

Master thesis : Generating Topic Models from Corpora Across Languages

Auteur : Thielen, Benoit

Promoteur(s) : Ittoo, Ashwin

Faculté : Faculté des Sciences appliquées

Diplôme : Master en sciences informatiques, à finalité spécialisée en "intelligent systems"

Année académique : 2021-2022

URI/URL : <http://hdl.handle.net/2268.2/13874>

Avertissement à l'attention des usagers :

Tous les documents placés en accès ouvert sur le site le site MatheO sont protégés par le droit d'auteur. Conformément aux principes énoncés par la "Budapest Open Access Initiative"(BOAI, 2002), l'utilisateur du site peut lire, télécharger, copier, transmettre, imprimer, chercher ou faire un lien vers le texte intégral de ces documents, les disséquer pour les indexer, s'en servir de données pour un logiciel, ou s'en servir à toute autre fin légale (ou prévue par la réglementation relative au droit d'auteur). Toute utilisation du document à des fins commerciales est strictement interdite.

Par ailleurs, l'utilisateur s'engage à respecter les droits moraux de l'auteur, principalement le droit à l'intégrité de l'oeuvre et le droit de paternité et ce dans toute utilisation que l'utilisateur entreprend. Ainsi, à titre d'exemple, lorsqu'il reproduira un document par extrait ou dans son intégralité, l'utilisateur citera de manière complète les sources telles que mentionnées ci-dessus. Toute utilisation non explicitement autorisée ci-avant (telle que par exemple, la modification du document ou son résumé) nécessite l'autorisation préalable et expresse des auteurs ou de leurs ayants droit.



UNIVERSITY OF LIEGE
FACULTY OF APPLIED SCIENCES

Generating Topic Models from Corpora Across Languages

Author:

Benoit THIELEN

Supervisor:

Prof. Ashwin ITTOO

End of study work carried out with a view to obtaining a Master's degree in
"Computer Science" by Benoit THIELEN.

Academic Year 2021-2022.

Abstract

Topic modeling is a learning process aiming to analyze texts to discover their topic composition by associating groups of correlated words. Historically, topic modeling has used unsupervised learning techniques. Bayesian generative models, such as Latent Dirichlet Allocation (LDA), have quickly proven their performance for representing with probabilities the distributions of words across topics and of topics across documents. Recently, new topic models based on LDA have emerged, like the Hierarchical Dirichlet Process (HDP) which self-determines the number of topics in the text and the nested Hierarchical Dirichlet Process (nHDP) which enables a hierarchical representation of the topics.

The performances in topic identification and hierarchical modeling of HDP and nHDP were evaluated in this work, on English and French corpora built from Wikipedia articles. A large number of very coherent and interesting topics were detected in both languages, despite the presence of some less coherent ones. Correlations have been highlighted between the statistics of the corpus and evaluation metrics such as coherence and model perplexity.

Additionally, a more recent approach of learning word embeddings in hyperbolic space, specifically in the Poincaré ball space, has been studied to determine if it could constitute a promising approach to hierarchical topic modeling. Poincaré embeddings of 10 dimensions were trained on hypernymy relations of our English corpus. Our analysis revealed clusters of words which can be linked to topics, unfortunately the 2D representation method we applied did not allow to show hierarchical relations between those clusters.

In conclusion, both HDP and nHDP models have shown good and similar learning performances when trained on French and English corpora, nHDP being also efficient in providing hierarchical representation of the topics. The Poincaré embeddings were successful in learning and representing the hypernymy relations in the Poincaré ball, however suffered from the constraints imposed by the data acquisition methods and required filtering processes.

Acknowledgements

I would like to express my sincere grateful thanks to my promoter Prof. Ashwin Ittoo for allowing me to carry out this work and for his time and guidance during our meetings. I want to thank M. Judicaël Poumay for answering my questions about hierarchical topic models. I would also like to express my thanks to everyone who spent time proofreading this work. Finally, I would like to express all my gratitude to my loved ones for their strong support throughout my studies and in everything I do.

Contents

Abstract	i
Acknowledgements	ii
1 Introduction	1
2 Theoretical background	4
2.1 Word embeddings	4
2.2 Embeddings in hyperbolic space	6
2.3 Bayesian topic modeling	8
2.3.1 Beta distribution	8
2.3.2 Dirichlet distribution	8
2.3.3 Dirichlet Process	10
2.3.4 Bayesian models	11
2.4 Topic models	11
2.4.1 Latent Dirichlet Allocation	12
2.4.2 Hierarchical Dirichlet Process	14
2.4.3 Nested Hierarchical Dirichlet Process	16
3 Experiments	17
3.1 Experimental setup	17
3.1.1 Data collection	17
3.1.2 Pre-processing	17
3.2 Methodology	18
3.2.1 HDP	18
3.2.2 nHDP	20
3.2.3 Poincaré embeddings	21
3.3 Evaluation	22
3.3.1 Topic models	22
3.3.2 Hypernyms and embeddings	24

4 Results	27
4.1 HDP	27
4.1.1 English corpus	27
4.1.2 French corpus	31
4.1.3 Log-likelihood	33
4.1.4 Discussions	34
4.2 nHDP	35
4.2.1 English corpus	36
4.2.2 French corpus	38
4.2.3 Perplexity	39
4.2.4 Discussions	39
4.3 Poincaré embeddings	41
4.3.1 Hypernyms evaluation	41
4.3.2 Embeddings evaluation	42
4.3.3 Discussions	44
5 Conclusion	46
A Supplementary examples of topics found by the HDP model	52
A.1 English corpus	52
A.1.1 ONE	52
A.1.2 IDF	54
A.1.3 PMI	57
A.2 French corpus	60
A.2.1 ONE	60
A.2.2 IDF	63
A.2.3 PMI	66
B Supplementary examples of topics found by the nHDP model	68
B.1 English corpus	68
B.2 French corpus	71
C Hearst Patterns used for hypernyms extraction	73
D Additional examples of clusters of word embeddings	74
Acronyms	78

Chapter 1

Introduction

In artificial intelligence, there are many different fields of research characterized by the tasks to be automated, the method of learning that is used, the human supervision level required to learn from data or the nature of processed data. The particular branch of artificial intelligence dealing with linguistic data, whatever the form it takes, is called Natural Language Processing (NLP). There are two main forms of linguistic data: it can be textual, like written or typed texts for example, or it can be audial, like voice recordings. For instance, speech recognition is a specific NLP task, often used for automatic subtitling, which aims to transcribe a meaningful vocal sequence into a piece of text corresponding to what has been said in the audio sequence. Machine translation is another NLP task which transforms a text from a given language into the equivalent text in a different language.

Topic modeling is one of the most important task in NLP. Topic models are employed for many jobs like classifying documents, processing queries in search browsers or producing suggestions related to a subject. More complex models are able to find hierarchies of topics in documents, those models are commonly called *hierarchical topic models*. There also exist *temporal topic models* that are used to situate topics and documents along a timeline.

In this work, our focus has been placed on topic modeling, aiming at associating groups of words to topics and discovering the topic composition of texts. More precisely, our research and experiments were concentrated on Bayesian topic models, one of which allowing hierarchical topic modeling (i.e. assign hierarchical relations between topics).

The first objective of this work has been to evaluate the performance of different topic models over texts written in English and in French from Wikipedia.

The models that were compared in this work are the Hierarchical Dirichlet Process (HDP) [1] and the nested Hierarchical Dirichlet Process (nHDP) [2], which are both extensions of the Latent Dirichlet Allocation (LDA) model [3]. Those 3 models have been described in Chapter 2. In order to study the influence of the frequency of occurrence of words on the model performances, the various models were also tested with different methods for assigning weight to words, depending

on their frequency in the corpus or in documents.

Most popular Bayesian topic models like LDA, HDP, nHDP and other variants are generative probabilistic models relying on the Dirichlet probability distribution. This particular probability distribution is well suited to represent the distribution of words and topics in a corpus of documents. Those models have shown very good performances and have been widely used in a lot of works since the early 2000s.

For example, Rosen-Zvi et al. [4] have build an author-topic model able to discover or re-attribute the authorship of documents. This model can be used to predict probable authors of documents or to find several authors related to the topics of a given document. Another example of model based on LDA is the Corr-LDA model proposed by Blei et al. [5]. Corr-LDA allows the creation of captions for images by splitting those images into regions and assigning them related description words.

In 2006, Teh et al. [1] introduced the HDP model, which is a non-parametric generalization of LDA. It assumes that the topic distribution over the corpus is generated by a Dirichlet Process (DP) [6] and that the words of each document are also distributed across these topics by a local DP specific to that document. The HDP model has inspired researchers to investigate further into the hierarchical representation of documents.

A popular hierarchical topic model is the hierarchical LDA (hLDA), introduced in 2004 by Blei et al. [7]. This model is built on the nested Chinese Restaurant Process (nCRP) framework which allows to represent the topic hierarchy of a corpus in a tree structure. In this tree, each node is a topic and child nodes can be seen as sub-topics of their parent. According to the hLDA model, each document follows one path from the root to one leaf in the tree, meaning that the model only distributes its topics across topic nodes in that path. This model is limited by itself in the sense that, with this configuration, a document cannot be composed of different sub-topics.

In 2015, Paisley et al. [2] introduced the nHDP model. With nHDP, the topic hierarchy is represented by a tree structure as in hLDA but each document can appear in different paths from the root. In this way, a document is not restricted in the topics it contains and it can choose several from all the branches of the general tree.

A number of researches have also been made to propose efficient inference techniques for those Bayesian models. We can split them into 3 main categories: Gibbs sampling, EM inference and Variational Bayes inference [8, 9, 10].

On the other hand, some interesting papers by Sia et al. and Onan [11, 12] have demonstrated that word embeddings could also be used to extract topics from texts by finding clusters of word vectors. More recently, a relatively new approach of learning numerical word representations in the Poincaré ball hyperbolic space was presented by M. Nickel and D. Kiela [13].

The second objective of our work has been to investigate whether these new methods of learning word embeddings in hyperbolic spaces, such as the Poincaré ball model, could be used as potential tools to model hierarchies in topics.

This part of our research has been mainly based on the review of the works made by Nickel and Kiela and Roller et al. in the last few years [13, 14, 15].

Following this short introduction, Chapter 2 will review some background notions to ease the further understanding of this work. In particular, the various topic models that were used in our experiments will be detailed. Then, the general setup, methodology and evaluation techniques used during our experiments will be described in Chapter 3. The results of our experiments will be presented and discussed in Chapter 4. Finally, the conclusion of our work will be proposed in Chapter 5, together with some ideas for improvement in future works.

Chapter 2

Theoretical background

2.1 Word embeddings

In this section, the notion of word embeddings will be reviewed and it will be explained how they are learnt and what differentiate the most popular embedding models.

To enable computers to deal with discrete data like words which have semantic meanings, different techniques have been introduced. The first one is named “tokenization”. It consists in extracting the words from the text to obtain a set of unique tokens. This set of unique tokens defines the *vocabulary* of the text. Then, the tokens may have to be converted into numeric values, to train a neural network for example. One technique proposed at first was to create one-hot vectors of the size of the vocabulary. With this solution, each index in the vector corresponds to a unique word of our vocabulary. For each word w , we have one vector where we set the value at the index of w to 1 and all other values to 0. Therefore, the text is seen as a sequence of one-hot vectors. The main drawback of this technique is that the vectors can be very large (as they grow with the size of the vocabulary) and become very quickly difficult to handle and computationally inefficient. A suitable solution to overcome this problem is the word embeddings which was first proposed by Bengio et al. [16] in 2003 and became really popular with the work of Mikolov et al. [17]. In the following section, the concept of word embeddings will be introduced and explained more in detail.

Word embeddings are vector representations of words which capture their semantic similarities. Those vectors have a fixed size and each index of the vectors can be seen as a feature. By this way, words with similar meaning should have similar embeddings. Their similarity is computed with distance measures and the difference in their vectors indicates their relationship. A famous example is the following puzzle analogy: “Man is to king as woman is to x ” where the embedding corresponding the most to x should be the embedding of the word *queen*.

The three most popular implementations of word embeddings are the neural-based methods

Word2Vec¹ [17], GloVe² [18] and FastText³ [19, 20]. The first method and its main differentiation from the two others are briefly explained below.

Two models can be used to generate Word2Vec embeddings, these are the Continuous Bag of Words (CBOW) and the Skip-gram (SG) models (see Figure 2.1). Both techniques use a shallow neural network (NN) to predict a target word or group of words given a context.

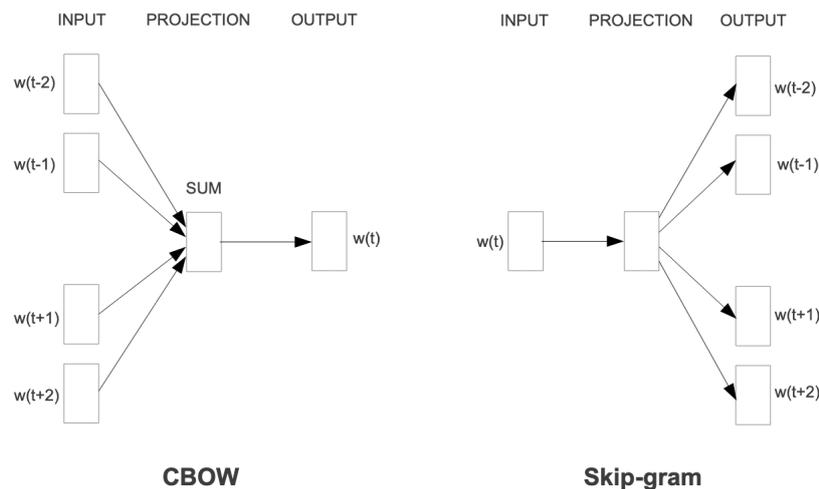


Figure 2.1: CBOW (left) and SG (right) models.⁴

In CBOW model, a target word is predicted given the context words around this target. For example, in the sentence “The quick brown ___ jumps over the lazy dog”, the hidden word *fox* should be predicted on the basis of its following and preceding words. The NN takes the one-hot vector of the context words as inputs and is supposed to output the one-hot vector of the target word. Between the input layer and the first hidden layer, there is an embedding matrix \mathcal{V} of size $n \times |V|$ where $|V|$ is the length of the vocabulary and n is the embedding size (e.g. 300). At each iteration, this matrix is multiplied by the one-hot input vectors. Embedding vectors are obtained for each input word and the states of the hidden layers are computed by averaging them. Then, the NN back-propagates its loss and the embedding matrix is updated to become more accurate in its predictions. At the end of training, the model has captured the relations between the words in its embedding matrix.

In the opposite, the SG model is trained to predict the context words given a target. The inner working is mostly the same, the embedding matrix is used to predict the output words and is updated after each iteration. The main difference is that SG does not require to average the embedding vectors of the input (which smooths the distributional information in the CBOW

¹<https://code.google.com/archive/p/word2vec/>

²<https://nlp.stanford.edu/projects/glove/>

³<https://fasttext.cc/>

⁴Image taken from the original paper of Mikolov et al. [17].

model) since there is only one input word here. This is the reason why SG is usually preferred over CBOW, at least for large datasets.

GloVe vectors aim to use words co-occurrence probabilities combined to the SG model to take advantage of the statistics of the dataset. GloVe embeddings outperform traditional embeddings on word analogy tasks.

FastText is another model which learns embedding on sub-words instead of words. It decomposes the words into n-grams of several characters and learns embeddings for those n-grams. The main advantage of FastText is that it can deal with unknown words by decomposing them into known sub-words.

Recently, new embedding vectors have emerged, like ELMo [21] or BERT [22] embeddings. Those methods produce contextualized embeddings, which means that a word with several meanings depending on the context will also have multiple context-dependant vectors. The hidden layers of ELMo and BERT are more sophisticated than the ones of Word2Vec. ELMo uses bidirectional LSTM [23] layers while BERT is composed of a sequence of Transformer [24] Encoder Blocks. Both techniques use the states of their hidden layers to produce the embedding vectors.

