

ORIGINAL ARTICLE OPEN ACCESS

New Heuristics for Stable LDA Parameter Search

Simon-Olivier Harel¹  | Erick Velazquez-Godinez² | Sylvie Ratté¹

¹Department of Software and Information Technology Engineering, École de technologie supérieure (ÉTS), Montreal, Quebec, Canada | ²Omnimed, Cookshire-Eaton, Quebec, Canada

Correspondence: Simon-Olivier Harel (simonolivierharel@gmail.com)

Received: 4 November 2023 | **Revised:** 17 December 2025 | **Accepted:** 2 April 2026

Keywords: latent Dirichlet allocation | parameter optimization | replication | stability | topic modeling

ABSTRACT

The Latent Dirichlet Allocation (LDA) algorithm automatically extracts latent topics from a textual corpus, but configuring its parameters can be difficult and time-consuming. Optimization algorithms can help determine the best parameters, but not necessarily the optimal ones. This research proposes a framework that estimates near-optimal parameters for the corpus under analysis and provides a way to justify the final model selection. To achieve this goal, we evaluated different combinations of fitness functions (traditional and novel) to guide the Genetic Algorithms (GA) to identify the quasi-optimal LDA parameters. A stability analysis is conducted to evaluate the quality of the parameters. The results show that different combinations of GA and fitness function can identify the expected number of latent topics and extract more distinct topics than traditional ones. However, using only combinations of different GA and fitness functions does not automatically produce the most stable results. We show that, instead of blindly relying on metaheuristic procedures, a thorough stability analysis is necessary to evaluate the quality of the parameters and obtain an optimal model. Finally, the choice of the fitness function should be based on the parameter tuning runtime and its ability to discover latent topics.

1 | Introduction

Making sense of or finding information in unstructured data, such as textual data, is challenging [1]. Finding, categorizing, and classifying the information in a meaningful way remains difficult as long as no structure is imposed. To see information emerge from textual data, we can rely on Natural Language Processing (NLP) techniques such as topic modeling to achieve this task. One of the standard techniques for finding related topics within texts is the Latent Dirichlet Allocation (LDA) algorithm [2]. Even in this era of deep learning techniques, the use of this probabilistic model remains relevant in the context of statistical analysis of a corpus due to its interpretability and grounded generative process [3].

However, the general problem when using LDA is selecting the values of the parameters $[K, \alpha, \beta, N]$ when applying it to a specific corpus in a situation where no universal parameterization exists [4–6]. Many heuristics have been proposed to find the nearest optimal parameters for a given task. However, most of them focus on the number of topics, denoted by K , while using fixed values (implementation related) for α and β [7, 8] or on fixing K while estimating α and β [9]. Researchers have recently proposed techniques based on metaheuristics to tune all the LDA parameters [4, 6, 10, 11]. Due to the very nature of this algorithm, researchers have pointed out the lack of stability that can lead to systematic errors and the difficulty of replicating results without proper parameters.

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2026 The Author(s). *Computational Intelligence* published by Wiley Periodicals LLC.

The motivation of our research is 2-fold. First, we want to continue discussing finding the nearest optimal LDA parameters started by Panichella [6] and Agrawal et al. [4]. Second, we propose a framework to find such optimal parameters. Our approach will focus on testing additional surrogate metrics used to calibrate LDA in an unsupervised fashion and on validating the parameters found by evaluating their stability based on replicated runs, word embedding, and clustering techniques [6, 8]. More precisely, we use the Genetic Algorithms (GA) combined with five different fitness functions (Silhouette, Coherence, R-square, Raw score, 2PJ) to determine a set of LDA parameters. We then evaluate the quality of the obtained parameters through a stability analysis based on replicated runs, convert the set of topics with word embedding to cluster them, and then quantify this analysis by a stability metric, which we will define as the *final score*.

Finally, we validate our proposed methodology and show the importance of the stability analysis with the following questions:

- Q1.** Do the different combinations of metaheuristic and fitness functions give similar results?
- Q2.** Are the run id of the maximum value of the fitness function and the run id of the maximum value of the stability analysis the same?
- Q3.** How do we explain the relation between a low number of topics and a high final score, and vice-versa?
- Q4.** Does hierarchical clustering find similar latent topics as Silhouette and Coherence?
- Q5.** Is it more advantageous to train a new GloVe than to use a pretrained GloVe?
- Q6.** How do we select the final parameters?

This paper is structured as follows. We start with a review of the literature in Section 2, discussing essential theoretical concepts related to our research. Section 3 introduces the methodology, data sets, and evaluation method. Then, in Section 4, we present and discuss the results of our experiments. In the last section, we state our conclusions and future work.

2 | Literature Review

2.1 | Latent Dirichlet Allocation

The Latent Dirichlet Allocation (LDA) algorithm was first introduced by Blei et al. [2]. LDA is a generative and probabilistic technique in the topic modeling branch of the Natural Language Processing (NLP) discipline. It seeks to capture the heterogeneity of ideas (latent semantic structure) prevailing in a set of documents (corpus). The corpus must be transformed into a document-term matrix (DTM) in which the frequencies of words in documents are counted. This process results in a matrix with dimensions $D \times V$, where D represents the document set in the corpus and V the vocabulary set. This transformation assumes that the words in a document are exchangeable, so their order is unimportant. LDA aims to model documents as a discrete distribution over K

latent topics, where every topic is modeled as a discrete distribution over the terms constituting the DTM (vocabulary).

Referring to the LDA graphical model in Figure 1, the nodes represent the random variables and the edges represent the dependence between those random variables. The shaded node represents the observed random variable (i.e., the words within the documents), the unshaded nodes represent the hidden random variables, and the rectangular boxes indicate replication. Specifically, the underlying latent semantic structure is expressed through the topic-word distribution ϕ , the per-document topic proportion θ , and the per-word topic assignments z . However, ϕ , θ and z are unobserved, and this algorithm aims to determine them from the observed variables. Note that this representation is called the smoothed LDA model. In our context, we prefer the smoothed LDA model instead of the original formulation, whose weakness is that it does not place a priori on each ϕ_k . By this modification, it can be qualified as a fully Bayesian model.

Because the LDA structure allows the observed variable to interact with the hidden variables (see Figure 1), the latent variables' posterior distribution can be inferred, given the observed documents. This interaction is exhibited by the generative process, in which the random process is assumed to have produced the observed data.

The generative process is described as follows [2]:

1. For every topic $k = \{1, \dots, K\}$,
 - a. Draw a distribution over words $\phi_k \sim \text{Dir}(\beta)$
2. For each document d in D ,
 - a. Draw a vector of topic proportions $\theta_d \sim \text{Dir}(\alpha)$ (i.e., per-document topic proportion)
 - b. For each word i within document d ,
 - i. Draw a topic assignment $z_{d,i} \sim \text{Mult}(\theta_d)$, where $z_{d,i} \in \{1, \dots, K\}$ (i.e., per-word topic assignment)
 - ii. Draw a word $w_{d,i} \sim \text{Mult}(\phi_{z_{d,i}})$, where $w_{d,i} \in \{1, \dots, V\}$

Finally, the goal of this generative process is to estimate the following two matrices:

1. Θ : Representing the distribution of topics over documents
 - $K \times D$ matrix, where the generic entry $\Theta(i, j)$ represents the probability of the j -th document being relevant to the i -th topic.
2. Φ : Representing the distribution of words over topics
 - $V \times K$ matrix, where the generic entry $\Phi(i, j)$ represents the probability of the i -th word belonging to the j -th topic.

2.1.1 | LDA Inference

Multiple mathematical methods approximate the intractable integral arising in a Bayesian inference. The two common techniques to infer LDA for a given corpus are variational expectation-maximization (VEM) [12] and Gibbs sampling [13].

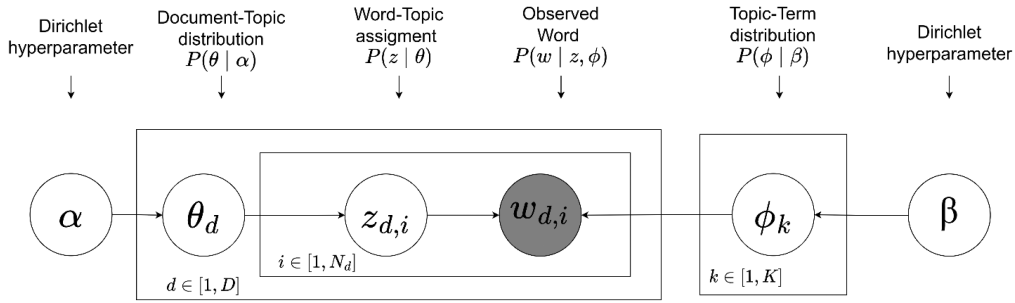


FIGURE 1 | Smoothed LDA in plates notation (adapted from Blei et al. [2]).

In this paper, we will rely on the Gibbs sampling proposed by Panichella [6] to estimate the θ and ϕ distributions. Panichella [6] justifies that choice because prior studies used it and showed it has a faster and better convergence towards the global optimum than VEM.

2.1.2 | The Four LDA Parameters to Set

When using LDA and Gibbs sampling, the practitioner must set:

- K : the number of topics to generate from the corpus.
- α : the hyperparameter that influences the document-topic density.
 - For a symmetric distribution, with a higher α , the documents will comprise more topics, and with a lower α , the documents will contain fewer topics.
 - For an asymmetric distribution, a higher α results in a more specific topic distribution per document.
- β : the hyperparameter that influences the topic-word density.
 - For a symmetric distribution, with a higher β , topics will comprise most of the words in the corpus, and with a lower β , they will consist of a few words.
 - For an asymmetric distribution, the β will result in a more specific word distribution per topic.
- N : the number of Gibbs iterations. This parameter is specific to the Gibbs sampling generative model.

