A Closer Look at Clustering Bilingual Comparable Corpora

Anna Laskina, Eric Gaussier, Gaelle Calvary
Univ. Grenoble Alpes, CNRS, Grenoble INP*, LIG
38000 Grenoble, France
{anna.laskina, eric.gaussier, gaelle.calvary}@univ-grenoble-alpes.fr

Abstract

We study in this paper the problem of clustering comparable corpora, building upon the observation that different types of clusters can be present in such corpora: monolingual clusters comprising documents in a single language, and bilingual or multilingual clusters comprising documents written in different languages. Based on a state-of-the-art deep variant of Kmeans, we propose new clustering models fully adapted to comparable corpora and illustrate their behavior on several bilingual collections (in English, French, German and Russian) created from Wikipedia.

Keywords: Text clustering, bilingual comparable corpora, Kmeans-based models

1. Introduction

Text clustering is a fundamental task of NLP and unsupervised machine learning aiming to unveil the structure underlying a collection of documents by identify clusters of similar documents. In addition to a better understanding of a collection through the clusters it contains, clustering comparable corpora can be useful to lexical and terminological studies as linguists can study the different usages of a term and its translations in specific topics. Many models and algorithms have been proposed for clustering purposes, from the original Kmeans algorithm (MacQueen et al., 1967) to latent topic models as LDA (Blei et al., 2003), the DBSCAN and HDBSCAN approaches (Ester et al., 1996; Campello et al., 2013), and the most recent Deep α-Kmeans proposal (Moradi Fard et al., 2020) which jointly learns document representations and cluster representatives. We are interested in this study in clustering comparable corpora, *i.e.*, corpora consisting of documents written in different languages without being translation of one another. For the sake of simplicity, we further focus on bilingual comparable corpora, in which documents can be written in either one of two languages, denoted by ℓ_1 and ℓ_2 in the remainder.

The main question we address here is whether or not one should rely on existing methods to cluster such corpora or develop dedicated methods. It is of course possible to directly apply existing clustering methods, as the ones mentioned above, to identify clusters in bilingual comparable corpora. However, the peculiarity of the clusters one is aiming at when clustering bilingual comparable corpora is that they are of different types: some are monolingual clusters containing documents written in either ℓ_1 or ℓ_2 but not both, whereas other are bilingual and contain documents written in ℓ_1 as well as documents written in ℓ_2. For example, in bilingual comparable collections comprising newspaper articles from different countries, some topics and clusters will be specific to a particular country and a particular language, while others will be shared across countries and languages: documents pertaining to national politics and economy may relate to specific, monolingual topics and clusters, whereas documents concerning international affairs will likely relate to topics and clusters common to the two corpora.

Standard clustering approaches are of course blind wrt this distinction; by considering all documents equivalently, without accounting for language distinctions, they may result in the inability to accurately identify clusters. The question is thus whether considering different types of clusters, solely in ℓ_1, solely in ℓ_2 or in both languages, may help cluster bilingual comparable corpora.

To address this question, we introduce two new variants of the Kmeans algorithm. These variants are based on the state-of-the-art text clustering method Deep Kmeans and its main building block, denoted here α-Kmeans, which provides a fully differentiable and soft version of the Kmeans problem of the form:

$$\arg\min_{\mathcal{R}} \sum_{x \in \mathcal{X}} \sum_{k=1}^{K} d\left(x, r_k\right) \frac{e^{-\alpha d(x, r_k)}}{\sum_{k'=1}^{K} e^{-\alpha d(x, r_{k'})}}, \quad (1)$$

where $\mathcal{R}$ is the set of cluster representatives $\{r_k\}_{k=1}^K$, $\mathcal{X}$ is the set of documents, d is a dissimilarity measure, typically the squared Euclidean distance $d(x, y) = \|x - y\|_2^2$, and α is a non-negative real number which plays the role of an inverse temperature (Moradi Fard et al., 2020) such that when α tends to $+\infty$ the fraction in Eq. 1 is 0 or 1 so that one recovers the original Kmeans formulation leading to a hard assignment of documents to clusters (whereas finite values of α result in a soft clustering of the documents over the different clusters). Note that both documents and cluster representatives are vectors in $\mathbb{R}^p$.

LREC-COLING 2024, pages 133–142
20-25 May, 2024.

We furthermore evaluate these two variants through a comprehensive series of experiments conducted on new bilingual corpora with ground truth clustering labels obtained from Wikipedia. Our contributions are thus two-fold:

- Firstly, we introduce two new models based on α-Kmeans dedicated to clustering bilingual comparable corpora by (a) relying on different cluster types (monolingual and bilingual), and (b) further improving document representations obtained from auto-encoders through masking;

- Secondly, we assess the quality of these models on new comparable corpora with ground truth clusters.

To the best of our knowledge, this is the first attempt to develop a clustering model dedicated to comparable corpora.

The remainder of the paper is organized as follows: Section 2 describes related work; Section 3 introduces our new comparable clustering models which are then evaluated through a series of experiments in Section 4; Section 5 concludes the paper.

2. Related Work

Clustering can be executed using various approaches. Algorithms, such as Gaussian Mixture Models (Reynolds et al., 2009), Dirichlet Process Mixture Models (DPMM), Bayesian Nonparametric Clustering (Hjort et al., 2010), leverage distribution-based clustering techniques. While these algorithms are well-suited for handling uncertainty, they are highly sensitive to the underlying data distribution and are less effective when dealing with large and high-dimensional corpora.

In density-based clustering algorithms such as DBSCAN (Ester et al., 1996) or OPTICS (Ankerst et al., 1999), data points are clustered within regions characterized by high data point density, with these high-density regions demarcated by regions of lower data point density. These algorithms demonstrate proficiency in managing outliers, although they may encounter challenges when confronted with clusters exhibiting similar density and high-dimensional data. Agglomerative Hierarchical Clustering and Divisive Hierarchical Clustering are clustering techniques used to group data points into a hierarchical structure of clusters. It can be computationally intensive, especially for large data sets, and the choice of parameters can have a significant impact on the results. Hierarchical density-based clustering combines aspects of density-based and hierarchical clustering, as seen in methods like HDBSCAN (Campello et al., 2013). Similar to hierarchical clustering, it can be computationally intensive, and parameter selection for both density-based clustering and hierarchical aggregation steps is crucial.

Centroid-based clustering models, represented by algorithms like Kmeans, may not perform as well in terms of initialization, handling irregularly shaped clusters, and addressing outliers. However, they rely on well-founded objective functions, excel in a wide range of applications, offer high scalability, and can handle large data sets with a relatively small memory footprint, making it vital for real-world applications involving big data. Furthermore, the fully differentiable, deep version of Kmeans, known as Deep α-Kmeans, has recently been shown to outperform many clustering alternatives on text collections (Moradi Fard et al., 2020). We thus rely on this family of models in the remainder.

