IDIAP RESEARCH REPORT

Textual Data Representation

Mikaela Keller [1]   Samy Bengio [2]

IDIAP–RR 03-74

December 27, 2003

Dalle Molle Institute for Perceptual Artificial Intelligence • P.O.Box 592 • Martigny • Valais • Switzerland

phone +41 − 27 − 721 77 11
fax +41 − 27 − 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

1 IDIAP, CP 592, 1920 Martigny, Switzerland, mkeller@idiap.ch
2 IDIAP, CP 592, 1920 Martigny, Switzerland, bengio@idiap.ch

IDIAP Research Report 03-74

Textual Data Representation

Mikaela Keller Samy Bengio

December 27, 2003

Abstract. We address in this report the problem of representing formally textual data. First, this problem is replaced in the context of automatic text processing. Then, the weaknesses of the basic document representation, *i.e.* the bag-of-words representation, are explained and some state-of-the-art methods claiming to overcome these weaknesses are reviewed. Moreover we propose a novel graphical model, the Theme Topic Mixture Model, which also claims to do so, in addition of giving a probabilistic framework in which documents are considered.

Contents

1 Introduction

Due to the increasing use of Internet, huge amount of texts in digital format are now available. In order to process them automatically, one must represent them formally in an appropriated manner.

In Text Categorization and other supervised related problems, *bag-of-words* document representation is common. Documents are represented by a vocabulary's sized vector $(w_1, ..., w_{|V|})$, w_i being in general a function of the number of times the i^{th} word of the vocabulary appears in the document.

There are at least two problems with this representation. The first one is based on the following idea: Imagine two documents discussing the same topic but this topic being described with many different words, in the *bag-of-words* representation they won't be close, when actually they are intuitively similar. This problem is referred to as the *synonymic* property of language, where for example the two words *car* and *automobile* are different but have the same meaning. On the other hand, human languages also have a property of *polysemy*, where the same word can refer to different concepts. For example, the word *surfing* can imply a sport activity or a search over Internet. Documents sharing words but dealing with different concepts will be considered as similar in this representation.

The second problem can be summarized as follows. Due to the size of the vocabulary (order of magnitude : 20000), the dimension of the representation space is very high. That means a lot of parameters to estimate which leads easily to a curse of dimensionality problem, unless you have enough examples to fill the document space. The problem is that due to the cost of labelling documents, annotated corpora usually contain a small number of documents, and thus the models of the data tend to overfit. Simultaneously, large corpora of non-annotated data are available, which could be used to better represent the distribution of words in the language.

To summarize these problems, we can say that there are too many words and that they have too many meanings to be basic components of a document. We would prefer a more compact and more general representation, one that highlights a little number of concepts or topics present in the documents. Linking that to the existence of important quantities of non-annotated data, we can think of a semi-supervised approach, where the information contained in unlabelled data helps improving the performance of the supervised task.

The basic idea is thus to use the unlabelled data to construct a representation of documents, in which, we can then map the labelled data. Extracting information from unlabelled data has previously been done using two kinds of approaches, departing both from the *bag-of-words* representation:

- **Exploratory data analysis methods** based for example on linear algebra, like the so-called LSI (Latent Semantic Indexing), which performs a Singular Value Decomposition (SVD) on the term-by-document matrix, and chooses the first resulting eigen vectors as new representation basis. The problem with this approach is that the complexity of SVD is directly related to the number of documents. Hence, it is not tractable for large databases.

- **Probabilistic models** that try to approximate the distribution of the data. Among these methods PLSA (Probabilistic Latent Semantic Analysis) and LDA (Latent Dirichlet Allocation) seem interesting with respect to our goal. They try to model the fact that words should be attached to common unknown topics. They model this idea with hidden variables.

We will here concentrate on the second approach since it is more interpretable from a theoretic point of view and more flexible. However, both PLSA and LDA have known difficulties. In PLSA the number of parameters to be estimated grows linearly with the number of documents, and thus the model tends to overfit the training data. LDA doesn't have this problem, but due to the complexity of the modelisation, its computation is intractable directly and several approximations need to be taken. We will propose in the following a model that tries to avoid these two difficulties.

Both categories of methods can be used in an unsupervised manner. Unfortunately, the performance of unsupervised methods is difficult to evaluate directly. It is more convenient to evaluate their perfomance in a supervised context. For example, we can use the representation obtained for documents by these methods for a supervised task and compare the performance to the one resulting from the same task with bag-of-words representation. Hence, we will concentrate on supervised issues such as Text Categorization and Information Retrieval. These two issues are well known and deeply studied. Text Categorization is the task of assigning one or several predefined categories to documents. The main difference with common classification is that each document can belong to several classes. The Information Retrieval task, is the one which consists of finding the documents in a corpus that are the most related to a user's query.

In the next section we present a review of the state-of-the-art literature which is related to this subject. Then follows the presentation of the Theme Topic Mixture Model (TTMM), a novel probabilistic model inspired by LDA, that tries to overcome the problems cited above. This section includes some preliminary results on a benchmark database. Finally in the last section some future directions are exposed.

2 State-of-the-Art

In the following, different subjects that are related to the textual data representation, are reviewed. We begin with a brief presentation of two automatic text processing tasks: Information Retrieval and Text Categorization. Then follows a description of three different document representation methods: Latent Semantic Indexing, Probabilistic Latent Semantic Analysis and Latent Dirichlet Allocation. The last two methods are brought together in a specific sub-section, to emphasize that they are both probabilistic models.

2.1 Automatic Text Processing Tasks

Both Information Retrieval (IR) and Text Categorization (TC) tasks make the assumption that the order of words in documents can be neglected and the frequencies of words in documents are sufficient information. Hence the preprocessing step and the document indexing (*i.e.* document representation), which reflect the implication of this assumption, are common to both tasks as shown in Fig.1.