2.2 Embeddings in hyperbolic space

All kinds of word embeddings described above are learnt in the Euclidean vector space, which allows to capture semantic similarities but do not account for hierarchical relations. In 2017, M. Nickel and D. Kiela [13] proposed a new approach for building word embeddings that capture the latent hierarchy of texts in hyperbolic space. The constant negative curvature of hyperbolic space have proven to enable representing tree-structured data while preserving the distances between the nodes [25]. In their work, the authors have used a n-dimensional Poincaré ball model and have developed a method for learning embeddings based on the Riemannian Stochastic Gradient Descent (RSGD) [26].

Using the equations and notations of Nickel and Kiela, the main points of their method for learning embeddings in the Poincaré ball is described below. For more details and explanations on this algorithm and hyperbolic space in general, the reader is suggested to refer to the original paper.

Formally, the open n-dimensional Poincaré ball is defined as $\mathcal{B}^n = \{x \in \mathbb{R}^n \mid \|x\| < 1\}$ with $\|\cdot\|$ being the Euclidean norm. The Riemannian manifold⁵ representing our Poincaré ball model is noted (\mathcal{B}^n, g_x) , where the Riemannian metric tensor g_x is the following:

$$g_x = \left(\frac{2}{1 - \|x\|^2} \right)^2 g^E$$

with g^E being the Euclidean metric tensor.

⁵A manifold is a high-dimensional space.

In this work, we are interested in learning embedding vectors which model the hidden hierarchies of words in texts. The model could have been trained to predict missing relations in networks or to learn lexical entailment by choosing different loss functions as described in the source paper cited above [13], but this is not the focus of this work. Our objective however, is to train the model on the reconstruction task, which consists in reconstructing the hypernymy relations of the words from their embeddings. As described in the initial paper, the following loss function should be minimized to that purpose:

$$\mathcal{L}(\Theta) = \sum_{(u,v) \in \mathcal{D}} \log \frac{e^{-d(u,v)}}{\sum_{v' \in \mathcal{N}(u)} e^{-d(u,v')}},$$

where $\mathcal{N}(u) = \{v \mid (u,v) \notin \mathcal{D}\} \cup \{u\}$ is a set of random negative samples of u which are not in \mathcal{D} . $d(u,v)$ represents the distance measure between the words u and v in the Poincaré ball and is equal to

$$d(u,v) = \operatorname{arcosh} \left(1 + 2 \frac{\|u-v\|^2}{(1-\|u\|^2)(1-\|v\|^2)} \right).$$

The objective is to optimize the distances between the words in order to minimize the loss function. In other words, minimizing this loss function encourages the embeddings of related words to be close to each other and embeddings of unrelated words to be further apart from each other. Therefore, the optimization procedure consists in updating the vectors θ such that

$$\theta_{t+1} = \Re_{\theta_t}(-\eta_t \nabla_R \mathcal{L}(\theta_t)),$$

where $\Re_{\theta}(v) = \theta + v$ is the retraction onto \mathcal{B} at θ , η_t is the learning rate at time t and $\nabla_R \mathcal{L}(\theta)$ is the Riemannian gradient of the loss function. The Riemannian gradient can be expressed as a rescaling of the Euclidean gradient ∇_E by the inverse of the Poincaré ball metric tensor g_{θ}^{-1} (g_{θ} being a scalar matrix, its inverse is trivial to compute).

The Euclidean gradient ∇_E is equal to $\frac{\partial \mathcal{L}(\theta)}{\partial d(\theta,x)} \frac{\partial d(\theta,x)}{\partial \theta}$.

By computing the partial derivative of the distance measure d by setting

$$\alpha = 1 - \|\theta\|^2, \quad \beta = 1 - \|x\|^2 \quad \text{and} \quad \gamma = 1 + \frac{2}{\alpha\beta} \|\theta - x\|^2,$$

the following expression is obtained:

$$\frac{\partial d(\theta,x)}{\partial \theta} = \frac{4}{\beta \sqrt{\gamma^2 - 1}} \left(\frac{\|x\|^2 - 2\langle \theta, x \rangle + 1}{\alpha^2} \theta - \frac{x}{\alpha} \right).$$

Finally, the embeddings are constraint to stay in the scope of our Poincaré ball by applying

the following projection:

$$\text{proj}(\theta) = \begin{cases} \theta/\|\theta\| - \epsilon & \text{if } \|\theta\| \geq 1 \\ \theta & \text{otherwise.} \end{cases}$$

The parameter update function can finally be reformulated as

$$\theta_{t+1} \leftarrow \text{proj}\left(\theta_t - \eta_t \frac{(1 - \|\theta_t\|^2)^2}{4} \nabla_E\right).$$

This optimization process is applied multiple times until the loss converges to a minimum value.

For the sake of completion, it should be mentioned that, in 2018, Nickel and Kiela [14] proposed a new and more efficient approach based on the Lorentz model instead of the Poincaré ball. The later work of Roller et al. [15] further improved the performance of this model by developing entailment cones [27] in the Lorentz manifold.

2.3 Bayesian topic modeling

Before describing the main topic models studied in this paper, it is important to understand what are the Beta and Dirichlet distributions.

2.3.1 Beta distribution

The Beta distribution is a continuous univariate distribution of probabilities defined on the interval $[0, 1]$. It takes 2 positive values as parameters: α and β . The Beta distribution is a useful distribution for modeling proportions. Its probability density function is defined by the equation:

$$f(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}, \quad B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha + \beta)} \quad (2.1)$$

where Γ is the Gamma function and B is the Beta function. $B(\alpha, \beta)$ is a normalization term ensuring that probabilities sum to 1. The shape of the distribution is controlled by the 2 positive parameters α and β . In the example of Figure 2.2, it can be seen that the probability density function (PDF) is symmetric when α and β are equal.

More concretely, Beta distribution is used to model prior expectations of an event. Parameters α and β define the range of probabilities which is the most likely for an event to occur.

2.3.2 Dirichlet distribution

The Dirichlet distribution is the generalization of the Beta distribution to multiple random variables. Thus, it is a multivariate and continuous distribution of probabilities. Its output

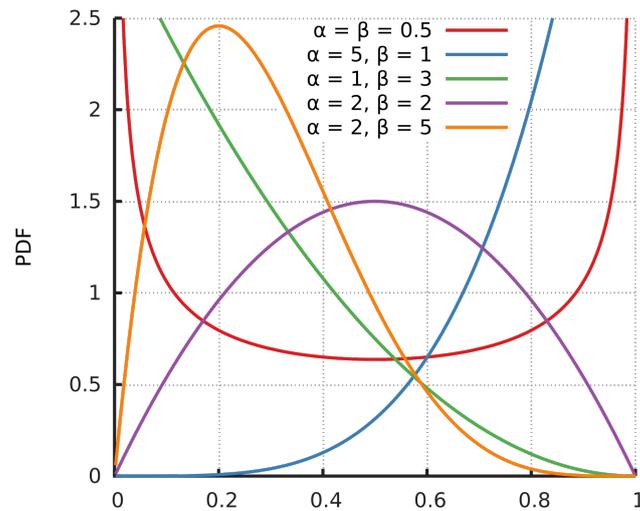


Figure 2.2: Probability density function of the Beta distribution with different α and β .⁶

is a K -dimensional vector of real values summing to 1 (one value for each of the K random variables). The Dirichlet distribution is parameterized by a K -dimensional vector α of positive values, which determines how the probability mass is distributed over the random variables. Figure 2.3 represents the probability density function of several Dirichlet distributions over 3 random variables (the 3 corners of the triangle) with different vector parameters α .

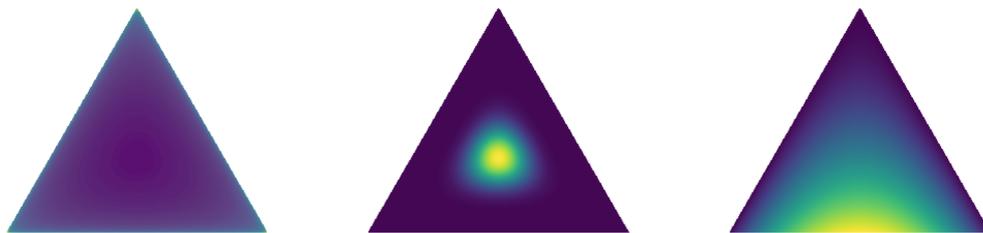


Figure 2.3: Probability density function of Dirichlet distributions with $\alpha = [0.9, 0.9, 0.9]$, $[10, 10, 10]$ and $[2.0, 2.0, 1.0]$ respectively (from left to right).

The Dirichlet distribution ($Dir(\alpha)$) is generally used as a prior in popular topic models like Latent Dirichlet Allocation (LDA) to represent the document-topic distribution and the topic-word distribution in a corpus of documents.

⁶Figure from https://en.wikipedia.org/wiki/Beta_distribution

2.3.3 Dirichlet Process

The Dirichlet Process $DP(\alpha_0, G_0)$ is a stochastic process over a space Θ , where G_0 is the base distribution over Θ and α_0 is a positive real number. It is often presented as the infinite extension of the Dirichlet distribution, as the distribution produced by the DP is discrete but of infinite size. A probability measure G drawn from $DP(\alpha_0, G_0)$ is also a probability distribution over the probability space Θ . Each draw G is thus a distribution over probability distributions. If Θ is split into finite measurable partitions (A_1, \dots, A_k) , the distribution $(G(A_1), \dots, G(A_k))$ is drawn from $Dir(\alpha_0 G_0(A_1), \dots, \alpha_0 G_0(A_k))$. This is what makes the DP a suitable tool for non-parametric Bayesian mixture models, which are widely used in topic modeling. The following methods have been introduced to construct a DP.

The **Stick-breaking construction** has been introduced by Sethuraman [28]. Starting with a stick of length 1, at each iteration k , the stick is broken at point β_k and set π_k to be equal to the length of the broken part of the stick. Then, this process is repeated with the remaining part of the stick to produce π_{k+1} , etc.

Mathematically, it can be represented by the following formula:

$$\pi_k = \beta_k \prod_{i=1}^{k-1} (1 - \beta_i), \quad \beta_k \sim Beta(1, \alpha_0) \quad (2.2)$$

with θ_k^* being a random variable drawn from the base distribution $\theta_k^* \sim G_0$ and $\delta_{\theta_k^*}$ being the indicator function for θ_k^* . The distribution G can be written as the sum of the length of every portions of the stick:

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k^*} \quad (2.3)$$

Therefore, $G \sim DP(\alpha_0, G_0)$.

The **Chinese Restaurant Process** (CRP) is a metaphor, attributed to Jim Pitman and Lester Dubin in Aldous' book [29], which explains a distribution over partitions. It assumes a sequence of people arriving one by one in an initially empty restaurant containing K tables. Every time a new customer arrives, (s)he has the choice to sit at an already occupied table k with probability $\frac{c_k}{n+\alpha}$ (with c_k the number of people already sitting at table k and n the total number of people in the restaurant) or to go to an empty table with probability $\frac{\alpha}{n+\alpha}$ where α is the dispersion factor. As deducted, the more customers in the restaurant, the less likely new customers are to choose an empty table and thus, the number of occupied tables stabilizes.

The **Blackwell-MacQueen urn scheme** [30] is an extension of the Pólya Urn scheme [31] with balls of multiple colours.

The Pólya urn scheme is a stochastic sampling scheme producing exchangeable but not in-

dependent draws. It operates on the following principle. A non-transparent urn contains an equal number of white balls and black balls at the beginning. At each step of the process, a ball is drawn. It is then returned in the urn with a new ball of the same color. As the initial configuration is balanced (i.e. there are as many white balls as black balls), the urn composition converges to a continuous uniform distribution. In other words, a “rich-get-richer” process is observed in the evolution of the urn composition and the proportion of balls converges to a Beta distribution.

The Blackwell-MacQueen urn scheme also uses a non-transparent urn containing balls, but of multiple colours, beyond black and white. The same principle of drawing is applied.

With K colors, the number of balls of color k contained in the urn is noted α_k . At the end of the process, the proportions of the balls of different colours are distributed according to $Dir(\alpha_0, \alpha_1, \dots, \alpha_K)$.

2.3.4 Bayesian models

All models that are described in this work are Bayesian probabilistic models, meaning that the uncertainty of the input and the output of each model is represented by probabilities. In Bayesian statistics, the prior knowledge is updated each time new data are observed. The aim of these models is to develop learnings about unobserved data (here, the document-topic distribution) by using some observations (i.e. the topic-word distribution). This process is called statistical *inference*. It can be expressed by the so-called Bayes theorem as follows:

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} \quad \text{with } p(x) = \int_{\theta} p(x|\theta)p(\theta)d\theta \quad (2.4)$$

where:

- $p(z|x)$ is the updated knowledge as known as the “posterior”,
- $p(x|z)$ is the “likelihood” or the probability of the observed data x given the parameter z ,
- $p(z)$ is called the “prior” (i.e. the probability to draw z , no matter what x is),
- and $p(x)$ is the “evidence” or the probability to observe x , independently of the value of z .

Usually, in most Bayesian problems, the prior and the likelihood are already known. However, the evidence, which acts as a regularization term, becomes difficult to compute in high dimensional problems and the exact inference process becomes practically impossible. In those cases, approximation or sampling techniques must be used to compute the posterior.

2.4 Topic models

Topic models are usually based on unsupervised methods, because words in documents are rarely annotated with a corresponding topic. Usually, the original structure of a corpus or a document

is unknown and to be discovered by the topic model. Topic models learn the inner structure of a corpus by analysing its terms and extracting information from their statistics. Moreover, the number of topics is not always known in advance and, while some models are working with fixed numbers of topics, some others are able to determine themselves the number of topics present in a corpus or a document.

In this section, 3 probabilistic topic models built on Dirichlet distributions are described, namely the Latent Dirichlet Allocation, the Hierarchical Dirichlet Process and the nested Hierarchical Dirichlet Process. The first model (LDA) is a parametric Bayesian model assuming Dirichlet prior distributions for topics over documents and for words over topics. The second model (HDP) is a non-parametric version of LDA based on Dirichlet Processes and the last model (nHDP) is a hierarchical version of HDP. The last two models have been used in our experiments.

2.4.1 Latent Dirichlet Allocation

Generative model

LDA is a popular model which has been widely used in NLP, it is based on the Dirichlet distribution and assumes a fixed and already known number of topics. It is called a generative model because words are supposed to be Dirichlet distributed over topics and, similarly, topics are Dirichlet distributed over documents (i.e. topics are seen as a mixture of words and documents as a mixture of topics). Since all observed documents and topics are assumed to be generated from these distributions, the latter could be used to generate new topics or new documents.

Figure 2.4 represents the generative process of documents with LDA. The document-topic distribution θ of each document can be seen as a sample drawn from the Dirichlet distribution parameterized by α . Similarly, the topic-word distribution ϕ is supposed to be drawn from the Dirichlet distribution parameterized by β .

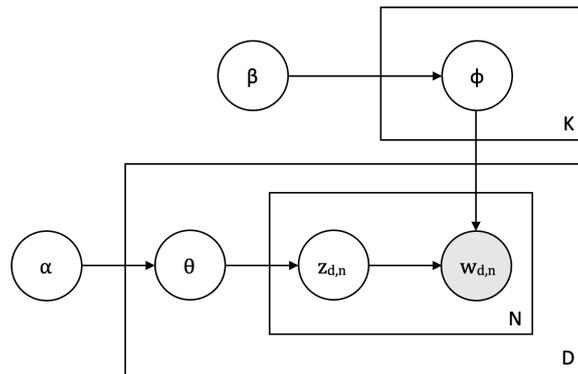


Figure 2.4: Generative process of documents with the Latent Dirichlet Allocation

With K being the total number of topics present in a corpus of documents, α the dispersion

parameter of the document-topic Dirichlet distribution and β the dispersion parameter of the topic-word Dirichlet distribution, LDA assigns each instance of a word to one of the topics. The matrix θ contains the proportion of each topic in each document while the matrix ϕ contains the word distribution of each topic. In a document d , the assignment of the n^{th} word $w_{d,n}$ to a topic k is represented by $z_{d,n}$ in the Figure 2.4, with $n \in 1, \dots, N$ and $d \in 1, \dots, D$ (N being the number of words in document d and D the number of documents in the corpus). Each document is thus a mix of some of the K topics.