Lastly, we know of no work specifically dedicated to clustering comparable corpora. We believe that this is due to the fact that any clustering algorithm can be applied on such corpora, provided one does not want to take into account the different cluster types inherent to them.

3. Clustering Comparable Corpora

We present in this section new models to cluster bilingual comparable corpora.

3.1. Monolingual and Bilingual Cluster Types

In the remainder, we will say that monolingual clusters in language ℓ_1 are of type t_1, that monolingual clusters in language ℓ_2 are of type t_2, and that bilingual clusters containing both documents in ℓ_1 and ℓ_2 are of type t_3. If a document in ℓ_1 (resp. ℓ_2) is close to other documents in ℓ_1 (resp. ℓ_2), and far away from documents in ℓ_2 (resp. ℓ_1), then it is likely that this document belongs to a cluster of type t_1 (resp. t_2). Conversely, if a document in ℓ_1 or ℓ_2 is close to both documents in ℓ_1 and ℓ_2, then it is likely that this document belongs to a cluster of type t_3.

We use here the ratio of the distances to the closest document in language ℓ_1 and to the closest document in language ℓ_2 to determine the cluster type, with the assumption that if this ratio for a document in ℓ_1 strongly favors ℓ_1 (resp. ℓ_2 for a document in ℓ_2), then the document likely belongs to a cluster of type t_1 (resp. t_2). On the other hand, if the ratio does not strongly favor any language, then the document likely belongs to a cluster of type t_3. This can be captured through the following quantity, denoted $F(v, i)$, where v denotes either a document or a cluster representative and i one of

the three cluster types:

$$F(v,i)_{i=1,2} = e^{\tau(\frac{1}{\mu} - \frac{d_i(v)}{d_{\bar{i}}(v)})} I(v,i),$$
$$F(v,3) = e^{\tau(min(\frac{d_1(v)}{d_2(v)} - \frac{1}{\mu}, \frac{d_2(v)}{d_1(v)} - \frac{1}{\mu}))} I(v,3). \quad (2)$$

$d_1(v)$ (resp. $d_2(v)$) is the distance of v to its closest document in ℓ_1 (resp. ℓ_2), and $\bar{i} = 2$ if $i = 1$ and 1 if $i = 2$. $1/\mu$, $\mu > 1$, represents the quantity controlling to which extent the ratio of closest distances favors one of the two monolingual cluster types: for a document in ℓ_1, if $d_1(v)/d_2(v)$ is smaller that $1/\mu$, that is if the distance to the closest document in ℓ_2 is greater, by a factor μ, than the distance to the closest document in ℓ_1, then the document is likely to belong to a cluster of type t_1 (and similarly for documents in ℓ_2 and clusters of type t_2). The other hyperparameter, τ, controls to which extent the assignment of documents and representatives to cluster types is harder (higher values of τ) or softer (smaller values of τ). Lastly, the function $I(v,i)$, $1 \leq i \leq 3$ allows one to avoid assigning documents in ℓ_1 (resp. ℓ_2) to clusters of type t_2 (resp. t_1). It is defined by: $I(v,1) = I(v,2) = I(v,3) = 1$ if v is a representative, $I(v,1) = I(v,3) = 1$ and $I(v,2) = 0$ if v is a document in ℓ_1, and $I(v,2) = I(v,3) = 1$ and $I(v,1) = 0$ if v is a document in ℓ_2.

We can then define the probability, for any document or cluster representative v, to belong to one of the three cluster types as:

$$P(t_i|v) = \frac{F(v,i)}{\sum_{i=1}^{3} F(v,i)}.$$

Lastly, in Eq. 1, each document x is assigned to each representative $\{r_k\}_{k=1}^{K}$ with a quantity which can be interpreted as the probability that r_k is the closest representative to x. However, when considering cluster types, one should try and assign a document x of type t_i, $1 \leq i \leq 3$, to representatives of the same type, and forbid assignment to representatives and clusters of different types. To do so, one can rewrite Eq. 1 as:

$$\arg\min_{\mathcal{R}} \sum_{x \in \mathcal{X}} \sum_{i=1}^{3} P(t_i|x) \sum_{k=1}^{K} d(x, r_k) \mathcal{A}(x, t_i, k; \alpha, \mathcal{R}), \quad (3)$$

with:

$$\mathcal{A}(x, t_i, k; \alpha, \mathcal{R}) = \frac{P(t_i|r_k) e^{-\alpha d(x, r_k)}}{\sum_{k'=1}^{K} P(t_i|r_{k'}) e^{-\alpha d(x, r_{k'})}}. \quad (4)$$

For a document x of type t_i, that is for which $P(t_i|x)$ is high, the above formulation privileges, in $\mathcal{A}(x, t_i, r_k; \alpha, \mathcal{R})$, representatives of the same type, that is representatives for which $P(t_i|x)$ is high. The solution to the optimization problem in Eq. 3, which can be obtained through standard gradient descent approaches, will be referred to as $c\alpha$-Kmeans.

3.2. Weak Representation Learning through Masking

Eq. 3 is valid for any vector-based representations of documents, in particular representations obtained with auto-encoders. In that case, the representation is learned beforehand, and used in the clustering process. One can wonder however whether it is possible to jointly learn a representation and cluster documents, as in Deep Kmeans, with the potential advantage of learning a representation fully adapted to the clustering task.

It turns out that is not possible to replicate, in the context of comparable corpora with different cluster types, the approach followed in Deep Kmeans, which relies on a joint loss function aiming to jointly minimize the reconstruction error of the representation obtained with an auto-encoder and the error of the clustering obtained with this representation. This is due to the fact that the optimization problem at the basis of Eq. 3 may change from one epoch to the other if the representations of documents change through an update of the auto-encoder; indeed, this change can yield different values for $d_1(x)$ and $d_2(x)$ and eventually different cluster types. In that case, there is no guarantee that the joint process converges.

This said, if one assumes that the original representation of documents, typically in the form of a vector obtained via an auto-encoder in our case, contains the necessary information to accurately cluster the comparable corpus, then one can try and improve this representation by identifying, for each cluster, the most relevant dimensions. Of course, the information relevant to a particular cluster may be widespread over different dimensions; however, we believe it unlikely that all relevant information are equally distributed over all dimensions, and more likely that most of the relevant information for a particular cluster be present in a subset of the original dimensions, specific to the cluster.

We thus introduce cluster masks, which are binary vectors representing, for each cluster, the dimensions that should be retained for this cluster. In order to avoid "concentrating" all documents and cluster representatives on the same point, which would minimize the objective function of Eqs. 1 and 3 but would lead to a degenerate solution, we further limit the number of dimensions not retained, so that the cluster masks are defined by:

$$\mathcal{M}_\eta = \{m^{(k)} | m^{(k)} \in \{0,1\}^p, \sum_{i=1}^{p} m_i^{(k)} \geq \eta \cdot p\}_{k=1}^{K} \quad (5)$$

In this formulation, the cluster masks serve as a filtering mechanism that selectively emphasizes the relevant dimensions within each cluster. By setting certain dimensions to 1 and others to 0, the mask effectively highlights the dimensions that

contribute significantly to the representation of each cluster. The parameter η controls the sparsity of the masks, determining the number of dimensions to be considered.