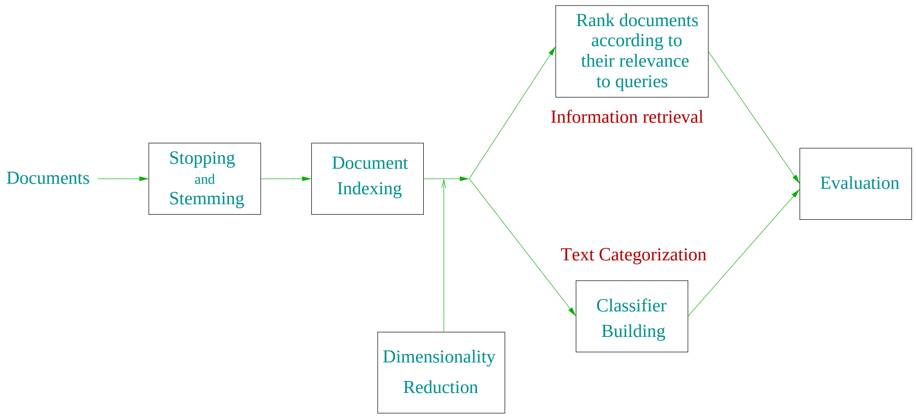


Figure 1: TC and IR Steps

The first box drawn in Fig.1 corresponds to the preprocessing step. The corpus documents are usually "stopped", *i.e.* chosen *topic neutral* words called **stopwords**, such as *a, the, about, as,* etc, are removed from documents. This is done in order to reduce the corpus size and attempts to remove the noise produced by the high frequency of poor informative words. In many approaches but not all, the words are then replaced in the corpus by their stem. For example *connecting, connected, connection,* *connections,* will be replaced by their common stem, *connect.* This step called **stemming** reduces the vocabulary size and attempts to reflet the fact that words with the same stem have similar meanings.

In most of TC and IR literature, the **bag-of-words** representation of document is reported. This representation is constructed, as explained in [13], from the training set Tr extracted from the corpus of texts $\mathcal{C}$. First $\mathcal{V}$ the vocabulary of Tr, that is the list of terms that appears at least once in at least one document of Tr, is collected. Then each document d of $\mathcal{C}$ is described as a vector of term weights $(w_1, \ldots, w_M)$, with $M = |\mathcal{V}|$. The component w_i represents the weight of the i^{th} vocabulary's word in the document. There are several ways to compute it, but its general form is the following:

$$w_i = Local(i, d) \cdot Global(i, Tr)$$

The local weight is for example an indicator of the presence/absence of the word in the document, or a count of the number of times it appears in the document, and the global weight can be a count of the number of documents the word appears in. Various combinations have been experimented as reported in [9], however the most commonly used is the so-called *tfidf* weighting,

$$w_i = tf(t_i, d) \cdot \log\left(\frac{|Tr|}{df(t_i)}\right)$$

where $tf(,)$ is the term frequency and $df()$ is the document frequency of a term in Tr.

The next box in Fig.1, corresponds to the **Dimensionality Reduction** (DR) step which is related to the data representation. It reduces the size of the representation space, in order to speed up computations and to try to avoid overfitting. Methods for DR can be divided into two categories:

1. DR by term selection
2. DR by term extraction

The first category of methods is generally used in TC. They attempt to select from the original vocabulary $\mathcal{V}$, the sub-set $\mathcal{V}'$ of terms (with $\mathcal{V} \ll \mathcal{V}'$) that, when used for document representation, provides the best results. As stated in [13], the choice of this subset is generally done by a filtering approach, that is selecting the terms that reach the highest scores according to a function that measures the "importance" of each term for the task. This function can simply be the *document frequency* [16][6], or a more complex information-theoretic function such as *information gain* [16][15] , *mutual information* [16][6], *chi-square*[12][16][15], etc.

The second approach, in which we are more interested, attempts to create new coordinates representing documents, from the old coordinates. This approach reduces the dimensionality of document space and in addition tries to avoid the problems of polysemy and synonymy reported above, characteristic of words. These methods should contain two steps, one in which the new terms are extracted from the old ones, in general in an unsupervised way, and one step for converting the original document representation in the representation based on the new synthesized dimension. Among these methods LSI, PLSA and LDA have been used in this scope and will be described in section 2.2 and 2.3.

After DR, comes the task specific step, which differs in accordance with the scope. Although other tasks are present in the literature, here we only present briefly TC and IR, because they are the more studied.

- The scope of TC is to assign to an unseen document one or several categories from a predefined set. For that aim a classifier is built on a training set of labelled documents. According to [13], in general a binary classifier is trained for each category.
- The scope of IR is to rank a set of documents according to their relevance to a user's query. As stated in [11] and [3], the most common ways of solving this problem are: The Vector Space Model where documents are ranked according to their computed similarity to the query, and the Probabilistic Modelling approach where the probability of relevance knowing the query and the document is modelled.

Finally, to complete both TC and IR processes an **evaluation** of the results is required. As shown in Fig.1, the same evaluation scheme is used for both. The performance of the systems is measured in terms of *precision* π and *recall* ρ, where:

$$\begin{aligned}
\rho &= \frac{\text{assigned and correct documents}}{\text{total correct documents}}, \text{ for TC results} \\
&= \frac{\text{retrieved and relevant documents}}{\text{total relevant documents}}, \text{ for IR results} \\
\pi &= \frac{\text{assigned and correct documents}}{\text{total assigned documents}}, \text{ for TC results} \\
&= \frac{\text{retrieved and relevant documents}}{\text{total retrieved documents}}, \text{ for IR results.}
\end{aligned}$$

The higher the precision and the recall, the better the systems are. Since these two values are linked, usually the recall-precision curve is plotted and some synthetic measures are derived from the curve.

As mentioned above, the topic of this work is related to the second kind of DR methods. In the following, three of these methods present in the literature are reviewed.

2.2 LSI

Latent Semantic Indexing (LSI), that is the application of Singular Value Decomposition (SVD) to the bag-of-words representation was originally proposed by [4]. The idea of LSI is to perform SVD of the term-by-documents matrix:

$$X_{N\times M} = \begin{pmatrix} w_{11} & \dots & w_{1M} \\ \vdots & \vdots & \vdots \\ w_{N1} & \dots & w_{NM} \end{pmatrix}$$

where, w_{ij} is the value of the i^{th} word in the j^{th} document, $N = |Tr|$ and $M = |\mathcal{V}|$, to find a new system of axes fitting the data, with the hope that the data is in fact well represented with **few** of these new dimensions.

SVD can be seen as a decomposition of the term-by-document matrix, in this way:

$$X = V\Sigma U', \qquad \Sigma = \begin{pmatrix} \sqrt{\lambda_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sqrt{\lambda_P} \end{pmatrix} \tag{1}$$

where $P = \min(M, N)$, $U = (\underline{u_1}, \dots, \underline{u_P})$, $\underline{u_j}$ is the eigen vector corresponding to the j^{th} eigen value λ_j of $X'X$ and $V = (\underline{v_1}, \dots, \underline{v_P})$, $\underline{v_j}$ is the eigen vector corresponding to the j^{th} eigen value λ_j of XX'.

