A high-reproducibility and high-accuracy method for automated topic classification

Andrea Lancichinetti[1,2], M. Irmak Sirer[2], Jane X. Wang[3],
Daniel Acuna[4], Konrad Körding[4], Luís A. Nunes Amaral[1,2,5,6]
[1]Howard Hughes Medical Institute (HHMI), [2] Department of Chemical and Biological Engineering, Northwestern University, Evanston, Illinois, USA, [3] Department of Medical Social Sciences Northwestern University Feinberg School of Medicine, Chicago, Illinois, USA, [4] Department of Physical Medicine and Rehabilitation, Rehabilitation Institute of Chicago Northwestern University, Chicago, Illinois, USA, [5] Northwestern Institute on Complex Systems, Northwestern University, Evanston, Illinois, USA, [6] Department of Physics and Astronomy, Northwestern University, Evanston, Illinois, USA.

Supplemental Material

OUTLINE

The supplementary material is organized as follows:

- Sec. S1 provides analytical insights on the likelihood landscape: in particular, we discuss the theoretical limitations of PLSA [1] and symmetric LDA [2] in finding the correct generative model. Also, in Sec. S1 F we present an additional example which suggests why equally sized topics often have better likelihood.
- Sec. S2 describes the network approach we take for topic modeling.
- Sec. S3 shows that standard LDA tends to overestimate the number of topics, and to find equally sized topics.
- Sec. S4 presents a more detailed analysis of the synthetic datasets: among other things, we visualize the algorithms' results.
- Sec. S5 shows the performance of asymmetric LDA [3].
- Sec. S6 discusses the hierarchical topics of Web of Science dataset and the role of the p-value for TopicMapping.
- Sec. S7 presents the computation complexity of TopicMapping.
- Sec. S8 shows the topics we found on a large sample of the English Wikipedia.
- Sec. S9 shows that TopicMapping often provides models with higher likelihood, if we compare models with the same effective number of topics.
- Sec. S10 is an appendix with some more technical information about the calculations presented in Sec. S1, some clarifications about Dirichlet distributions and measuring perplexity, and some technical information about the algorithms' usage.

S1. DEGENERACY PROBLEM IN INFERRING THE LATENT TOPIC STRUCTURE

A. Introduction

Most topic model optimizations are known to be computationally hard problems [4]. However, not much is known about how the roughness of the likelihood landscape affects the algorithms' performance.

We investigate this question by (i) defining a simple generative model, (ii) generating synthetic data accordingly and (iii) measuring how well the algorithms recover the generative model (which is considered the "ground truth").

In the whole study, we examine different generative models. In this section, we study the simplest among those, the language test. For this model, we prove that, *if the topics are not enough equally sized, the model which maximizes the likelihood optimized by* PLSA *and symmetric* LDA *can be different from the generative model.* More specifically, we show that it is possible to find an extremely large number of alternative models (with the same number of topics) which overfit some topics and underfit some others but have a better likelihood than the true generative model. Symmetric LDA is the version of LDA where the prior α is assumed to be the same for all topics, and it is probably the most commonly used. For asymmetric LDA, which allows different priors, the correct generative model has the highest likelihood, in the language test. However, we show that the ratio between the log-likelihood of the generative model and the one of the alternative models can be arbitrarily close to 1, even in the limit of infinite number of documents and infinite number of words per document. This implies that even increasing the amount of available information, the likelihood of the generative model will not increase relatively to the others. Below, we also give some quantitative estimates.

B. The simplest generative model

Let us call K the number of topics. Each topic has a vocabulary of N_w words, and for the sake of simplicity we assume all the words are equiprobable. We also assume that we cannot find the same word in two different topics, so that we are actually dealing with fully disambiguated languages. Then, each document is entirely written in one of the languages sampling L_d random words from the corresponding vocabulary (we use the same number of words L_d for each document). This should be a very simple problem, since there is neither mixing of words across topics, nor of topics across documents.

Let us compute the log-likelihood, $\log \mathcal{L}_{\text{true}}$, of the generative model. The process of generating a document works in two step. We first select a language with probability $p(L)$, and we then write a document with probability $p(doc|L)$:

$$\log \mathcal{L}_{\text{true}} = \log p(L) + \log p(doc|L). \tag{S1}$$

Let us focus on the second part, $\log \mathcal{L}^{'}_{\text{true}} = \log p(doc|L)$. After we selected the language that we are going to use, every document has the same probability of being generated:

$$\log \mathcal{L}^{'}_{\text{true}} = -L_d \log N_w. \tag{S2}$$

We will also consider $p(L)$ later. We stress that $\log \mathcal{L}^{'}_{\text{true}}$ is the log-likelihood per document. The symbol $'$ is to recall that the likelihood is computed given that we know which language we are using for the document.

Now, let us compute the log-likelihood of an alternative model, where one language (say English) is overfitted in two dialects, and two other languages (say French and Spanish) are merged. Fig. S1 illustrates how we construct the alternative model. French and Spanish are just one topic, in which each French and Spanish word is equiprobable. The English words instead are arbitrarily divided in two groups: the first English dialect makes use of words from the former group with probability f_1 and words from the second groups with probability g_1 and the second dialect has probabilities f_2 and g_2 for the two groups. We assume that the first group of words is more likely for the first dialect, i.e $f_1 \geqslant g_1$, while the situation is reversed for the second dialect: $g_2 \geqslant f_2$. The general idea is that if a document, just by chance, is using words from the first group with higher probability, it might be fitted better by the first dialect: overfitting the noise improves the likelihood and, if the English portion of the corpus is big enough, this improvement might overcome what we lose by underfitting French and Spanish.

In Sec. S10 A, we prove that the difference between the log-likelihood per English document of the generative model and the alternative model is bigger than $1/\pi$, regardless of the number of words per document, the size of the vocabulary or the number of documents. More precisely, if $N_w \geqslant L_d$, the difference can also be higher, $\simeq (\log 2)^2$. Calling $\mathcal{L}^{'}_{E}$, the likelihood per English document in the alternative model, we have that:

$$\langle \log \mathcal{L}^{'}_{E} \rangle = \log \mathcal{L}^{'}_{\text{true}} + C \quad \text{with} \quad C \in [\sim 0.3, \sim 0.5]. \tag{S3}$$

Fig. S2, shows the log likelihood difference per English document, as a function of L_d/N_w.

Keeping the same number of topics, the alternative model will pay some cost underfitting Spanish and French. Since the languages are merged, the size of the vocabulary is $2N_w$ and the log-likelihood per Spanish or French document is:

$$\log \mathcal{L}^{'}_{SF} = \log \mathcal{L}^{'}_{\text{true}} - L_d \log 2. \tag{S4}$$

Now, to compute the expected log-likelihood of the alternative model we also need to know how often we use the different languages. Let us call f_E the fraction of English documents, and f_U the fraction of documents written in Spanish or French (underfitted documents).

The average log-likelihood per document of the alternative model can then be written as:

$$\langle \log \mathcal{L}^{'}_{\text{alt}} \rangle = \log \mathcal{L}^{'}_{\text{true}} + f_E C - f_U L_d \log 2. \tag{S5}$$

We recall that, so far, we have not considered the probability that each document will pick a certain language, $p(L)$. Symmetric and asymmetric LDA make different assumptions at this point and we treat them both in the next two sections.

C. Symmetric LDA

PLSA does not account for the probability of picking a language $p(L)$ in the likelihood. LDA instead does consider that: the hyper parameters α_L are a global set of parameters (one per topic) which tune the probabilities that each document is making use of each topic. In our case, each document is uniquely assigned to a language: therefore, for each document, there is a language which has probability 1 and all the other languages have probability 0. This corresponds to the limiting case $\alpha_L = \kappa p(L)$ where the proportionality factor κ is very small.