Regarding the parameters α and β , typically, LDA implementations use symmetric Dirichlet priors over Θ and Φ with fixed concentration parameters α and β , respectively. Therefore, there is an implicit assumption that they have little practical effect. However, those priors may either be symmetric or asymmetric. A symmetric distribution refers to one which is fixed to a uniform distribution. Conversely, an asymmetric distribution is not fixed to a uniform distribution. Wallach et al. [14] tested the four combinations of symmetric and asymmetric concentration parameters combinations. They found that an asymmetric α has substantial advantages over a symmetric prior, while an asymmetric β provides no real benefit [14].

2.2 | LDA Instabilities

Model instability refers to the variability in the results when the same model is repeatedly fitted with identical data and

parameters. In the case of LDA, its probabilistic nature may lead to such variability, categorizing it as an unstable model. Researchers classify this type of instability as type-I [4, 5]. Conversely, a stable model reliably reproduces consistent results, enhancing the reproducibility of findings. To mitigate LDA's type-I instability, studies [4, 5, 15–17] propose various strategies, such as setting a consistent random seed to control the stochastic nature of the Gibbs sampling method, performing multiple Gibbs sampling restarts to avoid local optima, or running the model multiple times and aggregating the results (topic-terms matrix) of separate runs using clustering.

Additionally, Agrawal et al. [4] highlighted the phenomenon of type-II instability, who observe that different document orderings could lead to different topics when training LDA. To address this, they recommend to run LDA several times, each with a different ordering of the documents. The consistency of the topics extracted across these multiple runs can then be used as a metric (Raw score) to evaluate the stability of the LDA configuration. This method (Section 2.4.3) not only mitigates the effects of document ordering on topic generation but also helps in fine-tuning the LDA model using meta-heuristics to achieve a more stable modelisation.

In our study, we ran LDA numerous times and shuffled the document ordering before each LDA generation to consider these two types of instability.

Further details on our methodology are outlined in Section 3.

2.3 | Estimating LDA Parameters Based on Optimization Methods

In Panichella [6], his objective was to shed light on the impact of metaheuristic and fitness function combinations on the software engineering task. It should be noted that a software engineering task is a small corpus document containing a small amount of information (words), which is very specific. The authors compare the performance of seven metaheuristics namely Genetic Algorithms (GAs), Differential Evolution (DE), Particle Swarm Optimization (PSO), Simulated Annealing (SA), Random Search (Ran), Covariance Matrix Adaptation Evolution Strategy (CMA-ES), and Stochastic Hill Climbing (SHC) alongside three fitness functions (Silhouette, Coherence and Raw score). This comparison is based on their effectiveness in identifying duplicate bug reports and the time required to tune LDA.

TABLE 1 | Authors who used metaheuristic search to optimize LDA hyperparameters.

Author	Metaheuristic	Fitness function	Data set
Yarnguy and Kanarkard [11]	Ant colony optimization	Perplexity	UCI (KOS, NIPS, ENRON)
Agrawal et al. [4]	Differential evolution (DE) Genetic algorithms (GA)	Raw score Silhouette	Pits, citemap, Stackoverflow, software document
Panichella et al. [10]	Genetic algorithms (GA)	Silhouette	Bench4BL (software document)
Panichella [6]	Genetic algorithms (GA) Differential evolution (DE) Random search (RS) Simulated Annealing (SA) Particle Swarm Optimization (PSO) Stochastic Hill Climbing (SHC) CMA-ES	Silhouette Coherence Raw score	Bench4BL (software document)

They also keep the default metaheuristic parameters. Their results suggest no clear winner among the combinations of the different search-based approaches (GA, SA, SCH) and the Silhouette or Coherence fitness functions when tuning LDA. They observe that multiple metaheuristics could achieve equally good performances. However, using the Raw score fitness function leads to more calculation time before convergence. Also, they do not recommend using the CMA-ES, RS, PSO, or DE as these metaheuristics achieve significantly less accurate results.

On the other hand, Agrawal et al. [4] compare the combination of GA with Silhouette [10] to the combination of DE with Raw score, and in their case, DE with Raw score performs better. Moreover, they argue that Panichella et al. [10] did not consider the instability of type II, which could have led to the instability of LDA, when they proposed their method of LDA-GA with Silhouette.

This discrepancy between these researchers could be explained by the fact that each has a different research objective from an optimization point of view. Panichella et al. [10] use a fitness function based on the document-topic matrix (Silhouette), while Agrawal et al. [4] created their measure (Raw score), which uses the top nine words in the topic-word matrix.

This parameter optimization calculated by the metaheuristic procedures could lead to the belief that the resulting model is optimal. However, to validate the optimality of the model, Mantyla et al. [16] propose a complementary method to evaluate the type-I stability of the final model. Their approach is further described in Section 2.5. The following table presents studies that apply metaheuristics to optimize LDA parameters in an unsupervised manner. In Table 1, we present some of the research conducted. As we can see, the different authors do not all use the same data set to discuss the quality of the proposed solution (a combination of a metaheuristic and a fitness function).

2.4 | Fitness Functions

The combination of metaheuristic and fitness functions can guide the optimization process to find the nearest optimal parameters.

The following sections describe the fitness functions discussed in the present study.

2.4.1 | Topic Coherence

Several coherence measures have been proposed to evaluate the coherence of a set of words. We can mention two popular ones: the UCI measure proposed by Newman et al. [18] and the UMass measure proposed by Mimno et al. [19]. Both measures calculate the coherence of a topic as the sum of the pairwise distributional similarity for a top M most probable words in the topic. However, the main difference between those two measures is that the UCI measure calculates the word co-occurrence frequencies over an external corpus; meanwhile, the UMass calculates the co-occurrence over the corpus used to train the model. For this research, we will use the one proposed by Mimno et al. [19].

They defined:

- k as topic id
- $V^{(k)} = (v_1^{(k)}, \dots, v_M^{(k)})$ as a list of the M most probable words in topic k
- $D(v_l^{(k)})$ counts the number of documents containing $v_l^{(k)}$
- $D(v_m^{(k)}, v_l^{(k)})$ counts the number of documents containing $v_m^{(k)}$ and $v_l^{(k)}$

$$C(k; V^{(k)}) = \sum_{m=2}^M \sum_{l=1}^{m-1} \log \frac{D(v_m^{(k)}, v_l^{(k)}) + \epsilon}{D(v_l^{(k)})}$$

Originally, the ϵ (smoothing factor) was set to 1 to avoid taking the logarithm of zero, but Stevens et al. [20] found that the measure gives better results if this parameter is smaller. Because we rely on Matthew Denny's implementation for this measure,¹ the epsilon value is set to 1.

The overall coherence for an LDA model (with parameters $[K, \alpha, \beta, N]$) can then be computed as the arithmetic mean of the topic coherence scores of the K -generated topics. This final score takes a value between $-\infty, 0$ [where the value close to zero indicates a better score].

2.4.2 | Silhouette Coefficient

The silhouette coefficient for a cluster C is defined as follows:

$$s(d_i) = \frac{b(d_i) - a(d_i)}{\max(a(d_i), b(d_i))}$$

where:

- $s(d_i)$ denotes the silhouette coefficient for the document d_i in the corpus;
- $a(d_i)$ measures the maximum distance of the document d_i to the other documents in the same cluster (cluster cohesion);
- $b(d_i)$ measures the minimum distance of document d_i to another document in a different cluster (cluster separation).

The overall silhouette coefficient can be measured as the arithmetic mean of the coefficients $s(d_i)$ for all documents in the corpus [6, 10].

$$s(C) = \frac{1}{D} \sum_{i=1}^D s(d_i)$$

$s(C)$ takes a value between $[-1, +1]$, where a value close to 1 indicates a better score because (on average) the separation is larger than the cohesion of the clusters.

2.4.3 | Raw Score

Agrawal et al. [4] proposed an alternative surrogate metric, namely, the Raw score, to measure the quality of the LDA parameters. This method adapts the Jaccard measure to evaluate cross-run topic similarity. Specifically, they measure the stability of the topic model by calculating the median number of overlaps among the top nine words, denoted by the input o in Algorithm 1. To achieve this, they execute the LDA computation 10 times ($n=10$) on randomly shuffled data documents, formatted in tf-idf.² Subsequently, they calculate the overlap score of the aggregated topics from these runs. To minimize sampling bias, the procedure is repeated n times, allowing for the computation of the median of these 10 overlap scores. The pseudocode detailing these steps is provided in Algorithm 1.

The Raw score takes a value between $[0, 1]$, where a value close to 1 indicates a better score.

2.4.4 | Perplexity

We will not use the perplexity measure, despite its widespread use in information theory [11, 21, 22]. Perplexity measures how likely a model is to predict a held-out test set; however, accurately

ALGORITHM 1 | Raw Score Algorithm.

Input: $n, o, k, \alpha, \beta, Data$
Output: Raw Score
 $Score \leftarrow \emptyset$;
for $j \leftarrow 0$ **to** $n = 10$ **do**
 $Topics \leftarrow \emptyset$;
 for $i \leftarrow 0$ **to** $n = 10$ **do**
 $data \leftarrow \text{shuffle}(Data)$;
 $Topics.add(\text{lda}(k, \alpha, \beta, data))$;
 end
 $Score.add(\text{Overlap}(Topics, o, k))$;
end
 $RawScore \leftarrow \text{median}(Score)$;
return $RawScore$;

calculating the likelihood for an individual document is intractable. This limitation has led to the development of various sampling methods to estimate this probability [22]. Additionally, Chang et al. [23] have shown that perplexity does not correlate strongly with human judgment, indicating that the topics generated are often not effective at representing coherent ideas. For these reasons, we will not employ this measure.