Equation (1) can also be seen as the reconstruction of the initial data from the data projected in the new axes $\{\underline{u_1}, \dots, \underline{u_P}\}$. If instead of taking P, we choose the first Q, with $Q < P$, we have an approximation of the data:

$$\hat{X} = \hat{V}\hat{\Sigma}\hat{U}', \qquad \hat{\Sigma} = \begin{pmatrix} \sqrt{\lambda_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sqrt{\lambda_Q} \end{pmatrix}. \tag{2}$$

The reconstruction quality can be evaluated by the *variance percentage*:

$$\tau_Q = \frac{\sum_i \sum_j \hat{w}_{ij}^2}{\sum_i \sum_j w_{ij}^2} = \frac{\sum_{\alpha \le Q} \lambda_\alpha}{\sum_\alpha \lambda_\alpha}.$$

According to [4] and [7], LSI performs better in IR tasks than basic term matching with Vector Space Model in some databases, and worst in some others.

However, for large databases LSI becomes intractable, since the computation of eigen values and vectors requires the inversion of the matrix $X'X$ or the matrix XX', inversion which has a complexity of $\mathcal{O}(P^3)$. Hence, we cannot use LSI to extract information about relation between words in big databases.

2.3 Probabilistic Models

The remainder of this report is focused on probabilistic models of textual data. They try to overcome the problems mentioned in the introduction, that is polysemy and synonymy of words, and high dimensionality of vocabulary. They assume the existence of unobserved variables, the "topics", responsible of the generation of the observed data. It usually implies an unsupervised modelisation of the data. This modelisation is obtained by optimizing a criterion, such as the likelihood or the perplexity. Nevertheless, the most accurate way to evaluate whether it is a "good" representation of documents is to use it for a supervised task (such as TC or IR). They can be used, for example, as explained above as term extraction methods for Dimensionality Reduction of the document representation.

2.3.1 PLSA

In the Probabilistic Latent Semantic Analysis(PLSA), as explained by T. Hofmann in [7], we assume that we have two variables, one observed: the ocurrence of a word w in a particular document d, and one unobserved: t a latent topic or concept, with possible values in $\{1, \ldots, K\}$, K being an hyperparameter that must be chosen. PLSA defines a generative model for **word/document co-occurrences**. The assumption is that each word w_j in a given document d_δ is generated from a latent topic t, *i.e.* a word is conditionally independent from its original document given the latent topic it was generated from. The data generation process can be described as follows:

1. Select a document index δ with probability $P(\delta)$
2. Pick a latent topic $t = k$ with probability $P(t = k|d_\delta)$
3. Generate a word w_j with probability $P(w_j|t = k)$

This generative process is summarized by the joint distribution of a word w_j, a latent topic $t = k$, and a document d_δ :

$$P(w_j, t = k, d_\delta) = P(\delta)P(t = k|d_\delta)P(w_j|t = k),$$

and the joint distribution of the observed data:

$$P(d_\delta, w_j) = P(\delta) \sum_{k=1}^{K} P(t = k|d_\delta)P(w_j|t = k).$$

So each word in a document is seen as a sample from a mixture model where mixture components are multinomial $P(w_j|t = k)$, and the mixing proportions are $P(t = k|d_\delta)$.

The log-likelihood of the model,

$$\mathcal{L} = \sum_{\delta=1}^{N} \sum_{j=1}^{M} n(d_\delta, w_j) \log P(d_\delta, w_j)$$

$n(d_\delta, w_j)$ being the frequency of the word w_j in d_δ, is maximized by Expectation-Maximization algorithm [5].

The **E-step** consists on computing the posterior probabilities for the latent variable, based on the current estimates of the parameters, that is:

$$P(t=k|w_j,d_\delta)=\frac{P(w_j|t=k)P(t=k|d_\delta)}{\sum_l P(w_j|t=l)P(t=l|d_\delta)},$$

and the **M-step** consists on the maximization of the expected complete log-likelihood (*i.e.* the log-likelihood in the case where we knew which topic generated which word in documents), which is achieved with the following parameters re-estimation:

$$P(w_j|t=k)=\frac{\sum_{\delta=1}^{N} n(d_\delta,w_j)P(t=k|w_j,d_\delta)}{\sum_{m=1}^{M}\sum_{\delta=1}^{N} n(d_\delta,w_m)P(t=k|w_m,d_\delta)}$$

$$P(t=k|d_\delta)=\frac{\sum_{j=1}^{M} n(d_\delta,w_j)P(t=k|w_j,d_\delta)}{n(d_\delta)}$$

where $n(d_\delta)$ is the length of the document.

The PLSA model can be used to replace the original document representation by a representation in a low-dimensional "latent" space, to perform a TC or IR task. In [7], the components of the document in the low-dimensional space are $P(t=k|d), \forall k$, and for each unseen document or query they are computed by maximizing the log-likelihood with $P(w_j|t=k)$ fixed. This representation scheme is referred to as PLSI, for Probabilistic Latent Semantic Indexing. It is obvious that PLSA is not a well-defined generative model of documents, since there is no direct way to assign probability to an unseen document. However, some experiments in [7] report a comparison between LSI and PLSI, on several corpora. They conclude to a better performance of PLSI in all cases. In particular PLSI performs well even in the cases where LSI fails completely.

The other weakness of PLSA can be described as follows. The parameters of a K-topics PLSA model are the K multinomials of size $M=|\mathcal{V}|$ and the K mixing proportions for each of the N documents. Hence, the number of parameters equals $KM+KN$ and therefore grows lineary with the number of documents. This suggests that the model is prone to overfitting. In practice, to try to overcome this problem, a tempered EM (TEM) is performed instead of the EM. During the TEM iterations the parameters are smoothed in order to achieve an acceptable predictive performance on a validation set. However, according to [2], overfitting can occur even with the TEM version and it is likely with large corpora.

2.3.2 LDA

In the Latent Dirichlet allocation model the documents are assumed to be sampled from a random mixture over latent topics, where each topic is characterized by a distribution over words. In this modelisation the observed variable is the document d, seen as a set of words w_j, $j \in \{1,\ldots,|\mathcal{V}|\}$, and the unobserved variables are the topic t with possible values in $\{1,\ldots,K\}$, with K being an hyperparameter that must be chosen, and the mixing proportions $\theta=(\theta_1,\ldots,\theta_K)$, $\theta_k>0$, $\sum_{k=1}^{K}\theta_k=1$, of these topics for each document. The generative process for each document d is the following:

1. Choose $|d| \sim Poisson(\xi)$: the document size
2. Choose $\theta \sim Dir(\alpha)$: the random mixing proportions
3. For each of the $|d|$ words of d:
   (a) Choose a topic $t=k$ from $P(t|\theta)$, a multinomial probability with parameter θ
   (b) Choose a word w_j from $P(w|t=k)$, a multinomial probability conditioned on the topic $t=k$

where $Dir()$ refers to the Dirichlet distribution and $\alpha = (\alpha_1, \ldots, \alpha_K)$, $\alpha_k > 0$, is a parameter to estimate. The randomness of the document size $|d|$, modeled for example with a Poisson distribution with parameter ξ, is necessary for the generative process. However, given that $|d|$ is independent of all the other data generating variables (θ and t), it is not of real interest for the modelisation.[1] Hence, it will be ignored.