For symmetric LDA, however, all the α_L are equal. This implies that, regardless of the actual size of the languages, the algorithm fits the data with a model for which $p(L) = 1/K$ (we recall that K is the number of languages). Therefore:

$$\log \mathcal{L}_{\text{true}} = -\log K - L_d \log N_w \quad \text{and} \tag{S6}$$

$$\langle \log \mathcal{L}_{\text{alt}} \rangle = \log \mathcal{L}_{\text{true}} + f_E C - f_U L_d \log 2.$$

If f_E is big enough, the likelihood of the alternative model can be higher than the one of the generative model. To be more concrete, let us consider an example. If $L_d = 10$ and $N_w = 20$, in Sec. S10 A, we show that C can be as high as $\simeq 0.476$. Let us consider the simplest case of just three topics, $f_U = 1 - f_E$. Setting the right hand

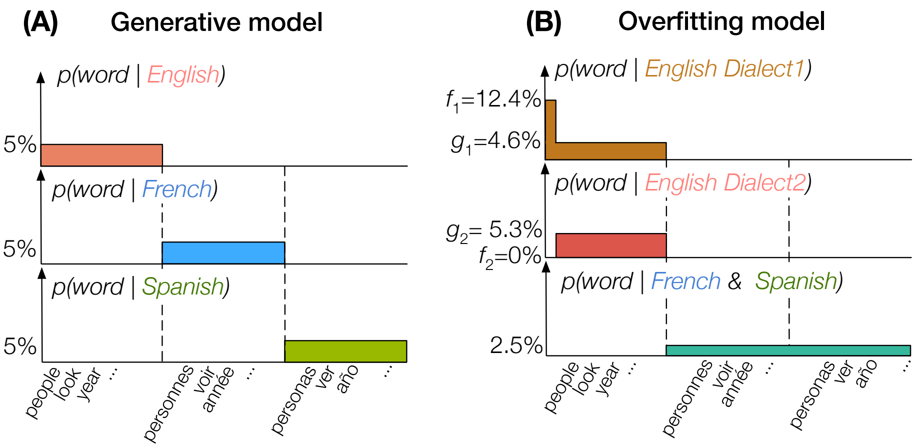

Figure S1: Distribution over words for the topics in the generative model (**A**) and in the most likely model (**B**). We set $N_w = 20$ words in each language's vocabulary.

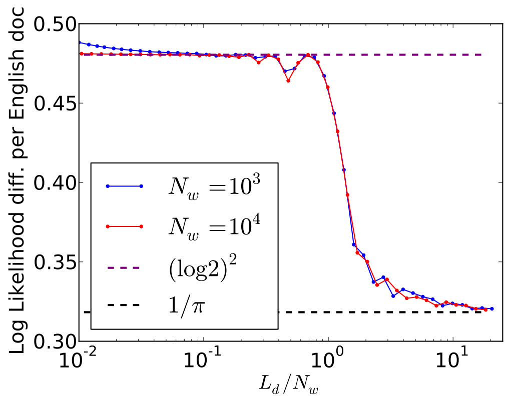

Figure S2: Average difference in the log-likelihood per English document of the alternative model and the generative model as a function of the ratio L_d/N_w, (words per document over vocabulary size). The function as well as the two dashed lines have been analytically computed in Sec. S10 A.

side of Eq. S6 to zero, we find that if $f_E \geqslant 0.936$ the alternative model has a better likelihood. If the topics are not balanced enough, symmetric LDA cannot find the right generative model, regardless of the absence of any sort of mixing. However, this critical value actually depends on L_d, and increasing L_d the generative model will eventually get a better likelihood. The case $L_d \gg 1$, is treated in detail below.

D. Asymmetric LDA

For asymmetric LDA, the average log-likelihood of the true model becomes:

$$\langle \log \mathcal{L}_{\rm true} \rangle = -H_{\rm true} + \log \mathcal{L}'_{\rm true}, \tag{S7}$$

where $H_{\rm true}$ is the entropy of the language probability distribution, $H_{\rm true} = -\sum_L p(L) \log p(L)$.

For the sake of simplicity, let us assume that French and Spanish are equiprobable, as well as the two English dialects (see Sec. S10 A). For the alternative model:

$$H_{\rm alt} = H_{\rm true} + (f_E - f_U) \log 2. \tag{S8}$$

From Eq. S5, we finally get:

$$\langle \log \mathcal{L}_{\rm alt} \rangle = \langle \log \mathcal{L}_{\rm true} \rangle - f_E(\log 2 - C) - f_U(L_d - 1) \log 2. \tag{S9}$$

Since $\log 2 > C$, now the generative model actually has the highest likelihood: in principle, asymmetric LDA is always able to find the generative model. The ratio of the two log-likelihoods, if the documents are long enough, becomes:

$$\frac{\langle \log \mathcal{L}_{\rm alt} \rangle}{\langle \log \mathcal{L}_{\rm true} \rangle} = 1 - \frac{f_U \log 2}{\log N_w}. \tag{S10}$$

The same equation holds for symmetric and asymmetric LDA, as well as PLSA. Therefore, even if we had infinite amount of information (infinite number of documents and words per document), the ratio of the two likelihoods can actually be very close to 1.

E. Finding the generative model in practice

The number of alternative models is huge. In Sec. S10 A, we show that if each language had a vocabulary size $N_w = 1000$, we can find $\sim K \times 10^{300}$ alternative models (this is a conservative estimate): assuming $f_U = 0.2$ (which would correspond to 10 equiprobable topics), the relative difference in their log-likelihood is $\sim 2\%$ as we can estimate from Eq. S10.

One might argue that, even if the relative difference of the log-likelihood is small, we have not considered that the basin of attraction of the generative model can be very large, so that optimization algorithms might actually be very effective in finding it anyway. Fig. S3 shows that the probability of finding the correct model for equiprobable languages is $\sim 20\%$, while in the heterogeneous case is $\sim 2\%$ (this was computed using variational inference [2]).

F. Model competition in hierarchical data

In the previous sections, we only discussed the difference in likelihood of the generative model and an alternative model with the same number of topics K. In this section, we consider a similar test case for which, however, we fit the data with a model with $K - 1$ topics.

The generative model we consider here is illustrated in Fig S4: we have $K - 1$ topics which have no words in common with any other topic and one bigger topic, say English, which has two subtopics, say "music" and "science", which share some words. Let us call U_M the number of words in one of the English subtopics (music) which cannot be found in the other subtopic, U_S the number of words which can only be found in the other subtopic (science), and C the number of words in common between the two subtopics. We further assume that $U_M = U_S = U$, the subtopics are equiprobable, and given a subtopic, each word is equiprobable. Let us call N_w the number of words in each non-English language, p_E the fraction of English documents and p_k the fraction of documents written in a different language (for sake of simplicity, all languages but English are equiprobable).

This model should be fitted with K topics. However, let us assume that we do not know the exact number of topics (as it is usually the case) and we try to fit the data with $K - 1$ topics. In Fig. S4 we show two possible competing models: the first model correctly finds all the languages, while the second correctly finds the English subtopics but merges two languages.

With similar calculations as above, we can prove (see Sec. S10 B) that the first model has higher likelihood if:

$$2p_k > p_E \frac{U}{C + U}. \tag{S11}$$

The previous equations holds for symmetric LDA, and also asymmetric LDA if $L_d \gg 1$ (the exact expression for asymmetric LDA can be found in Sec. S10 B). If $U = 0$, the first model is always better (there are no subtopics), if $C = 0$, one model is better than the other if it under-fits the smaller fraction of documents. In general, if English is used enough and $U > 0$, the second model better fits the data.

Let us consider a numerical example: consider $p_E = 50\%$, $U_M = U_S = 50$ words and $C = 900$ words ($1,000$ total words in the English vocabulary). This means that 90% of the English words are used by both subtopics. Eq. S11 tells us that we are going to split English in the two subtopics, if there are two other topics to merge with $2p_k < 2.6\%$.

We believe that this is the basic reason why big journals such as Cell and Astronomical journals are split by standard LDA in the Web of Science dataset (see Sec. S9). In general, since real-world topics are likely to display a hierarchical structure similar to the one described here, we argue that heterogeneity in the topic distribution makes standard algorithms prone to find subtopics of large topics before resolving smaller ones.

S2. A NETWORK APPROACH TO TOPIC MODELING

We give here a detailed description of TopicMapping. The method works in three steps.

First, we build a network of words, where links connect terms appearing in the same documents more often than what we could expect by chance. Second, we define the topics as clusters of words in such a network, using the Infomap method [5] and then we compute the probabilities $p(topic|doc)$ and $p(word|topic)$ locally maximizing a PLSA-like likelihood. Finally, we can refine the topics further optimizing the (asymmetric) LDA likelihood via variational inference [2].

a. How to define the network. A corpus can be seen as a weighted bipartite network of words and documents: every word a is connected to all documents where the word appears. The weight ω_a^d of the link is the number of times the word is repeated in document d.

From this network, we would like to define a unipartite network of words which have many documents in common. A very simple measure of similarity between any pair of words a and b is the dot product similarity:

$$z_{a,b} = \sum_d \omega_a^d \times \omega_b^d. \tag{S12}$$

From this definition, it is clear that generic words, like "to" or "of", will be strongly connected to lots of more specific words, putting close terms related to otherwise far semantic areas. A possible way to filter out generic words is to compare the corpus to a simple null model where all words are randomly shuffled among documents.