The comparison between each document and each cluster representative can then be based on the cluster mask by adapting the dissimilarity measure d used in Eqs. 1 and 3 to rely only on the dimensions retained for the cluster, through:

$$\forall k,\ 1 \leq k \leq K, d_{m^{(k)}}(x, r_k) = d(x \odot m^{(k)}, r_k \odot m^{(k)}),$$

where $\odot$ represents the element-wise product between two vectors.

Integrating this approach into Eq. 3 leads to the following optimization problem:

$$\arg\min_{\mathcal{R},\mathcal{M}} \sum_{x \in \mathcal{X}} \sum_{i=0}^{2} P(i|x) \sum_{k=1}^{K} d_{m^{(k)}}(x, r_k) \mathcal{A}_m(x, t_i, k; \alpha, \mathcal{R}, \mathcal{M}_\eta), \quad (6)$$

with:

$$\mathcal{A}_m(x, t_i, k; \alpha, \mathcal{R}, \mathcal{M}_\eta) = \frac{P(i|r_k) e^{-\alpha d_{m^{(k)}}(x, r_k)}}{\sum_{k'=1}^{K} P(i|r_{k'}) e^{-\alpha d_{m^{(k)}}(x, r_{k'})}}. \quad (7)$$

The solution of this problem, which is again fully differentiable, is a model we refer to as $mc\alpha$-Kmeans, which is designed to take into account the different cluster types inherent to clustering comparable corpora while identifying representations specific to each cluster.

Updating cluster masks We assume here that the dissimilarity used is the squared Euclidean distance, which corresponds to $d(x, r_k) = \sum_{j=1}^{p} d^j(x, r_k)$ with $d^j(x, r_k) = (x_j - r_{k_j})^2$. In both Eqs 4 and 7, the quantities $\mathcal{A}(x, t_i, k; \alpha, \mathcal{R})$ and $\mathcal{A}_m(x, t_i, k; \alpha, \mathcal{R}, \mathcal{M}_\eta)$ can be interpreted as the probabilities that x belongs to the cluster represented by r_k; when α tends to infinity, these two quantities tend to either 0 or 1 as only the dominating term, corresponding to the smallest distance in the sum, is kept in the denominator, which either dominates the numerator, leading a probability of 0, or is equal to the numerator, leading a probability of 1. The contribution of the j^{th} dimension of the k^{th} cluster to Eq. 6 can thus be approximated, for α sufficiently large, by:

$$\sum_{x \in \mathcal{X}} \sum_{i=0}^{2} P(i|x) d^j_{m^{(k)}}(x, r_k) \mathcal{A}_m(x, t_i, k; \alpha, \mathcal{R}, \mathcal{M}_\eta). \quad (8)$$

Starting with mask vectors with all coordinates set to 1, at the end of each epoch, for each cluster, we then simply update its mask by setting to 0 its coordinate on the dimension which deteriorates the most the clustering loss, that is the dimension for which the value given by Eq. 8 is the highest. We however do not update a mask $m(k)$ if the update would lead to violate the constraint $\sum_{i=1}^{p} m_i^{(k)} \geq \eta \cdot p$.

4. Experiments

We describe here our experimental protocol, present the results obtained and discuss important issues. All data sets and code are freely available through public repositories[1] .

4.1. Data Collection

To obtain comparable corpora with ground truth clusters and a variety of languages, we relied on Wikipedia and its underlying interlingual category system which is a graph with a tree backbone and a root corresponding to the category *Main topic classifications*. Articles in Wikipedia have different versions in different languages which, while being close to each other in terms of content, are usually not translations of each other. To this extent, Wikipedia can be seen as an easily accessible, high quality comparable corpus. In order to select both clusters and documents which are neither too generic nor too specific, we first filtered out all categories: (i) keeping only the ones such that the length of the shortest path to the root is comprised between 2 (level-2 categories) and 22 (level-22 categories), (ii) that either contain the empty words *by, in, from, about, and, after*, as otherwise they would be too specific, or the words *list, award, image, quotation, event, outline, redirect, people* as articles in these categories usually represent an enumeration, and (iii) keeping only the ones that are not ambiguous in the sense that they are not subcategories of several categories. From the level-2 categories and their subcategories, we then built several comparable corpora through the following process:

1. For each cluster type t_1, t_2, t_3, we first randomly select a number of clusters in {10, 15, 20, 25, 30}, and then associate each cluster to one level-2 category randomly selected without replacement,

2. For each level-2 category thus selected, randomly select a number of documents in {100, 250, 500, 750, 1000, 1250, 1500, 2000} for categories of types 1 and 2, and a number of

[1]The tools for creating data sets as well as the data sets used in our experiments are available at: https://github.com/anna-laskina/comparable_corpora_generator. The source codes of the proposed models and the evaluation measures are available at: https://github.com/anna-laskina/comparable_clustering.

document pairs in {100, 250, 500, 750, 1000, 1250, 1500, 2000} for categories of type 3,

3. Then, for each category, collect the articles, or pairs of articles, directly related to it in Wikipedia and which do not belong to more than ρ level-2 categories and to any other selected level-2 category of different types, focusing on articles in ℓ_1 (resp. ℓ_2) if the category is of type 1 (resp. 2), or using the pair of articles in ℓ_1 and ℓ_2 if the category is of type 3; we refer to the set of articles, or pairs of articles, thus obtained as $\mathcal{A}$.

4. Continue collecting all articles, or pairs of articles, directly related to the subcategories of the categories considered so far[2] and add them in $\mathcal{A}$ until $\mathcal{A}$ contains at least n articles (or pairs of articles) or there are no more subcategories, where n is the number of documents selected for the category at step 2.

5. Finally, if $\mathcal{A}$ contains more than 100 articles, which is the lower bound for the number of articles considered per cluster, randomly select n articles, or pairs of articles, from $\mathcal{A}$; otherwise do not consider the category.

The ranges considered for the number of clusters of different types and the number of documents per cluster allow one to obtain varied comparable corpora, with more or less balanced clusters and monolingual and bilingual parts. Furthermore, ρ allows one to control to which extent the comparable corpus obtained relies on hard ($\rho = 1$) or soft clusters ($\rho > 1$). As the average number of clusters per document in almost all corpora we finally considered is 1.02 and as our clustering methods derive from Kmeans, which is a hard clustering method, we present in this section the results obtained on comparable corpora built with $\rho = 1$. Lastly, we built 9 data sets for the English-French (En-Fr) language pair, one for the English-German (En-Ger) language pair and one for the French-Russian (Fr-Ru) language pair. Table 1 displays the main characteristics of these data sets and Table 2 presents an example of the cluster appearance in corpora.