2.4.5 | R-Square

Jones [24] proposes adapting the R-squared goodness-of-fit measure to topic models while keeping the same interpretation as the one used in a broader class of statistical models. For example, when applied to a linear model, this measure explains the proportion of variability in the data; the results are bound between 0 and 1, where a higher value indicates a better model fit.

The adaptation of R-square to topic modeling follows the same principle, and it is mathematically described as follows:

$$f_d = E(w_d) = n_d \cdot (\theta_d \cdot \Phi)$$

$$\bar{w} = \frac{1}{D} \sum_{d=1}^D w_d$$

$$SS_{tot} = \sum_{d=1}^D \text{dist}(w_d, \bar{w})^2$$

$$SS_{resid} = \sum_{d=1}^D \text{dist}(w_d, f_d)^2$$

$$R^2 \equiv 1 - \frac{SS_{resid}}{SS_{tot}}$$

where f_d is the document's fitted value representing the expected token counts under the topic model; n_d represents the total number of token draws (i.e., the documents length); Multiplying by n_d scales this probability distribution to expected counts matching the document length; $\text{dist}()$ is the Euclidian distance; SS_{tot} is the total sum of squares; SS_{resid} is the residual sum of squares.

We rely on the Tommy Jones implementation of R-square.³

2.5 | Measuring the Stability of LDA Topics Based on Replicated Runs

Recent research on LDA stability states that a high stability value in running LDA multiple times with fixed parameters means many topics with representatives in each LDA run can be identified. Although LDA is run multiple times with fixed parameters, its results can still be unstable due to different seeds and random restart strategies.

Panichella [6] and Panichella et al. [10] do not perform independent runs of LDA with the same parameters found by the optimization process. Rather, they run the metaheuristic/fitness function combination 30 and 100 times in their first and second papers, respectively. They use the resulting LDA parameters at the end of each run to generate the corresponding LDA model. Since they address the stability problem using the same seed and random restart strategies, their method ensures that the same results are obtained even when LDA is rerun several times with the same respective parameters.

For Agrawal et al. [4], the replicated runs process is incorporated into the fitness function they developed, namely, the Raw score defined in Section 2.4.3. Therefore, stability is implicitly addressed by the fitness function.

Mantyla et al. [16] propose a complementary method to evaluate the type-I stability of the final model. Their approach to providing information about the topic's stability must be used at the end of the optimization process. They perform 20 replicated runs of LDA with fixed parameters. Then, they extract the top 10 most probable words of each topic. These will be used to cluster the words from these 20 topic-word matrices via the k-medoid algorithm using the word embedding produced by the GloVe algorithm built on their corpus. Finally, they evaluate the similarity of the clustered topics with the suggested Rank Biased Overlap (RBO) measure [25]. The advantage of this method is that it offers the possibility of further investigation of the clusters as needed. A shortfall in their approach is that they do not consider type-II instability. Although RBO seems useful thanks to its implementation of a more flexible form of a Jaccard coefficient (since it considers the word ranking), it requires parameter optimization.

Rieger et al. [17] propose to evaluate the stability of LDA with clustering techniques applied to replicate LDA runs. They introduce a new similarity measure, S-CLOP (Similarity of multiple sets by Clustering with Local Pruning), based on a pruning algorithm applied to the hierarchical clustering results with complete linkage. Starting with a broad set of topics from all the LDA runs (replications), the ideal goal will be that the pruning algorithm results in clusters, each containing one topic for every replication of the modeling procedure. This increase in reliability is achieved by running the LDA several times (they suggest 50 times).

Finally, this concludes the related work that will help elaborate our methodological solution.

3 | Methodology and Implementation

The implementation of our experiment relies on R packages. We base our study on the previous works of Panichella [6] and Mantyla et al. [16], which can be found respectively at <http://doi.org/10.5281/zenodo.4016590> and <https://github.com/M3SOulu/Measuring-LDA-Topi>.

c-Stability. We use and adapt their code to our study needs. The R scripts used in our experiment are publicly available at <https://github.com/simonOHa/New-Heuristics-for-stable-LDA-parameter-search>.

The methodology of this research can be broken down into three independent steps:

1. The first step consists in preprocessing the corpus to build the document term matrix (DTM).
2. The second step focuses on finding the nearest optimal LDA parameters.
3. The third step focuses on evaluating the stability of the LDA model generated by the previously found parameters.

These phases are illustrated in the following figure, which outlines the full implementation pipeline (Figure 2).

The process begins with a raw textual corpus and applies a genetic algorithm to identify the optimal LDA configuration, including the number of topics (K), Dirichlet priors (α , β), and the number of Gibbs sampling iterations (N).

Once the optimal parameters are found, multiple topic models are generated by permuting the document order in the corpus. Each model produces a topic-term matrix (ST). To evaluate the semantic cohesion of the topics, we project each ST matrix into a vector space using a pretrained GloVe embedding matrix (G), producing topic-vector representations ($SV = ST \times G$).

We then cluster these topic vectors using the k -medoids algorithm and quantify topic stability with a custom score. This score aggregates internal and external clustering validation metrics, providing a numeric estimate of the consistency of topics across multiple model runs.

3.1 | Corpus Preprocessing

The corpus preprocessing aims to extract the relevant words from each document constituting the corpus, eliminating any potential noise. For evaluation purposes, we used four different corpora published⁴ by the researcher Greene et al. [8], which are presented in table 2.

The corpora based on 20 Newsgroups are artificially constructed to allow smaller documents and human-annotated “ground truths” by the topical categories. The annotation was done to derive smaller data sets for which the correct value of the number

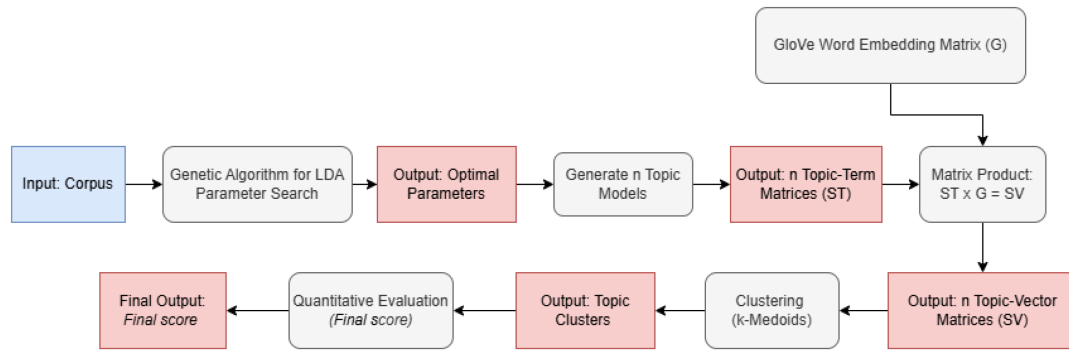


FIGURE 2 | Topic modeling pipeline with LDA optimization, embedding enrichment, and cluster stability evaluation.

TABLE 2 | Corpora information.

Corpus	DTM dimensions		Description
	Documents	Terms	
20 Newsgroups-ol-6-1	2191	8499	Natural classes (topics): 6 Document topical categories and counting: crypt (991), electronics (240), med (240), politics (240), religion (240), space (240) <i>o</i> * - * - *: data set with clusters that overlap considerably. <i>*I</i> - * - *: data set with unbalanced clusters where one cluster contains 60% of the documents.
20 Newsgroups-sb-6-1	3000	10,673	Natural classes (topics): 6 Document topical categories and counting: autos (500), Christian (500), crypt (500), forsale (500), hockey (500), windows (500) <i>s</i> * - * - *: data set with reasonably compact and well-separated clusters. <i>*b</i> - * - *: data set with balanced clusters containing 500 documents each.
BBC-Sport	737	384 8	Natural classes (topics): 5 Document topical categories and counting: athletics (101), cricket (124), football (265), rugby (147), tennis (100)
BBC-News	2225	8850	Natural classes (topics): 5 Document topical categories and counting: business (510), entertainment (386), politics (417), sport (511), tech (401)

of topics (K) is known. The corpora from BBC were studied by D. Greene in these research works [8, 26] and concern the news and sports sections. Using these corpora is advantageous as they offer a research result benchmark for comparison with our methodology results.

Regarding text preprocessing, we used the final DTM proposed by Greene et al. [8] for the two corpora related to 20 Newsgroups, where stop-word removal, stemming, and terms occurring in less than three documents were eliminated. We used the same steps for BBC, but here, we applied lemmatization instead of the stemming approach. This modification was possible because we had access to the two original corpora. Using lemmatization instead of stemming is motivated by the word embedding technique in the stability step. The resulting preprocessed documents were then converted into a DTM. We used the term frequency (tf) scheme instead of the term frequency with the inverse document frequency (tf-idf) scheme applied by Agrawal et al. [4] and

Panichella [6]. The LDA is a word generation model that implies a word is formed from a multinomial distribution. In the LDA context, we believe using the term frequencies (tf) instead of the inverse document frequency (tf-idf) makes more sense. It makes no sense to suggest that, for example, 32% of a word (tf-idf weight) is created by some distribution. However, for another context (e.g., in a search-engine context), it could make more sense to use the inverse document frequency (tf-idf).