For this purpose, we need to consider the probability distribution $p(z_{a,b})$ of the dot product similarity defined

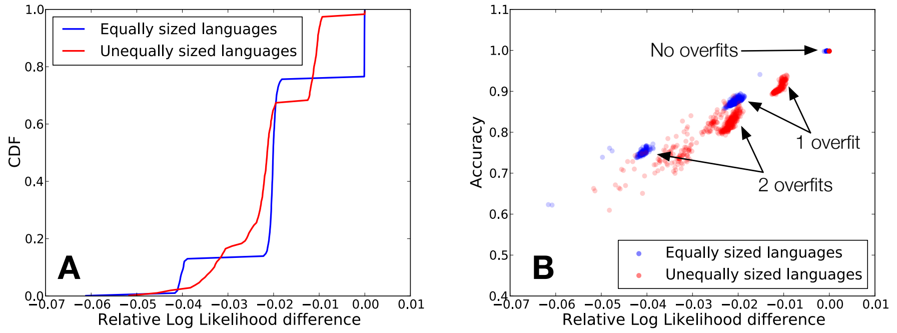

Figure S3: In this test, the corpus has 5000 documents of 100 words each, and the vocabulary of each language has 1000 equiprobable words. In the equally sized case, we consider 10 equiprobable languages, while in the heterogeneous case, we considered 2 languages with probability 30% each, and 8 languages with probability 5%. **A.** Cumulative probability of the relative difference of the log likelihood of the generative model and the one found by the algorithm. **B.** Scatter plot of the relative difference of the log likelihood versus the accuracy of the algorithm (accuracy is the Best Match similarity of the two models, see main text). Clear clusters are visible according to the how many languages are overfitted. Fig. 2 of the main paper, supports the same conclusion also after we removed the assumption that words are equiprobable.

in Eq. S12. We start considering that in the null model each weight ω_a^d is now a random variable which follows a hypergeometric distribution with parameters given by: the total number of words in document d, L_d, the total number of occurrences of word a in the whole corpus, $s_a = \sum_d \omega_a^d$, and the total number of words in the corpus $L_C = \sum_d L_d$. The mean $\langle \omega_a^d \rangle$ is:

$$\langle \omega_a^d \rangle = \frac{L_d \times s_a}{L_C}. \tag{S13}$$

Assuming a large enough number of documents, we can neglect the correlations among the variables ω_a^d and, from Eqs. S12 and S13, we get:

$$\langle z_{a,b} \rangle = \sum_d \langle \omega_a^d \rangle \times \langle \omega_b^d \rangle = \frac{s_a s_b}{L_C^2} \sum_d L_d^2. \tag{S14}$$

Since $z_{a,b}$ is the sum of rare events (if $L_C \gg 1$), its probability distribution can be well approximated by a Poisson distribution $\mathrm{Pois}_{\langle z_{a,b} \rangle}(z)$ with average given by Eq. S14, as shown in Fig. S5.

Finally, our procedure to filter out the noise consists in fixing a p-value, and for all pairs of words a and b which share at least one document, we compute $z_{a,b} - Z_p(s_a, s_b)$, where the latter term is the $(1-p)$-quantile of the Poisson distribution $\mathrm{Pois}_{\langle z_{a,b} \rangle}(z)$. Being more precise, $Z_p(s_a, s_b)$ is the largest non significant dot product similarity:

$$Z_p(s_a, s_b) = \max_x \left\{ x \text{ such that: } \sum_{z=x}^{\infty} \mathrm{Pois}_{\langle z_{a,b} \rangle}(z) > p \right\}. \tag{S15}$$

$z_{a,b} - Z_p(s_a, s_b)$ is the weight of the link between words a and b, if positive.

b. Finding the topics as clusters of words and Local Likelihood Optimization. Once the network is built, we detect clusters of highly connected nodes using the Infomap method [5]. This provides us with a hard partition of words, meaning that words can only belong to a single cluster.

We now discuss how we can compute the distributions $p(topic|doc)$ and $p(word|topic)$, given a partition of words.

We recall that in the probabilistic model of how documents are generated, we assume that every word w appearing in document d has been drawn from a certain topic. We are in the realm of the *bag of words* approximation, and therefore we are completely discarding any information about the structure of the documents. Then, it is reasonable to assume that every time we see a certain word in the same document, it was always generated by the same topic: let us denote this topic as $\tau(w, d)$.

We identify the topic $\tau(w, d)$ with the single module where word w is located by Infomap, $\tau(w)$: in fact, since the partition is hard (no words can sit in different modules), there is no dependency on the documents. Therefore, $p(t|w) = \delta_{t,\tau(w)}$ and:

$$p(w,t) = p(w)\,\delta_{t,\tau(w)} \quad \text{and} \quad p(t|d) = \frac{1}{L_d} \sum_w \omega_w^d\, \delta_{t,\tau(w)}. \tag{S16}$$

It is also useful to introduce $n(w, t) = L_C\, p(w, t)$, which is the number of times topic t was chosen and word w was drawn.

So far, we have got a model where all words are very specific to topics and documents use many topics, which is probably far from being a good candidate generative model. The model can be substantially improved optimizing the PLSA-like likelihood:

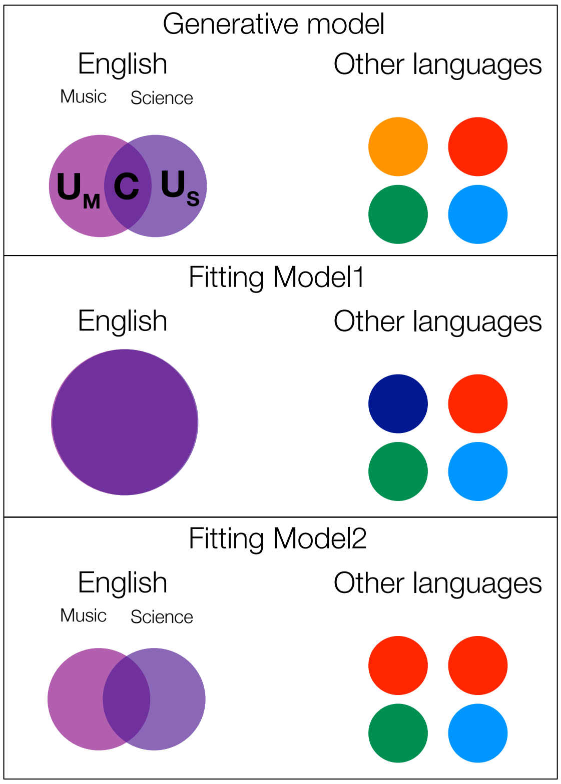

Figure S4: Generative model and two compiting models. In this example, we have $K-1$ languages but one language (English) is bigger than the others and have two subtopics ("music" and "science"). U_M is the number of words in the English vocabulary which can only be found in the music subtopics, U_S is the equivalent for science, whereas C is the number of common words between the two subtopics. If many documents are written in English, Model 2 has a better likelihood than Model 1.

$$\mathcal{L} = \prod_{w,d} p(w,d) = \prod_{w,d} \sum_t p(w|t)\, p(t|d)\, p(d). \tag{S17}$$

We then describe a series of very local moves aimed at improving the likelihood of the model. The local optimization algorithm aims at fuzzing the topics and making documents more specific to fewer topics. For that, it simply finds, for each document, topics which are infrequent (more precise definition follows) and "move" the words drawn from that topic to the most important one in that document.

1. For each document d, we find its most significant topic, τ_d: this is done selecting the topic with the smallest p-value, considering a null model where each word is independently sampled from topic t with probability $p(t) = \sum_w p(w)p(t|w)$. Calling x the number of words which actually come from topic t, ($x = L_d \times p(t|d)$, see Eq. S16), the p-value of topic t is then computed using a binomial distribution, $\mathrm{B}(x; L_d, p(t))$.

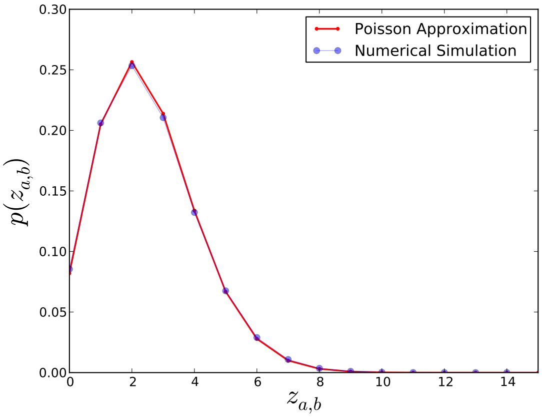

Figure S5: Poisson approximation of the probability distribution $p(z_{a,b})$ of the dot product similarity of words a and b in a randomly shuffled corpus. The occurrences of the words are $s_a = 10$, $s_b = 200$, and there are 1000 documents of length drawn uniformly between 10 and 100 words.