The base assumptions made in the LDA model are that the word-topic assignments are sampled in such a manner that $z_{d,n} \sim \text{Multinomial}(\theta)$ and the observed words $w_{d,n}$ of each document are drawn from $\text{Multinomial}(\phi)$.

Inference and training

Expressed in the terms of the Bayes rule, the inference formula of LDA can be defined by

$$p(\theta, \mathbf{z} | \mathbf{w}, \alpha, \beta) = \frac{p(\theta, \mathbf{z}, \mathbf{w} | \alpha, \beta)}{p(\mathbf{w} | \alpha, \beta)} \quad (2.5)$$

As described in the section about Bayesian models, the inference procedure is equivalent to finding the posterior distribution (in our case, the topic proportions of documents and the word-topic assignments) based on the prior, the likelihood and the evidence. However, the computation of the latter term is intractable. Therefore, two kinds of methods can be used to compute $p(\theta, \mathbf{z} | \mathbf{w}, \alpha, \beta)$: sampling and estimating.

Gibbs sampling and variational inference are the most popular methods of both types according to the literature.

Gibbs sampling is a Markov Chain Monte Carlo (MCMC) sampling technique, meaning that it generates samples in the form of a Markov chain. It consists of assigning each word of each document to a topic by looking at the number of times the current document has already used this topic (that we further note $n_{d,k}$) and the number of times this topic uses the given word (noted $v_{k,w}$). For LDA, the process is quite simple and takes place as follows:

1. The existence of a base word-topic assignment is assumed for each word in each document.
2. Then, iterations are performed on all words of all documents. For each word $w_{d,n}$:
 - (a) The word is first unassigned from its current topic
 - (b) A new topic is then sampled with probability proportional to

$$\frac{n_{d,k} + \alpha_k}{\sum_i^K n_{d,i} + \alpha_i} \frac{v_{k,w_{d,n}} + \beta_{w_{d,n}}}{\sum_i v_{k,i} + \beta_i}$$

- (c) The word is reassigned to this newly sampled topic

(d) The procedure is repeated for every words and every documents

In the equation of step (b) above, α is the Dirichlet parameter for document-topic distribution and β is the Dirichlet parameter for topic-word distribution as defined in the previous section.

Variational inference is an approximation method that is usually faster than sampling methods but that is not exact. Its objective is to find the optimal variational parameters of a function $q(z)$ to approximate at best the true posterior distribution $p(z|x)$. The choice of $q(z)$ is important in the sense that it should be complex enough to fit well the original distribution but should also be simple enough to be computationally tractable.

To find optimal variational parameters, the approach is to minimize the Kullback-Leibler (KL) divergence [32] between $q(z)$ and $p(z|x)$, which is defined by

$$KL(q(z)||p(z|x)) = \mathbb{E}_q \left[\log \frac{q(z)}{p(z|x)} \right] \quad (2.6)$$

It cannot be minimized directly because $p(z|x)$ is unknown and to be inferred. However, the ‘‘Evidence Lower Bound’’ (ELBO) of the model $p(z|x)$ which is derived as follows, can be minimized by applying the Jensen’s inequality $f(\mathbb{E}[x]) \geq \mathbb{E}[f(x)]$:

$$\begin{aligned} \log p(x) &= \log \int_z p(x, z) \frac{q(z)}{q(z)} \\ &= \log \left(\mathbb{E}_q \left[\frac{p(x, z)}{q(z)} \right] \right) \\ &\geq \mathbb{E}_q[\log p(x, z)] - \mathbb{E}_q[\log q(z)] \end{aligned}$$

It can be observed that, with some calculations, the KL divergence can be expressed as the negative ELBO plus a constant $\log p(x)$.

$$\begin{aligned} KL(q||p) &= \mathbb{E}_q \left[\log \frac{q(z)}{p(z|x)} \right] \\ &= \mathbb{E}_q[\log q(z)] - \mathbb{E}_q[\log p(z|x)] \\ &= \mathbb{E}_q[\log q(z)] - \mathbb{E}_q[\log p(x, z)] + \log p(x) \\ &= -(\mathbb{E}_q[\log p(x, z)] - \mathbb{E}_q[\log q(z)]) + \log p(x) \end{aligned}$$

Consequently, by maximizing the ELBO, the KL divergence is also minimized.

2.4.2 Hierarchical Dirichlet Process

As just explained, LDA is a parametric Bayesian topic model, constrained by a fixed number of topics. In many cases, the number of topics is not known in advance. To find the right K to

choose with LDA, the model has to be trained multiple times with different values of K to find which one fits the best the dataset.

The HDP model is an extension of LDA introduced by Teh et al. [1]. This model allows to work with an undetermined number of topics. With HDP, the corpus is seen as a draw of a Dirichlet Process itself (i.e. a distribution over topics), called the base DP. All documents are also considered as mixtures of topics distributed by Dirichlet processes. Each document-related DP takes as input a vector α_0 and the base DP. Distributing the base DP for each document ensures that all document-related DPs share the same topics as the base distribution. Each document is then a distribution of the topics themselves distributed over the corpus.

Figure 2.5 shows a graphical representation of the HDP model.

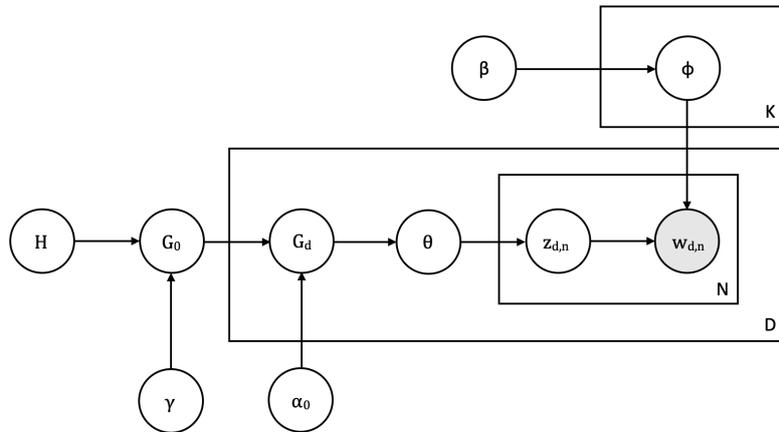


Figure 2.5: Generative process of documents with the Hierarchical Dirichlet Process

Chinese Restaurant Franchise

In their original paper [1], Teh et al. describe an analog of the CRP, called the “Chinese Restaurant Franchise” (CRF) where multiple restaurants share the same menu. In CRF, each table can take only one dish, chosen by the first customer sitting at that table, but several tables in the restaurants can order the same dish. When applied to topic modeling, the CRF represents the topic-word and the document-topic distribution processes in the HDP model. Customers characterize the words, dishes are associated to topics and restaurants can be linked to documents.

More formally, CRF is characterized by:

- $G_0 \sim DP(\gamma, H)$, with H the base distribution and γ the dispersion parameter of the base DP.
- $G_j \sim DP(\alpha_0, G_0)$, with α_0 the dispersion parameter of the restaurant-specific DP.

- θ_{ji} , the i^{th} customer arriving at restaurant j ;
- ϕ_1, \dots, ϕ_K being the set of dishes shared among the restaurants (distributed according to H , the base distribution);
- ψ_{jt} being the dish chosen at table t in restaurant j ;
- n_{jtk} being the number of customers in restaurant j sitting at table t where dish k is served. It is noted n_{jt} as the table in restaurant j is fixed to t because the dish is determined by the choice of the table.
- $m_{\cdot k}$, the number of tables, all restaurants combined, serving dish k and, analogously, $m_{\cdot\cdot}$ is the total number of tables in all restaurants.

The conditional distribution for customer θ_{ji} is drawn from:

$$\theta_{ji} | \theta_{j1}, \dots, \theta_{j,i-1}, \alpha_0, G_0 \sim \sum_{t=1}^{m_j} \frac{n_{jt}}{i-1+\alpha_0} \delta_{\psi_{jt}} + \frac{\alpha_0}{i-1+\alpha_0} G_0 \quad (2.7)$$

In the same idea, since G_0 is distributed according to $DP(\gamma, H)$, the dish assignment ψ_{jt} for table t in restaurant j is drawn from:

$$\psi_{jt} | \psi_{11}, \psi_{12}, \dots, \psi_{21}, \dots, \psi_{j,t-1}, \gamma, H \sim \sum_{k=1}^K \frac{m_{\cdot k}}{m_{\cdot\cdot} + \gamma} \delta_{\phi_k} + \frac{\gamma}{m_{\cdot\cdot} + \gamma} H \quad (2.8)$$

δ represents the indicator function.

The two equations of the CRF concept described above are used in HDP to represent the topic-word and the document-topic assignment processes.

2.4.3 Nested Hierarchical Dirichlet Process

In 2012, Paisley et al. [?] developed the nHDP model, allowing to find hierarchical topic representations. The nHDP model builds on the nCRP framework, which is a nested form of CRP where a table in a restaurant gives access to another restaurant whose tables give access to other restaurants and so forth. In topic modeling, the corpus is seen as a tree where nodes represent topics and each word follows a path along that tree. Each document is a sub-tree of the corpus starting at the root and whose branches define the topics it contains.

In their work, Paisley et al. have implemented a stochastic variational inference (SVI) algorithm to train their model. The SVI works like the variational inference but, at each iteration, it learns its local variational parameters on mini-batches instead of the whole corpus. For more details about SVI, it is advised to refer to the original articles on nHDP [2] and SVI [33].

Chapter 3

Experiments

Our experiments have been conducted on the HDP and nHDP models on two relatively similar datasets in English and in French. The coherence of the generated topics, the perplexity of both models and the impact of the language were compared. LDA was not included in our tests since it is only a parametric version of HDP.

In addition to the experiments made with the topic models, the training of Poincaré embeddings of the nouns extracted from the corpus was also studied in order to determine if hyperbolic embeddings could be used as an efficient topic modeling technique.

3.1 Experimental setup

3.1.1 Data collection

The experiments have been conducted on two subsets of Wikipedia’s articles. This choice was made because Wikipedia is the biggest encyclopedia of the Internet and most of its articles are available in many languages. Our objective was to have similar corpora in English and in French and it was intended to meticulously select corresponding articles in both languages among multiple categories. Unfortunately, we did not find any way to download or crawl corresponding English and French articles to build parallel corpora in an efficient and automatic manner. Consequently, we have resolved to select the 10,000 first pages of articles of both English¹ and French² Wikipedia dumps of November 1, 2021.

3.1.2 Pre-processing

Pre-processing is known to be an important step in most of NLP tasks, sometimes considerably improving results obtained with plain text. Some words, or families of words, may be discarded to

¹Available at: <https://dumps.wikimedia.org/enwiki/20211101/enwiki-20211101-pages-articles.xml.bz2>

²Available at: <https://dumps.wikimedia.org/frwiki/20211101/frwiki-20211101-pages-articles.xml.bz2>

ensure models focus on meaningful words. For example, in the 10,000 English Wikipedia articles that have been extracted, the most common words are “the”, “of”, “and”, “in”, “to”, etc., which are semantically useless for a topic modeling purpose. Those words are called “stop-words” and are usually removed during the pre-processing step.

The Gensim³ library was used for the extraction of the plain text from the compressed XML Wikipedia dumps but also for their tokenization. Gensim already applied a pre-processing job on the texts in order to ignore too small articles, to remove words with too many or too few characters, to suppress HTML tags, to lower-case all words and to remove numbers.

Afterwards, a second pre-processing step has been performed on our datasets. After the common stop-words were removed from the tokens, the 20 most frequent words of the vocabularies were then analysed and appeared to give a clearer idea of what the articles are about. It has then been decided to keep only words with certain Part-of-Speech (POS) tags, namely adverbs, adjectives, verbs and nouns. Those words have then been lemmatized, i.e. only the root of the words was kept and all their variations (e.g. “watch”, “watches”, “watching”, “watched”) were discarded. POS filtering and lemmatization have been performed with the Spacy⁴ library, which proposes those operations for multiple languages, including English and French.

Finally, a bi-gram model was used to keep composed words⁵ connected.

The final English vocabulary (i.e. the set of all unique words) contains 503,204 different tokens and the articles contain in average 1,657.28 words after the pre-processing step. Concerning the French corpus, its final vocabulary contains 433,746 distinct tokens and the articles are 1,034.17 words long in average after pre-processing. It can be observed that the English corpus is larger in its vocabulary and number of words per document than the French corpus. It is simply due to the greater richness of the English version of Wikipedia.

Finally, all documents of both corpora have been transformed according to the *Bag of Words* (BOW) model, which assumes that the order of the words in a sentence does not matter. In the BOW model, each document is seen as a set of tuples composed of the different words and their corresponding number of occurrences in the document.

Table 3.1 is presenting the 20 most common words of the vocabulary of both corpora after the pre-processing steps.

3.2 Methodology

3.2.1 HDP

For the HDP model, we used the python Tomotopy⁶ library, which provides a lot of topic models including LDA and HDP. The HDP model implemented by Tomotopy is inspired from the papers

³<https://radimrehurek.com/gensim/>

⁴<https://spacy.io/>

⁵i.e. words that are appearing frequently one after the other (e.g. “New-York”).

⁶<https://bab2min.github.io/tomotopy/v0.12.2/en/>

English corpus		French corpus	
Word	Occurrences	Word	Occurrences
include	60431	français	48540
year	55025	france	30285
time	53565	grand	29005
state	43277	ville	28427
work	40815	année	25758
world	38780	fichier	24713
form	34084	pays	23729
know	32550	catégorie	21637
early	31715	état	20851
system	31569	américain	19321
city	30862	ancien	18912
country	30680	devenir	18811
call	30536	commune	17885
american	29714	nouveau	17284
number	29430	pari	16463
large	28693	utiliser	16352
write	27865	saint	16125
base	27649	politique	15899
century	26862	bien	14884
people	26362	partie	14605

Table 3.1: Number of occurrences of the top 20 most frequent words in the French and English corpora.

of Newman et al. [10] and Teh et al. [1] and uses a Collapsed Gibbs sampler for inference. This version of the HDP model was preferred over the version implemented in Gensim which uses variational inference in order to observe the evolution of the number of topics along the training. We set the parameters of the model with similar values as used by Teh et al. [1] in their experiments for both of our corpora (i.e. English and French). The only parameter that we have made vary in our experiments is the weighting scheme of the words.

Various weighting schemes were indeed applied to the tokens to determine their influence on the HDP model performances. The first weighting scheme, called “ONE”, simply counts the number of occurrences of each word in the document. Thus, each word is weighted proportionally to its number of occurrences.

The second way to assign a weight to each token is to use the inverse document frequency (IDF) of the word, that can be represented as:

$$IDF(w) = \log \frac{|D|}{N_d(w)}$$

with $|D|$ being the number of documents in the corpus and $N_d(w)$ being the number of document which contain word w . With IDF, words appearing in fewer documents have more weight than those appearing in many documents.

The third weighting scheme that was used in our experiments is the point-wise mutual information (PMI) of a word and a given document, formulated as:

$$PMI(w, d) = -\log \frac{p(x_i|d)}{p(x_i)} = -\log \frac{\#[\text{tokens } x_i \text{ in } d]}{\#[\text{tokens } x_i]}$$

With PMI, when a word appears only a few times in a document but is frequent in the corpus, this word receives a small weight. Contrarily, rare words and very document-specific words get more weight.

Those weighting schemes were already implemented in Tomotopy and are inspired from the work of Wilson and Chew [34].

3.2.2 nHDP

For our experiments with the nHDP model, the original model implementation of John Paisley et al. [2] was used⁷. For this model, only the ONE weighting scheme was used for testing its performances. However, IDF and PMI could also have been applied but due to time constraints, it has not been done in this work.