3.2 | Finding the Nearest Optimal Parameters

This section will outline the methodology for identifying the quasi-optimal LDA parameter using various combinations of GA and fitness functions, including previously discussed metrics such as Silhouette, Coherence, Raw score, and R-square. Additionally, we introduce a new fitness function named 2PJ, which will be detailed subsequently. Following this introduction, we will discuss the methodology and parameter selection

for executing these GA-fitness function combinations on our selected corpora.

3.2.1 | Proposal for a New Fitness Function: Phi-Phi-Prime-Mean-Jaccard (2PJ)

Panichella [6] highlights the need for more refined surrogate metrics (fitness functions) to guide the unsupervised calibration and evaluation of LDA models. In response, we propose a novel metric: Phi-Phi-Prime-mean-Jaccard (2PJ). This score is designed to quantify the semantic alignment between the topic-word distribution (Φ) and the word-topic distribution (Φ') learned by LDA, thus capturing the bidirectional relationship between topics and words.

3.2.1.1 | Definition of the 2PJ Score. Given the topic-word distribution matrix Φ , where each entry represents the conditional probability $P(w_{d,i}|k)$, we define the corresponding word-topic distribution Φ' using Bayes' theorem:

$$\Phi' = P(k|w_{d,i}) = \frac{P(w_{d,i}|k) \cdot P(k)}{P(w_{d,i})}$$

Assuming that all documents in the corpus are equally probable, that is, $P(d_i) = \frac{1}{D}$, we compute:

$$P(k) = \sum_{i=1}^D P(k|d_i) \cdot P(d_i) \text{ and } P(w_{d,i}) = \sum_{k=1}^K P(w_{d,i}|k) \cdot P(k)$$

Once Φ' is derived, we identify the top M words for each topic from both Φ and the transpose of Φ' (denoted Φ'^T). For each topic, we compute the Jaccard similarity between these two sets of top words. Averaging over all K topics gives us the 2PJ score:

$$2PJ = \frac{1}{K} \sum_{k=1}^K \text{Jaccard}(\text{Top}_M(\phi_k), \text{Top}_M(\phi'_k))$$

The resulting 2PJ score lies in the range $[0, 1]$, where higher values indicate greater alignment between Φ and Φ' – and thus, better topic model quality.

3.2.1.2 | Example. Consider a machine learning corpus with $K = 3$ topics and $M = 3$ top words. From the topic-word distribution Φ , suppose Topic 1 has top words [*algorithm*, *data*, *model*]. Now, examining the word-topic distribution Φ' , we might find:

- Word “*algorithm*” has highest probabilities for [Topic 1, Topic 3, Topic 2]
- Word “*data*” has highest probabilities for [Topic 1, Topic 2, Topic 3]
- Word “*learning*” has highest probabilities for [Topic 1, Topic 3, Topic 2]

From Φ'^T , Topic 1's top-3 words (those most indicative of Topic 1) would be [*algorithm*, *data*, *learning*]. Comparing the two perspectives:

- **Forward (Φ):** Topic 1 $\rightarrow$ [*algorithm*, *data*, *model*]
- **Reverse (Φ'):** Topic 1 $\leftarrow$ [*algorithm*, *data*, *learning*]

The Jaccard similarity would be $\frac{2}{4} = 0.5$ (2 shared words: *algorithm*, *data* out of 4 unique words total: *algorithm*, *data*, *model*, *learning*). This bidirectional consistency check ensures that Topic 1's most representative words are also the words most indicative of Topic 1, providing a more robust assessment of topic quality than unidirectional measures.

3.2.1.3 | Motivation and Benefits. The primary advantage of the 2PJ score is its bidirectional assessment of the semantic relationships encoded in LDA. Whereas traditional metrics like Coherence evaluate word co-occurrence patterns within documents without directly leveraging the learned topic-word probability distributions, and Silhouette ignores the probabilistic nature of topic models entirely, 2PJ explicitly models both the generation and inversion of topics, providing:

- A comprehensive representation of the semantic consistency between topics and words. By comparing $P(w|k)$ and $P(k|w)$ perspectives, 2PJ identifies topics where the most representative words are also the most discriminative words, ensuring robust topic-word associations that work in both directions.
- Improved interpretability by ensuring that a topic's most probable words align with the words most indicative of that topic. This bidirectional validation helps detect and penalize topics that may have high-probability words that are not uniquely characteristic of that topic (e.g., words that appear with high probability across multiple topics).
- Direct utilization of learned probability distributions. Unlike Coherence, which measures document-level co-occurrence frequencies of top topic words, 2PJ operates directly on the topic-word and word-topic probability distributions learned during LDA inference, providing a more direct assessment of the model's internal semantic structure.
- Enhanced sensitivity to topic distinctiveness. 2PJ inherently favors topics with words that are both highly probable within the topic (from Φ) and uniquely associated with that topic (from Φ'), leading to more distinct and interpretable topic boundaries.
- Flexibility in selecting the parameter M , which defines the number of top terms used for similarity comparison and can be tailored to dataset size or domain specificity. Smaller M values focus on core topic terms, while larger M values capture broader semantic relationships.

Overall, 2PJ is an LDA-aware fitness function that addresses key limitations of existing metrics by providing a principled way to evaluate the bidirectional consistency of topic-word relationships. This makes it particularly valuable for applications requiring high topic interpretability and distinctiveness, and can be leveraged for more effective model evaluation and selection in unsupervised topic modeling scenarios.

3.2.2 | Methodology Related to the Run Execution of the Possible GA-Fitness Function Combinations

Following Panichella's suggestion (2021), we opted for GA instead of SHC or SA in our pursuit of the nearest optimal LDA parameters $[K, \alpha, \beta, N]$ through Gibbs sampling. While all methods produce equivalent results, GA offers faster convergence to the final solution.

We used the TextmineR library [27] for the LDA implementation because it allowed us to modify the symmetry assumption on prior distributions. As Wallach et al. [14] suggested, we forced α to be asymmetric and β to be symmetric. We experimented with the five following fitness functions to guide the GA in the optimization process. We use four previously discussed fitness functions (Silhouette, Coherence, Raw score, and R-square) and the new proposed one, 2PJ.

We executed several independent runs for each combination to allow a fair comparison between the possible combinations of GA-fitness functions. The number of LDA models generated during the optimization process is based on the design of the fitness functions. As a reminder, for n execution of the Raw score, n^2 LDA models will be generated, versus one for the other fitness functions. Given that ratio, we executed only one independent run for the GA-Raw score and 25 for the other combinations.

The followings are the Genetic Algorithms' parameter settings and fitness functions:

- For the GA, we used the same parameters as described in Panichella [6]:
 - population size of 10 LDA configurations;
 - number of GA iterations to 5;
 - binary simulated crossover with probability $p_c=0.9$;
 - polynomial mutation with mutation probability $p_m=0.25$; and
 - mutation index $\eta_m = 20$.
- For Silhouette, we use the Euclidean distance to compute the distance between documents in the topics space.
- For 2PJ, Coherence, and Raw score, we use the top 10 words.
- For Raw score, we use 7 overlap words and $n = 5$.
- For R-square, no parameters are needed.

In this step, we only address the type-II instability problem by applying a random shuffle on the DTM (documents order) before fitting the LDA model. This applies to all the fitness functions except for the Raw score. The two types of instabilities are implicitly considered for this latter due to their design.

Finally, that procedure is applied to all four data sets, with the LDA parameter boundaries set as follows (see table 3):

The range of topics to extract (K) was selected based on the number of ground truth labels in the corpora listed in Table 2, which fall within the range [5, 6]. Additionally, a maximum of 20 topics was chosen to strike a balance between granularity and comprehensiveness, ensuring that the extracted topics

TABLE 3 | List of LDA parameters and their ranges.

Parameter	Range	Description
K	[5, 20]	Number of topics to extract
α	[0.0000001, 1]	Distribution of topics over documents
β	[0.0000001, 1]	Distribution of topics over words
N	[200, 300]	Number of Gibbs iterations

capture a diverse range of themes without overwhelming the analysis with excessive detail. Recall from Section 2.1.2 that a higher α results in a more specific topic distribution per document. The α range is in accordance with the small number of topics identified by Greene et al. [8]. Since β represents the topic-word density, we preferred a lower β (i.e., few words per topic). We used the same Gibbs iterations (N) range as Panichella [6].

3.3 | Stability Evaluation

Following the identification of a set of LDA parameters, the application of stability analysis ensues. This analytical framework, drawing upon the work of Mantyla et al. [16], comprises four independent steps applied sequentially: model replication, integration of word embeddings, employment of word clustering technique, and quantitative evaluation of the resultant clusters.

Upon completion of this procedure, we derive a final metric, termed the *final score*, which serves as a criterion for selecting the quasi-optimal set of LDA parameters. Below, we elaborate on each sequential step of this methodological framework:

3.3.1 | Replication of the Model

Model replication involves generating the LDA model multiple times with the previously identified parameters and aggregating the topic-word matrices. Following the approach of Mantyla et al. [16], we replicate the model 20 times. To address both types of instability, random shuffling is applied to the DTM (type-II) before each model generation, while the 20 replications handle type-I instability.

3.3.2 | Word Embedding Integration

It has been shown that the vector space produced by a word embedding algorithm such as GloVe tends to cluster similar words [28]. Our idea is to take advantage of the potential synonymic and semantic similarity of the vector space generated by GloVe to validate the cohesion of the most significant terms constituting the different latent topics discovered [16] over the model replications.

