta \to \infty$ corresponds to the case of a *K-means* clustering using only the linguistic dissimilarities defined on textual objects. Decreasing both α and β increases the

fuzziness of groups, resulting in a mixture of groups defined over textual objects. Figure 1 gives insights on how the clustering behaves depending on α and β. The hyperparameter κ controls the spatial objective by interpolating between the N-cut objective and the modularity criterion, and its effect is more difficult to interpret. These three hyperparameters are tuned with grid search in case studies.

FIGURE 1 – Semantic clustering of word-tokens into 3 groups on a part of a MANIFESTO file (see section 3). Left: β is high relatively to α, resulting in small sequences. Middle: α is low relatively to β resulting in large sequences. Right: α and β are low, resulting in high group fuzziness, represented by the mixture of colors.

3 Case studies

3.1 Tasks

In case studies, in order to compare methods derived from our formalism with existing ones, we are interested in how they perform in two particular tasks, namely *topic clustering* and *text segmentation*.

Topic clustering, to our knowledge, is not a term used in literature, although existing methods, such as *LDA* (Blei et al., 2003) or *NMF* (Arora et al., 2012), can be adapted to perform it. In this article, we define it as the task of assigning groups to textual objects, with unsupervised methods, so that objects in the same group express a similar topic. Unlike topic modeling approaches, an affinity matrix is computed between every word-token and topic, taking into consideration the wor-token's position in text, and not only between word-types and topics. Validation of this task must use unsupervised measures of adequation, such as the *Normalized Mutual Information (NMI)* (Strehl & Ghosh, 2002) used

here. Note that measures of self-coherence, such as the *Perplexity* often used in topic modeling, do not apply here as our model is not generative. Section 3.4.1 explores the topic clustering capabilities of our methods. By contrast, the text segmentation task is well known in literature (Arnold et al., 2019; Chen et al., 2009; Choi, 2000; Eisenstein & Barzilay, 2008; Glavaš et al., 2016; Koshorek et al., 2018; Riedl & Biemann, 2012). It generally consists in finding breakpoints in a text, such that the resulting segments address different topics. Supervised and unsupervised methods for this task exist. Generally, apart from a few exceptions (e.g. Arnold et al., 2019; Chen et al., 2009), these methods do not label segments with a topic. Hence, the result does not indicate if the document consists of two alternating topics, a number of topics equal to the number of segments, or a situation in-between. Validation is generally made through the use of the P_k index (Beeferman et al., 1999) or the *Window diff (WD)* index (Pevzner & Hearst, 2002), measuring if separation marks between segments correspond to ground truth. Section 3.4.2 will compare the text segmentation performances of our algorithm against cutting edge methods.

To our knowledge, no other algorithm performs both tasks at the same time in an unsupervised way, especially *without using contrastive information between documents*. While sharing information over several documents can be an asset in some cases, and usually give better performances, using our method enables the user to find text segments, with their assigned topics (with word-types defining them, see section 2.2.3), in a single document without any training (though the choice of hyperparameters can still be subject to tuning).

3.2 Corpora

While it is possible to create artificial datasets to validate both topic clustering and text segmentation tasks (Choi, 2000), we favor here real datasets, as methods seem to give over-confident results on artificial ones (Glavaš et al., 2016). We will use five datasets, which can all be found in our Github repository.

The MANIFESTO dataset (Volkens et al., 2017)[3] consists in textual

3 https://manifesto-project.wzb.eu/, accessed Jan. 2025.

parties policy positions from different countries. Some of them are manually annotated with topics along the text, which is a premium resource to test methods in a topic clustering or classification task. Topical annotations are divided between 7 super-topics and several sub-topics. We chose here to use only super-topic classes, as the number of groups seems reasonable for our algorithm. For coherence in language and culture, we extracted the annotated documents from US parties, which corresponds to 9 documents: the manifestos from the Democratic Party from 1992, 2004, 2012, 2016, and 2020, and the Republican Party from 2004, 2008, 2012, and 2016.

The WIKI50 dataset, introduced in (Koshorek et al., 2018), consists in a set of 50 randomly sampled test documents from the largest WIKI-727K. It consists in 50 English Wikipedia articles and their segmentation corresponding to their table of content.

The CITIES and ELEMENTS datasets, introduced in Chen et al. (2009), are two datasets which are also extracted from English Wikipedia. The first consists in 100 articles about large cities, and the second in 119 articles about chemical elements in the periodic table.