The model implementation is the code written by the authors for conducting their experiments described in the source paper and was not intended to be used for any other purpose. What is meant here is that the code was very difficult to handle by anyone other than its authors because of the poor documentation and the lack of clarity in the names of the variables. Primarily, being more used to code in Python, a first attempt was made to translate the code from Matlab to Python using Numpy⁸ and SciPy⁹ toolboxes to deal with vectorial computation. The new code was doing a good job for very small datasets but was unsuccessful at dealing with datasets bigger than several hundreds of documents. Therefore, it was decided to come back to the original code written in Matlab.

Even with the original Matlab implementation of nHDP, only datasets of maximum 1,000 documents could be trained by the model. With bigger datasets, the machine used for those experiments¹⁰ always ran out of memory and the executions of the code were stopped. Given those limitations, our nHDP model evaluation was then conducted on reduced versions of our datasets of 1,000 documents for each language. The size of the vocabulary and number of words per document of those new corpora have been reported in Table 3.2.

It is notable that the reduced corpora are more balanced than the original one (i.e. the vocabulary size and the average number of words per document are very similar in French and in English).

During our experiments, all hyper-parameters were kept as they were set in the original

⁷The code is available here: <http://www.columbia.edu/~jwp2128/code/nHDP.zip>

⁸<https://numpy.org/>

⁹<https://scipy.org/>

¹⁰Apple iMac 2019 3.7GHz Intel Core i5, 8Go RAM.

	Vocabulary	Nb words/doc
English	147,448	1,667.52
French	142,302	1,566.01

Table 3.2: Statistics of corpora used with nHDP.

implementation. The only changes made in the code are the batch size (that we set to 200 documents), the number of iterations (chosen to be 50), and the addition of a piece of code to compute the training perplexity of the model. All other parameters, like the dispersion parameters, the number of levels of the topic-tree, the number of child-nodes per level and the learning rate were left unchanged.

The topic tree produced by the algorithm has been organised in 3 levels (under the root which is not represented as it would contain all tokens of the corpus), starting from the top:

- Level 1, containing “general” topics;
- Level 2, containing “specific” sub-topics;
- Level 3, grouping “specialized” sub-topics.

The number of nodes per branch is limited per level. The tree can contain a maximum of twenty “general” topics. Each general topic can have at most ten child nodes (i.e. ten “specific” sub-topics) and each “specific” topic can subdivide into five “specialized” sub-topics.

It should also be mentioned that batches were composed of randomly picked documents from the corpus, as it was already the case in the original code. As a consequence, some documents of the corpus could have not been used to train the model, while others could have been chosen multiple times (but at most once per batch). The original tree initialization phase using K -means clustering with L1 distance measure proposed in the implementation was also used to get a prior distribution of words over the tree. For more details on the initialization step, the reader is advised to check out the original paper of the nHDP model [2].

3.2.3 Poincaré embeddings

Poincaré embeddings have been learnt on the English corpus of 10,000 documents.

To learn the Poincaré embeddings of the words composing our corpus, their hypernymy relations should first be learnt. Several techniques exist to do so but the two most used methods are based on Hearst Pattern matching [35] or distributional representations (usually inspired from the Distributional Inclusion Hypothesis, abbreviated DIH [36]). In this work, the Hearst Patterns matching method was used in order to scan the texts and find the latent *is-a* relations of our corpus. Only nouns and nominal groups have such hypernymy relations. For this reason, we have used patterns which only extract pairs of nouns or nominal groups.

To detect those patterns, we have performed a lighter pre-processing job than we did for topic models. After filtering all special characters, only the lemmatization of the words have

been performed. In this case, the BOW assumptions does not apply because the order of the words determines if the sentence matches one of the Hearst Patterns or not. Therefore, all tokens have been kept, the order of the words has been preserved and all sentences have been checked to be well separated by white spaces. Nouns and nominal groups have been identified with the Spacy library. Nominal groups have been transformed into a single nominal token and all nominal tokens (i.e. nouns and transformed nominal groups) have been tagged with the flag “NP_” to be recognized by the patterns.

The hypernymy relations were detected by using the code of `mmichelsonIF`¹¹. This code provided a well-constructed list of Hearst Patterns that were tested and modified to match our needs. All patterns used to detect our hypernyms have been listed in Table C.1 presented in Appendix C. For each observed pair of words, the number of times it has been extracted from the text has been counted. If a pair (u, v) and its inverse direction pair (i.e. the pair (v, u)) have been extracted both from the text, only the most occurring pair was kept. The quality of our relations was then evaluated on different datasets by following the procedure of Roller et al. [15]. The results obtained with the complete set of extracted and filtered hypernyms were compared with 2 smaller subsets where only pairs occurring more than 2 and 3 times were kept. It was then decided to learn multiple word embeddings with these different lists of pairs in order to evaluate which one gives the best results.

The model that we have used to learn the Poincaré embeddings is the one from Nickel and Kiela described in Chapter 2 [13]. It is the original implementation of the authors, written in PyTorch¹². Embeddings of size 10 have been trained in order to evaluate them on the reconstruction task as described in the initial paper [13] and embeddings of size 2 have also been trained in order to represent them on a 2-dimensional projection of the Poincaré ball to visualize their hierarchical relations. This implementation has been chosen against the one of Gensim because the original PyTorch implementation came with the testing process ready and easy to use.

3.3 Evaluation

3.3.1 Topic models

Evaluation metrics have been a field of research on their own in topic modeling. There is not a standard evaluation metric to assess the performances of a topic model. Perplexity is commonly used to evaluate probabilistic language models, it is usually measured on a separate test set and indicates how the model is perplex about the unseen documents. The formula of the perplexity measure can be expressed as the exponential of the negative log-likelihood. Therefore, the lower the perplexity, the better the model. Perplexity has often been used at first in topic modeling too. However, this measure has shown to be uncorrelated with human judgement concerning topic evaluation [37]. Then, several coherence measures based on PMI or NPMI (i.e. normalized

¹¹The code can be downloaded here: https://github.com/mmichelsonIF/hearst_patterns_python

¹²The code is available at: <https://github.com/facebookresearch/poincare-embeddings>

PMI) appeared to better evaluate the quality of the topics. Recent works have also focused on the use of human judgement, as in the word intrusion and topic intrusion tasks [38]. Those methods give interesting results but suffer from long manual set-up and data collection phases via surveys completion.

In this work, the U_{mass} coherence score was used to assess the quality of the topics found by our models. The main reason of this choice is the quick computation time. Unlike other traditional coherence measures (like C_v , C_{uci} , etc.), the U_{mass} score computes the coherence by looking at occurrences and co-occurrences of words of the same topics in the whole documents, rather than in a sliding window.

Coherence measures are not standardised, meaning that various sources may not necessarily agree on the way to compute them and their interpretation may change from author to author. In this work, the U_{mass} score of a topic is computed as described below.

The words of the topics are sorted by popularity and only the N most popular ones are selected (in our experiments, we set $N = 20$). Then, for each word w_v , from the most popular to the least popular, a search is performed to find co-occurrences in the documents with the topic words that are more popular than the word w_v itself (i.e. the ones appearing before in the list, noted as w_u in the following formula):

$$U_{mass} = \log \frac{D(w_u, w_v) + \epsilon}{D(w_v)}$$

$D(w_u, w_v)$ represents the number of documents where w_v and w_u are both appearing,

$D(w_v)$ is the number of documents containing word w_v ,

and ϵ is a small number added for numeric stability (we used $\epsilon = 1e-12$ in our experiments).

For example, in a topic “Education”, if the 5 top words are *student*, *course*, *university*, and *book* sorted in order of popularity, the model will first look at co-occurrences of pair (*student*, *course*), then (*university*, *course*) and (*university*, *student*), and so on. Finally, the mean of the scores obtained with all pairs of words is used to determine the U_{mass} score for that topic.

The formula reveals that the U_{mass} score will be negative (because of the log) and more coherent topics should have a score close to 0.

For the HDP model, the mean U_{mass} coherence score of its generated topics has been calculated based on their 20 most used words. In the case of our nHDP model, the chosen metric has been the per-level mean U_{mass} score, also based on the 20 most used words of each topic.

The log-likelihood and the perplexity of the model have also been calculated on both corpora and respective test sets in order to compare them to the average coherence scores. More precisely, the log-likelihood of the HDP model has been computed on the corpora and on French and English held-out test sets of 2,000 documents each. However, the perplexity has been calculated for the nHDP model. Due to the reduced size of corpora used with this model, the size of the

test sets for these experiments has been reduced to 200 documents each.

To summarize, for each experiment with HDP, the model has been trained on 10,000 documents. The average U_{mass} score has been calculated with the top 20 words of each topic and the log-likelihood of the model has been measured.

For the nHDP model evaluation, experiments were conducted on the model trained with a reduced corpus. The average coherence score of the topics has been calculated per level with the U_{mass} measure and based on their top 20 words. The perplexity of the model has been measured. Statistics of the test sets used to compute the log-likelihood and perplexity can be found in Table 3.3.

Model	Language	Nb documents	Vocabulary	Nb words/doc
HDP	English	2,000	173,055	1,616.40
	French	2,000	113,411	731.97
nHDP	English	200	40,493	1,553.15
	French	200	17,039	451.795

Table 3.3: Statistics of the test sets used with HDP and nHDP.

3.3.2 Hypernyms and embeddings

To evaluate the quality of the hypernyms detected by our Hearst Patterns, the evaluation process used by Roller et al. [15] was used. It consists in computing the score obtained by the model for 3 different hypernymy tasks: Detection, Direction and Graded Entailment.

The **Detection** task evaluates how good the model is at finding the pairs of words which are in a real hypernymy relation. The metric used for this task is the Average Precision. It is calculated on 5 different annotated datasets containing word pairs, which are namely BLESS, EVAL, LEDS, SHWARTZ and WBLESS, as defined in the source paper. The detection of many pairs that are in a true hypernymy relation indicates a large Average Precision of the task, and therefore a high score for this task.

The **Direction** task is intended to evaluate whether the direction of the relations in the evaluated dataset are correct or not, by comparing them to those existing in two annotated test datasets, which are namely WBLESS and BIBLESS. As an example, the direction of the relation “a cat is an animal” should be detected as correct while the inverse pair “an animal is a cat” should not be correct. The third dataset BLESS used by the authors to compute the accuracy on all pairs has not been used in our evaluations as this latter requires to keep inverse relations that we have decided to discard. The larger the accuracy of the pair direction, the larger the score for this task.

In the **Graded Entailment** task, a set of annotated pairs which are graded from 0 to 6 is used to evaluate the weighting of the relations found (i.e. the number of occurrences indicates the level of certainty for a given pair). For this task, the Spearman’s rank correlation (ρ) has been computed and used as the task performance indicator. Here again, a good performing task will obtain a large score.

The following four lists of hypernymy relations were compared by using the above three tasks:

- WIKI (unfiltered): the unmodified complete list extracted by scanning the texts with the Hearst Patterns.
- WIKI (filtered): a variant of the former list where the inverse relations have been removed.
- WIKI (≥ 2): a variant of WIKI (filtered) containing only pairs occurring more than once in the text (inverse relations also removed)
- WIKI (≥ 3): another variant of WIKI (filtered) containing only pairs occurring more than twice in the text (inverse relations have been removed too)

The comparison of the unfiltered list to the 3 variants has been motivated by the observation that the model was extracting a significant number of pairs occurring only a few times in the complete list but characterized by “incoherent” hypernymy relations, which therefore somehow were polluting the quality of the analysis. It was thought that pairs detected many times are more valuable than pairs extracted once or twice from the whole corpus.

Table 3.4 presents the vocabulary size, number of distinct pairs and total number of pairs for each of the four lists. The list of hypernymy relations from the mammals’ sub-tree of the WORDNET dataset [39] is also reported to be used as a baseline to compare the results obtained with the different lists.

	Vocabulary size	Nb of distinct pairs	Total nb of pairs
WIKI (unfiltered)	207,601	241,771	252,039
WIKI (filtered)	144,824	115,185	117,272
WIKI (≥ 2)	4,136	3,293	7,652
WIKI (≥ 3)	1,115	915	3,800
WORDNET MAMMALS	<i>1,152</i>	<i>6,540</i>	<i>6,540</i>

Table 3.4: Vocabulary sizes, number of distinct pairs and total number of pairs of each list of hypernymy relations.

For more information on the tasks and datasets used to evaluate the hypernyms, the reader is advised to refer to the evaluation part of the original paper from Roller et al. [15].

The same evaluation method as used in the work of Nickel and Kiela [13] was then conducted to evaluate the performance of our Poincaré embeddings at reconstructing the extracted taxonomy. In order to associate a score to this reconstruction task, we rank the distance between

each pair of words and the negative samples. The mean rank of the pairs and the Mean Average Precision (MAP) of the ranking have then been measured. For this task, a low mean rank and a high MAP are preferred.

Chapter 4

Results

4.1 HDP

4.1.1 English corpus

Our experiments started with the application of the HDP topic model on the English corpus. Three different executions of the model were carried out, using our three different weighting schemes (ONE, IDF and PMI). The execution time for training the model was of 8 hours in average per weighting scheme. The evolution of the topics' coherence and number of topics are plotted in Figure 4.1 as a function of the number of iterations. It is reminded that the coherence score has been computed on the basis of the 20 most used words for each topic.

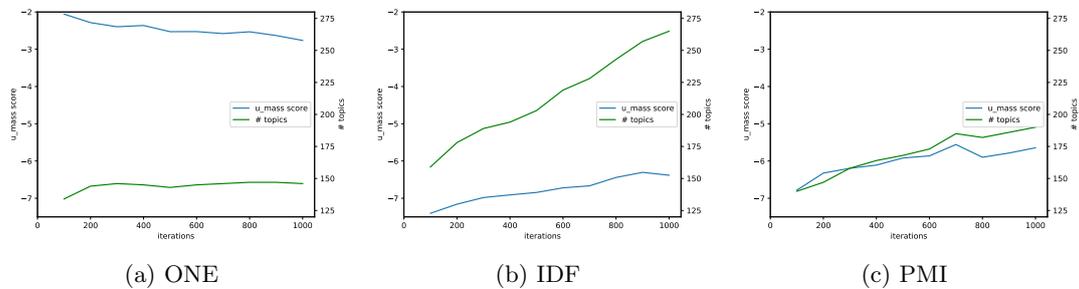


Figure 4.1: Evolution of the coherence score (blue color) and number of topics (green color) over iterations while training on the English corpus.

At first glance, when looking at the weighting scheme ONE, it could be surprising to see that the coherence (blue line) is decreasing as the training progresses. However, it should be noted that the amplitude of this effect is very small and that the absolute value of the coherence score varies only between -2 after 100 iterations and $-2,7$ at the end of the 1,000 training iterations. Those absolute values are in fact depicting a very good coherence of the topics in the entire range

of the training process. Figure 4.1 is also showing that the more the HDP model iterates over the corpus, the more it creates new topics. This increase in number of topics is however pretty limited with the ONE weighting scheme (only 12 additional topics have been created between iteration 100 and 1,000). When the number of topics increases, the top words of the topics may change because they are distributed among the topics. Consequently, the top words of the topics are less frequently used and less likely to co-occur together in the same documents. Such effect could create a slight drop in coherence as visible in the graph of the ONE weighting method, but it may be smoothed out for IDF and PMI which produce much lower coherent topics at the beginning of the training.

Indeed, with IDF and PMI, the average coherence of the topics is very low at the beginning of the training (around -7) and increases with the number of iterations. With those weighting methods, frequent terms are not receiving large weights like it is the case with ONE. IDF gives weight to terms appearing in few documents while PMI gives more weight to terms when they appear very infrequently in a document (compared to their frequency in the whole corpus). In our opinion, this difference in the weighting philosophy makes the IDF and PMI models start with a low coherence score because rarer words are pushed at the top of the topics' words list and are thus influencing negatively the value of the U_{mass} score.

For both PMI and IDF, the number of topics increased between iterations 100 and 1,000, moderately for the PMI weighting scheme (about 50 new topics created) but more significantly for the model using the IDF weighting scheme (more than 100 topics created).