We used the GloVe model trained on Common Crawl data offered by Stanford.⁵ We also created two new GloVe models trained on the BBC corpora (sport and news) separately and one other by combining the two corpora. The four different GloVe models are used to see if they impact the stability analysis. No preprocessing was applied to avoid losing semantic information in the BBC

corpora. Moreover, as the GloVe paper [28] suggested, Table 4 shows the used parameters.

Table 5 shows basic information about the dimensions of the vectors.

The stability analysis was performed using the four different GloVe models. The following table shows the percentage of each DTM vocabulary included in the terms constituting the GloVe vector representations.

The lower percentages associated with the 20 Newsgroups can be explained by the reduction of their vocabulary to their stem forms.

3.3.3 | Clustering LDA Topics

The word clustering technique is the process of clustering LDA topics derived from replicated runs. Each LDA replication generates a distinct set of topics (Φ), where Φ encapsulates the word distribution over the vocabulary (V) for each topic and each topic can be described as a list of words. By systematically repeating this process RN times, we obtain RN repeated sets of K topics. Consequently, the total number of topics (word lists) to be clustered, denoted as t , is the product of the number of replication runs (RN) and the number of topics (K), which can be expressed as $t = RN \times K$.

TABLE 4 | GloVe parameters.

Parameter	Value
Skip-grams window	10
Skip-grams window context	Symmetric
X max	100
Learning rate	0.05
Alpha	0.75
Lambda	0

TABLE 5 | GloVe basic information.

GloVe model	Total words	Dimension
GloVe-Common-crawl	2,196,018	300
GloVe-BBC-sport	14,158	300
GloVe-BBC-news	32,286	300
GloVe-BBC-sport-news	33,863	300

TABLE 6 | Terms intersection between GloVe and DTM (in %).

	GloVe-Common-Crawl	GloVe-BBC-Sport	GloVe-BBC-News	GloVe-BBC-Sport-News
20 Newsgroups-ol-6-1	72.44	23.92	37.91	38.41
20 Newsgroups-sb-6-1	71.20	20.63	33.69	34.31
BBC-Sport	92.83	96.10	92.13	98.67
BBC-News	95.40	54.80	96.96	97.08

To facilitate clustering, we harness GloVe word embeddings for our entire vocabulary, establishing a word vector space with v words and w weight vectors, depicted as a matrix G with dimensions ($v \times w$).

To prepare our topic-word matrix T of dimensions ($t \times v$) for clustering, we convert it into a topic-vector matrix W ($t \times w$) through matrix multiplication with G ($v \times w$). This transformation enables us to portray each word as a word embedding. Subsequently, we employ the K -medoids clustering algorithm to group each word embedding of W ($t \times w$) into K clusters.

3.3.4 | Quantitative Clustering Analysis

At this stage, we can analyze the results either across the K clusters or within each respective cluster. To consider both aspects, we propose a composite metric based on the top n words (10) of each topic, named *final score*. This metric is designed to quantify stability analysis and facilitate the identification of the most stable LDA parameters.

Initially, we integrate the Silhouette and Jaccard metrics to form the *mean-sil-jacc* score. This combination is chosen because it provides a robust estimation of clustering quality. The Silhouette coefficient measures both the cohesion within clusters and the separation between them, while the Jaccard coefficient assesses stability by evaluating the similarity of elements within a cluster.

For each cluster, identified through K -medoids clustering, we calculate a Silhouette score and compute the average of these scores. To integrate this with the Jaccard metric, we normalize the average between 0 and 1. We also compute the Jaccard coefficient, which quantifies the similarity between topic sets within each cluster based on term intersections. This mean Jaccard coefficient for all clusters is similarly scaled from 0 to 1. The combined mean scores of these two coefficients produce the *mean-sil-jacc* metric, providing a comprehensive view of cluster quality.

Additionally, to assess the consistency and the order of the top words representation within clusters, we employ the Kendall coefficient. This metric is used for pairwise comparisons between topic sets within the same cluster, focusing on the consistency of the presence and order of top words. The results are aggregated and normalized, resulting in a measure we call *Kendall-centered*, which also ranges from 0 to 1.

Furthermore, we use a clustering measure termed *n-topic-per-clusters* to evaluate each cluster's cohesion by comparing the actual number of topics within a cluster to the expected

number based on replicated runs. This metric is computed as follows:

$$\max \left(1 - \frac{\left| \frac{\text{all topics in the cluster under analysis} - \text{number of replicated runs}}{\text{number of replicated runs}} \right|, 0 \right)$$

After calculating this metric for all clusters, we compute the average to provide a global assessment of cohesion. Finally, we calculated the **average arithmetic mean of the Kendall-centered, mean-sil-jacc, and n-topic-per-clusters metrics to create the final score measure.**

Together, these metrics form a comprehensive assessment tool that not only reflects the cohesion and separation of clusters but also the internal consistency of topic presentation, making the *final score* a robust indicator of parameter stability in LDA models. As we can see, the *n-topic-per-clusters* acts as a penalty term to maximize the expected number of topic-per cluster.

Using this final metric, we refine the methodology proposed by Mantyla et al. [16], who suggest using Rank Biased Overlap (RBO) to compare ranked lists. Although the RBO measure offers a more flexible form of a Jaccard coefficient, as noted by Rieger et al. [17], it requires extensive parameter optimization, making it time-consuming.

4 | Results and Discussion

In this section, we present the results obtained during our experimentation. We conduct our discussion considering the two main steps of our methodology: parameters tuning and stability evaluation. This section will allow us to answer the research questions we stated in this paper's introduction.

4.1 | Parameter Tuning

In this section, we analyze the number of topics found and the duration for each combination of metaheuristic-fitness functions from a statistical point of view. As a reminder, the dispersion measures are based on 25 independent runs per combination, except for the Raw score fitness function, which uses only 1 run. This is equivalent to the number of LDA models evaluated during the optimization process. The statistical results are presented in Table 7, followed by their discussion.

4.1.1 | Number of Topics

For all corpora, regarding the number of topics found by the optimization process (see Table 7, columns median and mean number of topics found), we observed that the Coherence and Silhouette fitness functions found smaller numbers of topics than the three other fitness functions. The higher values may suggest that these fitness functions are more likely to generate hierarchical clustering. Based on Rieger et al. [17], we employed hierarchical clustering with complete linkage. This method is well-suited for organizing data into hierarchical structures based on similarity metrics. In our analysis, we utilized the

Hellinger distance to quantify the similarities between topics. The Hellinger distance is particularly effective for comparing probability distributions, making it suitable for our topic modeling context. This approach enabled us to effectively organize the data into hierarchical clusters based on topic similarities.

The following graphs illustrate this clustering approach in one executed run of the BBC-News corpus for those three fitness functions: 2PJ, R-square, and Raw score. Based on the expected number of topics reported in Table 7, we pruned the resulting dendrogram in five to see if they could be retrieved. The results are presented below (see figure 3), showing clear evidence that we can retrieve the expected number of topics.

This observation between the two groups of fitness functions is interesting because the emergence of latent topics found by Silhouette and Coherence seems to be on a more general level. In comparison, the other fitness functions find the general topics, and furthermore, they have more fine-grained latent information on the analyzed corpus.

The mean and median (of the number of topics) show similar results, indicating that the distributions are not skewed. Furthermore, if we compare the expected number of topics found by D. Greene, the variance coefficient for Silhouette and Coherence indicates that our results agree with the expected number of topics.

4.1.2 | Duration

The execution time was calculated for the entire process of the optimization task, and the mean duration time was calculated over the 25 independent runs. The parallel computation (synchronization, communication, thread creation/destruction) influences the estimation of the durations used by the GA implementation on the server. Moreover, the LDA calculation time is influenced by the vocabulary length (see Table 2), the number of documents (see Table 2), and the number of topics dictated by the solution under optimization (see Table 3).

As reported in Table 7, we can see that the fitness functions that allow more topics take more time than the Coherence measure. However, what is counter-intuitive is that the Silhouette measurement is slower overall. Furthermore, the fact that Coherence is faster than Silhouette does not coincide with Panichella [6] since their results show that Silhouette should be faster.

Following this analysis, we can now answer the first question:

Q1. Do the different combinations of metaheuristic and fitness functions give similar results?

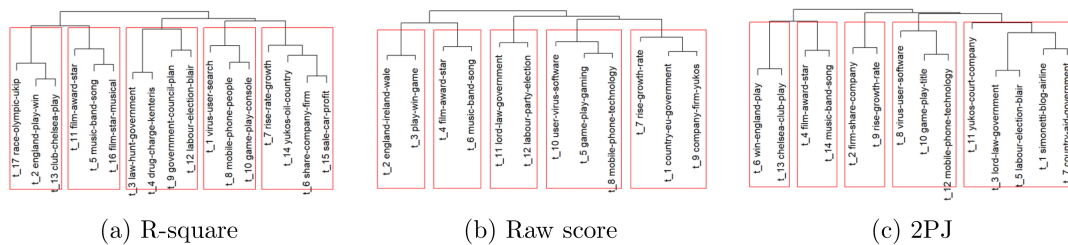
We believe that the combinations between the GA and the different fitness functions provide similar results to those expected by Greene et al. [8]'s studies. Although we observed that the R-square, 2PJ, and Raw score fitness functions find more topics. The hierarchical analysis (Figure 3) of these clusters shows that it is possible to find more general latent topics such as the ones discovered by traditional methods such as Silhouette and Coherence. Finally, in the context of this analysis, we have just

TABLE 7 | Summary of the combinations of GA and fitness functions per data set.