Given this generative model, we can write the joint distribution of a word and a topic as:

$$P(w_j, t = k|\theta, \alpha) = P(w_j|t = k)P(t = k|\theta).$$

Summing over k, we obtain the marginal distribution of a word,

$$P(w_j|\theta, \alpha) = \sum_{k=1}^{K} P(w_j|t = k)P(t = k|\theta).$$

Hence, the joint distribution of document d (*i.e.* a set of $|d|$ words) and a topic mixture θ is given by:

$$P(\theta, d|\alpha) = P(\theta|\alpha) \prod_{w_j \in d} \left[\sum_{k=1}^{K} P(w_j|t = k)P(t = k|\theta) \right]^{n(w_j,d)}$$

where $n(w_j, d)$ is the frequency of the word w_j in d and $P(\theta|\alpha)$ the Dirichlet probability density of θ. Finally, integrating over θ, we obtain the marginal distribution of a document,

$$P(d|\alpha) = \int P(\theta|\alpha) \prod_{w_j \in d} \left[\sum_{k=1}^{K} P(w_j|t = k)P(t = k|\theta) \right]^{n(w_j,d)} d\theta.$$

LDA, contrary to PLSA, is a true generative model of documents since both observed and unseen documents can be generated by the process described above. The parameters of the model are α and $P(w_j|t = k)$, $\forall k$, that is $K + K|\mathcal{V}|$ parameters, which is less than for PLSA and independent of the number of documents.

However, in order to estimate these parameters one has to compute the posterior distribution $P(\theta, t|d, \alpha, \beta)$, $\beta = (\beta_{jk})$, $\beta_{jk} = P(w_j|t = k)$, which is intractable in general, according to [2]. Therefore, instead of doing an exact inference for LDA, the authors of the paper, propose an approximate inference algorithm based on a variational method [10]. Indeed, their algorithm maximizes a lower bound on the log-likelihood based on a variational distribution which approximates the posterior distribution $P(\theta, t|d, \alpha, \beta)$. This maximization is done by a so-called variational EM algorithm, which consists in the iteration of these two steps:

1. **(E-step)** Variational approximation of the posterior distribution. This is performed by an iterative algorithm, which requires approximatively $|d|^2K$ operations for each document, according to [2].

2. **(M-step)** Maximize the resulting lower bound on the log-likelihood with respect to the parameters α and β.

In order to represent the documents in a space with a lower dimension than the bag-of-words space (*eg* to perform a supervised task), the authors of the paper have chosen K variational parameters from the posterior distribution approximation. This gives a representation of documents in terms of topics instead of a representation in terms of words. Experiments reported in [2] show that with a reduction of the representation space of 99.6% using LDA, a classifier for a TC task needs less training data to perform better or or at least the same than one using a bag-of-word representation. They also show that LDA has a lower perplexity than PLSA, for different values of K, on two different corpora.

For many reasons mentioned above LDA is an interesting model of document density. However, the approximate inference algorithm is not easy to implement. Therefore, as a first step, another model, tractable by exact inference, is proposed in the following section.

[1] In fact the log-likelihood will have this form: $\mathcal{L} = A(|d|) + B(\theta, t)$ and thus maximizing it will lead to two distinct problems.

3 Theme Topics Mixture Model

In this section, we present a novel probabilistic model, the Theme Topic Mixture Model (TTMM), which can be used to represent documents in corpora. It has been designed to overcome the problems explained in the introduction, which are related to document representation such as bag-of-words.

We first introduce the formal model, then report some preliminary results comparing TTMMs with other models.

3.1 The Model

TTMM is very similar to LDA, but instead of taking a continuous space for the choice of the mixing proportions of the topics, the choice is constrained to a discrete finite set. The generative process for each document is the following:

1. Choose $|d| \sim Poisson(\xi)$: the document size
2. Choose a theme $h = j$ from $P(h)$, a multinomial probability with parameter $\pi = (\pi_1, ..., \pi_J)$: the mixing proportions
3. For each of the $|d|$ words in d:
   (a) Choose a topic $t = k$ in $\{1, \ldots, K\}$ from $P(t|h = j)$, a multinomial probability conditioned on the theme $h = j$
   (b) Choose a word w_j from $P(w|t = k)$, a multinomial probability conditioned on the topic $t = k$

where J and K are hyperparameters that must be chosen. For the same reasons than with LDA, the randomness of $|d|$ is ignored.

According to the generative process, each word w is seen as mixture of topics t, with different mixture proportions depending on the document's theme h:

$$P(w|h = j) = \sum_{k=1}^{K} \tau_{jk} P(w|t = k, \beta), \quad \tau_{jk} = P(t = k|h = j).$$

The probability of a document d given that it was generated by the theme $h = j$:

$$P(d|h = j) = \prod_{w_l \in d} [P(w_l|h = j)]^{n(w_l,d)},$$

where $n(w_l, d)$ is the frequency of the term w_l in d, and each document d is seen as a mixture of themes h:

$$\begin{aligned} P(d) &= \sum_{j=1}^{J} \pi_j P(d|h = j), \quad \pi_j = P(h = j) \\ &= \sum_{j=1}^{J} \pi_j \prod_{w_l \in d} [P(w_l|h = j)]^{n(w_l,d)} \\ &= \sum_{j=1}^{J} \pi_j \prod_{w_l \in d} \left[\sum_{k=1}^{K} \tau_{jk} P(w_l|t = k, \beta) \right]^{n(w_l,d)}. \end{aligned} \tag{3}$$

The log-likelihood of the model on the corpus then becomes:

$$\mathcal{L}(\pi, \tau, \beta) = \sum_{i=1}^{N} \ln \sum_{j=1}^{J} \pi_j \prod_{w_l \in d_i} \left[\sum_{k=1}^{K} \tau_{jk} P(w_l|t = k, \beta) \right]^{n(w_l,d_i)}. \tag{4}$$