Corpora id	t_1		t_2		t_3	
	doc.	k	doc.	k	doc.	k
En-Fr №1	6240	16	2342	9	6160	7
En-Fr №2	1657	8	3193	6	4354	8
En-Fr №3	7986	16	2068	8	5132	8
En-Fr №4	3472	12	925	7	5178	9
En-Fr №5	2601	12	1048	4	4150	20
En-Fr №6	5487	19	882	7	5460	12
En-Fr №7	8972	15	507	5	8026	13
En-Fr №8	2988	8	947	8	10682	15
En-Fr №9	2205	7	1045	5	2388	6
En-Ger №1	1620	6	1012	7	3004	7
Fr-Ru №1	1618	6	945	7	3002	7

Table 1: Data sets used in the experiments. For each data set, we provide the number of documents and number of clusters per cluster type. The numbers of clusters below 10 are due to the potential pruning in step 5 of the data collection process.

4.2. Experimental Protocol

Models compared In order to evaluate our proposal, we compared our approach with different variants of the Kmeans algorithm, as (a) other baselines as HDBSCAN performed badly on the data sets retained[3], and (b) Deep α-Kmeans model is a state-of-the-art model for text clustering. The models retained are: the scikit-learn implementation of the standard Kmeans algorithm (Sklearn), a variant of the Kmeans algorithm which uses batches (Kmeans batch) as all other models but the previous one, the fully differentiable and soft version α-Kmeans, our proposals $c\alpha$-Kmeans and $mc\alpha$–Kmeans, and the pre-trained version of the deep Kmeans model proposed by Moradi Fard et al. (2020) (referred to here as Deep α-Kmeans^p). Furthermore, in order to assess the importance of the masking mechanism (Section 3.2), we also included it in the α-Kmeans model, leading to the $m\alpha$-Kmeans model, and in the Deep α-Kmeans^p model, leading to the Deep $m\alpha$-Kmeans^p model. To produce comparable results when using non-deep versions of Kmeans, we used the same auto-encoder with the same settings as for the deep version Moradi Fard et al. (2020), using pre-trained BERT-base embeddings as input to the auto-encoder.

As users usually interact with clustering methods in order to select the best hyperparameters of the method, we ran each model 10 times and compared them according to their best run, the average of their 3 best runs and the average of their 10 runs. On each data set, the hyperparameters were chosen on the following sets: {1.00, 5.00, 10.00, 10000.00} for α, {0.0, 1.0, 5.00, 10.00} for τ, and {1.00, 0.95, 0.90, 0.80, 0.75} for η. For $m\alpha$Kmeans, $mc\alpha$-Kmeans and Deep $m\alpha$-Kmeans, we set $\mu = 2.0$. To ensure reproducibility, identical seed values were applied to all models for initialization.

Evaluation measures We compared the clusters obtained by the above models with the ground truth labels available in the data sets we constructed. Apart from the standard versions of

[2] At first only one category is considered, then all its subcategories, then all the subcategories of its subcategories, ...

[3] In particular, in all the configurations we tested, HDBSCAN provided a high number of outliers, resulting in a lot of documents not being assigned to any cluster and finally poor clustering performance.

Type	Cluster category	Document titles
t_1	Wisdom, Social concepts, Musical composition, **Information theory**, Information management, Sports stubs, Technology development	*Entropic vector*, *Pinsker's inequality*, *Pointwise mutual information*
t_2	Periodic phenomena, **Historiography**, Historic preservation, Books, Business software	*Révisionnisme*, *Histoire comparée*, *Âge d'or des comics*
t_3	Geographical areas, History of religion, **Philosophical theories**, Change, Information Age, Time zones	*Scientific realism*, *Quantum mind*, *Acatalepsy*, *Réalisme scientifique*, *Esprit quantique*, *Acatalepsie*

Table 2: Category representations within *En-Fr №9* corpus by monolingual English (t_1), monolingual French (t_2), and bilingual (t_3) types. The second column provides lists of the Wikipedia categories that formed clusters in the collection. For categories in bold, the third column lists some of the document titles assigned to them. Underlined titles refer to texts in English, while ununderlined titles refer to texts in French.

Kmeans, all models in fact yield soft assignments of documents to clusters. In addition to these soft assignments, we also consider hard assignments obtained by selecting the best cluster for each document. For evaluating the hard clustering results, we used two standard measures, namely the Matrices Adjusted Rand Index (ARI) (Hubert and Arabie, 1985), and Adjusted Mutual information (AMI) (Vinh et al., 2009). The prevailing methods for evaluating soft clustering algorithms typically involves expanding on the Rand Index (Rand, 1971). There are indeed numerous fuzzy adaptations of the Rand Index (Campello, 2007; Frigui et al., 2007; Brouwer, 2009; Hullermeier and Rifqi, 2009; Anderson et al., 2010). In this work, we employed the Fuzzy Rand Index by Hullermeier and Rifqi (2009) (H-FRI), which is based on the comparison of the document distributions on the true clusters and the clusters obtained with a model, and our own variation of it called O-FRI and specifically tailored for collections in which documents belong to a limited number of clusters. Indeed, H-FRI tends to give high scores to distributions concentrated on a few clusters and fail to discriminate between different models. Following the notation introduced by Hullermeier et al. (2011), we use in O-FRI the cosine siimlarity for the *fuzzy equivalence relations* E_P and E_Q, and renormalize the *degree of discordance* as:

$$disc(x,x') = \lambda(E_P(x,x') - E_Q(x,x'))^2$$
$$\lambda = \begin{cases} \frac{b}{n(n-1)/2-b} & if E_P(x,x') = 0 \\ \frac{n(n-1)/2-b}{b} & \text{otherwise} \end{cases} \quad (9)$$

where b is a number of pairs $(x,x') \in \mathcal{X}^2$ of documents for which $E_P(x,x') = 0$. Here, P denotes the ground truth partition, while Q denotes the predicted partition. Figure 1 illustrates whether the different measures retained concentrate or not on a particular range of values. As one can note, both H-FRI and AMI tend to concentrate in a given region and fail to discriminate different models. For space reasons, we thus report here the results obtained with ARI and O-FRI.

Code All models and measures were implemented in Python (v. 3.8.10), using the scikit-learn (v. 1.1.3), numpy (v. 1.22.2), scipy (v. 1.10.1) and pytorch (v. 1.10.2) libraries.

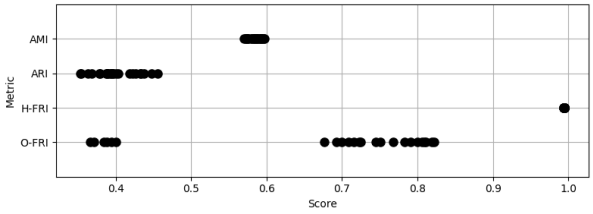

Figure 1: Concentration of the evaluation measures for all models on the 9 En-Fr data sets.