Finally, the CLINICAL dataset, introduced in Eisenstein & Barzilay (2008), consists in 226 chapters extracted from the *Clinical Textbook*, which are mainly used in the topic segmentation task because the different sections are not labeled.

All these datasets are well designed for text segmentation, but, with the exception of the MANIFESTO dataset, they are less suitable for topic clustering, especially if information is not shared across documents. As a matter of fact, each document extracted from Wikipedia (WIKI50, CITIES, ELEMENTS), when looked at individually, has a unique label for each segment. This means that, when a method of clustering is applied on a unique file, these documents operate like the CLINICAL dataset with unlabeled segments: the number of groups found in a document is always equal to the number of segments. This situation is not ideal, as a topic should be a recurring subject, appearing at multiple places in a document, as found in the MANIFESTO dataset. Nevertheless, because of the lack of other real datasets and because these corpora are largely used to evaluate methods in literature, we will use them here

for comparison purposes.
All datasets are preprocessed the same way: case is lowered, stop-words, numbers and punctuation marks are removed, while information about where each sentence ends is kept in order to apply the method on sentences.

3.3 Methodology

For both tasks, we used two versions of our algorithm, which work on two different textual objects: we named *SpatialWord* our method when applied on word-tokens, and *SpatialSent* when used on sentences. Relative weight $\mathbf{f}$ is defined as uniform for word-tokens, and proportional to the number of words for sentences. For each document, the real number of groups is given to our method. Both methods are tried with a semi-supervised version with 5% and 10% random labeling rate.
For word-tokens, we wanted to use semantic similarities which do not use their local context, as we wanted to rely solely on the exchange matrix to express spatial dependencies. This excludes word-token embeddings, e.g. based on BERT (Devlin et al., 2019), because of their use of the context of a token to build its vector. We selected 3 kinds of semantic similarities, computed as the cosine between word-type vectors extracted from pre-trained embeddings: *w2v*, which is a 300d Word2Vec Skip-Gram model trained on the English Wikipedia 2018, as found in Wikipedia2Vec (Yamada et al., 2020)[4]; *glv*, which is 300d GloVe model trained on Common Crawl (Pennington et al., 2014)[5]; and *ftx*, which is a 300d FastText model trained on Wikipedia 2017 (Bojanowski et al., 2017).[6] The choice of the similarity between those three will be a hyperparameter to tune. For *SpatialSent*, semantic similarities are computed with the cosine between vectors obtained from a pre-trained sentence embedding model, named *all-mpnet-base-v2* (Reimers & Gurevych, 2019), which is based on BERT (Devlin et al., 2019). All similarities are transformed into dissimilarities with $d_{ij} = \max_{kl} s_{kl} - s_{ij}$. For both methods, we

4 https://wikipedia2vec.github.io/wikipedia2vec/, accessed Jan. 2025.
5 https://nlp.stanford.edu/data/glove.42B.300d.zip, accessed Jan. 2025.
6 https://fasttext.cc/docs/en/english-vectors.html, accessed Jan. 2025.

used the uniform exchange matrix, as it seems to consistently give better results on these tasks.

For each dataset, each task, and each method, the tuning of hyperparameters is done with a grid search on one file of the dataset. This file is selected to be the closest to the typical values found in the dataset in terms of number of tokens, number of sentences, and average topical sequence length. Hyperparameters consist in the choice of $r \in \{5, 10, 15\}$, $\alpha \in \{1, 2, 5, 10, 30\}$, $\beta \in \{5, 10, 50, 100, 200\}$, and $\kappa \in \{0, 0.25, 0.5, 0.75, 1\}$. Moreover, for the *SpatialWord* method, the choice between {*w2v*, *glv*, *ftx*} for the semantic dissimilarity is also tuned.

For the topic clustering task, our methods are compared to Latent Dirichlet Allocation (*LDA*) (Blei et al., 2003) and Non-negative Matrix Factorization (*NMF*) (Arora et al., 2012). However, these methods do not give good results if used as intended, i.e. when using the whole corpus to extract topics. A more efficient way to use them in this task is to split one document into small chunks, and to consider these chunks as different parts containing a mixture of topics. The length of these chunks, which can be selected between fractions $\{\frac{1}{20}, \frac{2}{20}, \frac{3}{20}, \frac{4}{20}, \frac{5}{20}, \frac{6}{20}, \frac{7}{20}, \frac{8}{20}, \frac{9}{20}, \frac{10}{20}\}$ of document length, is tuned on the same documents as our methods. Using this scheme for these methods allows us to assign different topics to different occurrences of the same word-type, as the probability for a token w to be in group g is $\mathrm{P}(T = g|W = w) \sim \mathrm{P}(W = w|T = g)\mathrm{P}(T = g)$, with a different $\mathrm{P}(T = g)$ for each chunk.