Except in the case of the model using the ONE weighting method, Figure 4.1 is also suggesting that both coherence score and number of generated topics could have developed further if the training had been continued beyond the 1,000 iterations. As a matter of fact, both the curves of coherence and number of topics are showing a steep slope and no plateau during the whole training phase of 1,000 iterations. It can be assumed that coherence would have been further improved with a longer training process and that ultimately the increases in number of topics would have stabilized. A longer training with these two last weighting schemes should therefore allow the model to construct more coherent topics.

In order to determine which topics obtained better coherence scores, the per-topic coherence score of the models after 1,000 iterations have been plotted in Figure 4.2. In parallel, top words of the topics created with the three weighting scheme have been analysed. To illustrate our following analysis, some samples from the topic lists have been attached in Appendix A.

With the weighting scheme ONE, it can be observed that the coherence score is reasonably good for the 100 first topics. Typical values between -2 and -1 are observed. Beyond this threshold limit, the coherence sharply decreases. For example, around topic 120, the coherence is very low and its score is reaching values as low as -13 . We have inspected the top words of topic 120 and found that, although the topic is not easily definable, we can affirm that it is about primates and Africa (and maybe more precisely Congo – see Figure 4.3a). Therefore, even though this topic has the lowest coherence score, it is not completely incoherent in the human

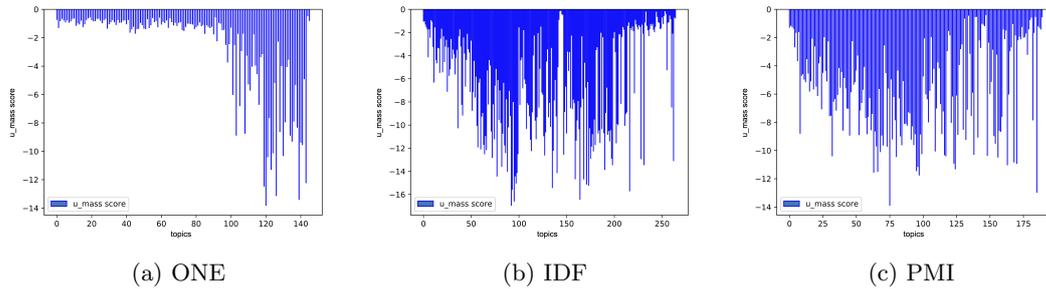


Figure 4.2: Per-topic coherence scores after training on the English corpus with the 3 weighting schemes.

point of view. In contrast, topic 121 shown in Figure 4.3b, which has a coherence score around -10 , is completely incoherent for human beings. On the other hand, topic 19, which has the best coherence score (-0.53), is very hard to interpret and its top 5 words displayed in Figure 4.3c are very confusing. Moreover, it can be observed that 4 of the 5 words are in the table of the 20 most frequent words of the corpus (Table 3.1).

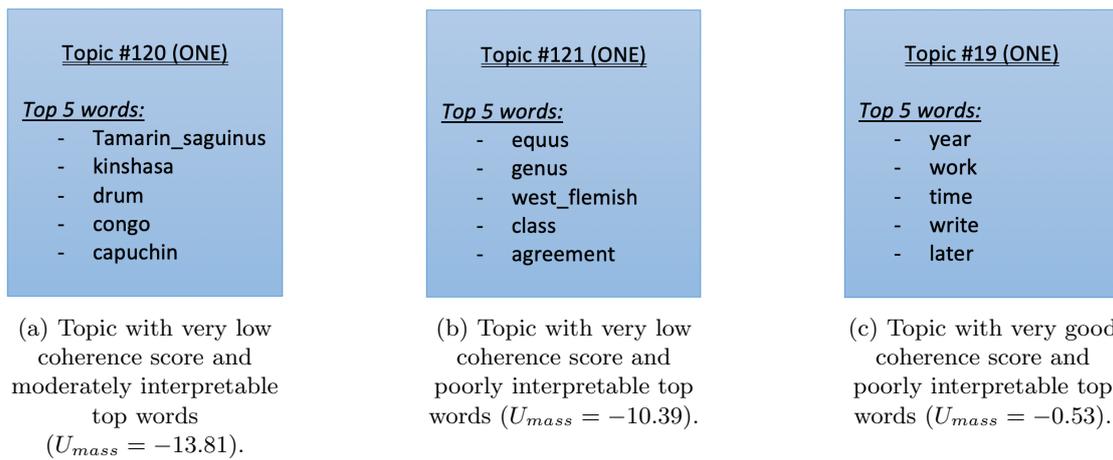


Figure 4.3: Some topics found in the English corpus by the HDP model with the weighting scheme ONE.

With IDF and PMI, the very first topics are really coherent, which is in agreement with their coherence score. After a deep analysis of their top words, the main topics seem to be found within the first 50 topics. However, when approaching the 100th topic, the coherence scores become lower and the interpretability of the topics becomes worse too. After this topic threshold value, the coherence score varies much more from topic to topic but it still seems to be correlated with the human interpretability in most of the cases. Some consistent topics found after topic 100 are very specific. We observe that after topic 200 with IDF, the average coherence score gets higher again but most of the topics are not consistent at all. Actually, these topics are only composed of

infrequent words, which could explain their high U_{mass} score. According to the U_{mass} formula, if the words of a topic appear all together in only one document of the corpus, the score will be very close to 0. With PMI, we also observe an increasing tendency of the coherence score in the last topics and the few of them which are consistent are also very specific. Others are mostly composed of very rare words.

Across the 3 weighting schemes, we found similar topics in the first appearing ones, despite small differences sometimes in their top words. Table 4.1 includes some equivalent topics found with our different weighting schemes, represented by their top 5 words. Top words of topics generated with the ONE weighting scheme are more general and often include words that are very frequent words of the corpus (highlighted in orange color). Among those very frequent words, we found some “intruders” (like *include*, *city* or *example*) which are not really correlated with the topics. It is an observation made on numerous topics from the 100 first ones generated with ONE. Many topics are globally coherent but include intruders in their top words.

Weighting scheme	Topic 1 (IT)	Topic 2 (Mathematics)	Topic 3 (Education)	Topic 4 (Medication)
ONE	system code language number program	number function example theory point	university city college student include	include cause effect drug treatment
IDF	datum user code network system	function theorem algebra equation integer	university student college polk campus	drug patient disease symptom treatment
PMI	system computer datum code software	function theory number point define	education university student college theory	disease treatment patient disorder cause

Table 4.1: Examples of similar topics found in the English corpus with our different weighting schemes.

For the sake of completeness, the coherence score obtained with 20 top words per topic, as computed previously, was compared to a coherence score based on only 10 top words per topic. Although the coherence score obtained with 20 top words is generally a bit lower, no large differences were evidenced. The above observation was expected as the deeper we go down in the list of top words, the lower their use in the topic. It was decided to continue to use the top 20 words of each topic because it takes more words into consideration to evaluate the topic coherence.

4.1.2 French corpus

The same experiments were repeated on the French corpus in order to determine if similar results would be obtained in another language. The average computation time was of 7 hours for training the model (per weighting scheme). Figure 4.4 plots the evolution of the average coherence score of the topics and number of topics while training on the French corpus.

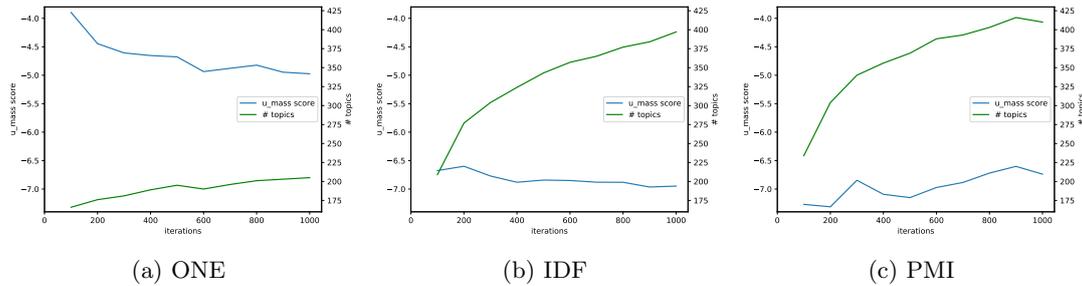


Figure 4.4: Evolution of the coherence score and number of topics over iterations while training on the French corpus with different weighting schemes.

Considering first the ONE weighting scheme, it can be seen that the average coherence score of topics in the French corpus is lower than in English and that more topics have been found (about 205 topics in French versus 150 topics in English). The model presents the same behavior in both languages as the coherence decreases with the increasing number of topics.

When looking at the per-topic coherence scores shown in Figure 4.5, the graphs look very similar to those of the English corpus. The weighting scheme ONE has very good coherence scores for its first 100 topics but beyond that point the coherence score falls down significantly.

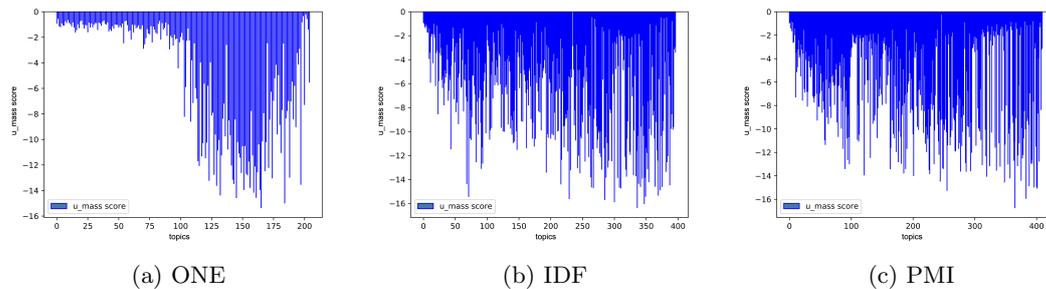


Figure 4.5: Per-topic coherence scores after training on the French corpus with the 3 weighting schemes.

Here again, we manually reviewed the topics found in the French corpus. With the weighting scheme ONE, it was noticed that the most popular topics found by the model were grouping very frequent words but are less semantically consistent than some topics with lower coherence score, as it was the case on the English corpus. For example, in Figure 4.6b, three of the top 5

words (*français*, *américain*, *saint*) are in the 20 most frequent words of the vocabulary (see Table 3.1) and the two remaining ones (*homme_politique*, *acteur_américain*) are not really correlated with others. Obviously, very coherent topics were found as well where even the most frequent words make sense. An example is depicted in Figure 4.6a, where the topic could be labelled as “Software” (i.e. “Logiciel informatique” in French) and whose top 5 words include 2 from the 20 most frequent words of the corpus (*utiliser*, *fichier*). topics with ID greater than 100 (i.e. topics with lower coherence score) have also been examined and plenty of inconsistent topics were found, like topic 184 which makes no sense at all (see Figure 4.6c). It is another behaviour that has already been spotted on the English corpus. On the other hand, it was pointed out that among the topics with ID greater than 100, some topics with higher coherence scores like topic 137 (Figure 4.6d) were consistent and very specific.

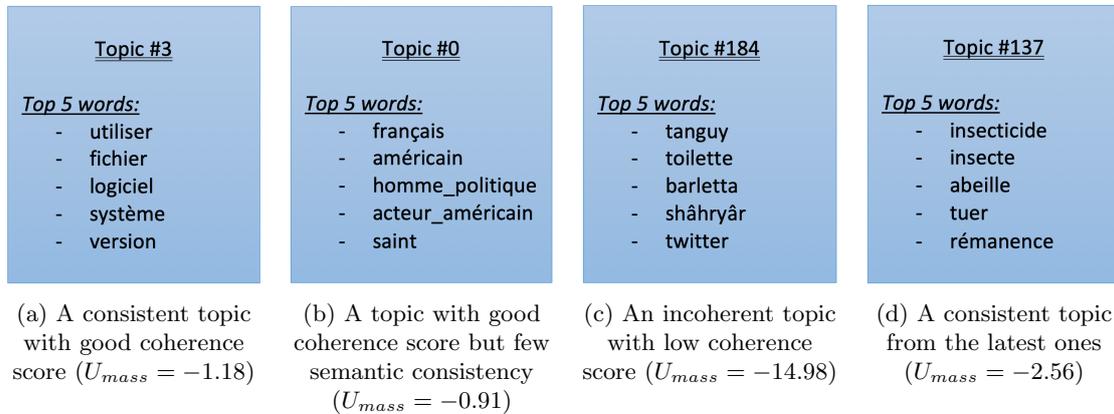


Figure 4.6: Several topics found in the French corpus with the weighting scheme ONE.

Looking back at Figure 4.4, it can be noticed that the IDF weighting method is producing a different model behavior in the French corpus in comparison to the English corpus analysed before. As a matter of fact, the coherence score decreases along the training whilst it was increasing on the English corpus. It starts however from a better coherence score than in the English corpus in the first iterations. The PMI weighting method, on the other hand, induces the same model behavior across the two languages, as the coherence score is increasing as a function of the number of iterations, together with the number of topics that are generated. For both IDF and PMI methods, the models are producing much more topics in French than in English, over the same number of iterations (this was also observed for the model using the ONE weighting method). The per-topic coherence graphs of Figure 4.5 reveals that both IDF and PMI weighting methods lead to lower coherence scores in average, even if it can be noted that the coherence score seems to be a little better in the first topics.

Although IDF and PMI weighting methods lead to a lower average coherence score, many consistent topics have been found, even beyond the 100 first topics. We found also more incoherent topics but this is expected as the number of identified topics is almost twice as large as

the number obtained with the ONE weighting method. Among the consistent topics, there are very specific ones, like the one depicted in Figure 4.7a, which can be associated to cryptography. In this particular case, no equivalent topic was found with the weighting scheme ONE.

Also, there are many topics which could be grouped into a single one (e.g. in Figure 4.7, topic 102 and 125 both group tokens corresponding to German states).

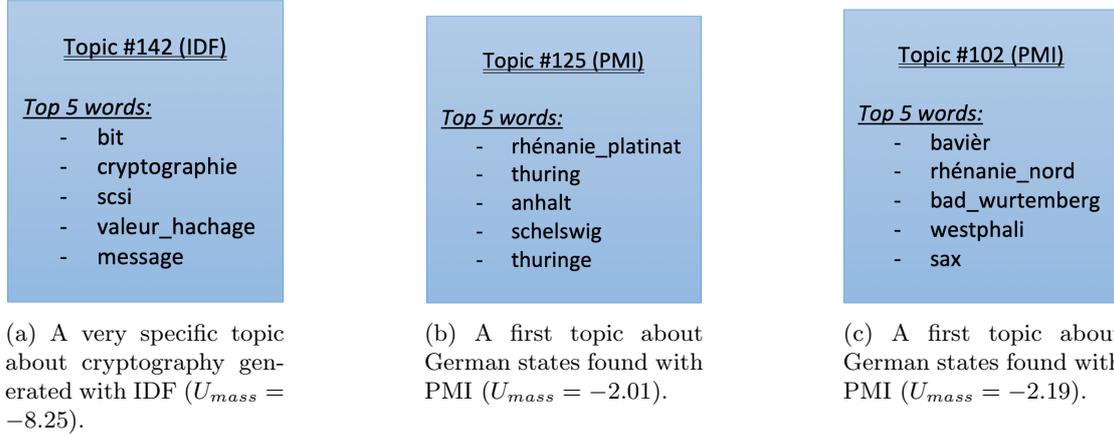


Figure 4.7: Several topics found in the French corpus with the weighting schemes IDF and PMI.

4.1.3 Log-likelihood

In order to compare with the coherence measure of the topics found in both the French and English corpora, computation of the average log-likelihood per word was performed on each whole corpus. The average log-likelihood per-word has also been calculated on a held-out test set of 2,000 documents for both languages. For these experiments, as large absolute values of the log-likelihood were obtained, the determination of the log-likelihood has been preferred over the perplexity. Indeed, as the perplexity is computed from the exponential of the negative log-likelihood, its computation would have generated huge values and very big differences from experiment to experiment. Figure 4.8 illustrates the comparison between the log-likelihood per word on the corpus and the log-likelihood per word on the test set after training our model on both corpora with our 3 weighting schemes.