4.3. Results

The main results are summarized in Table 4 for the En-Ger and Fr-Ru collections, and Table 3 for the En-Fr collections. As one can note, for ARI, the best results over all 9 En-Fr collections, as well as for the En-Ger collection, for the best run, the three best runs and the 10 best runs, are obtained with our complete proposal $mc\alpha$-Kmeans, followed by Deep $m\alpha$-Kmeans^p. Furthermore, for the last collection (Fr-Ru), $mc\alpha$-Kmeans is the best model when comparing the best run and the three best runs, and second after Deep $m\alpha$-Kmeans^p when comparing the 10 best runs. For O-FRI, for the En-Fr collections, $mc\alpha$-Kmean is the best model for the 10 best runs on 7 out of 9 collections, the best model for the 3 best runs on 6 out of 9 collections, and the best model for the best run on 5 out of 9 collections. It is in average (last column of Table 3) the best model for the 3 best and 10 best runs, and is second and close (0.819 vs 0.822) to best model (Deep $m\alpha$-Kmeans^p) in average for the best run. The two models, $mc\alpha$-Kmean and Deep $m\alpha$-Kmeans^p, are the best models for all En-Fr collections on all runs. Contrary to the En-Fr collections, the Fr-Ru and En-Ger data sets benefit from the capacity of the deep versions to jointly learn a representation and perform clustering. Indeed, Deep $m\alpha$-Kmeans^p is the best model here, over all runs, followed by Deep α-Kmeans^p and