For the text segmentation task, resulting P_k scores for our methods are compared to a random baseline, and scores are reported from *BayesSeg* (Eisenstein & Barzilay, 2008), *GraphSeg* (Glavaš et al., 2016), *TextSeg* (Koshorek et al., 2018), and *Sector* (Arnold et al., 2019) methods. Note that the last two methods, *TextSeg* and *Sector*, are supervised methods and are expected to show better results. Except for *Sector*, a text segmentation and topic classification method, none of these methods give class labels for resulting segments.

3.4 Results

3.4.1 Topic clustering results

Results for the topic clustering task are shown into table 1. The *SpatialSent* consistently performs better than other methods, while *SpatialWord* method is better than *LDA* but less good than *NMF*. Surprisingly, when using some labels, the *SpatialWord* now outperforms *SpatialSent* and gives very strong results. We observe that all method give a low NMI on the MANIFESTO dataset, which is the only annotated dataset permitting a serious validation of the topic clustering task. However, when looking at words defining clusters on a file, even with the *SpatialWord* method as shown in table 2, we can clearly identify pertinent topics using $\mathrm{P}(T = g|W = w)$ or $\mathrm{P}(W = w|T = g)$ (the latter gives the general tone of each topic, which is quite redundant in this case).

	MANIFESTO	WIKI50	CITIES	ELEMENTS	CLINICAL
LDA	12.4	38.0	56.7	48.8	35.0
NMF	19.9	54.7	68.4	61.6	47.6
SpatialWord	14.8	50.2	56.9	45.0	29.0
SpatialSent	**26.9**	**66.6**	**81.5**	**75.9**	**58.1**
SpatialWord, 5%	45.9	72.0	87.2	69.4	67.4
SpatialSent, 5%	30.0	61.9	77.9	76.7	60.4
SpatialWord, 10%	51.7	76.2	91.6	77.0	74.5
SpatialSent, 10%	30.8	66.2	80.2	75.2	68.3

TABLE 1 – Mean NMI results for method × dataset. Higher is better, best results (without considering semi-supervised version of methods) are in boldface.

3.4.2 Text segmentation results

Text segmentation results are found in table 3. Globally, the *TextSeg* seems to perform better than other methods, with the exception of *GraphSeg* for the MANIFESTO dataset (in fact, *TextSeg* was not tested on this dataset) and of *SpatialSent* on the ELEMENTS dataset. If we strictly look at unsupervised methods, we can see that *SpatialSent* generally gives the best results, while *SpatialWord* is in the average. The only other method to also give group labels, *Sector*, gives generally better results than our methods, but these results must be put in perspective because, unlike our methods, it is supervised. When looking at semi-

Group 1	**Group 2**	**Group 3**	**Group 4**	**Group 5**	**Group 6**	**Group 7**
Top 5 word-types regarding P($T = g\|W = w$)						
qadhafi	*america*	*cleaner*	*love*	*withstand*	*bp*	*uninsured*
muammar	*colombia*	*generating*	*tell*	*adversary*	*tsunami*	*counseling*
syrian	*indonesia*	*electricity*	*telling*	*reinforce*	*fissile*	*funds*
prodemocracy	*malaysia*	*cheap*	*splipped*	*involve*	*gm*	*funded*
iraqi	*chile*	*optimizing*	*story*	*reveal*	*nasa*	*refinancing*
Top 5 word-types regarding P($W = w\|T = g$)						
president	*trade*	*energy*	*president*	*president*	*president*	*health*
obama	*president*	*jobs*	*middle*	*continue*	*obama*	*president*
states	*american*	*education*	*back*	*security*	*nuclear*	*care*
support	*economic*	*tax*	*america*	*obama*	*states*	*democrats*
rights	*global*	*businesses*	*work*	*nuclear*	*american*	*support*

TABLE 2 – Word-types defining clusters on the file Democratic 2012 from the MANIFESTO dataset, as found with the *SpatialWord* method (NMI = 11.2).

supervised results, the *SpatialWord* method quickly outperforms all methods, while *SpatialSent* remains about the same, as already seen in the topic clustering results.