2. For document d, we define the *infrequent* topics t_{in} as those which are used with probability smaller than a parameter: $p(t_{in}|d) < \eta$.

   We consider the most significant topic τ_d (see above) and we increment $p(\tau_d|d)$ by the sum of the probabilities of the infrequent topics, while all $p(t_{in}|d)$ are set to zero. Similarly, $n(w,t)$ has to be decreased by ω_w^d for each word w which belongs to an infrequent topic, and $n(w,\tau_d)$ is increased accordingly.

3. We repeat the previous step for all documents. We then compute $p(w,t) = n(w,t)/L_C$, as well as the the likelihood of the model, $\mathcal{L}_\eta$, where we made explicit its dependency on η.

4. We loop over all possible values of η (from 0% to 50% with steps of 1%) and we pick the model which maximizes $\mathcal{L}_\eta$.

c. LDA Likelihood optimization. The model we find, at this point, can be refined further via iterations of the Expectation-Maximization algorithm optimizing the LDA likelihood. The algorithm follows closely the implementation from [2]. The main difference, however, is that, for computing efficiency, we use sparse data structure, where words and documents are assigned to only a subset of the topics.

In most cases, the model does not change very much and the algorithm converges very quickly. However, if topics are very heterogenous in size, we might encounter situations similar to the one described in Sec. S1 F (see Sec. S8 for an example). In practice, the software records models every few iterations, allowing users to better explore the data.

d. Implementation details. Here, we would like to make a few points more precise.

1. The filtering procedure and the LDA likelihood optimization in TopicMapping are deterministic. Instead, optimizing Infomap's code length uses a Monte Carlo technique, which can be performed multiple times. The number of runs for Infomap's optimization was set to 10 in most tests, although most results barely change with a single run. For measuring the reproducibility in Sec. S6, instead, we used 100 runs, because the topic structure is less sharp and we need some more runs to achieve good reproducibility (each run takes about a minute).

2. After running Infomap, we might find that some words have not been assigned to any topics, because all their possible connections to other words have not been considered significant. In each document which uses any of them, we automatically assign these words to its most significant topic, τ_d.

3. Some (small) topics might have not been selected as the most significant by any document. We remove these topics before the filtering procedure: if we do not, high values of the filter η will yield models where these topics do not appear at all, and this might penalize their likelihood just because the number of topics is diminished.

4. Depending on the application, it might also be useful to remove very small topics even if they were selected as the most significant by a handful of documents (this is especially important to avoid the following LDA optimization to inflate them, see Sec. S4 D). We used no threshold for the synthetic datasets, but we selected a threshold of 10 documents for the journals in Web of Science, and 100 documents for Wikipedia. In the implementation of the software, we let the users choose a threshold for removing small topics.

5. The initial α for LDA optimization was set to 0.01 for all topics.

S3. HELD-OUT LIKELIHOOD AND EFFECTIVE NUMBER OF TOPICS

The most used method for selecting the right number of topics, consists in (i) holding out a certain fraction of documents (say 10% of the corpus), (ii) training the algorithm on the remainder of the dataset, (iii) measuring the likelihood of the held-out corpus for the model obtained on the training set. The best number of topics should be the one for which the held-out likelihood is maximum. Fig. S6 shows that this method tends to give a higher number of topics that the actual one.

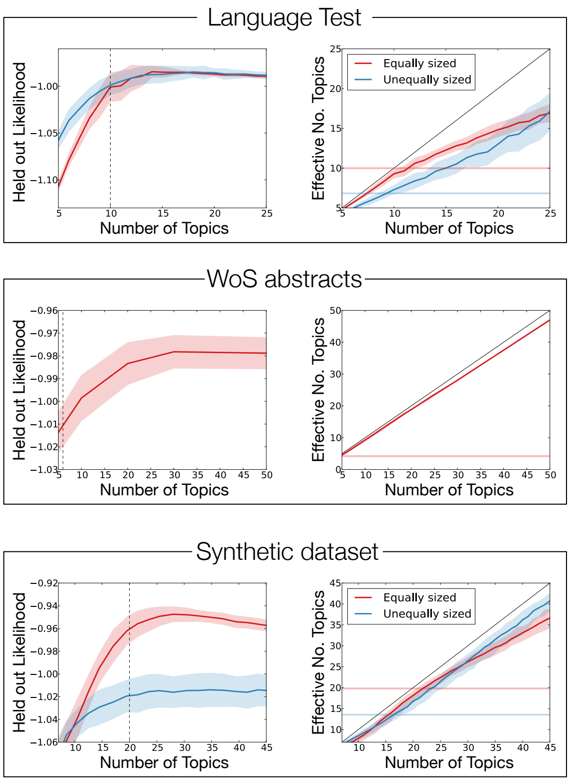

Figure S6: Held-out likelihood and effective number of topics for the three datasets we considered in the main paper. In the language test, we considered 5,000 documents, while, in the synthetic dataset, we set $\alpha = 10^{-3}$ and the fraction of generic words to 25%. The dashed black lines on the left indicated the number of topics K that should have been selected by the method. The black line on the right-hand panels is $y = x$ (the highest achievable value of the effective number of topics) and the horizontal lines are the actual effective number of topics.

We also show that LDA tends to provide models in which $p(topic)$ is fairly close to a uniform distribution. To assess this, we compare the entropy of the topic distribution,

$$h(p_t) = -\sum_{topic=1}^{K} p(topic) \log_2 p(topic), \qquad \text{(S18)}$$

with the maximum possible entropy, i.e. those achieved by equally probable topics: $h_u = \log_2(K)$. In fact, it is easier to compare the exponential entropy [6] of the topic probability distributions: $2^{h(p_t)}$ versus K. The former can be seen as an effective number of topics: it is the number of topics needed by a uniform distribution to achieve the same entropy. Fig. S6 shows that indeed, the effective number of topics is rather close to the input K.

S4. ADDITIONAL ANALYSIS ON THE SYNTHETIC DATASETS

In this section, we present five supplementary sets of results related to the synthetic datasets, presented in Fig. 4 in the main paper. In the first section, we measure the performance of the algorithms in terms of perplexity [2] (a standard measure of quality for topic models) and we show that, for our case, this evaluating method has a fairly low discriminatory power. We then propose a visualization of the comparison between the correct generative model and the ones found by the algorithms we considered. The third section is dedicated to measuring the performance of the methods in case we do not have information about the correct number of topics to input. In the fourth section, we study how the performance of LDA is affected by the initial conditions of the optimization procedure, and we show that they are crucial, as expected. Finally, we compare the performance of TopicMapping before and after running LDA as a refinement step.

A. Perplexity

Fig. S7 shows the performance of the algorithms on the synthetic datasets in terms of perplexity (in Sec. S10 D we explain in detail how perplexity is defined). Algorithms which yield a lower perplexity are considered to achieve a better performance because the model they provide is less "surprised" by a portion of the datasets which they have never seen before. The advantage of this approach is that it can be implemented for generic real-world datasets, where the actual generative model is unknown. However, in the study of our interest, the measure performs poorly in discriminating the methods.

B. Visualizing topic models

Fig. S8 shows a visualization of the performance of the methods on the synthetic datasets. We selected a few runs where the algorithms have got an average performance. The colors allow to show in which way standard LDA and PLSA fail in getting the generative model. Similarly to what happens in the language test, some (small) topics are merged together (indicated by a "*" symbol) and some other topics are overfitted in two or more dialects.

C. Performances for different number of topics.

Here we discuss how the performance of LDA and PLSA changes if we do not know the exact number of topics. In the main paper, we have fed the algorithms the right number of topics, although we have shown (Sec. S3) that it is hard to guess this information. Here, we show what we get setting a different number of topics, but still reasonably close to the right value ($K = 20$). In general, the performance gets worse as we move further from the correct number, although 15 or 25 topics sometimes give

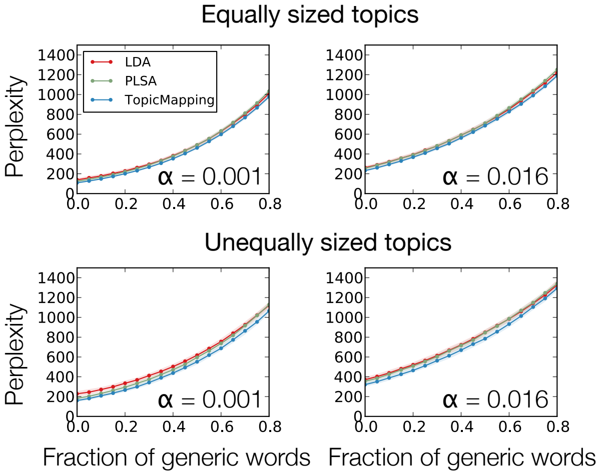

Figure S7: Evaluating the performance of several algorithms on synthetic corpuses measuring perplexity for several values of the parameters (the other parameters are the same as in Fig. 4 in the main paper). Perplexity seems to have low discriminatory power in this test.