The bar plots look very similar between the two corpora, although slight differences can be noted (as the log-likelihood is better when it approaches 0, it is considered to be lower when its absolute value is larger, and vice-versa). First of all, it seems that whatever the weighting schemes, the log-likelihood is always slightly better (lower value) on the English corpus. Then, it is evident that the weighting scheme ONE has better log-likelihoods than the other schemes, in both corpora. Moreover, with the model based on the ONE scheme, the log-likelihood on the test set appears to be slightly better (lower absolute value) than the log-likelihood computed on the corpus itself. The other weighting schemes present significantly worse test log-likelihoods (higher

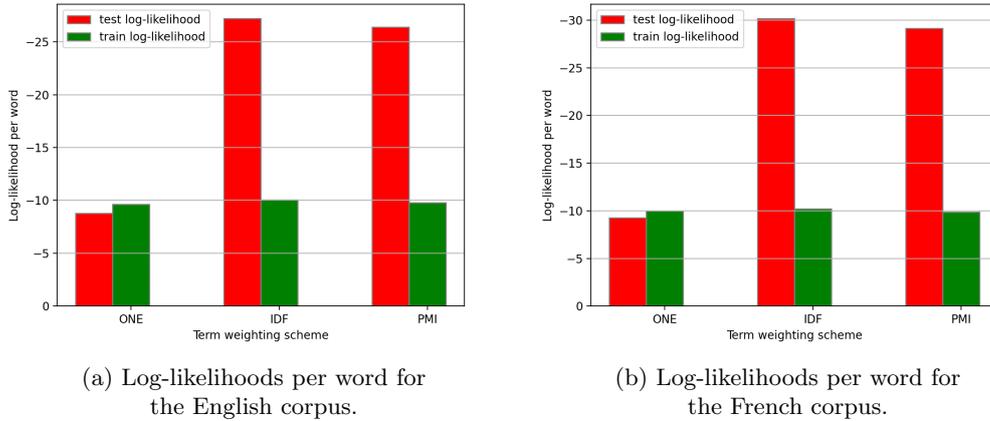


Figure 4.8: Comparison between the log-likelihood per word on the corpus and the log-likelihood per word on the test set after training on our 2 corpora.

absolute values) than those determined on the whole corpora. Finally, although characterized by pretty high values, the PMI scheme seems to provide slightly better log-likelihood than the IDF weighting scheme, on the test samples in both the French and English corpora.

Finally, comparing these values to the average coherence scores of the topics, we observe that the trend is the same for both evaluation methods. With the weighting scheme ONE, a high average U_{mass} score is observed as well as good log-likelihood of the model (compared to IDF and PMI). For IDF and PMI, the average coherence score is lower and we can see that, with these weighting schemes, the log-likelihood of the model is also lower. When using PMI, the model has a slightly better log-likelihood than with IDF and the same observation can be done for the average coherence score of the topics. When comparing both corpora together, it has been observed that the HDP model produces more coherent topic in general on the English corpus than on the French corpus, according to the coherence score. The log-likelihood is also better on the English corpus than on the French one, whatever the weighting scheme.

4.1.4 Discussions

The results obtained with the HDP model were not as expected. Actually, a coherence score increasing along the training iterations and ultimately approaching 0 at the end of the training process was expected, regardless of the weighting scheme. However, this expected behaviour was not observed for 3 of our 6 experiments. This is likely due to the generation of new topics during the training process. In view of the number of incoherent topics generated, it is assumed that a better coherence score would have been obtained with less topics by fine-tuning the dispersion parameters. However, the goal of this work was not to find the best model fitting the data but to compare and evaluate models among different corpora and languages with several weighting schemes.

It was also noticed that the topics' coherence score was strongly related to the frequency of occurrence of their top words in the whole corpus. Topics with top words that were very frequent words in the corpus itself obtained good coherence scores. On the other hand, topics grouping rare words usually got very low coherence scores, except when those words appear together in the same documents, which gives a better U_{mass} score. It should be noted that both those high and low coherence score topics were not necessarily easily interpretable from a human analysis. For these reasons, it is suggested that the HDP model would be performing better on corpora where the most frequent and rarest words have been removed in a pre-processing step.

The above suggestion is reinforced by the observation that there is a better match between the coherence measured by the U_{mass} score and the human interpretation of coherence applied to topics which are composed of words that are neither very frequent nor very rare.

We noted that the number of topics found is always larger in the French corpus. Again, these results are surprising because the French corpus has a smaller vocabulary than the English corpus. No evidence was found on the reason of this observation but, in our opinion, it is simply a characteristic of the French language.

The French topics generally have a lower coherence score compared to the English ones, what we believe to be due to the gap between their vocabulary size and their number of words per document. The coherence score of the French topics could also be affected by the larger number of topic produced.

When the weighting schemes are compared, it can be seen that ONE is producing topics with a higher coherence score but being very general and not specific. Contrariwise, PMI and IDF produce more topics, including very coherent and specific ones, but also a lot of inconsistent ones.

Finally, a correlation was found between the average log-likelihood per word and the average U_{mass} score of the topics.

As a preliminary conclusion, the HDP model seems to perform well in both languages, with regard to the amount of coherent and humanly interpretable topics it produces. However, it could perform even better with fine-tuned parameters.

4.2 nHDP

As described in the Methodology section, the French and English corpora used for the experiments with the nHDP model are ten times smaller than the ones used with the HDP model. However, they contain a sufficient number of documents (i.e. 1,000 each) to enable the model to develop a tree-structured representation of their inner topic hierarchy.

As the model makes use of a variational inference method, the number of topics do not vary like it is the case with Gibbs sampling. The number of topics (i.e. the number of nodes of the tree) that is used during training is determined by the K-means initialization procedure which makes a first distribution of the words across the tree to cluster related word together. The model

initialized the trees with 1184 topics for the English corpus and 1204 for the French corpus. The model took approximately 3 hours to train on each corpus.

4.2.1 English corpus

Figure 4.9 presents the evolution of the per-level coherence of the topics found by the model in the English corpus at the 3 levels of the topic hierarchy (which are namely *General*, *Specific* and *Specialized* levels).

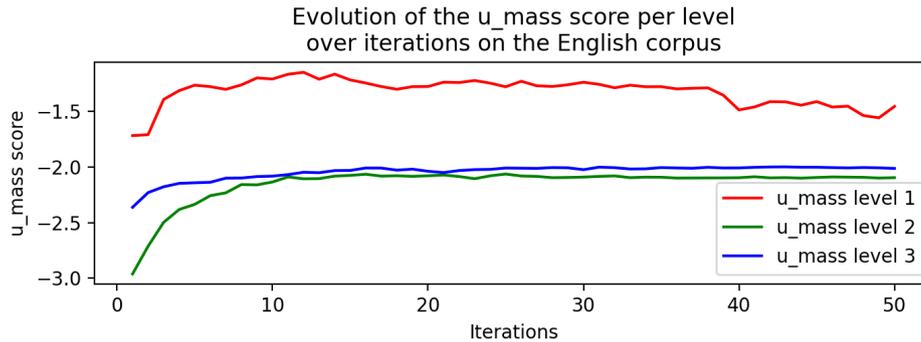


Figure 4.9: Evolution of the mean per-level U_{mass} score of the topics over 50 iterations on the English corpus.

For each level, the coherence quickly increases at the beginning of the training and stabilizes at a plateau after approximately 10 to 15 iterations. Level 1 is characterized by the larger coherence score and appears to contain the most coherent topics. This observation could be explained by the “general” nature of the topics at level 1. In fact, the most popular words in topics of level 1 are very common words of the corpus (many of the top words of level 1 topics are present in Table 3.1), which have greater probability to co-occur in the same documents. This would explain why they achieve a better U_{mass} score than topics from other levels. Level 2 and level 3 topics are characterized by coherence scores that are very close to each others at their plateau value, with a slight advantage to the level 3 topics, which is not really surprising since level 3 topics are supposed to be specializations of level 2 topics.

The manual analysis of the topics obtained by the nHDP model after the 50 iterations on the English corpus evidenced some very interesting topic hierarchies that are reported in Figure 4.10. At the first level of the hierarchy, topic 3 has been identified as a general topic dealing with “Sciences”. When only considering this level 1 topic, the very general nature of its top words would make its identification and labelling pretty challenging as few of those words are really about sciences. However, when looking at its child nodes at level 2, several branches of sciences like biology, mathematics, physics, etc. can be immediately recognized. In the same way, at level 3, the children of the “Biology” topic (topic 53) are clearly showing branches of biology like

micro-organisms, animal species, plant species, etc.

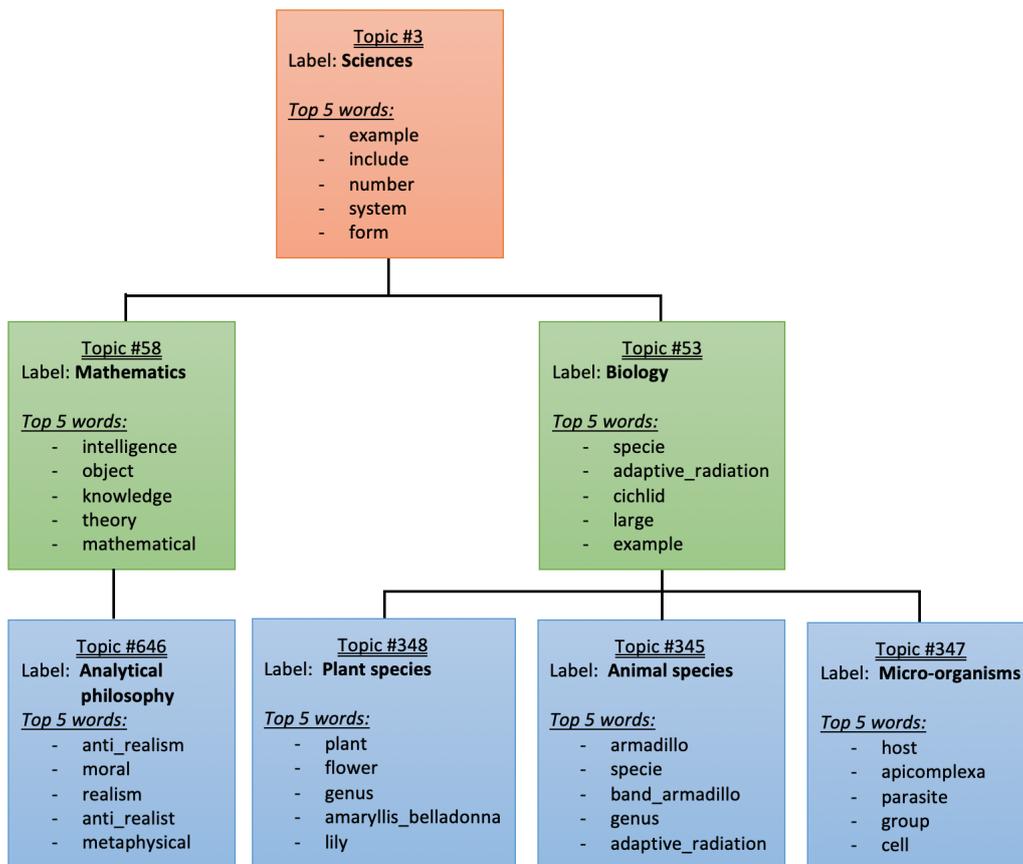


Figure 4.10: Example of topic hierarchies found in the English corpus.

As shown in Figure 4.11, some incoherent topics or incoherent hierarchical relations (i.e. coherent topics which have no real correlation with their parent) were also identified.

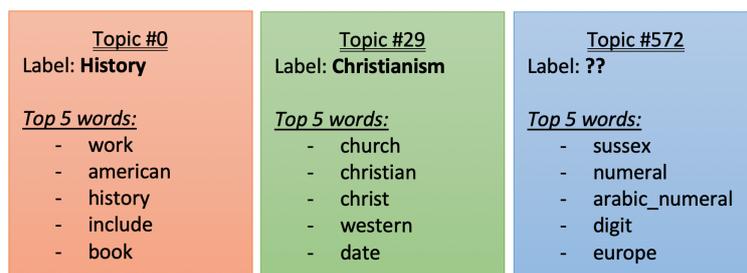


Figure 4.11: Example of an inconsistent topic hierarchy found in the English corpus.

As an example, topic 0 (orange box) is a general level 1 topic that can be related to “History”. Its first level 2 specification (topic 29, green box) seems to be consistent with “Christianism”

as a sub-category of “History”. On the contrary, topic 572, which is supposed to be a level 3 specialized sub-topic of “Christianism”, can hardly be assigned any correlation to its parent and is hard to interpret.

More topic hierarchies found by the nHDP models are available in Appendix B.

4.2.2 French corpus

In the French corpus, trends similar to the English corpus have been observed, with even higher coherence scores observed in the French corpus, as depicted in Figure 4.12. Level 1 is again presenting the best average coherence score and all 3 levels’ average coherence scores stabilize at a plateau after about 10 to 15 iterations. Interestingly, it was noted that the average coherence score of levels 1 and 3 suddenly increased between iterations 26 and 27 while level 2 dropped at the same time. This sudden change in coherence scores is suggesting a rearrangement of the hierarchical tree after iteration 26. In order to verify our assumption, the structure of the tree was analysed around iterations 26 and 27. It was found that 5 topics from level 2 actually switched their positions with 5 topics from level 1 between iteration 26 and iteration 27. This sudden variation in the coherence scores is effectively corresponding to a tree rearrangement, leading to an overall better coherence at level 1 and a situation very similar to the English corpus where level 2 and 3 coherence scores were found almost identical at the end of the training.

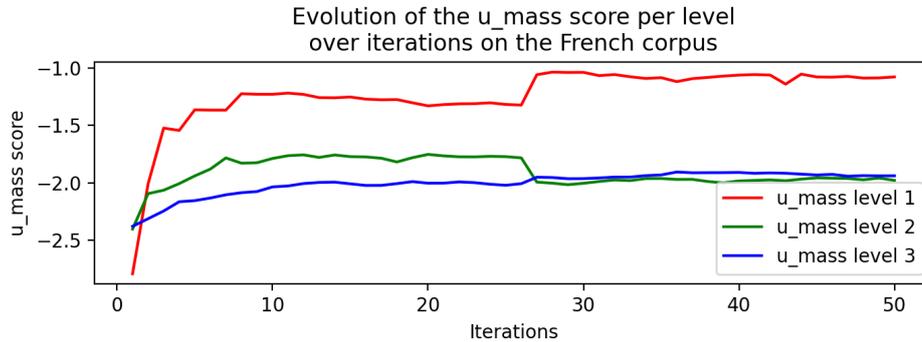


Figure 4.12: Evolution of the mean per-level U_{mass} score of the topics over 50 iterations on the French corpus.

In the hierarchy of topics found in the French corpus, a topic about “Olympic games” was spotted at the first level (topic 11 in Figure 4.13), which has a sub-topic grouping “Team sports” at level 2, this specific sub-topic being also divided into specialized ones like “Handball” and “Basketball”. Those hierarchies are very interesting because they respect the notions of specification and specialization. The hierarchy of topic 7 is another interesting example shown in Figure 4.13.