	MANIFESTO	WIKI50	CITIES	ELEMENTS	CLINICAL
Random baseline	60.4	61	59	58.5	66.9
GraphSeg	**28.1**	63.6	40	49.1	-
BayesSeg	-	49.2	36.2	35.6	57.8
TextSeg	-	**18.2**	**19.7**	41.6	**30.8**
Sector	-	28.6	21.4	39.2	35.6
SpatialWord	39.6	50.2	49.9	45.4	33
SpatialSent	38.8	43.7	33.9	**28.4**	40.2
SpatialWord, 5%	41.6	33.8	13.2	23.4	27.2
SpatialSent, 5%	38.4	40.2	32.8	30.2	37
SpatialWord, 10%	31.8	24	5.6	10.5	25.5
SpatialSent, 10%	37.3	41	32.1	28.3	30.9

TABLE 3 – Mean P_k results for method × dataset. Lower is better, best results (without considering semi-supervised version of methods) are in boldface.

3.4.3 Method scaling

The method complexity is $\mathcal{O}(n^2 m)$ and does not scale well on large files. However, as seen in section 2.1.5, it is possible to divide a text file into p overlapping blocks of n_b tokens, and proceed sequentially on blocks while transferring predicted labels from the previous block as fixed labels on the next block. We tested this process with *SpatialWord*

on the largest file in our datasets, i.e. the Republican 2020 from the MANIFESTO dataset, which has 25'870 tokens (without stopwords). When using block sizes of n_b, we define each block to have $n_b/2$ overlapping tokens with its predecessor, giving a theoretical complexity of $\mathcal{O}(n_b^2 nm)$. Results are found in figure 2. We see that the computing time is reduced, as well as performances in clustering, as shown by NMI. However, this loss in performances becomes acceptable for largest block sizes. By contrast, P_k seems less affected by the computation on blocks and gives comparable results, even with low block sizes. This block method has not been tested on *SpatialSent*, as the number of sentences is much lower than the number of words and the computing time is reasonable for every document in our datasets.

4 Conclusion

We have presented a very general, classical, formalism which is able to fuzzily cluster textual objects by taking into account a balance between object similarity and position in text. We proposed two methods derived from this formalism: *SpatialWord*, which applies on word-tokens, and *SpatialSent*, operating on sentences. These methods showed good performances for automatically retrieving topics, associated vocabulary, as well as textual segments where these topics appear. Hence, these methods could be used as a new distant reading tool in order to extract topical information on a single document, without any previous training. When compared to state-of-the-art methods on two different tasks, topic clustering and text segmentation, the proposed methods give good results considering they perform both tasks at the same time, in an unsupervised way, and without sharing information across documents. The number of hyperparameters, while permitting these methods to be highly flexible, can however becomes problematic if these methods are applied without knowing ground truth, as no self-validation indices (such as perplexity for topic modeling or inter-group variance for k-means) have been developed for the moment. However, experiments on these datasets on both tasks have already shown some regularities for the studied corpora, and we recommend using *ftx* semantic similarities while setting $r = 15, \alpha = 10, \beta = 100, \kappa = 0.25$ for *SpatialWord*