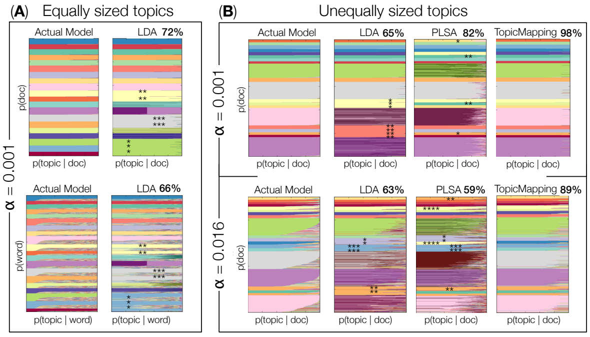

Figure S8: Topic comparison for the synthetic datasets. All parameters are the same as in Fig. 4 in the main paper, and we set the fraction of *generic* words (words which are used uniformly across documents) to 40%. Every rectangle is split in 1000 horizontal bars, one for document. Each bar is divided in color blocks representing topics, with block size proportional to $p(topic|doc)$. The documents are sorted according to their most prominent topic. **A.** Performance of LDA, for equally sized topics and $\alpha = 0.001$. The "*" symbols indicate topics inferred by LDA in which two or more actual topics are merged. Top: comparison for documents. Bottom: same procedure for words: generic words are clearly distinguishable from specific ones. The numbers on the corners are obtained from the topic similarity (see main text). **B.** Unequally sized topics. We show results for two values of α, 0.001 and 0.016. Comparison of documents only is shown. We compare LDA, PLSA and TopicMapping.

slightly better results. We also show that the results do not change very much if we increase the number of documents to 5,000.

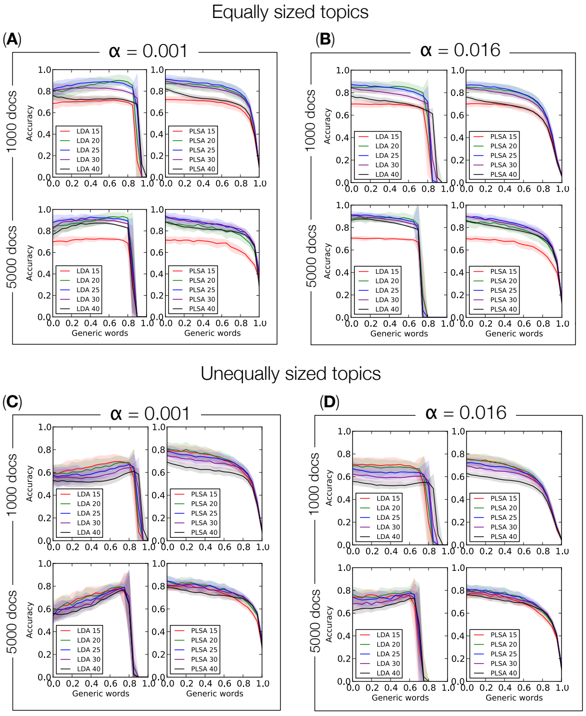

Figure S9: Performance of LDA and PLSA when we input different number of topics. The number of topics in the generative model is 20.

D. LDA initial conditions

In this section, we discuss how the initial conditions affect the performance of LDA optimization. Two standard different ways of initializing the topics have been considered: *random* and *seeded*. The former assigns random initial conditions while the latter uses randomly sampled documents as seeds. We used both throughout the whole study, but we have only shown the *seeded* version in the WoS dataset (the difference in performance is not appreciable, though). Here we compare these two initializations with the performance of the method when we guess the best possible initial conditions, meaning we start from the actual generative model (Fig. S10).

Similarly to the language test, starting from the generative model as initial conditions, we get an outstanding performance, which is also the optimal one in terms of likelihood. However, we checked that if we slightly change the number of topics, the performance gets worse and the likelihood improves. In Fig. S10, we show both performance and likelihood. 24 *topics* refers to a model close to the generative one, but where we added 4 small topics, for which only one single word can be drawn: more precisely, we pick a word at random w_r and we define these small topics with word probability distributions $p(word|topic) = \delta_{word,w_r}$. LDA will grow these small topics to increase the likelihood, overfitting the data and getting a worse performance. This is the main reason why we decided to threshold small topics in the Web of Science dataset (see Sec. S2).

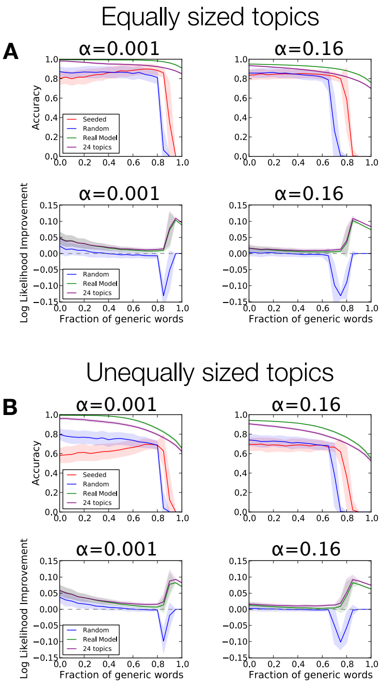

Figure S10: How the initial conditions affect the performance of LDA. We checked four different ways of initializing the topics: *random* and *seeded* are the basic provided options. Real model refers to setting the underlying true parameters as initial conditions. 24 *topics* refers to the right initial conditions where we added 4 small topics peaked on a single randomly chosen word. The log likelihood improvement is defined as the relative difference in the log likelihood we get with the different initial conditions compared to the *seeded* initialization. The plot shows mean values and standard deviations.

E. TopicMapping guess

Here, we show the performance of TopicMapping just for the guess, i.e. before running the LDA optimization (see Fig. S11). We do not show the results for the language test because, in that case, there is no difference at all. In the systematic tests, instead, running LDA as a last step slightly improves the performance of the algorithm, although the difference is not dramatic. We found a remarkable difference only in the Wikipedia dataset (see Sec. S8), where the topic distribution provided by the guess was highly heterogeneous.

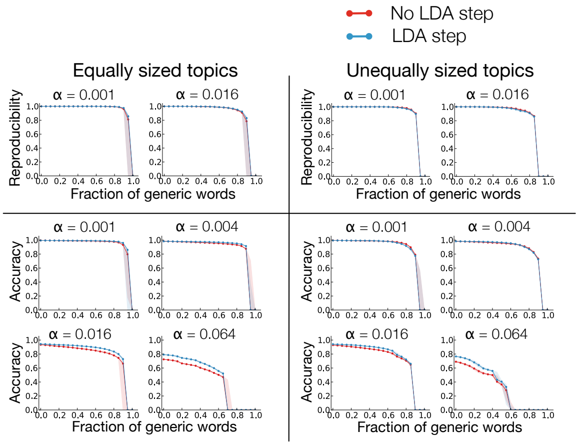

Figure S11: Performance of TopicMapping on the synthetic datasets, before and after running LDA.

S5. ASYMMETRIC LDA

In this section we discuss the results we obtain using asymmetric LDA [3] (`http://mallet.cs.umass.edu`). The algorithm has two main differences respect with the other LDA method we used throughout the study: first, the prior probabilities of using a certain topic are not all equal, and, second, the optimization algorithm is based on Gibbs-sampling rather than variational inference [2].

Fig. S12 shows that the algorithm performs better than symmetric LDA in the language test, although it still struggles recognizing the languages if the number of documents is large and the language probabilities are unequal. The performance on the synthetic graphs is better to standard LDA, (see Fig. S13) for certain parameters only.