Let $\{h_{ij}\}$ be the indicator variable specifying which theme j the document d_i was generated from, and $\{t_{jlk}\}$ the indicator variable specifying which topic k the word w_l was generated from given that we are in the theme j context. The complete log-likelihood can be written as:

$$\begin{aligned}\mathcal{L}_{comp}(\pi,\tau,\beta) &= \sum_{i=1}^{N}\sum_{j=1}^{J} h_{ij}\ln\left(\pi_j\prod_{w_l\in d_i}\left[\sum_{k=1}^{K}\tau_{jk}P(w_l|t=k,\beta)\right]^{n(w_l,d_i)}\right)\\ &= \sum_{i=1}^{N}\sum_{j=1}^{J} h_{ij}\left(\ln(\pi_j)+\sum_{w_l\in d_i}\ln\left[\sum_{k=1}^{K}\tau_{jk}P(w_l|t=k,\beta)\right]^{n(w_l,d_i)}\right)\\ &= \sum_{i=1}^{N}\sum_{j=1}^{J} h_{ij}\left(\ln(\pi_j)+\sum_{w_l\in d_i}\sum_{k=1}^{K}t_{jlk}[n(w_l,d_i)]\ln\left[\tau_{jk}P(w_l|t=k,\beta)\right]\right) \qquad (5)\end{aligned}$$

The maximization of the log-likelihood can be performed with Expectation-Maximization algorithm as follows:

In the **E-step** the complete log-likelihood is estimated, by estimating the posteriors as follows:

$$\begin{aligned}E(h_{ij}) = P(h=j|d_i) &= \frac{\pi_j P(d_i|h=j)}{\sum_{q=1}^{J}\pi_q P(d_i|h=q)}\\ &= \frac{\pi_j\prod_{w_l\in d_i}\left[\sum_{k=1}^{K}\tau_{jk}P(w_l|t=k,\beta)\right]^{n(w_l,d_i)}}{\sum_{q=1}^{J}\pi_q\prod_{w_l\in d_i}\left[\sum_{k=1}^{K}\tau_{qk}P(w_l|t=k,\beta)\right]^{n(w_l,d_i)}}\\ E(t_{jlk}) = P(t=k|w,h=j) &= \frac{\tau_{jk}P(w|t=k)}{\sum_{p=1}^{K}\tau_{jp}P(w|t=p)}.\end{aligned}$$

In the **M-step** the expected log-likelihood is maximized:

$$E[\mathcal{L}_{comp}] = \sum_{i=1}^{N}\sum_{j=1}^{J}P(h=j|d_i)\left(\ln(\pi_j)+\sum_{w_l\in d_i}\sum_{k=1}^{K}P(t=k|w_l,h=j)[n(w_l,d)]\ln\left[\tau_{jk}P(w_l|t=k,\beta)\right]\right) \qquad (6)$$

under the normalization constraints. Which leads to maximise:

$$\begin{aligned}\mathcal{H} = E[\mathcal{L}_{comp}] &+ \rho\left(1-\sum_{j=1}^{J}\pi_j\right)\\ &+\sum_{j=1}^{J}\lambda_j\left(1-\sum_{k=1}^{K}\tau_{jk}\right)+\sum_{k=1}^{K}\eta_k\left(1-\sum_{l=1}^{M}P(w_l|t=k)\right). \qquad (7)\end{aligned}$$

Which is equivalent to solve the following system of equations:

$$\begin{aligned}\sum_{i=1}^{N}P(h=j|d_i)-\pi_j\rho &= 0, \quad 1\le j\le J\\ \sum_{i=1}^{N}P(h=j|d_i)\sum_{w_l\in d_i}n(w_l,d_i)P(t=k|w_l,h=j)-\tau_{jk}\lambda_j &= 0, \quad 1\le j\le J, \quad 1\le k\le K\\ \sum_{i=1}^{N}\sum_{j=1}^{J}n(w_l,d_i)P(h=j|d_i)P(t=k|w_l,h=j)-P(w_l|t=k)\eta_k &= 0, \quad 1\le l\le M, \quad 1\le k\le K\end{aligned}$$

$\Longleftrightarrow$

$$
\begin{aligned}
\pi_j = P(h=j) &= \frac{\sum_{i=1}^N P(h=j|d_i)}{\sum_{q=1}^J \sum_{i=1}^N P(h=q|d_i)} \\
&= \frac{\sum_{i=1}^N P(h=j|d_i)}{N}, \text{ given that } \sum_{q=1}^J P(h=q|d_i) = 1 \\
\tau_{jk} = P(t=k|h=j) &= \frac{\sum_{i=1}^N P(h=j|d_i) \sum_{w_l \in d_i} P(t=k|w_l, h=j) n(w_l, d_i)}{\sum_{p=1}^K \sum_{i=1}^N P(h=j|d_i) \sum_{w_l \in d_i} P(t=p|w_l, h=j) n(w_l, d_i)} \\
&= \frac{\sum_{i=1}^N P(h=j|d_i) \sum_{w_l \in d_i} P(t=k|w_l, h=j) n(w_l, d_i)}{\sum_{i=1}^N P(h=j|d_i) \#(d_i)}, \\
& \quad \text{given that } \sum_{p=1}^K P(t=p|w_l, h=j) = 1 \\
P(w_l|t=k) &= \frac{\sum_{i=1}^N \sum_{j=1}^J P(h=j|d_i) P(t=k|w_l, h=j) n(w_l, d_i)}{\sum_{m=1}^M \sum_{i=1}^N \sum_{j=1}^J P(h=j|d_i) P(t=k|w_m, h=j) n(w_m, d_i)}.
\end{aligned}
$$

The TTMM model has $J(1+K) + K|\mathcal{V}|$ parameters. Which is about J times more than LDA (number of parameters: $K + K|\mathcal{V}|$). This is due to the fact that the continuous distribution with one parameter, which generates the mixing proportions θ in LDA, is replaced in TTMM by two discrete distributions, the multinomials with parameters $\pi = (\pi_j)_j$ and $\tau = (\tau_{jk})_{j,k}$. The number of parameters is possibly less than with PLSA (number of parameters: $K|\mathcal{V}| + KN$), since we hope that documents can be clustered together by themes, and so $J < N$.