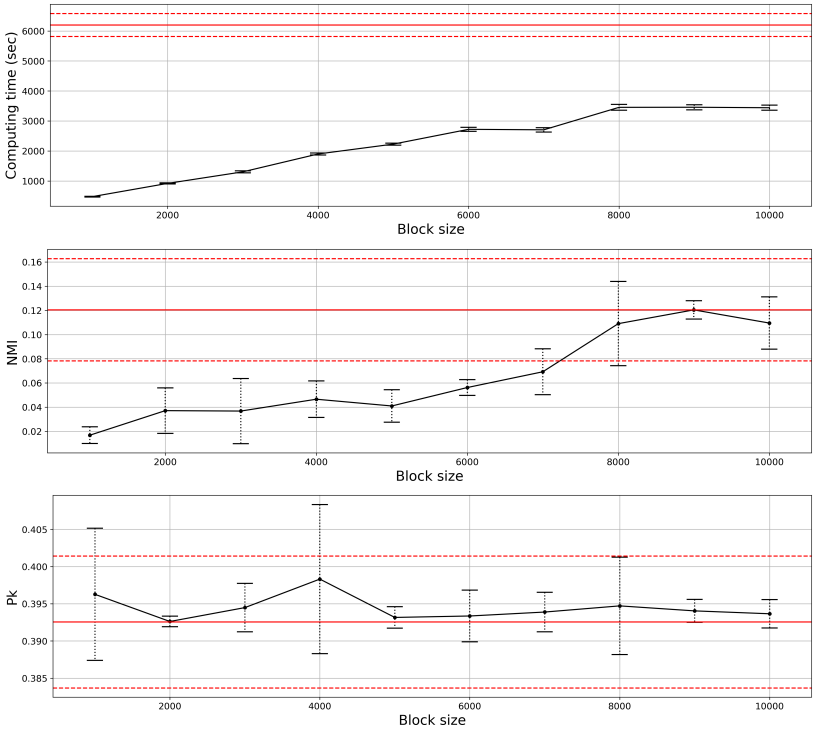

FIGURE 2 – Computing time (left), NMI (middle), and P_k (right) vs the number of tokens in blocks when run on the Republican 2020 file from the MANIFESTO dataset (with 95% CI). Horizontal lines correspond to results when the algorithm is run on the whole file. Results are computed on a single thread, 2.6GHz, i7-9750H CPU.

method, and $r = 5, \alpha = 10, \beta = 100, \kappa = 0.75$ for *SpatialSent* in order to obtain decent results on both tasks at once. Experiments also showed that *SpatialSent* performs consistently better than *SpatialWord* when used without information, which seems obvious as manually annotated segments are generally delimited by punctuation. Surprisingly, *SpatialWord* outperforms greatly *SpatialSent* when having access to some labeled words. While the situation of having direct access to token labels seems unrealistic in real world situations, this could still have applications: a user could provide short lists of typical words defining expected topics (e.g. 10 words for every topic), label corresponding

tokens, and find a pertinent segmentation for his query, as well as the rest of the associated vocabulary. The only remaining difficulty with these methods is a very large computing time. We suggested a way to alleviate this problem, if someone desires to apply them to very large files, but we showed that the performances in the topic segmentation task then decreased. Nevertheless, this should not be problematic if this method is used as an exploratory tool on relatively small corpora, which is the usual setting for a digital humanities researcher.

References

Agarwal, S. & Yu, H. (2009). Automatically classifying sentences in full-text biomedical articles into introduction, methods, results and discussion. *Bioinformatics*, 25(23):3174–3180.

Anselin, L. (2010). Local indicators of spatial association-LISA. *Geographical Analysis*, 27(2):93–115.

Arnold, S., Schneider, R., Cudré-Mauroux, P., Gers, F. A., & Löser, A. (2019). Sector: A neural model for coherent topic segmentation and classification. *Transactions of the Association for Computational Linguistics*, 7:169–184.

Arora, S., Ge, R., & Moitra, A. (2012). Learning topic models – going beyond SVD. In *Proceedings of the 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, FOCS '12, pages 1–10, USA. IEEE Computer Society.

Bavaud, F., Cocco, C., & Xanthos, A. (2015). Textual navigation and autocorrelation. In Mikros, G. K. & Macutek, J. (Eds.), *Sequences in Language and Text*, pages 35–56. De Gruyter Mouton, Berlin, München, Boston.

Beeferman, D., Berger, A., & Lafferty, J. (1999). Statistical models for text segmentation. *Machine Learning*, 34(1):177–210.

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

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Ceré, R. & Bavaud, F. (2018). Soft image segmentation: on the clustering of irregular, weighted, multivariate marked networks. In *Communications in*

Computer and Information Science, pages 85–109. Springer International Publishing.

Chen, H., Branavan, S., Barzilay, R., & Karger, D. R. (2009). Global models of document structure using latent permutations. In Ostendorf, M., Collins, M., Narayanan, S., Oard, D. W., & Vanderwende, L. (Eds.), *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 371–379, Boulder, Colorado. Association for Computational Linguistics.

Choi, F. Y. Y. (2000). Advances in domain independent linear text segmentation. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, NAACL 2000, pages 26–33, USA. Association for Computational Linguistics.

Cressie, N. A. C. (1993). *Statistics for spatial data*. John Wiley & Sons, Inc., New York, NY.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., & Solorio, T. (Eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Eisenstein, J. & Barzilay, R. (2008). Bayesian unsupervised topic segmentation. In Lapata, M. & Ng, H. T. (Eds.), *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 334–343, Honolulu, Hawaii. Association for Computational Linguistics.