	Median number of topics found	Mean number of topics found	Variance coefficient mean topic found (%)	Mean duration (seconds)	Variance coefficient mean duration (%)	Expected number of topics (D. Greene)
20 Newsgroups-sb-6-1 corpus						
Coherence	7	7	16.27	296	12.94	6
2PJ	6	6	28.46	280	14.21	6
R-square	17	16	19.94	375	13.05	6
Raw score	10	10	NA*	245**	NA*	6
Silhouette	7	7	23.08	533	11.71	6
20 Newsgroups-ol-6-1 corpus						
Coherence	6	6	28.17	251	11.59	6
2PJ	18	18	11.79	330	11.58	6
R-square	19	18	7.93	341	8.65	6
Raw score	13	13	NA*	296**	NA*	6
Silhouette	7	8	39.47	474	17.70	6
BBC-Sport corpus						
Coherence	9	9	30.74	86	11.27	5
2PJ	18	17	18.75	111	14.92	5
R-square	19	19	6.73	113	8.44	5
Raw score	11	11	NA*	89**	NA*	5
Silhouette	7	7	26.08	157	14.27	5
BBC-News corpus						
Coherence	7	7	28.07	322	12.54	5
2PJ	15	14	34.57	371	20.55	5
R-square	19	19	7.64	433	12.16	5
Raw score	12	12	NA*	343**	NA*	5
Silhouette	7	8	30.94	588	20.97	5

*Statistics calculated in one execution.

**Total run time divided by 25 to allow easier comparison.

**FIGURE 3** | Hierarchical clustering results on BBC-News data set.

demonstrated the potential of using new fitness functions to discover latent topics by LDA.

4.2 | Stability

Building upon the insights outlined in Section 3.3, this section delves into verifying the efficacy of the stability analysis. The stability analysis serves as a tool to identify the most suitable run among the 25 iterations of the optimization process. As a

reminder, the stability analysis involves the generation of 20 LDA models using the identified parameters, followed by the transformation of topic-word matrices into vector representations utilizing GloVe models. Subsequently, employing the k-medoid algorithm, we cluster these word vectors and evaluate these groupings using clustering metrics. Together, these assessments result in a final metric that we called the *final score*. This metric serves as a determinant in discerning the most stable run id, thereby facilitating the selection of optimal parameters. To critically evaluate whether we can rely solely on metaheuristic

TABLE 8 | Top two runs on the BBC-News corpus using the Silhouette fitness function.

Run id	K	Fitness score	Final score (%)			
			GloVe models			
			Common-Crawl	BBC-Sport	BBC-News	BBC-Sport-News
Sorted by Fitness score						
*14	6	0.69	79.49	78.51	79.76	79.47
7	6	0.68	81.16	80.56	81.51	81.14
Sorted by Common-Crawl						
*23	5	0.53	95.24	95.00	95.25	95.18
17	6	0.57	83.94	83.13	84.0	83.4
Sorted by BBC-Sport						
23	5	0.53	95.24	95.00	95.25	95.18
17	6	0.57	83.94	83.13	84.0	83.4
Sorted by BBC-News						
23	5	0.53	95.24	95.00	95.25	95.18
17	6	0.57	83.94	83.13	84.0	83.4
Sorted by BBC-Sport-News						
23	5	0.53	95.24	95.00	95.25	95.18
17	6	0.57	83.94	83.13	84.0	83.4

procedures for parameter optimization, a thorough stability analysis is indispensable. This leads us to address the following question:

Q2. Are the run id of the maximum value of the fitness function and the run id of the maximum value of the stability analysis the same?

This comparison helps us determine whether the optimal results suggested by the fitness function are consistent with those indicated by the stability analysis. To answer this question, we compared the fitness score (the final value obtained by the optimization process) to the *final score* obtained by the stability analysis. The range of the fitness score is driven by the nature of the fitness function used. In comparison, the *final score* calculated will always be between 0 and 1.

The tables 8 and 9 were obtained by selecting the two highest values sorted by the fitness score or the stability score obtained by each GloVe model. For this comparison, we use only Silhouette and Coherence.

Looking at the run id and the number of topics in Tables 8 and 9, we can see the non-concordance between the highest fitness and *final scores*. Therefore, we can deduce that the optimization process does not automatically lead to the most stable LDA model. Furthermore, for the selected corpus (BBC-News and 20 Newsgroups-sb-6-1), we can note that the two highest selected values all have the same run id (run id 23 in Table 8 and run id 1 in Table 9), independently of the GloVe models used. The advantage of using the *final score* value instead of the fitness score is that using the former provides a more stable model.

Furthermore, we ensured that all underlying metrics constituting the *final score* agree (i.e., not only one or two scores push the *final score* high). Tables 10 and 11 compare the underlying metrics of the best run selected by the fitness function versus the selection based on the *final score* (run id prefixed by * in Tables 8 and 9).

Finally, we can conclude that the best solution found by the optimization process does not point out the most stable model and that applying the stability analysis seems to improve the reliability/replicability. At this point, we can select the most stable parameters to generate the LDA for each fitness function used. Based on this selective approach, the results presented in the following table are based on the maximum value among all runs as a function of the fitness score and *final score* of the four different GloVe models.

A remarkable feature of the results presented in Table 12 is the relationship between the number of topics discovered (K) and the *final score* results. This observation leads to the following question:

Q3. How do we explain the relation between a low number of topics and a high final score, and vice-versa?

Let us compare the different *final scores* obtained by the different fitness functions for a given data set. Notably, the number of topics discovered influences the value of the *final score*. As we can observe in Table 12, the higher the number of topics, the lower the *final score*, and vice-versa. This relationship can be explained by the difficulty of the clustering algorithm to correctly group the hierarchical elements that then influence the different metrics constituting the *final score*. Specifically, the

TABLE 9 | Top two runs on the 20 Newsgroups-sb-6-1 corpus using the Coherence fitness function.

Run id	K	Fitness score	Final score (%)			
			GloVe models			
			Common-Crawl	BBC-Sport	BBC-News	BBC-Sport-News
Sorted by Fitness						
*2	7	−51.47	78.16	76.38	77.55	77.03
12	7	−51.48	79.32	80.84	80.9	80.84
Sorted by Common-Crawl						
*1	6	−56.25	95.57	94.68	94.81	94.92
19	6	−54.43	93.91	93.34	93.02	93.10
Sorted by BBC-Sport						
1	6	−56.25	95.57	94.68	94.81	94.92
19	6	−54.43	93.91	93.34	93.02	93.10
Sorted by BBC-News						
1	6	−56.25	95.57	94.68	94.81	94.92
19	6	−54.43	93.91	93.34	93.02	93.10
Sorted by BBC-Sport-News						
1	6	−56.25	95.57	94.68	94.81	94.92
19	6	−54.43	93.91	93.34	93.02	93.10

TABLE 10 | Final score underlying metrics comparison for the best run on the BBC-News corpus using the Silhouette fitness function.

Run id	Mean-Kendall-centered	mean-sil-jacc	n-topic-per-clusters	Final score
14	0.777316	0.748714	0.866667	79.76
23	0.933253	0.924393	1	95.25

TABLE 11 | Final score underlying metrics comparison for the best run on the 20 Newsgroups-SB-6-1 corpus using the Coherence fitness function.

Run id	Mean-Kendall-centered	mean-sil-jacc	n-topic-per-clusters	Final score
2	0.78	0.72	0.84	78.16
1	0.95	0.92	1.0	95.57

relation between a low number of topics and a high *final score* happens when the topics discovered are more general. Then, the projection of the terms constituting the groups of topics (via word embedding) facilitates the task of the clustering algorithm (k-medoid) because their positioning in the word vector space does not impose any overlap. Conversely, when the number of topics discovered comprises hierarchical refinement, the projection of the terms into the word vector space does not distinguish between the parent and child groupings of the different hierarchies discovered. This feature adds difficulty to the clustering algorithm.

The apparent contradiction between 2PJ's sophisticated bidirectional assessment and its lower stability scores reveals an important insight about topic modeling evaluation. 2PJ's strength lies in discovering semantically coherent but fine-grained topics that traditional clustering-based stability measures struggle to validate. This suggests that different fitness functions may require different stability evaluation approaches, or that stability metrics should be interpreted differently based on the granularity of discovered topics.

Q4. Does hierarchical clustering find similar latent topics as Silhouette and Coherence?

We cannot rely on the *final score* value to compare the two groups of fitness functions: the ones discovering topics at a less low level of granularity (Silhouette and Coherence) and the ones discovering the highest level of granularity (R-square, 2PJ, and Raw score). First, the *final score* is influenced by the number of topics discovered. Second, the Silhouette and Coherence fitness functions tend to discover fewer latent topics (described as a basic understanding of the different corpora) than the other three. On the other hand, their hierarchical characteristics can be exploited to convert many topics into smaller ones. When two models (a reference model and a comparison model) have the same number of topics, it is easier to compute a similarity value between the groups of terms discovered.

The following approach is used to enable this comparison:

- Compute the dendrogram of the comparison models, that is, the models generated by R-square, 2PJ, and Raw score fitness functions.

TABLE 12 | Best run for all corpus and fitness functions.