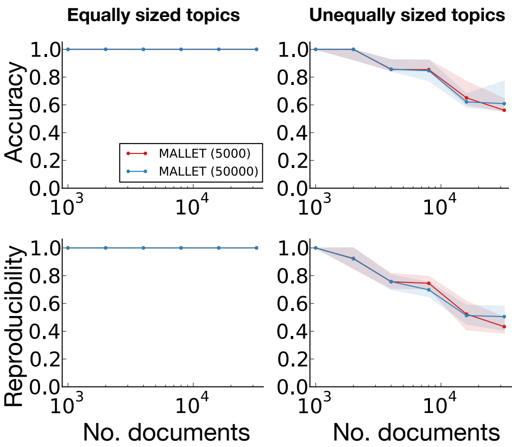

Figure S12: Performance of asymmetric LDA in the language test (same as Fig. 2 in the main text). We used $5,000$ and $50,000$ iteration for Gibbs sampling and we input the correct number of languages in the algorithm. We optimize the hyper parameters each 100 iterations but performance is barely affected by the optimization interval. Curves are the median values and the shaded areas indicate 25th and 75th percentiles.

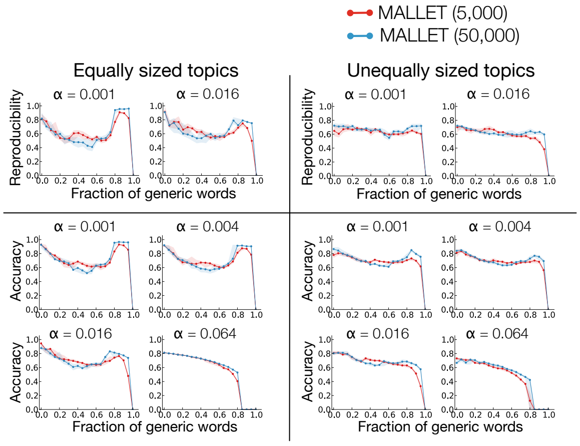

Figure S13: Performance of asymmetric LDA on the tests presented in Fig. 4 in the main paper. Curves are median values and colored areas are 25th and 75th percentiles.

S6. THE HIERARCHY OF WOS DATASET

In this section, we study the subtopic structure of the Web of Science dataset. In fact, we expect to find subtopics in each journal. Although we do not know any "real" topic model to compare with, we can still measure the reproducibility of the algorithm.

Similarly to what we observed above, we find again that standard LDA is not reproducible and the effective number of topics is strongly affected by the input number of topics, see Fig. S14.

For TopicMapping, we observe that the number of topics is affected by the p-value we choose for filtering the noisy words. This is not what happens in all the other tests we have presented so far, which have a rather clear topic structure: therefore, choosing a p-value of 5% or 1% barely makes any difference. Instead, in analyzing Astronomical Journal abstracts, for instance, the topic structure is not so sharp anymore and we do observe that reducing the p-value provides a higher number of topics. Fig. S14 shows the results. For Astronomical Journal, with a p-value of 5% we only observe one topic. Decreasing the p-value to 1% we start observing sub-topics like: "galaxi* observ* emiss*", "star cluster metal" or "orbit system planet". For Cell, we also observe that the effective number of topics increases for smaller p-values. However, in both cases, TopicMapping is much more reproducible.

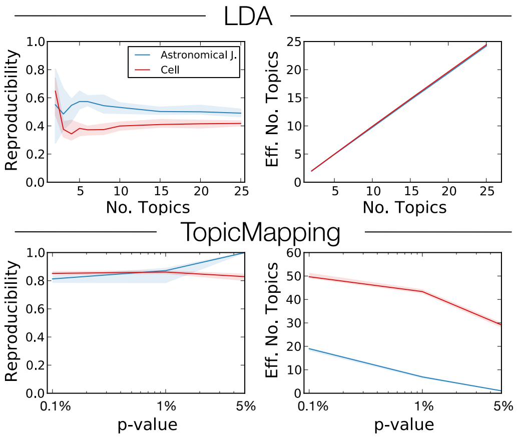

Figure S14: Reproducibility and effective number of topics for LDA and TopicMapping for the scientific abstracts of Astronomical Journal and Cell. The number of topics can be tuned in LDA changing the input number of topics. Similarly, in TopicMapping the resolution can be tuned to some extent filtering words with different p-values. However, this effect is present only in corpora with a less defined topic structures than the language test or the synthetic graphs, for instance. Median and 25th and 75th percentiles are shown.

S7. COMPUTATIONAL COMPLEXITY

For a given vocabulary size, LDA's complexity is proportional to the number of documents times the number of topics.

The computational complexity of TopicMapping's guess is also linear with the number of documents. In particular, building the graph costs $O(\sum_d u_d^2)$, where u_d is the number of unique words in document d. Infomap's complexity is of the same order of magnitude (smaller if we filter links), because the algorithm runs in a time proportional to the number of edges in the graph. Local PLSA-likelihood optimization is also linear in the number of documents, and can scale better than LDA with the number of topics, if the assignments of words to topics is sparse. In fact, we use sparse data structures to compute the topics for each document and each word, meaning that for each document, for instance, we do not handle a list of all topics (including never used topics), but only a list of the topics the document actually makes use of. Indeed, this enables the algorithm to scale much better with the number of topics (see Fig. S15) on the synthetic datasets.

As a further example, to analyze the WoS corpus, TopicMapping takes $\sim$ 25 minutes on a standard desktop computer. LDA takes $\sim$ 20 minutes for finding models with 6 topics and 120 minutes for models with 24 topics.

S8. TOPICS IN WIKIPEDIA

In the main paper, we have shown the results of TopicMapping after running LDA optimization for one single iteration. The inset was obtained running the algorithm on the sub-corpus consisting of all words which were more likely drawn from the first topic. Fig. S16, instead, shows the results after the full LDA optimization. For comparison, we also show the results starting the algorithm with random initial conditions. Interestingly, in this dataset, LDA optimization changed our guess significantly. This is not what happens in any of the other datasets we have tested, for which the topics in our guess were less heterogeneous (see Sec. S1 F).

S9. TOPICMAPPING AS A LIKELIHOOD OPTIMIZATION METHOD

Here we discuss to which extend TopicMapping provides models with better likelihood compared to standard LDA. Indeed, in controlled test cases as the synthetic tests we have presented in this work, TopicMapping generally finds better models in terms of likelihood and this explains why it performs better (the actual generative model has the highest likelihood).

In real cases, as we discussed in Sec. S1 F, the likelihood can be maximized splitting large topics in subtopics and merging smaller topics. Therefore, if we compare the likelihood found by TopicMapping and the one found by variation inference [2] as a function of the number of topics, TopicMapping does not provide models with higher likelihood. However, this comparison heavily penalizes TopicMapping, which often provides models with a broad distribution of topics, and many of them are barely used at all. We then argue that comparing models with the same number of effective topics is a more fair comparison. Doing so, Fig. S17 shows that, indeed, TopicMapping's models have often higher likelihood. However, the difference is not dramatic as we can see from the inset of Fig. S17, because of the degeneracy of the likelihood landscape.

S10. APPENDICES

A. Likelihood of English documents in the language test

In this section, we compute the likelihood of the alternative model for the English documents (Sec. S1). Let us call a and b the number of English words in the first group and in the second group respectively. We have that:

$$a + b = N_w \qquad \text{and} \qquad af_1 + bg_1 = 1,$$

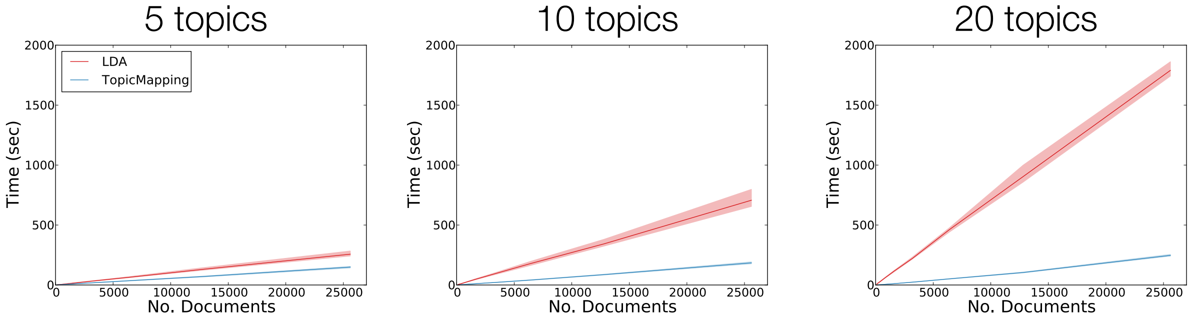

Figure S15: Time needed for the execution of standard LDA and TopicMapping (before the LDA step) on synthetic corpora. Similarly to the other tests, we used a fixed vocabulary of 2000 unique words and 50 words per document. We set $\alpha = 10^{-3}$ and the generic words are 30%. Both algorithms' complexity is linear in the number of documents. However, TopicMapping can be significantly faster if the number of topics is large.