ARI									
En-Fr №1	En-Fr №2	En-Fr №3	En-Fr №4	En-Fr №5	En-Fr №6	En-Fr №7	En-Fr №8	En-Fr №9	avg
AE+Sklearn									
0.369	0.546	0.435	0.275	0.349	0.396	0.257	0.282	0.693	0.400 ± 0.134
0.353 ± 0.013	0.501 ± 0.032	0.433 ± 0.003	0.267 ± 0.010	0.345 ± 0.003	0.393 ± 0.003	0.230 ± 0.019	0.274 ± 0.006	0.690 ± 0.003	0.387 ± 0.134
0.329 ± 0.018	0.452 ± 0.043	0.401 ± 0.027	0.249 ± 0.015	0.339 ± 0.005	0.379 ± 0.011	0.212 ± 0.016	0.254 ± 0.017	0.656 ± 0.034	0.363 ± 0.127
AE+Kmeans batch									
0.366	0.518	0.431	0.285	0.357	0.419	0.273	0.286	0.694	0.403 ± 0.128
0.345 ± 0.015	0.494 ± 0.017	0.428 ± 0.004	0.275 ± 0.008	0.353 ± 0.003	0.411 ± 0.007	0.239 ± 0.025	0.274 ± 0.009	0.686 ± 0.007	0.389 ± 0.130
0.331 ± 0.016	0.454 ± 0.033	0.404 ± 0.025	0.254 ± 0.015	0.344 ± 0.007	0.396 ± 0.012	0.216 ± 0.021	0.253 ± 0.018	0.660 ± 0.028	0.368 ± 0.127
AE+α–Kmeans									
0.374	0.539	0.463	**0.432**	0.380	0.427	0.275	0.316	0.698	0.434 ± 0.119
0.353 ± 0.010	0.512 ± 0.025	0.446 ± 0.012	**0.400 ± 0.028**	0.375 ± 0.004	0.418 ± 0.009	0.267 ± 0.008	0.300 ± 0.012	0.691 ± 0.006	0.418 ± 0.119
0.339 ± 0.015	0.465 ± 0.031	0.417 ± 0.030	**0.338 ± 0.050**	0.359 ± 0.012	**0.401 ± 0.013**	0.234 ± 0.024	**0.278 ± 0.019**	0.665 ± 0.025	0.388 ± 0.118
AE+$m\alpha$–Kmeans									
0.413	0.573	0.477	**0.432**	0.381	0.427	**0.313**	0.316	0.716	0.449 ± 0.122
0.391 ± 0.012	0.535 ± 0.027	0.460 ± 0.017	**0.400 ± 0.026**	0.375 ± 0.006	0.418 ± 0.005	**0.279 ± 0.024**	0.300 ± 0.008	0.698 ± 0.016	0.428 ± 0.120
0.357 ± 0.022	**0.473 ± 0.054**	**0.429 ± 0.026**	**0.338 ± 0.046**	0.359 ± 0.012	**0.401 ± 0.011**	**0.248 ± 0.029**	**0.278 ± 0.019**	0.666 ± 0.028	0.394 ± 0.116
AE+$c\alpha$–Kmeans									
0.376	0.539	0.463	**0.432**	**0.390**	0.427	0.275	0.316	**0.720**	0.438 ± 0.124
0.366 ± 0.007	0.512 ± 0.025	0.458 ± 0.006	**0.400 ± 0.028**	0.375 ± 0.004	0.418 ± 0.009	0.267 ± 0.008	0.300 ± 0.012	0.706 ± 0.014	0.422 ± 0.122
0.340 ± 0.021	0.465 ± 0.031	0.421 ± 0.036	**0.338 ± 0.050**	0.359 ± 0.012	**0.401 ± 0.013**	0.234 ± 0.024	**0.278 ± 0.019**	**0.669 ± 0.033**	0.389 ± 0.119
AE+$mc\alpha$–Kmeans									
0.424	**0.591**	**0.480**	**0.432**	**0.390**	**0.428**	**0.313**	**0.326**	**0.720**	**0.456 ± 0.121**
0.398 ± 0.022	**0.546 ± 0.042**	**0.463 ± 0.016**	**0.400 ± 0.026**	**0.382 ± 0.011**	**0.419 ± 0.007**	**0.279 ± 0.024**	**0.306 ± 0.016**	**0.708 ± 0.017**	**0.433 ± 0.122**
0.358 ± 0.033	**0.473 ± 0.054**	**0.429 ± 0.026**	**0.338 ± 0.046**	**0.360 ± 0.008**	**0.401 ± 0.011**	**0.248 ± 0.029**	**0.278 ± 0.019**	**0.669 ± 0.036**	**0.395 ± 0.117**
Deep α–Kmeans^p									
0.324	0.486	0.400	0.374	0.354	0.389	0.267	0.300	0.677	0.397 ± 0.116
0.303 ± 0.015	0.467 ± 0.014	0.385 ± 0.006	0.336 ± 0.017	0.344 ± 0.007	0.384 ± 0.004	0.241 ± 0.019	0.281 ± 0.019	0.669 ± 0.006	0.379 ± 0.120
0.282 ± 0.018	0.441 ± 0.022	0.365 ± 0.017	0.288 ± 0.042	0.337 ± 0.006	0.368 ± 0.011	0.226 ± 0.015	0.244 ± 0.014	0.638 ± 0.037	0.354 ± 0.119
Deep $m\alpha$–Kmeans^p									
0.324	0.471	0.406	0.374	0.357	0.383	0.268	0.301	0.680	0.396 ± 0.115
0.305 ± 0.013	0.467 ± 0.005	0.387 ± 0.005	0.338 ± 0.015	0.345 ± 0.008	0.379 ± 0.004	0.242 ± 0.018	0.276 ± 0.017	0.667 ± 0.009	0.378 ± 0.119
0.282 ± 0.019	0.435 ± 0.018	0.365 ± 0.021	0.288 ± 0.043	0.338 ± 0.007	0.367 ± 0.013	0.228 ± 0.014	0.243 ± 0.014	0.637 ± 0.037	0.353 ± 0.118
O-FRI									
En-Fr №1	En-Fr №2	En-Fr №3	En-Fr №4	En-Fr №5	En-Fr №6	En-Fr №7	En-Fr №8	En-Fr №9	avg
AE+Sklearn									
0.356	0.517	0.395	0.318	0.328	0.373	0.281	0.306	0.677	0.395 ± 0.119
0.342 ± 0.010	0.483 ± 0.025	0.393 ± 0.003	0.314 ± 0.005	0.326 ± 0.002	0.368 ± 0.004	0.261 ± 0.014	0.301 ± 0.004	0.672 ± 0.004	0.384 ± 0.118
0.325 ± 0.013	0.447 ± 0.033	0.366 ± 0.024	0.303 ± 0.010	0.322 ± 0.004	0.355 ± 0.010	0.248 ± 0.011	0.287 ± 0.011	0.640 ± 0.032	0.366 ± 0.110
AE+Kmeans batch									
0.353	0.500	0.392	0.324	0.338	0.396	0.294	0.317	0.686	0.400 ± 0.116
0.339 ± 0.010	0.480 ± 0.014	0.389 ± 0.003	0.319 ± 0.004	0.333 ± 0.003	0.388 ± 0.006	0.268 ± 0.018	0.305 ± 0.008	0.678 ± 0.006	0.389 ± 0.117
0.328 ± 0.010	0.451 ± 0.026	0.369 ± 0.022	0.306 ± 0.010	0.327 ± 0.005	0.371 ± 0.012	0.251 ± 0.015	0.291 ± 0.013	0.649 ± 0.028	0.371 ± 0.112
AE+α–Kmeans									
0.637	0.577	0.800	0.591	0.459	0.782	0.903	0.850	0.778	0.709 ± 0.139
0.631 ± 0.004	0.561 ± 0.011	0.788 ± 0.013	0.576 ± 0.013	0.452 ± 0.006	0.759 ± 0.016	0.900 ± 0.002	0.850 ± 0.000	0.777 ± 0.001	0.699 ± 0.142
0.614 ± 0.015	0.530 ± 0.029	0.756 ± 0.025	0.552 ± 0.019	0.433 ± 0.015	0.722 ± 0.033	0.885 ± 0.019	0.843 ± 0.008	0.752 ± 0.023	0.676 ± 0.144
AE+$m\alpha$–Kmeans									
0.877	0.643	0.936	0.752	0.521	0.930	0.914	0.899	0.805	0.809 ± 0.137
0.871 ± 0.004	0.617 ± 0.019	0.936 ± 0.000	0.748 ± 0.003	0.505 ± 0.013	0.918 ± 0.009	0.911 ± 0.002	0.898 ± 0.001	0.796 ± 0.008	0.800 ± 0.142
0.861 ± 0.010	0.574 ± 0.040	0.934 ± 0.002	0.720 ± 0.035	0.484 ± 0.017	0.907 ± 0.010	0.906 ± 0.004	0.894 ± 0.005	0.769 ± 0.024	0.783 ± 0.152
AE+$c\alpha$–Kmeans									
0.695	0.577	0.816	0.596	0.459	0.819	0.919	0.857	0.785	0.725 ± 0.144
0.681 ± 0.011	0.562 ± 0.009	0.809 ± 0.008	0.586 ± 0.008	0.452 ± 0.006	0.804 ± 0.013	0.916 ± 0.002	0.850 ± 0.000	0.782 ± 0.002	0.716 ± 0.146
0.653 ± 0.026	0.530 ± 0.029	0.778 ± 0.025	0.561 ± 0.018	0.433 ± 0.015	0.769 ± 0.030	0.910 ± 0.007	0.843 ± 0.008	0.758 ± 0.026	0.693 ± 0.149
AE+$mc\alpha$–Kmeans									
0.891	**0.654**	**0.941**	0.788	0.524	**0.936**	**0.920**	**0.906**	0.812	0.819 ± 0.136
0.886 ± 0.004	**0.629 ± 0.029**	**0.938 ± 0.002**	0.784 ± 0.004	.506 ± 0.013	**0.936 ± 0.000**	**0.917 ± 0.002**	**0.902 ± 0.003**	.803 ± 0.009	**0.811 ± 0.143**
0.877 ± 0.008	**0.574 ± 0.040**	**0.934 ± 0.002**	**0.742 ± 0.039**	0.489 ± 0.015	**0.925 ± 0.009**	**0.911 ± 0.009**	**0.897 ± 0.006**	0.773 ± 0.024	**0.791 ± 0.154**
Deep α–Kmeans^p									
0.683	0.519	0.925	0.674	0.571	0.835	0.902	0.862	0.786	0.751 ± 0.138
0.669 ± 0.010	0.514 ± 0.004	0.921 ± 0.005	0.670 ± 0.001	0.557 ± 0.010	0.833 ± 0.001	0.902 ± 0.000	0.861 ± 0.000	0.771 ± 0.010	0.744 ± 0.140
0.632 ± 0.037	0.490 ± 0.021	0.863 ± 0.047	0.660 ± 0.019	0.533 ± 0.021	0.817 ± 0.023	0.902 ± 0.000	0.857 ± 0.005	0.746 ± 0.023	0.722 ± 0.142
Deep $m\alpha$–Kmeans^p									
0.899	0.545	0.925	**0.862**	**0.602**	0.836	0.902	0.900	**0.928**	**0.822 ± 0.136**
0.765 ± 0.046	0.538 ± 0.005	0.925 ± 0.000	**0.862 ± 0.000**	**0.599 ± 0.003**	0.834 ± 0.002	0.902 ± 0.000	0.900 ± 0.000	**0.926 ± 0.002**	0.806 ± 0.136
0.710 ± 0.044	0.524 ± 0.009	0.900 ± 0.019	0.685 ± 0.181	**0.586 ± 0.015**	0.828 ± 0.004	0.902 ± 0.000	0.860 ± 0.002	**0.917 ± 0.010**	0.768 ± 0.138

Table 3: Clustering results in terms of ARI and O-FRI on the 9 En-Fr collections (higher is better). Each cell contains the best run and the average of the 3 and 10 best runs (with std deviation).