	Coherence			Silhouette			R-square			2PJ			Raw score		
	Run id	Final	K	Run id	Final	K	Run id	Final	K	Run id	Final	K	Run id	Final	K
		score			score			score			score			score	
BBC-News corpus															
Fitness score	18	—	5	14	—	6	1	—	20	8	—	18	1	—	12
GloVe-Common-Crawl	18	95.38	5	23	95.24	5	23	69.97	20	7	90.15	5	1	72.40	12
GloVe-BBC-Sport	22	95.57	5	23	95.00	5	23	71.26	20	7	89.73	5	1	70.47	12
*GloVe-BBC-News	22	95.28	5	23	95.25	5	13	71.49	17	7	90.06	5	1	72.65	12
GloVe-BBC-Sport-News	22	95.28	5	23	95.18	5	13	71.57	17	7	90.03	5	1	72.68	12
BBC-Sport corpus															
Fitness score	10	—	5	15	—	7	18	—	20	1	—	7	1	—	11
GloVe-Common-Crawl	15	80.32	6	4	80.27	6	16	69.24	16	20	72.23	14	1	76.84	11
*GloVe-BBC-Sport	15	80.78	6	4	79.44	6	16	70.81	16	15	74.85	14	1	77.50	11
GloVe-BBC-News	15	80.80	6	4	80.25	6	16	69.32	16	15	74.48	14	1	76.97	11
GloVe-BBC-Sport-News	15	80.76	6	4	80.25	6	16	70.39	16	15	74.94	14	1	77.07	11
20 Newsgroups-ol-6-1 corpus															
Fitness score	7	—	5	10	—	6	11	—	18	9	—	17	1	—	13
*GloVe-Common-Crawl	11	71.77	5	22	69.39	6	6	65.28	19	16	69.18	12	1	67.40	13
GloVe-BBC-Sport	1	72.55	5	22	69.74	6	1	63.85	19	16	68.92	12	1	65.60	13
GloVe-BBC-News	1	71.65	5	2	67.90	7	20	64.57	20	16	68.08	12	1	62.61	13
GloVe-BBC-Sport-News	1	71.74	5	2	67.67	7	20	64.10	20	16	67.65	12	1	62.82	13
20 Newsgroups-sb-6-1 corpus															
Fitness score	2	—	7	19	—	7	19	—	20	14	—	5	1	—	10
*GloVe-Common-Crawl	1	95.57	6	1	91.75	6	3	77.42	8	12	92.43	5	1	74.55	10
GloVe-BBC-Sport	1	94.68	6	1	91.04	6	3	76.63	8	18	91.58	6	1	73.52	10
GloVe-BBC-News	1	94.81	6	1	91.27	6	3	76.19	8	4	91.51	6	1	73.00	10
GloVe-BBC-Sport-News	1	94.92	6	1	91.38	6	3	76.12	8	4	91.65	6	1	72.73	10

- Prune the dendrogram to obtain the number of clusters equal to the number of topics of the reference model, that is, the models generated by the Silhouette and Coherence fitness functions.
- Extract the 10 most important terms from each branch constituting each cluster, for example, if there are three topics in a cluster and only 10 words in the reference model, we select 30 words for the comparison model.
- For the list of terms constituting each cluster in the comparison model (e.g., 30 words), we determine its most similar list of terms in the reference model (e.g., 10 words) by doing a pairwise comparison using the Jaccard similarity measure. Then, an association is carried out based on the highest similarity value. For each association, calculate the asymmetric similarity as follows:

$$\frac{|\text{list of terms in comparison model} \cap \text{list of terms in reference model}|}{|\text{list of terms in comparison model}|}$$

- Finally, calculate the arithmetic mean of each association similarity value over all the clusters to obtain the global similarity score.

We used the LDA parameters obtained by each * prefixed row from Table 12 to produce the following results. Their selection

is based on the higher percentage of vocabulary intersection between the GloVe model and the data set under analysis (see Table 6), except for the BBC-Sport corpus used by the BBC-Sport GloVe model. The results below show the arithmetic mean of each association similarity value. The columns are for the reference models, while the rows are for the comparison model (Tables 13–16).

If we look at the similarity results for cases where the reference models are Silhouette and Coherence, we can conclude that there is a similarity among the categories of fitness functions. Generally, we can note that the fitness functions R-square, Raw score, and 2PJ are more similar to Silhouette. To conclude, there is a correspondence between the methods that hierarchically describe the corpora and the so-called general interpretation methods (Silhouette, Coherence).

Q5. Is training a new GloVe more advantageous than using a pretrained GloVe?

Given that all our data sets consist of newspaper corpora, we anticipate minimal differences in the results when comparing the final scores among various GloVe models, whether pre-trained or newly trained, used in the stability evaluation step. It is noteworthy that, despite the slight variations in vocabulary intersection percentages detailed in Table 6—where intersections of

TABLE 13 | Percentage of word similarity for BBC-Sport corpus.

	Coherence	Silhouette
R-square	0.73	0.80
Raw score	0.70	0.78
2PJ	0.68	0.68

TABLE 14 | Percentage of word similarity for BBC-News corpus.

	Coherence	Silhouette
R-square	0.80	0.82
Raw score	0.96	0.98
2PJ	0.96	0.96

TABLE 15 | Percentage of word similarity for 20 Newgroup-ol-6-1 corpus.

	Coherence	Silhouette
R-square	0.60	0.67
Raw score	0.45	0.55
2PJ	0.63	0.68

TABLE 16 | Percentage of word similarity for 20 Newgroup-sb-6-1 corpus.

	Coherence	Silhouette
R-square	0.87	0.83
Raw score	0.80	0.73
2PJ	0.80	0.87

common words are more than 90%—the outcomes for the best run id selection are remarkably consistent (refer to Table 12). This consistency underscores the effectiveness of pre-trained models in capturing the essential semantic characteristics of news-related content. The decision to use a pre-trained model [29] or to train a model specifically tailored to a particular corpus depends on the specific characteristics and requirements of that corpus.

In our study, we utilized the Stanford pre-trained GloVe model based on, at least, the 2013 Common Crawl data set,⁶ chosen for its extensive coverage of general content. This selection is particularly relevant given that our analysis focuses on the BBC data set from 2004 to 2005,⁷ which is categorized into five broad thematic areas: business, entertainment, politics, sport, and technology. The BBC sport sub-corpus includes topics such as athletics, cricket, football, rugby, and tennis. Additionally, the general nature of the terms and their relevance across different time periods support the use of the pre-trained GloVe model for this study. The fact that these themes are generic and recur regularly in news cycles reduces the impact of any temporal discrepancies between the training data of the GloVe model and the BBC corpus.

The rationale for using the pre-trained GloVe model lies in its comprehensive linguistic representation derived from a vast web

corpus, which includes significant news-related content. Such content ensures that major themes typically covered in news reports. This broad thematic alignment suggests that the linguistic biases and semantic relationships inherent in the pre-trained GloVe model are likely to be in consonance with those in the BBC news corpus.

Our objective is not to capture specific events from a particular period but rather to analyze overarching thematic trends and language usage within the news domain. Therefore, the pre-trained GloVe model, with its broad training base is ideally suited for our analysis. By utilizing a model trained on such a rich and varied data set, we ensure that our semantic analysis benefits from a robust and relevant linguistic foundation, thereby enhancing the reliability and depth of our study's insights into the newspaper corpus.

This decision was based on the understanding that such content typically encompasses a broad spectrum of topics, akin to what is termed “common knowledge.” The inherent bias of GloVe models, meaning their domain-specific knowledge, is crucial in our context. Pre-trained GloVe models are developed using extensive and diverse data sets that often include substantial news content. This broad training base equips the models with a comprehensive linguistic and contextual understanding that is particularly suited to the semantic fields found in newspaper articles.

Q6. How do we select the final parameters?

Following the above analyses, selecting the best set of parameters for a corpus will depend on the depth of analysis the user would like to acquire (i.e., more or fewer topics). We have demonstrated the distinction between two main categories of fitness functions and shown their power of interpretation. Selecting the final parameters will undoubtedly require a stability analysis; it will simply require selecting the maximum final score value.

5 | Conclusion

The primary aim of this paper was to explore the use of Genetic Algorithms for determining near-optimal LDA parameters, following the approach suggested by Panichella [6]. We evaluated five different fitness functions: two traditional ones (Silhouette and Coherence), two less-known ones (R-square and Raw score), and one that we developed (2PJ). LDA is known for its instability due to its generative and probabilistic nature, a point noted by several researchers. To mitigate this shortcoming, we modified the order of the DTM before each LDA model generation (addressing Type-II instability). We also repeatedly generated models with the same parameters to assess the topic-terms matrices replicability (addressing Type-I instability).

Our comparison of different combinations of fitness functions revealed an interesting advantage in uncovering latent themes in texts. Traditional methods such as Silhouette and Coherence tend to identify more general topics, whereas R-square, 2PJ, and

Raw score uncover a broader range of latent topics. Our analysis reveals that 2PJ offers a complementary approach to traditional fitness functions rather than a replacement. While it achieves lower stability scores due to its fine-grained topic discovery, it provides valuable semantic insights that can be leveraged through hierarchical post-processing to achieve both detailed topic exploration and practical topic numbers. By using a hierarchical representation, we found that these fitness functions are comparable to traditional methods in effectiveness. Consequently, their use offers greater descriptive potential for automatic topic analysis in texts. Employing the data sets proposed and analyzed by Greene et al. [8], we demonstrated that it is possible to achieve consistent results in terms of the number of topics (K) with various GA-fitness function combinations, even after applying a pruning algorithm to the non-traditional fitness functions.

While using an optimization algorithm to determine near-optimal parameters automatically may seem like a magic solution to a given problem. However, when using LDA, we have demonstrated the importance of adding a stability analysis step to evaluate the parameters' quality.