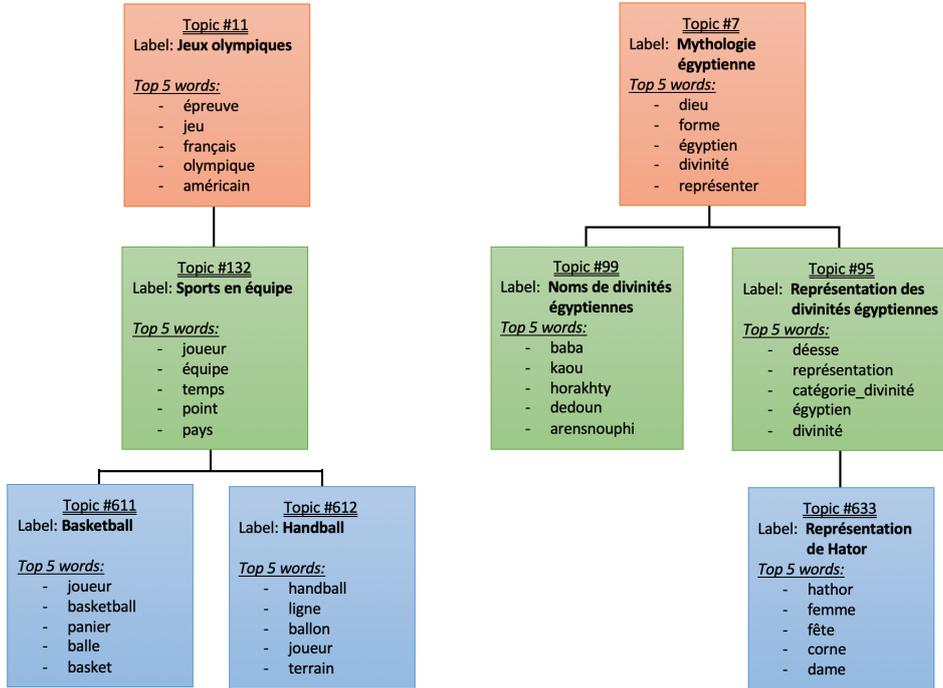


Figure 4.13: Example of topic hierarchies found in the French corpus.

As for the English corpus, other topic hierarchies found in the French corpus by the nHDP model are available in Appendix B.

4.2.3 Perplexity

In order to determine if the perplexity of the nHDP model and the coherence scores of its topics are correlated, in comparison with the HDP model, the evolution of the training perplexity per word of the nHDP model was monitored as a function of the number of iterations. The evolution of the perplexity on both English and French corpora, depicted in Figures 4.14 and 4.15 respectively, demonstrates that the perplexity of the model does not decrease as the coherence gets better, but even tends to increase slightly over iterations. The values obtained on the French and the English corpora are very similar. However, the evaluation of the model after training, performed on a held-out test set of 200 documents provided a mean perplexity per word of 13,601.02 for the English corpus and 20,347.23 for the French corpus.

4.2.4 Discussions

When testing the nHDP model, the coherence score of topics was also related to the frequency of its top words. In average, the topics of level 1 had higher U_{mass} scores than those of lower levels. During the K-means initialization step, a greater number of topics was found in the French

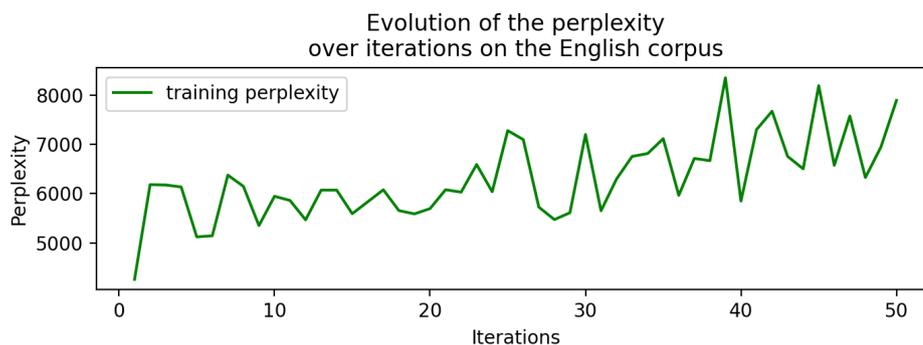


Figure 4.14: Perplexity of the model over 50 iterations on the English corpus.

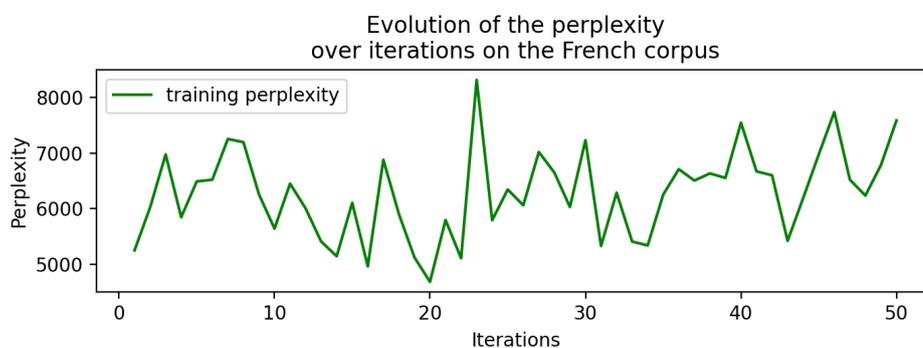


Figure 4.15: Perplexity of the model over 50 iterations on the French corpus.

corpus than in the English one as it was observed with HDP.

The global coherence scores are relatively similar in both languages for topics found by the nHDP model, although the French scores appeared to be slightly better. This is different from what has been observed for the HDP model, where the English scores were better. It is supposed that this difference between HDP and nHDP is due to the reduced size and new statistics of nHDP's corpora. The size of the vocabulary and the number of words per document are much more balanced between English and French in the reduced corpora used by the nHDP model, resulting in very close coherence scores.

The effect of size reduction on the statistics of both corpora and their respective test sets can also explain the similar observations made on the perplexity measures. While the training perplexity of nHDP is very similar in French and in English, the testing perplexity is higher on the French test set. Looking back at the statistics of the test sets presented in Table 3.3, the greater perplexity on the French test set can be explained by a less rich vocabulary and the

smaller size of the documents.

It should be noted that the parameters of the nHDP model have not been fine-tuned to best fit the data of our corpora, most of the parameters were left unchanged from their original setting. Despite no optimization, the nHDP model still delivered very valuable results and found very interesting topic hierarchies in both corpora.

4.3 Poincaré embeddings

Besides the topic modeling experiences, some attempts were made to model the latent hierarchy relations between words of the 10,000 articles of the English corpus by means of the Poincaré embeddings as described in the Methodology section.

4.3.1 Hypernyms evaluation

After extraction of the hypernyms from the corpus with the Hearst Patterns prepared for that purpose, the pairs of words were examined and it was found that a lot of pair relations seemed incoherent, mostly in pairs detected only once. As explained in the Evaluation section, going around this situation was possible by evaluating the list itself, as well as a set of modified versions of this list, where only the most valuable pairs were considered for analysis. Our results have been reported in Table 4.2.

	Detection (AP)					Direction (Acc.)		Graded (ρ)
	BLESS	EVAL	LEDS	SHWARTZ	WBLESS	WBLESS	BIBLESS	HYPERLEX
WIKI (unfiltered)	.17	.23	.54	.26	.54	.60	.56	.24
WIKI (filtered)	.10	.22	.49	.25	.42	.53	.53	.10
WIKI (≥ 2)	.10	.22	.50	.25	.43	.65	.64	.06
WIKI (≥ 3)	.10	.22	.49	.25	.42	.64	.71	.20
WORDNET MAMMALS	.12	.23	.50	.25	.44	.66	.77	/

Table 4.2: Results of the different hypernymy tasks performed on multiple versions of the set of word pairs detected with our Hearst Patterns. Last line is the score obtained with the mammals’ sub-tree of the WORDNET dataset.

It can be observed that WIKI (unfiltered) obtains the highest score for the Detection task, whatever the dataset under consideration. This observation is expected, as the wider vocabulary of the unfiltered list enables the detection of more relations. However, as explained in the Methodology section, the overview of the list has revealed that, among these relations, a large number is incorrect and many of them are inverse relations. WIKI (unfiltered) also shows the best score on the Graded Entailment task which can also be explained by the larger number of relations favoring entailment.

When looking at the filtered lists, WIKI (filtered), WIKI (≥ 2) and WIKI (≥ 3) all present similar but much weaker scores in the Detection task, which is again expected as those lists were filtered and numerous pairs were removed in the filtering process. The Detection task is therefore given less focus in the further analysis of the filtered lists below.

WIKI (≥ 2) and WIKI (≥ 3) are showing the highest scores in the Direction task, for both of the datasets under consideration. By averaging the accuracy scores obtained on both datasets, WIKI (≥ 3) is however slightly outperforming WIKI (≥ 2). WIKI (≥ 3) also achieves a relatively good score (second best score) on the Graded Entailment task.

Finally, when comparing the scores for the various tasks of WIKI (≥ 3) with the baseline scores computed for the mammals sub-tree of WORDNET’s taxonomy, it can be noted that, although WIKI (≥ 3) achieves slightly lower results than WORDNET MAMMALS, it is really not far off. Note that the Graded Entailment task has not been performed with WORDNET’s hypernyms because all pairs occur exactly once.

WIKI (≥ 3) appeared to be the best list for representing consistent hypernymy relations of the corpus. However, in the following experiments, the 3 filtered lists have been used to build Poincaré embeddings in order to compare their consequences on the performance of the model for the reconstruction task described in the Evaluation section.

4.3.2 Embeddings evaluation

With a demonstration purpose, our first experiment consisted in learning simple 2D vectors in order to visualize their relations on the Poincaré ball representation. Figure 4.16 is showing the word embeddings learnt after 100, 2,000 and 5,000 iterations. When looking at the evolution of the word’s position during training, it can be observed that the majority of word embeddings is moving away from the center of the image to the edges of the ball when training is progressing. This is explained by the fact that the training process is defining the position of words in the multi-dimensional space to ensure they are close to their parents and siblings, but far from words they are not linked to. As a reminder, the disks represented in Figure 4.16 are only projections of the Poincaré ball in a 2-dimensional space. In the real manifold, the Poincaré ball is multi-dimensional and there is much more room between the words present near the edges of the disks. In the Poincaré representation, words that are close to the center of the image are higher in the hierarchy.

In our second experiment, 10-dimensional embeddings have been trained and their performance on the reconstruction task was evaluated. The model has been trained on 300 epochs, a batch size of 10 was used, a constant learning rate of 0.3 was set and 50 negative samples were used in the loss function. The word embeddings learning process was applied on our 3 filtered lists of hypernyms (WIKI (filtered), WIKI (≥ 2) and WIKI (≥ 3)). The results obtained on the reconstruction task are reported in Table 4.3. The time required for training the model was 13 hours with WIKI (filtered), about 30 minutes with WIKI (≥ 2) and 5 minutes with WIKI (≥ 3).

As observed in Table 4.3, WIKI (≥ 3) seems to be the list of hypernyms providing the most

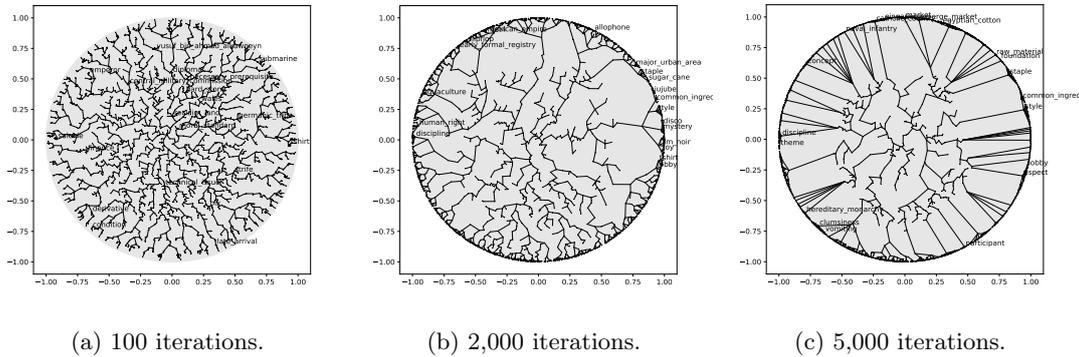


Figure 4.16: Evolution of word embeddings positions in the Poincaré ball during training after 100, 2000 and 5000 iterations on the corpus.

	Mean Rank	MAP
WIKI (filtered)	18.380	.807
WIKI (≥ 2)	1.605	.873
WIKI (≥ 3)	1.105	.950
WORDNET MAMMALS	1.126	.978

Table 4.3: Results of the reconstruction task performed by Poincaré embeddings learnt with multiple versions of the set of word pairs detected with our Hearst Patterns.

interesting results. It presents the lowest Mean Rank and highest MAP scores, and those are really close to the baseline values obtained with the WORDNET MAMMALS. However, it must be kept in mind that WIKI (≥ 3) contains much less pairs and has a smaller vocabulary than WIKI (filtered) and WIKI (≥ 2).

Finally, a qualitative analysis has been carried out on the embeddings of size 10 which performed best at the reconstruction task, namely those learnt with WIKI (≥ 3). In order to visualize the embeddings of size 10, they were transformed by applying a well known dimensionality reduction technique called T-Distributed Stochastic Neighbor Embedding (*T-SNE*) [40]. *T-SNE* aims to find the most appropriate representation for high dimensional data in a space of reduced dimension (usually two or three dimensions) by maintaining the proximity between the points. The *T-SNE* model of the Scikit-Learn¹³ library was used to transform our embeddings. The model was trained for 2,500 epochs and the point cloud depicted in Figure 4.17 was obtained. Each point represents a word, the transparency of the points was set to 0.5 in order to enable more colorful regions when many points are overlapping. Each word was also associated to its Euclidean norm (squared) in the Poincaré model, which determines the color of its point in the graph. This is enabling an easier visualization of the words that are higher in the hierarchy.

¹³<https://scikit-learn.org/>

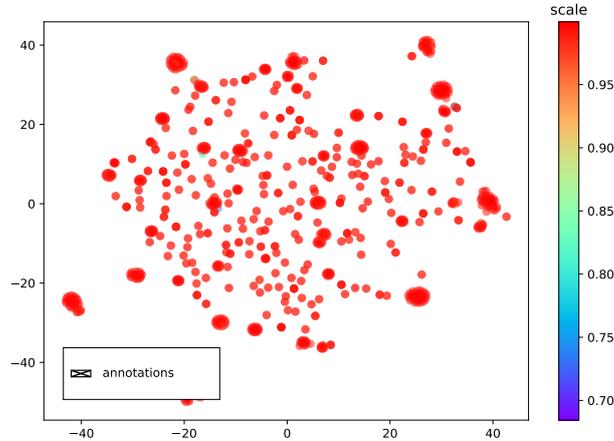


Figure 4.17: Point cloud of the 10-dimensional embeddings of WIKI (≥ 3) reduced to two dimensions with *T-SNE*.

What can be observed in the first place is that most of the words have a very high norm (i.e. the majority of the points in the cloud are red). This indicates that very few words are at the top of the hierarchy. The second most obvious observation is that some points have been grouped together, as revealed by the very distinct clusters appearing in the cloud of points.

A zoom was performed in some of these clusters and their points were annotated with their name and norm. It was clearly observed that those groups are composed of very correlated words, which could be associated to topics. In some cases, words related to a cluster were found farther than the other words. In fact, these words are usually shared with other topics so they are positioned between groups of words related to those different topics.

As exemplified in Figure 4.18, some clusters are very well constructed.

It can be seen that the word “company”, characterized by a lower norm, is positioned in the center of the cluster and almost all other words are scattered around it, forming an oval shape. This was interpreted by looking in the WIKI (≥ 3) list where we found that nearly all the words on the oval shape are hyponyms of “company”. Actually, only the word “search_engine” is not an hyponym of “company”, but it is located between “Google” and “Yahoo” which are hyponyms of “search_engine”.

Many other topics found in the graph are shown in Figure D.1 in Appendix D.

4.3.3 Discussions

The model delivered very good results when trained on the reconstruction task with the list of pairs occurring at least three times in the corpus (i.e. WIKI (≥ 3)). In that case, the hierarchy

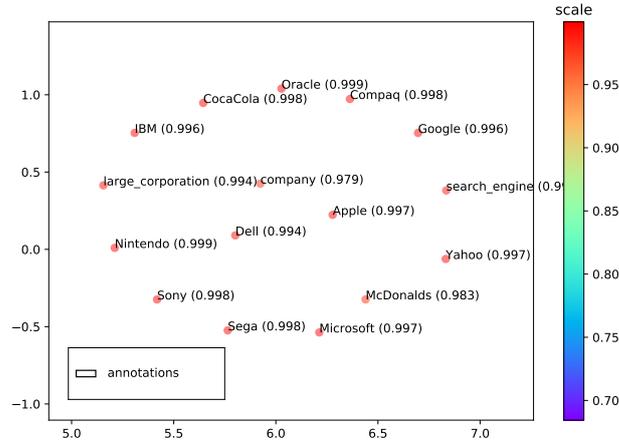


Figure 4.18: A cluster of the point cloud in which company names are grouped around the word “company”.

of the words have been well captured. However, when the model was trained on less filtered relations, the Mean Rank significantly increased and the MAP rank dropped strongly, evidencing a deterioration of the model performances. This is suggesting that the Poincaré embeddings model has more difficulty in representing the hierarchy with larger or noisier datasets.