$mc\alpha$-Kmeans.

To complement the above analysis, we display in Figure 2 the Critical Difference diagrams (Demšar, 2006) for the En-Fr collections for both evaluation measures, ARI and O-FRI. These diagrams are based on the non-parametric Friedman test (Friedman, 1937). In the event that the null hypothesis (indicating that all ranks are not significantly different) is rejected, the Nemenyi test (Nemenyi, 1963) is employed as a post-hoc test. According to the Nemenyi test, two models are deemed significantly different if the corresponding average ranks differ by at least the critical difference. These critical difference diagrams show that the best method overall is indeed $mc\alpha$-Kmeans, with a clear, even though not significant overall collections, difference with the second best model, which is $c\alpha$-Kmeans for ARI, and Deep α-Kmeans^p for O-FRI.

Table 5 provides results for En-Fr collections by cluster type. For both metrics, ARI and O-FRI, $mc\alpha$-Kmean outperforms the other models, sharing this place with α-Kmean, $m\alpha$-Kmean and $c\alpha$-Kmean for

En-Ger №1		Fr-Ru №2	
ARI	O-FRI	ARI	O-FRI
AE+Sklearn			
0.466	0.482	0.406	0.425
0.463 ± 0.003	0.474 ± 0.006	0.399 ± 0.006	0.420 ± 0.005
0.447 ± 0.011	0.463 ± 0.009	0.386 ± 0.010	0.409 ± 0.008
AE+Kmeans batch			
0.471	0.485	0.446	0.459
0.461 ± 0.008	0.474 ± 0.008	0.418 ± 0.022	0.451 ± 0.006
0.447 ± 0.011	0.464 ± 0.008	0.387 ± 0.025	0.426 ± 0.019
AE+α–Kmeans			
0.474	0.542	0.451	0.621
0.463 ± 0.008	0.538 ± 0.005	0.429 ± 0.015	0.620 ± 0.000
0.448 ± 0.012	0.527 ± 0.008	0.392 ± 0.028	0.600 ± 0.021
AE+mα–Kmeans			
0.474	0.556	**0.457**	0.667
0.463 ± 0.009	0.555 ± 0.001	0.431 ± 0.020	0.664 ± 0.003
0.448 ± 0.012	0.545 ± 0.009	0.392 ± 0.030	0.646 ± 0.017
AE+cα–Kmeans			
0.494	0.544	0.451	0.640
0.468 ± 0.018	0.538 ± 0.005	0.429 ± 0.015	0.622 ± 0.015
0.449 ± 0.012	0.527 ± 0.008	0.392 ± 0.000	0.600 ± 0.021
AE+mcα–Kmeans			
0.495	0.558	**0.457**	0.688
0.468 ± 0.006	0.556 ± 0.002	**0.432 ± 0.014**	0.681 ± 0.008
0.449 ± 0.012	0.545 ± 0.009	0.394 ± 0.029	0.661 ± 0.019
Deep α–Kmeans^p			
0.405	0.607	0.436	0.751
0.391 ± 0.008	0.592 ± 0.011	0.429 ± 0.008	0.740 ± 0.012
0.373 ± 0.016	0.576 ± 0.014	0.391 ± 0.031	0.701 ± 0.042
Deep $m\alpha$–Kmeans^p			
0.405	**0.922**	0.442	**0.923**
0.392 ± 0.012	**0.727 ± 0.003**	0.428 ± 0.007	**0.828 ± 0.000**
0.374 ± 0.017	**0.712 ± 0.011**	**0.398 ± 0.021**	**0.811 ± 0.020**

Table 4: Clustering results on the En-Ger and Fr-Ru collections (higher is better). Each cell contains the best run as well as the average of the 3 and 10 best runs (with std deviation).

ARI and type t_3, and with Deep mα–Kmeans^p for O-FRI and type t_2. We suspect that for ARI this is due to a concentration problem (Fig. 1). In almost all models for O-FRI except Deep mα–Kmeans^p the results for type t_1 are better than for type t_2 and for type t_2 better than for type t_3. For ARI this is only true in half of the cases, namely Sklearn, Kmeans batch, cα–Kmeans, and $mc\alpha$-Kmean, but it is true that for all models it achieves better results for types t_1 and t_2 than for type t_3. These results highlight the difficulty of clustering comparable corpus, suggesting that it is more difficult than clustering monolingual corpus.

4.4. Discussion

Overall, the above results indicate that $mc\alpha$-Kmeans outperforms the other models on almost all collections, followed by Deep $m\alpha$-Kmeans^p and $c\alpha$-Kmeans. This provides a positive answer to the general question we addressed, namely whether or not dedicated clustering models, as $c\alpha$-Kmeans and $mc\alpha$-Kmeans, able to take into account the different cluster types inherent to comparable corpora can improve the clusters obtained. This also shows that it is important to somehow learn a representation adapted to a given collection, either through a joint approach as in Deep α-Kmeans or through the masking mechanism we introduced which is able to improve the joint learning of Deep α-Kmeans. This said, the main disadvantage of the masking mechanism compared to the joint learning is that it assumes that the original embedding space, obtained here with a pre-trained, shared auto-encoder, captures all the correct cluster representations as its only role is to prune, in a cluster dependent way, some of the dimensions of the original embedding space. This may not be true in practice and we believe that this explains why the joint approach clearly outperforms the other approaches on the En-Ger and Fr-Ru collections for O-FRI.

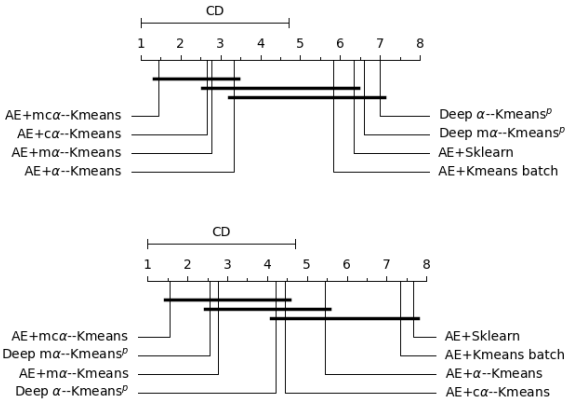

Figure 2: Critical difference diagrams of the mean ranks of the different models, in terms of ARI (top), and O-FRI (bottom). Horizontal bars denote lack of significant differences among models.

In regards to the complexity of the proposed models in comparison to the baseline models, one epoch of our complete model ($mc\alpha$-Kmeans) has a complexity of $3(C + K)$ where K denotes the number of clusters and C denote the complexity of one epoch of the baseline model with α-Kmeans. As $K << C$, the complete model is roughly three times slower than the α-Kmeans baseline, presuming that the number of epochs is the same for all models.