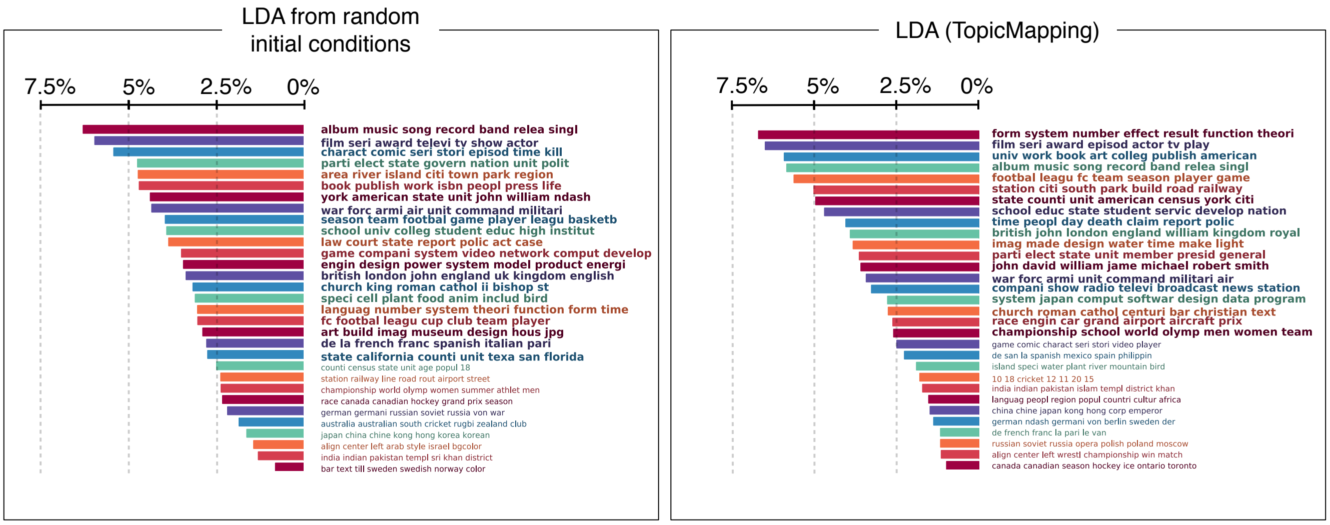

Figure S16: Topic found in a large Wikipedia sample by standard LDA and TopicMapping with full LDA optimization. For each topic, we show the top 7 words. Bold fonts are used for the top topics which account for 80% of the total.

and the equivalent holds for f_2 and g_2. When we write an English document, we randomly sample words from the English vocabulary. This means that the probability that n_a words fall in the first group, and $n_b = L_d - n_a$ in the second, follows a binomial distribution:

$$p(n_a) = \frac{L_d!}{n_a!(L_d - n_a)!}\left(\frac{a}{N_w}\right)^{n_a}\left(1 - \frac{a}{N_w}\right)^{L_d - n_a}. \quad \text{(S19)}$$

The last ingredient is how to decide which dialect a document should be fitted with. Let us define a threshold T such that, if $n_a \geqslant T$ we use the first dialect, and we use the second otherwise. Without loss of generality, we also assume that $T \geqslant 1$, because otherwise we go back to the one single dialect case (Eq. S1).

Let us call $\mathcal{L}'_E$ the likelihood of an English document in this model. Its average can be written as:

$$\langle \log \mathcal{L}'_E \rangle = \sum_{n_a = T}^{L_d} p(n_a) \log \mathcal{L}'_1(n_a) + \sum_{n_a = 0}^{T-1} p(n_a) \log \mathcal{L}'_2(n_a), \quad \text{(S20)}$$

where

$$\log \mathcal{L}'_1(n_a) = n_a \log f_1 + (L_d - n_a) \log g_1,$$

and the same equation holds for $\mathcal{L}'_2$ replacing f_1 with f_2 and g_1 with g_2.

We can compute the optimal values for f_1 and f_2 simply setting derivatives to zero:

$$\frac{\partial \log \mathcal{L}'}{\partial f_1} = \sum_{n_a = T}^{L_d - 1} p(n_a)\left(\frac{n_a}{f_1} + \frac{a(L_d - n_a)}{a f_1 - 1}\right) + p(L_d)\frac{L_d}{f_1} = 0.$$

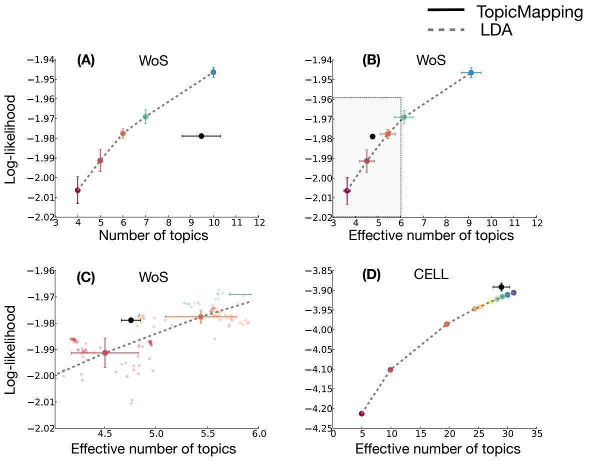

Figure S17: Comparison between TopicMapping and standard LDA in terms of likelihood, for the Web of Science dataset we described in the main paper, and for Cell. Colors represent the results from standard LDA with different number of topics as input. **A**. TopicMapping does not provide better likelihood if we compare models with the same number of topics. **B**. Comparison for Web of Science between model with the same effective number of topics. **C**. Zooming in the shaded area in **B**, we can see that TopicMapping performs better on average, although standard LDA is sometimes comparable. Indeed splitting Cell and merging Schizophrenia Bullettin and American Economic Review gives very comparable results in terms of likelihood. **D**. Comparison for Cell.

If we call:

$$\omega_1 = \sum_{n_a=T}^{L_d} p(n_a), \quad m_{a_1} = \frac{\sum_{n_a=T}^{L_d} p(n_a) n_a}{\omega_1},$$

$$p_{a_1} = \frac{m_{a_1}}{L_d} \text{ and } \mu_a = L_d a / N_w,$$

the optimal f_1 and f_2 can be written as:

$$f_1 = \frac{p_{a_1}}{a} \quad \text{and} \quad f_2 = \frac{p_{a_2}}{a}.$$

ω_1 is how often we use the first dialect, μ_a is the expected number of words which fall in the first group of English words, p_{a_1} is the probability of using words from the first group, given that we are using the first dialect. We also have that:

$$\omega_2 = 1 - \omega_1 \quad \text{and} \quad \omega_1 p_{a_1} + \omega_2 p_{a_2} = p_a = \frac{a}{N_w}.$$

For group b, we have:

$$p_{b_1} = 1 - p_{a_1} \quad \text{and} \quad p_{b_2} = 1 - p_{a_2}.$$

We can now compute the expected log-likelihood of Eq. S20:

$$\frac{\langle \log \mathcal{L}_E^{'} \rangle}{L_d} = \omega_1 [p_{a_1} \log \frac{p_{a_1}}{a} + p_{b_1} \log \frac{p_{b_1}}{b}] +$$

$$\omega_2 [p_{a_2} \log \frac{p_{a_2}}{a} + p_{b_2} \log \frac{p_{b_2}}{b}].$$

Calling the entropy of a binary variable $h(p) = -p \log p - (1-p) \log(1-p)$, we get:

$$\frac{\langle \log \mathcal{L}_E^{'} \rangle - \mathcal{L}_{\text{true}}^{'}}{L_d} = -\omega_1 h(p_{a_1}) - \omega_2 h(p_{a_2}) + h(p_a). \quad \text{(S21)}$$

Now, the problem is to find, for given L_d and N_w, which choice of the parameters a and T maximizes Eq. S21. It turns out that there are two different regimes depending on the condition $N_w \geqslant L_d$.

If $N_w \geqslant L_d$, a possible strategy is to assume $T = 1$. This means that we use the second dialect if (and only if) there are no words in the first group. This means that $p_{a_2} = 0$.