Fellbaum, C. (1998). *WordNet: An electronic lexical database*. MIT Press, Cambridge, MA.

Glavaš, G., Nanni, F., & Ponzetto, S. P. (2016). Unsupervised text segmentation using semantic relatedness graphs. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 125–130. Association for Computational Linguistics.

Koshorek, O., Cohen, A., Mor, N., Rotman, M., & Berant, J. (2018). Text segmentation as a supervised learning task. In Walker, M., Ji, H., & Stent, A. (Eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 469–473, New Orleans, Louisiana. Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., & Weinberger, K. (Eds.), *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Nadler, B., Lafon, S., Kevrekidis, I., & Coifman, R. (2005). Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. In Weiss, Y., Schölkopf, B., & Platt, J. (Eds.), *Advances in Neural Information Processing Systems*, volume 18. MIT Press.

Newman, M. E. J. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582.

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global vectors for word representation. In Moschitti, A., Pang, B., & Daelemans, W. (Eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Pevzner, L. & Hearst, M. A. (2002). A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36.

Reimers, N. & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Inui, K., Jiang, J., Ng, V., & Wan, X. (Eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Riedl, M. & Biemann, C. (2012). TopicTiling: A text segmentation algorithm based on LDA. In *Proceedings of ACL 2012 Student Research Workshop*, pages 37–42, Jeju Island, Korea. Association for Computational Linguistics.

Shi, J. & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

Strehl, A. & Ghosh, J. (2002). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3(Dec):583–617.

Tepper, M., Capurro, D., Xia, F., Vanderwende, L., & Yetisgen-Yildiz, M. (2012). Statistical section segmentation in free-text clinical records. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2001–2008, Istanbul, Turkey. European Language Resources Association (ELRA).

Volkens, A., Lehmann, P., Matthieß, T., Merz, N., Regel, S., & Weßels, B. (2017). Manifesto project dataset (version 2017b). *Berlin: Wissenschaftszentrum Berlin Für Sozialforschung*.

Yamada, I., Asai, A., Sakuma, J., Shindo, H., Takeda, H., Takefuji, Y., & Matsumoto, Y. (2020). Wikipedia2Vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from Wikipedia. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 23–30. Association for Computational Linguistics.

A Appendices

A.1 Spatial clustering/semi-supervised classification algorithm

Input: dissimilarity matrix $\mathbf{D}$, exchange matrix $\mathbf{E}$, initial membership matrix $\mathbf{Z}^0$, the set of tagged objects $\mathcal{T}$ (can be empty in case of clustering), hyperparameters α, β, κ, learning parameter $\lambda \in (0, 1]$, stopping threshold ϵ.
Output: The fuzzy membership matrix of objects $\mathbf{Z}$