Another point concerns the generalization of the approach proposed to multilingual corpora, comprising documents in more than two languages. The main problem with such corpora is that the number of cluster types can be of the order 2^ℓ, where ℓ is the number of different languages, as all combinations of languages can yield different cluster types. A possible approach would be to select just a few cluster types based on the proximity of the different documents in the embedding space obtained by a pre-trained, shared auto-encoder.

	AE+Sklearn	AE+Kmeans batch	AE+α–Kmeans	AE+mα–Kmeans	AE+cα–Kmeans	AE+mcα–Kmeans	Deep α–Kmeans^p	Deep mα–Kmeans^p
				ARI				
t_1	0.428 ± 0.162	0.436 ± 0.160	0.447 ± 0.167	0.459 ± 0.171	0.487 ± 0.190	**0.505 ± 0.185**	0.441 ± 0.235	0.448 ± 0.230
t_2	0.426 ± 0.115	0.430 ± 0.115	0.457 ± 0.112	0.466 ± 0.101	0.470 ± 0.113	**0.478 ± 0.101**	0.459 ± 0.117	0.461 ± 0.114
t_3	0.304 ± 0.173	0.311 ± 0.175	**0.348 ± 0.164**	**0.348 ± 0.166**	**0.348 ± 0.164**	**0.348 ± 0.166**	0.312 ± 0.174	0.313 ± 0.176
				O-FRI				
t_1	0.619 ± 0.103	0.624 ± 0.101	0.795 ± 0.071	0.820 ± 0.063	0.816 ± 0.079	**0.838 ± 0.069**	0.797 ± 0.101	0.801 ± 0.109
t_2	0.491 ± 0.106	0.495 ± 0.107	0.743 ± 0.086	0.808 ± 0.098	0.758 ± 0.087	**0.814 ± 0.098**	0.780 ± 0.089	**0.814 ± 0.080**
t_3	0.448 ± 0.131	0.454 ± 0.132	0.685 ± 0.116	0.753 ± 0.108	0.697 ± 0.118	**0.758 ± 0.110**	0.700 ± 0.101	0.730 ± 0.099

Table 5: Clustering results by cluster type in terms of ARI and O-FRI (higher is better). Each cell contains the average of 10 runs over 9 En-Fr collections (with std deviation).

Lastly, we also assessed to which extent the different models are sensitive to the setting of their hyperparameters. Figure 3 illustrates this for $mc\alpha$-Kmeans for O-FRI and η. As once can note, there exists an important interval in which all η values provide good models. This is true for all the hyperparameters of $mc\alpha$-Kmeans (but not other models) for O-FRI, and to a lesser extent for ARI.

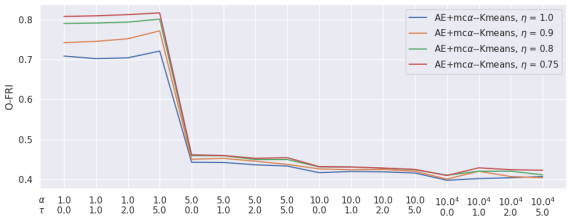

Figure 3: Sensitivity of $mc\alpha$-Kmeans (for O-FRI) wrt the hyperparameter η averaged over the 9 En-Fr collections.

5. Conclusion

We have revisited in this paper the problem of clustering bilingual comparable corpora, building upon the fact that comparable corpora can yield different types of clusters. We have designed a new model, based on α-Kmeans, to take into account this fact, and have shown how one can further adapt, through a masking mechanism, input representations to obtain cluster dependent representations for both documents and cluster representatives. This last point is important as there is in general no guarantee that a given representation is well adapted to all clusters of a given collection, or even to the clustering task. We further developed a suite of tools to create, from Wikipedia, new bilingual comparable corpora with ground truth clusters and different distributions over the cluster types and clusters in a given type. Our results, on 9 En-Fr, one En-Ger and one Fr-Ru collections, indicate that the model we proposed outperforms all the other models on almost all collections.

Our future work will focus on trying to come up with a formulation of the problem that can lead to a true joint learning of the representation and clustering of the documents. This is however not easy as there one needs to estimate, for each document, the probability that it belongs to a particular cluster type. We will also consider the problem of clustering multilingual comparable corpora (with more than two languages) through an appropriate selection of the possible cluster types.

Ethical considerations Our study aims to propose and evaluate models for clustering bilingual comparable corpora and our work is mostly methodological. In particular, the developments of our models and tools raised no ethical concerns. It is nevertheless possible, and beyond our control, that the clusters obtained by the models and tools proposed be used for purposes which may not be entirely ethical.

6. Acknowledgements

This work has been funded by the French projects ANR-20-IDES-0005 IDéES@UGA and ANR-19-P3IA-0003 MIAI@Grenoble Alpes.

7. Bibliographical References

Derek T Anderson, James C Bezdek, Mihail Popescu, and James M Keller. 2010. Comparing fuzzy, probabilistic, and possibilistic partitions. *IEEE Transactions on Fuzzy Systems*, 18(5):906–918.

Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. Optics: Ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2):49–60.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022.

Roelof K Brouwer. 2009. Extending the rand, adjusted rand and jaccard indices to fuzzy partitions. *Journal of Intelligent Information Systems*, 32:213–235.

Ricardo JGB Campello. 2007. A fuzzy extension of the rand index and other related indexes for clustering and classification assessment. *Pattern Recognition Letters*, 28(7):833–841.

Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. 2013. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer.

Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, 7:1–30.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231.

Milton Friedman. 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701.

Hichem Frigui, Cheul Hwang, and Frank Chung-Hoon Rhee. 2007. Clustering and aggregation of relational data with applications to image database categorization. *Pattern Recognition*, 40(11):3053–3068.

Nils Lid Hjort, Chris Holmes, Peter Müller, and Stephen G Walker. 2010. *Bayesian nonparametrics*, volume 28. Cambridge University Press.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification*, 2:193–218.

Eyke Hullermeier and Maria Rifqi. 2009. A fuzzy variant of the rand index for comparing clustering structures. In *Joint 2009 International Fuzzy Systems Association World Congress and 2009 European Society of Fuzzy Logic and Technology Conference, IFSA-EUSFLAT 2009*, pages 1294–1298.

Eyke Hullermeier, Maria Rifqi, Sascha Henzgen, and Robin Senge. 2011. Comparing fuzzy partitions: A generalization of the rand index and related measures. *IEEE Transactions on Fuzzy Systems*, 20(3):546–556.

James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

Maziar Moradi Fard, Thibaut Thonet, and Eric Gaussier. 2020. Deep k-means: Jointly clustering with k-means and learning representations. *Pattern Recognition Letters*, 138:185–192.

Peter Bjorn Nemenyi. 1963. *Distribution-free multiple comparisons.* Princeton University.

William M Rand. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850.

Douglas A Reynolds et al. 2009. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663).

Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2009. Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 1073–1080, New York, NY, USA. Association for Computing Machinery.