In fact, using the equations above, we get:

$$\omega_1 = 1 - \left(1 - \frac{a}{N_w}\right)^{L_d} \quad p_{a_1} = \frac{a}{N_w \omega_1} \quad \text{and} \quad p_{a_2} = 0.$$

It is possible to prove that for $L_d \gg 1$ and $N_w \geqslant L_d$ the maximum is attained for:

$$\omega_1 = \frac{1}{2} \quad \text{and} \quad p_a = \frac{\log 2}{L_d},$$

and disregarding size effect due to a being an integer:

$$\langle \log \mathcal{L}_E^{'} \rangle - \mathcal{L}_{\text{true}}^{'} \simeq (\log 2)^2 \simeq 0.4804.$$

In the second case, $L_d \geqslant N_w$, we restrict ourselves to considering $T = L_d / N_w$. In the limit $L_d \gg 1$, using the Gaussian approximation of the binomial distribution in Eq. S19, we get:

$$\omega_1 = \frac{1}{2} \quad \text{and} \quad p_a \simeq \frac{a}{N_w} + \sqrt{\frac{2a}{\pi N_w L_d}},$$

$$\langle \log \mathcal{L}_E^{'} \rangle - \mathcal{L}_{\text{true}}^{'} \simeq \frac{N_w}{\pi (N_w - a)}.$$

If also $N_w \gg 1$, the difference is independent of a:

$$\langle \log \mathcal{L}_E^{'} \rangle - \mathcal{L}_{\text{true}}^{'} \simeq \frac{1}{\pi} \simeq 0.3183.$$

In conclusion, the log-likelihood per document of the alternative model (given that we use English), is bigger that the one of the generative model, and, remarkably, the difference varies from roughly 0.5 to 0.3, so that it is substantially independent of all the parameters of the model. Since we can divide the English words in two arbitrarily groups, we can actually have a large number of alternative models. For instance, if we have $L_d = 100$, and $N_w = 1000$, the model with highest likelihood splits English in two groups of 7 and 993 words, so that the number of alternative models becomes:

$$\binom{1000}{7} \simeq 10^{17},$$

and there are many more alternative models with slightly smaller likelihood: for instance using $a = 500$, the likelihood of the alternative model is 99.6% the likelihood we obtain for $a = 7$, but the number is $\simeq 10^{300}$. All these models are likely local maxima of the log-likelihood for Expectation-Maximization algorithms.

B. Derivation of Eq. S11

Let us start computing $\log \mathcal{L}'_{\mathrm{M1}}$, the log-likelihood per document for the model where the subtopics are merged and all languages are recovered. We recall that the symbol $'$ means that the likelihood is computed given that we know the topics of the document.

If we merge the two English subtopics, the common words (C) have probability $1/(U + C)$ while the words only used in one of the two subtopics ($2U$) have probability $1/(2U + 2C)$. Therefore, the average log-likelihood per English document in Model 1 is:

$$\langle \log \mathcal{L}'_E \rangle = -L_d \frac{C}{U+C} \log(U+C) - L_d \frac{U}{U+C} \log(2U+2C),$$

which can be re-written as:

$$\langle \log \mathcal{L}'_E \rangle = -L_d \log(U+C) - L_d \frac{U}{U+C} \log 2.$$

The log-likelihood per non-English document is:

$$\log \mathcal{L}'_{NE} = -L_d \log N_w.$$

Instead merging two languages which are not English (Model 2), we get:

$$\log \mathcal{L}'_{M} = -L_d \log 2N_w.$$

The difference in the average log-likelihood between the two models becomes (we recall that p_k is the probability of any non-English language):

$$\langle \log \mathcal{L}'_{\mathrm{M1}} \rangle - \langle \log \mathcal{L}'_{\mathrm{M2}} \rangle = -p_E L_d \frac{U}{U+C} \log 2 + 2 p_k L_d \log 2.$$

Eq. S11 follows from the equation above. For asymmetric LDA, we also have to consider the difference in the language entropies. Accounting for that, we get:

$$\langle \log \mathcal{L}_{\mathrm{M1}} \rangle - \langle \log \mathcal{L}_{\mathrm{M2}} \rangle =$$

$$-p_E (L_d \frac{U}{U+C} - 1) \log 2 + 2 p_k (L_d - 1) \log 2.$$

Then, Model 1 has higher likelihood than Model 2 if:

$$2 p_k (L_d - 1) > p_E (L_d \frac{U}{U+C} - 1).$$

The correction from Eq. S11 is $O(L_d^{-1})$.

C. The Dirichlet distribution

The Dirichlet distribution is frequently used in Bayesian statistics since it is the conjugate prior of the multinomial distribution. The distribution is parameterized by K values α_{topic}, and the support of the function is the standard $(K-1)-$simplex, i.e. the set of vectors $\boldsymbol{x}$ of dimension K such that $\sum_i x_i = 1$ and all $x_i \geqslant 0$. Clearly, $\boldsymbol{x}$ can be interpreted as a probability distribution. Moreover $\langle x_i \rangle = \alpha_i / \sum_{topic} \alpha_{topic}$.

In generating the synthetic corpus, for each document, we use the same $\alpha_{topic} = K \times p(topic) \times \alpha$ to draw $p(topic|doc)$ from the Dirichlet distribution. In fact, even letting α depend on documents (but not on topics), this definition makes sure we get back the pre-defined topic probabilities, since $\langle p(topic|doc) \rangle = p(topic)$. Fig. S18 shows how $p(topic|doc)$ depends on α in the simple case of 20 equiprobable topics.

D. Measuring Perplexity

Perplexity is the conventional way to evaluate topic models' accuracy [2]. Here, we briefly review how it is computed.

The spirit is to cross validate the model, whose parameters have been computed on a trained set of documents, looking at how well the model fits a small set of unseen documents. Therefore, the procedure is (i) to held out a fraction of documents from the corpus (typically 10%), (ii) train the algorithm using the remaining 90% of the documents, (iii) infer the topic probabilities for the unseen documents $p(topic|doc)$ without changing the topics, i.e. $p(word|topic)$, (iv) compare the actual word frequencies $p(word|doc)$ of the unseen documents with the topic mixture $q(word|doc) = \sum_{topic} p(topic|doc) \times p(word|topic)$.

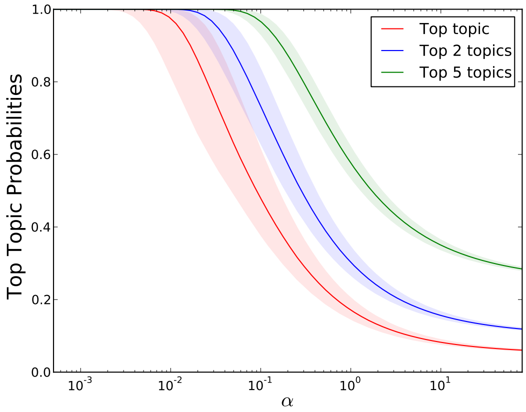

Figure S18: For each document, we extract $p(topic|doc)$ from the Dirichlet distribution for several values of α, setting 20 equally probable topics. We then measure the probability of its most prominent topic (red curve) as well as the sum of the two and five largest topic probabilities (blue and green curves). The plot shows the median together with the 25% and 75% quantiles. Small values of α lead to documents which are mostly assigned to one single topic: for instance, for $\alpha = 10^{-3}$, the probability of the top topic is basically 1 and all others are zero. For $\alpha = 10^{-1}$, the top topic has roughly 0.5 probability, the second one has 0.2 and all the last 15 combined have less than 0.05. For large values like $\alpha \gtrsim 10^{2}$, all topic probabilities tend towards equality, $p(topic|doc) = 0.05$: this means that documents cannot be classified as they use all topics with equal probability.

E. Algorithms' usage details

For LDA, we used the implementation that can be found from `http://www.cs.princeton.edu/\~\blei/lda-c/index.html`. The stopping criterion in running LDA and PLSA was that the relative improvement of the log likelihood bound was less than 10^{-5} with respect to the previous iteration. In running LDA we let the algorithm optimize α as well. The initial value was set to $\alpha = 1$.

[1] T. Hofmann, in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence* (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999), UAI'99, pp. 289–296, ISBN 1-55860-614-9.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan, J. Mach. Learn. Res. **3**, 993 (2003), ISSN 1532-4435.

[3] H. Wallach, D. Mimno, and A. McCallum, Advances in Neural Information Processing Systems **22**, 1973 (2009).

[4] D. Sontag and D. Roy, Advances in Neural Information Processing Systems **24**, 1008 (2011).

[5] M. Rosvall and C. Bergstrom, Proc. Natl. Acad. Sci. USA **105**, 1118 (2008).

[6] L. Campbell, Probability Theory and Related Fields **5**, 217 (1966).