Regarding future work, the proposed methodology offers several avenues for improvement through its two main steps: the search for optimal parameters and the stability analysis. Additionally, we suggest benchmarking against BERTopic and other topic models. Below we provide concrete suggestions for each.

Optimization algorithms:

1. Alternative algorithms exploration:
Although we currently use the Genetic Algorithm (GA) as suggested by Panichella [6], future work could experiment with Simulated Annealing (SA) and Stochastic Hill Climbing (SHC). Efforts could focus on testing other metaheuristics that minimize computational burden while maintaining accuracy.
2. Enhanced GA techniques:
Following Scrucca's [30] recommendation, integrating an efficient local search in the last iteration of GA could refine the parameter optimization process. Investigating asynchronous local searches that run in parallel across different segments of the parameter space might overcome the increased computational times we observed.
3. Parallel computing frameworks:
Implementing and comparing different parallel GA frameworks, specifically the island model GA, could be valuable. This should include analyzing how different island sizes and communication frequencies affect the exploration and exploitation phases of the algorithm.

Stability analysis:

1. Increased LDA runs:
As per Mantyla et al. [16], increasing the number of LDA replication runs beyond 20 and analyzing the impact on topic consistency and stability would be insightful. A statistical analysis of the variance in the *final scores* calculated

across different numbers of replication runs using the same parameter set would help address the question of whether 20 runs are sufficient.

2. Advanced word embedding models:
Incorporating larger word embedding models such as BERT, FastText, and GPT for projecting word clusters should be explored. Comparative studies to evaluate the impact of these embeddings on the semantic quality of clusters would help determine their necessity.
3. Development of robustness metrics:
While we introduced a *final score* metric for cluster evaluation, future investigations could benefit from the incorporation or modification of additional metrics to comprehensively assess cluster quality and robustness.

As we conclude our investigation, the necessity of benchmarking our methodology against contemporary topic modeling approaches, such as BERTopic, becomes increasingly apparent. Through this comparative analysis, we aimed to evaluate crucial aspects such as the correspondence in the number of topics discovered, coherence in identified topics, and the necessity of stability analysis. This effort not only provides valuable information, but also sets a benchmark for future progress in topic modeling methodologies in the context of the pervasive influence of word insertion techniques.

Acknowledgments

The authors thank the Fondation Les Petits Trésors, the Laurent Mottron laboratory from the University of Montreal, the LiNCS laboratory of Ecole de Technologie Supérieure, and IBM Canada Center for Advanced Studies (CAS) for their support.

Funding

The research presented in this paper was financially supported by MITACS (Mathematics of Information Technology and Complex Systems) IT25026.

Data Availability Statement

The data can be found here: <https://github.com/simonOHa/New-Heuristics-for-stable-LDA-parameter-search/tree/main/dtm>.

Endnotes

¹<https://github.com/matthewjdenny/SpeedReader>.

²The DTM uses a term frequency (tf) scheme or a term frequency with the inverse document frequency (tf-idf) scheme. The tf shows the term counts in the documents, and the tf-idf shows the weighted counts. The weights are based on how important a term is in the corpus. For example, the weight of a given term will be based on the number of occurrences in a given document but also the number of occurrences over the whole corpus.

³<https://cloud.r-project.org/web/packages/textmineR/index.html>.

⁴Dereck Greene resources website: <http://mlg.ucd.ie/howmanytopics/index.html>.

⁵<https://github.com/stanfordnlp/GloVe>.

⁶<https://commoncrawl.org/overview>.

⁷<http://mlg.ucd.ie/datasets/bbc.html>.

References

1. V. Rajaraman, "Big Data Analytics," *Resonance* 21, no. 8 (2016): 695–716, <https://doi.org/10.1007/s12045-016-0376-7>.
2. D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research* 3 (2003): 993–1022.
3. J. Boyd-Graber, Y. Hu, and D. Mimno, "Applications of Topic Models," *Foundations and Trends in Information Retrieval* 11, no. 2–3 (2017): 143–296.
4. A. Agrawal, W. Fu, and T. Menzies, "What Is Wrong With Topic Modeling? And How to Fix It Using Search-Based Software Engineering," *Information and Software Technology* 98 (2018): 74–88, <https://doi.org/10.1016/j.infsof.2018.02.005>, <http://www.sciencedirect.com/science/article/pii/S0950584917300861>.
5. M. Hughes, D. I. Kim, and E. Sudderth, "Reliable and Scalable Variational Inference for the Hierarchical Dirichlet Process," in *Artificial Intelligence and Statistics* (PMLR, 2015), 370–378.
6. A. Panichella, "A Systematic Comparison of Search-Based Approaches for Lda Hyperparameter Tuning," *Information and Software Technology* 130 (2021): 106411.
7. R. Arun, V. Suresh, C. V. Madhavan, and M. N. Murthy, "On Finding the Natural Number of Topics With Latent Dirichlet Allocation: Some Observations," in *Advances in Knowledge Discovery and Data Mining: 14th Pacific-Asia Conference, PAKDD 2010, Hyderabad, India, June 21–24, 2010. Proceedings. Part I 14, Pages* (Springer, 2010), 391–402.
8. D. Greene, D. O'Callaghan, and P. Cunningham, "How Many Topics? Stability Analysis for Topic Models," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15–19, 2014. Proceedings, Part I 14, Pages* (Springer, 2014), 498–513.
9. S. Terragni, E. Fersini, B. G. Galuzzi, P. Tropeano, and A. Candelieri, "Octis: Comparing and Optimizing Topic Models Is Simple!," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations* (2021), 263–270.
10. A. Panichella, B. Dit, R. Oliveto, M. Di Penta, D. Poshynanyk, and A. de Lucia, "How to Effectively Use Topic Models for Software Engineering Tasks? An Approach Based on Genetic Algorithms," in *In 2013 35th International Conference on Software Engineering (ICSE)* (IEEE, 2013), 522–531.
11. T. Yarnaguy and W. Kanarkard, "Tuning Latent Dirichlet Allocation Parameters Using Ant Colony Optimization," *Journal of Telecommunication, Electronic and Computer Engineering* 10, no. 1–9 (2018): 21–24.
12. T. P. Minka and J. Lafferty, "Expectation-Propogation for the Generative Aspect Model," *arXiv Preprint arXiv:1301.0588* (2012).
13. T. L. Griffiths and M. Steyvers, "Finding Scientific Topics," *Proceedings of the National Academy of Sciences* 101, no. 1 (2004): 5228–5235.
14. H. Wallach, D. Mimno, and A. McCallum, "Rethinking Lda: Why Priors Matter," *Advances in Neural Information Processing Systems* 22 (2009): 1973–1981.
15. D. Binkley, D. Heinz, D. Lawrie, and J. Overfelt, "Source Code Analysis With Lda," *Journal of Software: Evolution and Process* 28, no. 10 (2016): 893–920.
16. M. V. Mantyla, M. Claes, and U. Farooq, "Measuring Lda Topic Stability From Clusters of Replicated Runs," in *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (ACM, 2018), 1–4.
17. J. Rieger, L. Koppers, C. Jentsch, and J. Rahnenführer, "Improving Reliability of Latent Dirichlet Allocation by Assessing Its Stability Using Clustering Techniques on Replicated Runs," *arXiv Preprint arXiv:2003.04980* (2020).
18. D. Newman, J. H. Lau, K. Grieser, and T. Baldwin, "Automatic Evaluation of Topic Coherence," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (Association for Computational Linguistics, 2010), 100–108.
19. D. Mimno, H. Wallach, E. Talley, M. Leenders, and A. McCallum, "Optimizing Semantic Coherence in Topic Models," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, 2011), 262–272.
20. K. Stevens, P. Kegelmeyer, D. Andrzejewski, and D. Buttler, "Exploring Topic Coherence Over Many Models and Many Topics," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (Association for Computational Linguistics, 2012), 952–961.
21. A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh, "On Smoothing and Inference for Topic Models," *arXiv Preprint arXiv:1205.2662* (2012).
22. H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno, "Evaluation Methods for Topic Models," in *Proceedings of the 26th Annual International Conference on Machine Learning* (ACM, 2009), 1105–1112.
23. J. Chang, S. Gerrish, C. Wang, J. Boyd-Graber, and D. Blei, *Reading Tea Leaves: How Humans Interpret Topic Models*, vol. 22 (Advances in neural information processing systems, 2009).
24. T. Jones, "A Coefficient of Determination for Probabilistic Topic Models," *arXiv Preprint arXiv:1911.11061* (2019).
25. W. Webber, A. Moffat, and J. Zobel, "A Similarity Measure for Indefinite Rankings," *ACM Transactions on Information Systems* 28, no. 4 (2010): 1–38.
26. D. Greene and P. Cunningham, "Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering," in *Proceedings of the 23rd International Conference on Machine Learning* (Association for Computing Machinery (ACM), 2006), 377–384.
27. T. Jones, W. Doane, and M. T. Jones, "Package 'Textminer,' Functions for Text Mining and Topic Modeling," (2016).
28. J. Pennington, R. Socher, and C. D. Manning, "Glove: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Association for Computational Linguistics, 2014), 1532–1543.
29. D. Nguyen, A. S. Doğruöz, C. P. Rosé, and F. de Jong, "Computational Sociolinguistics: A Survey," *Computational Linguistics* 42, no. 3 (2016): 537–593, https://doi.org/10.1162/COLI_a_00258.
30. L. Scrucca, "On Some Extensions to Ga Package: Hybrid Optimisation, Parallelisation and Islands Evolution," *arXiv Preprint arXiv:1605.01931* (2016).