This density estimation method can be used as a Dimentionality Reduction method on the *bag-of-words* representation. The idea is that instead of considering words as basic units of document representation we will consider a topic basis, with the hope that a few topics will capture more information than the huge amount of words. We can choose as topic component its posterior given the document:

$$P(t=k|d) = \frac{P(t=k,d)}{P(d)},$$

where,

$$
\begin{aligned}
P(t=k,d) &= \sum_{j=1}^J \pi_j P(t=k, d|h=j) \\
&= \sum_{j=1}^J \pi_j \tau_{jk} P(d|t=k, h=j) \\
&= \sum_{j=1}^J \pi_j \tau_{jk} \prod_{w_l \in d} [P(w_l|t=k, h=j)]^{tf(w_l,d)} \\
&= \sum_{j=1}^J \pi_j \tau_{jk} \prod_{w_l \in d} \left[\frac{P(w_l, h=j|t=k)}{P(h=j|t=k)}\right]^{tf(w_l,d)} \\
&= \sum_{j=1}^J \pi_j \tau_{jk} \prod_{w_l \in d} \left[\frac{P(w_l|t=k) P(h=j|t=k)}{P(h=j|t=k)}\right]^{tf(w_l,d)}
\end{aligned}
$$

$$= \prod_{w_l \in d} [P(w_l|t=k)]^{tf(w_l,d)} \sum_{j=1}^{J} \pi_j \tau_{jk} \tag{8}$$

3.2 Preliminary Results

In the following, TTMM preliminary experiments on the Reuter-21578 dataset are presented. First, a method for tuning the hyperparameters and some training measures are explained. Then, a comparison beetween TTMM and LDA is achieved in the Text Categorization context. This comparison is based on the repetition, using TTMM, of an experiment presented in [2] under the name "Document Classification". The authors of [2] have done other choices for the text preprocessing than I have, therefore the data used for the comparison is theirs, which is different from the one used for the tuning experiment.

3.2.1 TTMM's Hyperparameters Tuning

Before training a TTMM, one has to decide some structural values (also called hyperparameters): the number of themes J and the number of topics K. As shown in Fig. 2, two TTMMs with different hyperparameters can give both intuitively "suitable" results. It is clear thus that the choice of the hyperparameters is important, since it can lead to very different models, and that this choice must be done by an objective criterion. The obvious criterion is the one optimized during training, that is Maximum Likehood criterion or its equivalent, Minimum Negative Log-Likehood (NLL), computed on a validation set, different from the one used for training, to monitor the model fitting to unseen data.

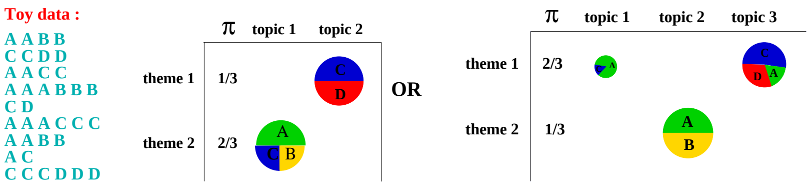


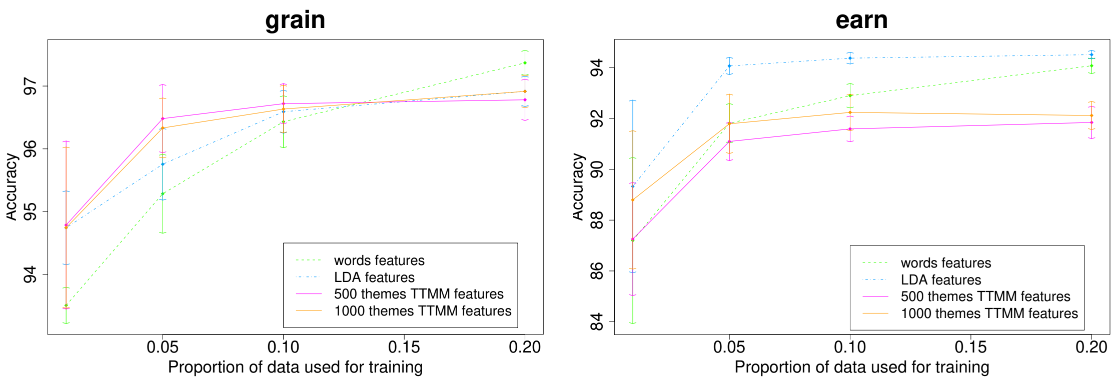
Figure 2: Illustration of two different Toy data TTMM models with respectives hyperparameters $J = |\text{themes}| = 2$, $K = |\text{topics}| = 2$ and $J = 2$, $K = 3$, pie charts representing $\tau_{jk} \cdot P(w_l|t=k)$ for fixed j and k. As can be seen both models enligthen the existence of the 3 kind of documents AB, AC and CD, but by two different ways

Here, a TTMM's hyperparameter tuning method is presented, performed on few data for an exploratory purpose, since learning time for multiple choice topic's and theme's numbers is quite long. A subset $\mathcal{S}$ of 1246 documents in ModApte split's training set of Reuter-21578 dataset has been chosen randomly. Documents have been stopped and stemmed as explained in section 2.1. The vocabulary $\mathcal{V} = (t_1, \ldots, t_M)$ has then been extracted from this preprocessed data and the documents represented by $d = (w_1, \ldots, w_M)$, with $w_l = tf(t_l, d)$. There are 6269 words in $\mathcal{V}$. $\mathcal{S}$ has been then divided in a training set and a validation set. Models with hyperparameters $\Upsilon = (J, K) = (|\text{themes}|, |\text{topics}|) \in \Xi^2$, $\Xi = \{10, 50, 100, 200, 300, 500, 1000, 1500, 2000\}$, have been trained. The choice of Ξ is a heuristic. Measures have been collected during and at the end of each training, on training and validation sets.

Fig. 3 shows these measures for a fixed theme's number, $J = 50$ and a variable topic's number $K \in \Xi$. In general, the hyperparameter Υ is chosen in order to obtain the optimal model's capacity, by taking the Υ for which the NLL on validation set is lowest. For this, the curves A and B of Fig. 3, are almost ideal. We can see that the NLL on the training set decreases while the capacity increases. Simultaneously, on the validation set the NLL decreases up to $K = 1500$, then it increases when the model begins to overfit the training data. However, we probably need more points in the curve B to say that the NLL for $J = 50$ has a minimum at $K = 1500$. Furthermore, to choose our optimal Υ we also need to vary the value of J, which has been done, and the results are not always as ideals. During training, however, we can see in curves C and D of Fig. 3, that the NLL follows the same curious pattern across time for each value of K in training and validation set. We expect the NLL to decrease in training set and to decrease and then increase (or at least stay flat) in validation set during training, which it does for all values of K. What is curious and not explained yet is the sill the NLL stays in for many iterations, in which the parameters of the models are changed, before decreasing at last. This will be studied further later.

3.2.2 TTMM vs LDA

In this section, a comparison of LDA and TTMM is reported. In [2], LDA's features and bag-of-words document representations are compared in a Text Categorization task using SVMs. Using the same data and splits the experiment is repeated here with TTMM's features document representation. For this experiment, the authors of [2] have selected 8529 Reuters-21578's documents (almost all the training data), they stopped, but did not stemmed the data, and from the resulting vocabulary they discarded the less frequent words to finally obtain a vocabulary of 15810 words. An LDA model with 50 topics is trained on all the documents, without reference to their class labels. For each proportion $p \in \{0.01, 0.05, 0.1, 0.2\}$ of the training data, they trained a support vector machine (SVM) on the LDA's features documents representation, as mentioned in section 2.3.2. Then, they compared the results of these SVMs on each remaining data (*i.e.* proportion $1 - p$ of the data), for two binary classification problems[2], to the ones of SVMs trained on the bag-of-words representation. They repeated the experiment for various splits. Their results for GRAIN and EARN are illustrated in Fig.4.