1 $\mathbf{f} \leftarrow \mathbf{E}\mathbf{e}_n$; // Object weights. $\mathbf{e}_n$ is the size n vector of ones.
2 $\mathbf{Z} \leftarrow \mathbf{Z}^0$; // Initialize membership matrix.
3 $\mathcal{F} \leftarrow 10^9$, $\mathcal{F}^{old} \leftarrow 10^{10}$; // Initialize free energy.
4 **while** $|\mathcal{F} - \mathcal{F}^{old}| > \epsilon$ **do**
5 $\quad \mathbf{Z}^{old} \leftarrow \mathbf{Z}$; // Save old membership matrix.
6 $\quad \mathcal{F}^{old} \leftarrow \mathcal{F}$; // Save old free energy value.
7 $\quad \boldsymbol{\rho} \leftarrow \mathbf{Z}^\top \mathbf{f}$; // Group weights.
8 $\quad \mathbf{F} \leftarrow (\mathbf{f}(\mathbf{e}_n \oslash \boldsymbol{\rho})^\top) \odot \mathbf{Z}$; // Within-group distibutions.[7]
9 $\quad \boldsymbol{\delta} \leftarrow \mathbf{diag}(\mathbf{F}^\top \mathbf{D}\mathbf{F})$; // Group inertias.[8]
10 $\quad \mathbf{c} \leftarrow (\boldsymbol{\rho}^2 - \mathbf{Z}^\top \mathbf{E}\mathbf{Z}) \oslash \boldsymbol{\rho}^\kappa$; // Generalized cut values for groups.[9]
11 $\quad \mathbf{H} \leftarrow \beta(\mathbf{D}\mathbf{F} - \frac{1}{2}\mathbf{e}_n\boldsymbol{\delta}) + \mathbf{e}_n[\alpha\boldsymbol{\rho}^{-\kappa} \odot (\boldsymbol{\rho} - \mathbf{E}\mathbf{Z} \oslash \mathbf{f}\mathbf{e}_m^\top) - \frac{\alpha\kappa}{2}\mathbf{c} \oslash \boldsymbol{\rho}]^\top$; // Matrix $\mathbf{H}$.
12 $\quad \mathbf{Z} \leftarrow (\mathbf{e}_n\boldsymbol{\rho}^\top) \odot \mathbf{Exp}(-\mathbf{H})$; // Unnormalized membership matrix.[10]
13 $\quad \mathbf{Z} \leftarrow \mathbf{Z} \oslash \mathbf{Z}\mathbf{e}_m\mathbf{e}_n^\top$; // Normalize the membership matrix.
14 $\quad \mathbf{Z} = \lambda\mathbf{Z} + (1 - \lambda)\mathbf{Z}^{old}$; // Move it according to the learning rate.
15 $\quad$ **if** $\mathcal{T} \neq \emptyset$ **then**
16 $\quad\quad z_{ig} \leftarrow z_{ig}^0,\ \forall i \in \mathcal{T}, \forall g.$; // Reset tagged objects (if any) to initial values.
17 $\quad$ **end**
18 $\quad \mathcal{F} \leftarrow \beta\mathbf{f}^\top\mathbf{Z}\boldsymbol{\delta} + \frac{\alpha}{2}\mathbf{e}_m^\top\mathbf{c} + \mathbf{e}_n^\top(\mathbf{F} \odot \mathbf{Log}(\mathbf{F} \oslash \mathbf{f}\mathbf{e}_m^\top))\boldsymbol{\rho}$; // Free energy update.
19 **end**
20 `Return(Z)`

7 $\oslash$ and $\odot$ are componentwise (Hadamard) division and multiplication respectively.
8 $\mathbf{diag}()$ gives the vector with the diagonal of the matrix.
9 Powers of vectors are componentwise.
10 $\mathbf{Exp}()$ and $\mathbf{Log}()$ (on line 18) are componentwise.

A.2 Computations of exchange matrices

Uniform exchange matrix : The uniform exchange matrix $\mathbf{E}^{\mathrm{U}(r)}$ can be obtained with the Metropolis-Hastings algorithm from an adjacency matrix $\mathbf{A}^r = (a^r_{ij}) = (\mathbb{1}(|i-j| \leq r]))$, given the stationary distribution $\mathbf{f}$. It reads

$$\mathbf{E}^{\mathrm{U}(r)} = \mathbf{Diag}(\mathbf{f}) - \mathbf{LB} \quad \text{where} \quad \mathbf{B} = (b_{ij}) = \left(\min \left(\frac{f_i a^r_{ij}}{a^r_{i\bullet}}, \frac{f_j a^r_{ji}}{a^r_{j\bullet}} \right) \right)$$

where $\mathbf{Diag}(\mathbf{f})$ is the diagonal matrix containing $\mathbf{f}$ and $(\mathbf{LB})_{ij} := \delta_{ij} b_{i\bullet} - b_{ij}$ the Laplacian of $\mathbf{B}$.

Diffusive exchange matrix : Here, we use a diffusive process from the adjacency matrix $\mathbf{A} = (a_{ij}) = (\mathbb{1}(|i-j| = 1]))$. It gives

$$\mathbf{E}^{\mathrm{D}(t)} = \mathbf{Diag}(\mathbf{f})^{1/2} \mathbf{Exp}(-t\mathbf{\Psi}) \mathbf{Diag}(\mathbf{f})^{1/2}$$

where the $\mathbf{Exp}()$ is the matrix exponentiation and

$$\mathbf{\Psi} := \mathbf{Diag}(\mathbf{f})^{-1/2} \frac{\mathbf{LA}}{\mathrm{tr}(\mathbf{LA})} \mathbf{Diag}(\mathbf{f})^{-1/2}$$

with $(\mathbf{LA})_{ij} = \delta_{ij} a_{i\bullet} - a_{ij}$ the Laplacian of $\mathbf{A}$.