(a) GRAIN vs. NOT GRAIN

(b) EARN vs. NOT EARN

Figure 4: Classification results on two binary classification problems from the Reuters-21578 dataset for different proportions of training data.

These results are optimistic, and not comparable with others Text Categorization results publications, since the vocabulary is extracted from both training and test sets. Thus the problem of having unseen words in the test set is not addressed. Nevertheless, to make a comparison between TTMM and LDA, we followed the same experimental protocol. A TTMM was trained on all the documents without references to their class labels, using the vocabulary they have extracted. The TTMM's

[2]Reuters-21578 documents are labelled with one or several categories among 105 possibles. In [2] a one-against-the-others approach is considered for two of these categories: EARN and GRAIN.

hyperparameters were chosen without tuning, because of a lack of time. However, since TTMM derives from LDA we tried to keep similar structural values. The TTMM topics correspond to LDA topics, thus we chose to have the same number than in [2], that is $K = 50$. The TTMM themes correspond to the infinite number of possible mixtures in LDA, hence we chose a "high" number of themes, for which we tried several values. For each proportion $p \in \{0.01, 0.05, 0.1, 0.2\}$ of data in each split (the ones chosen in [2]), SVMs hyperparameters were tuned by a 5-fold cross-validation using the TTMM's features document representations, defined in section 3.1. Then, again for each proportion $p \in \{0.01, 0.05, 0.1, 0.2\}$ of data in each split, the optimal SVMs were trained, and then tested on the remaining $1 - p$ proportion of the data. The results can be seen in Fig. 4.

For GRAIN category, the accuracy is higher with LDA and TTMM features than with bag-of-word representation for the classifiers trained with few data (less than 10% of the data). TTMM features achieve a performance as good as LDA features and even a better one for proportion $p = 0.05$. For EARN category, classifiers trained with TTMM features give poorer results than the ones trained with LDA features and even, except for $p = \{0.01, 0.05\}$, than the ones trained with bag-of-words representation. However, we can suppose that we obtained these poor results because we have not taken a high enough value for J. This assumption is reinforced by the fact that the higher J the better the accuracy. Furthermore, we can conclude from this experiment that TTMM does capture some information from the data, since even with 99.6% less features than the bag-of-words representation (50 vs 15810), the results are better for small values of p. This last point confirms our assumption, explained in the Introduction, that unlabelled data can be useful to construct a representation to be used in a supervised task. We expect the results to be better or at least the need of labeled data to be fewer, if the unlabelled database is bigger.

4 Conclusion and Future Research Directions

In this report we have explained the problem of textual data representation. The current methods used to solve this problem have been exposed, with a special focus on probabilistic approaches. A novel probabilistic model has been proposed and an experiment has been presented, comparing its performance to LDA and bag-of-words. The results are encouraging but further research is still needed. In the following some interesting future research directions are explained.

4.1 Taking Advantage of Large Unlabeled Corpora

As explained in the introduction, large unlabeled corpora are easily available while annotated databases, oriented to a specific supervised task, are usually of little sizes. Probabilistic models such as PLSA, LDA and TTMM can be used to learn a document representation in the scope of solving a supervised task. For such a task, the labels of documents are useless. Although, PLSA and LDA have been tested and have given interesting results, it was always on the task specific annotated corpora. It would be interesting to test TTMM on a larger corpus such as the new Reuters database (RCV1 : 800 000 annotated news stories). Most of the data can be used, without caring of the labels, for document density estimation. Then this estimation can be used to represent documents in the framework of Text Categorization on the remaining data (using for instance $P(t|d)$ as explained in section 3.1). A comparison of the results of a classifier on this representation versus a classifier on bag-of-words representation will show whether using probabilistic models for document representation is an efficient way of taking advantage of large unlabeled corpora or not.

We can hope that in this specific case it should work since the probabilistic model training set and the classifier training set are part of a homogene corpus. But it would be probably a different story if the unlabeled document set is composed, let say, of 20 years old news stories, and the labeled document set of this year's news stories. So it would be interesting to use the TTMM estimations of document density in different sub-sets to detect temporal drifts of concepts, by using a measure of distance beetween distributions, such as the Kullback-Leibler divergence.

Another interesting issue could be to measure whether using a representation based on probabilistic models learned on a large unlabeled database will reduce the number of needed annoted documents.

Finally, TTMM and related models can also be used for Language Modeling (instead of overused n-grams [8] or the recent NNLM [1] models) since these models define links between words through the topics. For example, in [2], the authors proposed to change the modelisation made in LDA so that the model will take into account the order of the words. Instead of estimating the probability of a word given a topic $P(w_l|t = k)$ we can estimate the probability of a word given a topic and the n preceding words $P(w^t = w_l|t = k, w^{t-n}, \ldots, w^{t-1})$. Thus, we will have a modelisation similar to n-grams but conditioned on topics. This idea can also be applied to TTMM. Another idea could be to have specific Language Modeling for each document depending on its most probable theme according to a TTMM. Since we can assign a list of most probable words for each theme j (through $P(t|h = j)$ and $P(w|t)$), we could have a theme specific Language Model with this word's list used as lexicon. We can assume that perhaps we will have a smaller lexicon with a better document coverage, which will improve the document recognition.

4.2 Supervised Probabilistic Models

Instead of learning a general document representation as proposed in the previous section, probabilistic models can also be used to solve directly a specific supervised task. For example in Text Categorization, TTMM can be used to construct a Bayes Classifier, which will thus be trained on labeled data. We can assume that the "themes" of the model are the categories $\{c_1, \ldots, c_J\}$ present in the data, and TTMM will then try to model the distributions of "topics" in each category. To categorize an unseen document d one will compute:

$$P(d|c_j)P(c_j), \quad \forall j,$$

and the categories for which this value is greater than a threshold will be chosen as document's categories. Note that in that case the training algorithm must be slighty modified since during training $P(h|d)$ is observed and not hidden any more.

4.3 Reconsidering the Implementation of TTMM

For the moment TTMM is a model using hidden variables where the likelihood of the data is maximised by EM scheme. The priors of the parameters are represented by tables. Since tables need a lot of parameters for their representation, and do not take into account the possible relations between the elements of the table, they may not be a convenient representation of the priors. Another solution would be to replace these tables by Multi-Layers Perceptrons (MLPs), where $P(t|h)$ would be for instance represented by $MLP(h; \eta)$ with parameters η and with K outputs constrained to sum to 1 using a softmax output function. In order to train this new representation, the Maximization step of the EM algorithm has to be changed since one cannot directly maximize an MLP in one step, but can instead compute the gradient that goes in the right direction. The resulting algorithm is called a Generalysed EM. A stochastic training of the MLPs could be interesting, since it is proned to be more robust to noise and often faster than batch training on large redundent databases. Hence, we would like to test some ideas from [14], in which an other optimisation method than EM is proposed to maximize the likehood of a Gaussian Mixture Model, which allows incremental learning.

References

[1] Y. Bengio, R. Ducharme, P. Vincent, and C. Gauvin. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3:1137–1155, February 2003.

[2] David Blei, Andrew Ng, and Michael Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, January 2003.

[3] F. Crestani, M. Lalmas, C. Rijsbergen, and I. Campbell. Is this document relevant?...probably: A survey of probabilistic models in information retrieval, 1998.

[4] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statist. Soc. B.*, 39:1–38, 1977.

[6] Susan Dumais, John Platt, David Heckerman, and Mehran Sahami. Inductive learning algorithms and representations for text categorization. In George Gardarin, James French, Niki Pissinou, Kia Makki, and Luc Bouganim, editors, *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM-98)*, pages 148–155, New York, November 3–7 1998. ACM Press.

[7] Thomas Hofmann. Unsupervised learning by Probabilistic Latent Semantic Analysis. *Machine Learning*, 42:177–196, 2001.

[8] F. Jelinek. Continuous Speech Recognition by Statisical Methods. In *IEEE Proceedings 64:4*, pages 532–556, 1976.

[9] Thorsten Joachims. *Learning to Classify Text using Support Vector Machines*. Kluwer Academic Publishers, Dordrecht, NL, 2002.

[10] Michael I. Jordan, Zoubin Ghahramani, Tommi Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

[11] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.

[12] Hinrich Schutze, David A. Hull, and Jan O. Pedersen. A comparison of classifiers and document representations for the routing problem. In *Research and Development in Information Retrieval*, pages 229–237, 1995.

[13] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[14] Yoram Singer and Manfred K. Warmuth. A New Parameter Estimation Method for Gaussian Mixtures. *Advances in Neural Information Processing Systems 11 (NIPS*98)*, 1998.

[15] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In Marti A. Hearst, Fredric Gey, and Richard Tong, editors, *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, pages 42–49, Berkeley, US, 1999. ACM Press, New York, US.

[16] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In Douglas H. Fisher, editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 412–420, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.

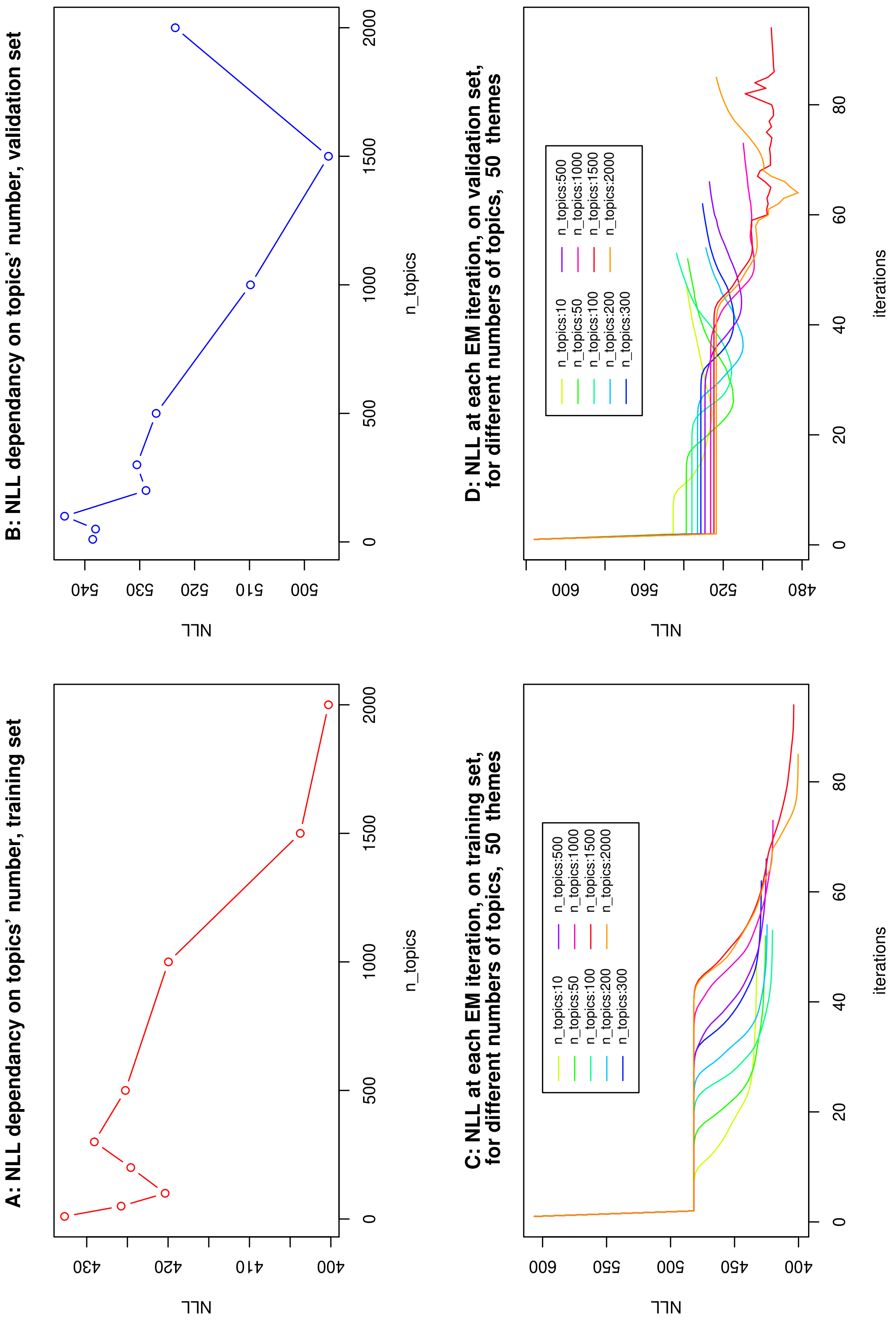


Figure 3: Negative Log-Likehood for 50 themes and differents number of topics