During our qualitative analysis, it was found, with the help of dimensionality reduction techniques, that Poincaré embeddings are good at clustering related words together. Several clusters of words identified by the Poincaré embeddings were relatively close and similar to topics found by our other topic models. Moreover, it appeared that the squared norm of a word can be a good indicator of words’ position in the hierarchy. The hypernyms with the lowest norms of the cluster appeared to be a good description of the cluster subject and may often be used to label the cluster, as it was the case for the cluster of *company* in our example.

Note that the norm is not actually a distance provided directly within the embeddings vector, but can be computed separately from this vector.

It should be noted as well that, although the representation in the graph generated by *T-SNE* is providing evidence of some local hierarchy relations in clusters of words, there is hardly any information about the existence of potential hierarchy between different clusters or outside of the clusters.

Chapter 5

Conclusion

The performances of the HDP and nHDP topic models have been evaluated to determine their ability to extract coherent topics from English and French corpora, built from the 10,000 first Wikipedia articles in both languages. For both models, the coherence of the topics generated has been measured with the U_{mass} score. The log-likelihood and perplexity have also been used as a metric of the model performances.

Then, in order to determine if the Poincaré embeddings could be used as potential tools to model topic hierarchies, pairs of hypernyms have been extracted from the English corpus and used to learn Poincaré embeddings. The performance of the embeddings on the reconstruction task was evaluated and their ability to capture and represent a latent hierarchy of topics was analysed manually.

For the HDP model, 3 different word weighting schemes were tested, namely ONE, IDF and PMI. The HDP model revealed to be effective in detecting very specific topics among the large number of topics found, when using the IDF and PMI weighting schemes. With the ONE weighting scheme, it found less topics and the latter were more general, despite a better average U_{mass} coherence score. This U_{mass} score has shown to be strongly correlated with the frequency of the topics' top words in the corpus. Additionally, a correlation between the average U_{mass} score of the topics and the log-likelihood measure of the model has been put in evidence.

The nHDP model was found very promising for hierarchical topic modeling. It has delivered good performances by providing qualitative topic hierarchies, in both English and French languages. The training perplexity of the model has been shown to be correlated with the average U_{mass} score of the topics.

However, in its current implementation, nHDP is not adapted for common use. As explained in the Methodology section, the code is very difficult to understand and to use. Moreover, the algorithm is unusable with a desktop machine when the amount of data exceeds a thousand of

documents.

As a conclusion, the evaluation metrics used on both corpora were suggesting good performances with an advantage for the English corpus. However, the comparison of the results of both languages is biased by intrinsic differences existing in both corpora. These differences in the two corpora can be explained by the generally lower amount of resources in languages other than English (mainly shorter articles in French and less developed pre-processing tools).

Additionally, we have not been able to compare our observations to similar works in the literature because of the lack of studies comparing the performance of topic models across different languages in general and on topic modeling in French in particular. It is however worth noting that, despite a smaller vocabulary and less words per document in average, our French corpus appeared to contain more topics than the English one, whatever the model used, what we think is a characteristic of the French language.

Poincaré embeddings have demonstrated a good ability to capture the semantics and the hierarchical relations of the words they are trained on. However, their use is not really appropriate for extracting a topic hierarchy.

As a matter of fact, it has been clearly shown that the quality of the Poincaré embeddings is highly dependant on the quality of the dataset used to learn representative word vectors, and more particularly on the sanity of the hypernymy relations. The model used for learning Poincaré embeddings is limited by the data acquisition method used to extract the hypernymy relations. This method requires the raw text to be clear and free of typos and the used patterns to be well-constructed with a view to keep only pairs of nouns or nominal groups linked by hypernymy relations. Moreover, additional filtering steps are required to leverage the quality of the hypernyms.

The net result of those limitations is a reduction of the size of the dataset, which becomes much smaller than the size of the original corpus. In our case, the vocabulary used for this task was reduced by a factor 450, in comparison to the one used to train the HDP model. As a direct consequence, the topics modeled by the clustering of the Poincaré embeddings are fewer in number and less rich than those found by HDP.

We have also shown that 2-dimensional representation techniques like *T-SNE* are not well suited for hierarchical clustering.

For future works, some areas of potential improvement have been identified.

First, the parameters of the HDP model could be fine-tuned to better fit the dataset it is trained on. With a larger amount of iterations, the coherence of the topics could also be improved.

Second, the code of the nHDP model could be optimized to be more easily adopted by the general public. Ideas of improvement would mainly focus on the memory usage in order to enable

management of larger datasets. For example, dealing with large and dense matrices should be avoided. Also, some tasks could be processed in parallel on multicore processors to reduce the computation time. Moreover, we would suggest the transcription of the code to a more open language than Matlab which requires costly licenses.

Finally, it would be interesting in our opinion to try training a Bayesian topic model using the Poincaré embeddings instead of the tokens as done by Dieng et al. and Batmanghelich et al. for LDA and HDP with Euclidean embeddings [41, 42]. In the same way, nHDP could also be modified to make use of word embeddings and the results obtained with different types of word embeddings, including Poincaré embeddings, could be compared.

Bibliography

- [1] Y. Teh, M. Jordan, M. Beal, and D. Blei, “Hierarchical dirichlet processes,” *Journal of the American Statistical Association*, vol. 101, pp. 1566–1581, 01 2006.
- [2] J. Paisley, C. Wang, D. M. Blei, and M. I. Jordan, “Nested hierarchical dirichlet processes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, p. 256–270, Feb 2015.
- [3] D. Blei, A. Ng, and M. Jordan, “Latent dirichlet allocation,” vol. 3, pp. 601–608, 01 2001.
- [4] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, “The author-topic model for authors and documents,” 2012.
- [5] D. Blei and M. Jordan, “Modeling annotated data,” *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*, 09 2003.
- [6] Y. W. Teh, *Dirichlet Process*, pp. 280–287. Boston, MA: Springer US, 2010.
- [7] D. Blei, T. Griffiths, M. Jordan, and J. Tenenbaum, “Hierarchical topic models and the nested chinese restaurant process,” *Advances in Neural Information Processing Systems*, vol. 16, 05 2004.
- [8] C. Wang, J. Paisley, and D. Blei, “Online variational inference for the hierarchical dirichlet process,” in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (G. Gordon, D. Dunson, and M. Dudík, eds.), vol. 15 of *Proceedings of Machine Learning Research*, (Fort Lauderdale, FL, USA), pp. 752–760, PMLR, 11–13 Apr 2011.
- [9] M. Hoffman, D. M. Blei, C. Wang, and J. Paisley, “Stochastic variational inference,” 2013.
- [10] J. Liu, “The collapsed gibbs sampler in bayesian computations with applications to a gene regulation problem,” *Journal of The American Statistical Association - J AMER STATIST ASSN*, vol. 89, pp. 958–966, 09 1994.
- [11] S. Sia, A. Dalmia, and S. J. Mielke, “Tired of topic models? clusters of pretrained word embeddings make for fast and good topics too!,” 2020.

- [12] A. Onan, “Two-stage topic extraction model for bibliometric data analysis based on word embeddings and clustering,” *IEEE Access*, vol. 7, pp. 145614–145633, 2019.
- [13] M. Nickel and D. Kiela, “Poincaré embeddings for learning hierarchical representations,” 2017.
- [14] M. Nickel and D. Kiela, “Learning continuous hierarchies in the lorentz model of hyperbolic geometry,” 2018.
- [15] S. Roller, D. Kiela, and M. Nickel, “Hearst patterns revisited: Automatic hypernym detection from large text corpora,” 2018.
- [16] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A neural probabilistic language model,” *JOURNAL OF MACHINE LEARNING RESEARCH*, vol. 3, pp. 1137–1155, 2003.
- [17] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” 2013.
- [18] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” vol. 14, pp. 1532–1543, 01 2014.
- [19] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching word vectors with subword information,” 2017.
- [20] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, “Bag of tricks for efficient text classification,” 2016.
- [21] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” 2018.
- [22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2019.
- [23] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2017.
- [25] M. Gromov, *Hyperbolic Groups*. New York, NY: Springer New York, 1987.
- [26] S. Bonnabel, “Stochastic gradient descent on riemannian manifolds,” *IEEE Transactions on Automatic Control*, vol. 58, p. 2217–2229, Sep 2013.
- [27] O.-E. Ganea, G. Bécigneul, and T. Hofmann, “Hyperbolic entailment cones for learning hierarchical embeddings,” 2018.

- [28] J. Sethuraman, “A constructive definition of the dirichlet prior,” *Statistica Sinica*, vol. 4, pp. 639–650, 01 1994.
- [29] D. J. Aldous, “Exchangeability and related topics,” in *École d’été de probabilités de Saint-Flour, XIII—1983*, vol. 1117 of *Lecture Notes in Math.*, pp. 1–198, Berlin: Springer, 1985.
- [30] D. Blackwell and J. MacQueen, “Ferguson distributions via Polya urn schemes,” *The Annals of Statistics*, vol. 1, pp. 353–355, 1973.
- [31] F. Eggenberger and G. Pólya, “Über die statistik verketteter vorgänge,” *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 3, no. 4, pp. 279–289, 1923.
- [32] S. Kullback and R. A. Leibler, “On information and sufficiency,” *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.
- [33] M. Hoffman, D. M. Blei, C. Wang, and J. Paisley, “Stochastic variational inference,” 2013.
- [34] A. T. Wilson and P. A. Chew, “Term weighting schemes for latent dirichlet allocation,” *HLT ’10, (USA)*, p. 465–473, Association for Computational Linguistics, 2010.
- [35] M. A. Hearst, “Automatic acquisition of hyponyms from large text corpora,” in *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*, 1992.
- [36] M. Geffet and I. Dagan, “The distributional inclusion hypotheses and lexical entailment,” in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*, (Ann Arbor, Michigan), pp. 107–114, Association for Computational Linguistics, June 2005.
- [37] J. Chang, J. Boyd-Graber, S. Gerrish, C. Wang, and D. Blei, “Reading tea leaves: How humans interpret topic models,” vol. 32, pp. 288–296, 01 2009.
- [38] J. Lau, D. Newman, and T. Baldwin, “Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality,” *14th Conference of the European Chapter of the Association for Computational Linguistics 2014, EACL 2014*, pp. 530–539, 01 2014.
- [39] D. Soergel, “Wordnet. an electronic lexical database,” 10 1998.
- [40] L. van der Maaten and G. Hinton, “Viuualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 11 2008.
- [41] A. B. Dieng, F. J. R. Ruiz, and D. M. Blei, “Topic modeling in embedding spaces,” 2019.
- [42] K. Batmanghelich, A. Saeedi, K. Narasimhan, and S. Gershman, “Nonparametric spherical topic modeling with word embeddings,” 2016.

Appendix A

Supplementary examples of topics found by the HDP model

A.1 English corpus

A.1.1 ONE

ID	Top words	U_{mass} score
0	number, function, example, theory, point	-0.72
1	american, english, bear, player, french	-1.28
2	system, code, language, number, program	-0.81
3	theory, human, study, world, work	-0.64
4	country, government, world, economy, large	-0.76
5	form, element, produce, metal, process	-0.89
6	state, government, party, president, election	-0.74
7	system, include, game, datum, release	-0.91
8	language, word, english, form, example	-0.82
9	specie, include, island, plant, large	-0.65
10	state, member, government, united_states, court	-0.62

Table A.1: Examples of topics with very good coherence score found by the HDP model with the weighting scheme ONE in the English corpus.

ID	Top words	U_{mass} score
110	angel, wing, depiction, archangel_gabriel, gabriel	-5.26
111	amalric, surname, alter, miser, tube	-5.69
112	form, disperser, mean, stem, ulterior	-4.01
113	baldr, völva, bald, odin, baldur	-1.75
114	tischendorf, leipzig, maecena, novum_testamentum, manuscript	-3.93
115	gent, word, hair, superior, ghent_gent	-4.54
116	alexander, afrikaans, dutch, afrikaan, south_africa	-6.69
117	dictionary, college, cuny, brooklyn, york	-3.35
118	saint, list, innocent, alexander, pope	-3.13
119	clitic, host, enclitic, attach, affix	-12.45
120	tamarin_saguinus, kinshasa, drum, congo, capuchin	-13.81
121	equus, genus, west_flemish, class, agreement	-10.39
122	barlaam, saga, zorn, ailanthus, version	-7.64
123	cedar_fall, anastasio, myanmar, anastasio, kyun	-11.27
124	dionysius, benedetto, paragraph, grammatical, dionysius_thrax	-10.11
125	direct, callisto, jean, robert, andré	-5.16
126	balch, oromo, gysin, seagram, accidentally_discover	-13.12
127	judoka, modern_pentathlete, liège, theoretical_physicist, andorran	-2.25
128	alexis, fragment, odysseus, ankylion, athenaeus	-8.61
129	portugal, lusitania, municipium, lusitanian, lusitani	-3.97
130	gibberish, gobbledygook, irish, jargon, tanaka	-10.30
131	canon_quod, lateran, commelinaceae, clericorum, payment	-7.93
132	karume, đinđić, oldcorne, jafri, konietzka	-4.04
133	chadic, proto_chadic, newman, chadic_branch, chadic_language	-3.52
134	emperor_theodosius, collider, bangui, convict_crime, ryanair	-8.87
135	aalborg, aalborg_municipality, vildmose, teleorman, lille	-9.29
136	rievaulx, zanzibar, religious_freedom, colliery, schöner	-7.88
137	light, bethany, anatolius, kilpatrick, pickett_charge	-5.30
138	gnaeus, kharkov, layard, newburgh, cochinchina	-8.30
139	schwenckfeld, agouti, schwenckfelder, silesia, caspar_schwenckfeld	-13.39

Table A.2: Topics around ID 120 with very low coherence score (found by the HDP model with the weighting scheme ONE in the English corpus).

A.1.2 IDF

ID	Top words	U_{mass} score
0	government, party, election, president, soviet	-1.03
1	company, economy, market, country, government	-1.00
2	datum, user, code, network, system	-1.28
3	university, student, college, polk, campus	-1.52
4	theory, philosophy, knowledge, wittgenstein, hayek	-1.71
5	marx, chiang, goldwater, rohrabacher, hemingway	-4.11
6	particle, energy, earth, electron, mass	-1.69
7	haiti, government, roosevelt, iran, nehru	-2.42
8	american_actor, american, player, american_singer, politician	-0.89
9	computer, software, intel, processor, atari	-1.04
10	album, music, band, song, game	-3.21
11	gandhi, film, novel, story, nietzsche	-6.29
12	function, theorem, algebra, equation, integer	-1.52
13	giraffe, sagan, eindhoven, mirror, dance	-4.35
14	disraeli, party, aaliyah, aalen, election	-4.63
15	ribbentrop, king, england, henry, circe	-3.30
16	album, film, sinatra, band, song	-3.34
17	theory, economic, value, anarchist, rothbard	-5.46
18	rommel, army, military, force, hitler	-2.50
19	aircraft, missile, vehicle, engine, navy	-1.65
20	church, jesus, christ, chaucer, christian	-5.40
21	drug, patient, disease, symptom, treatment	-2.29
22	film, presley, album, episode, capp	-4.52
23	lake, forest, island, river, water	-3.19
24	player, game, team, ball, aircraft	-2.60

Table A.3: Examples of the first topics with moderately good coherence score found by the HDP model with the weighting scheme IDF in the English corpus.

ID	Top words	U_{mass} score
90	kabul, afghanistan, language, afghan, indo_european	-7.01
91	cagney, landis, baseball, dodo, nansen	-9.20
92	less_poland, dredd, wollstonecraft, ford, elijah	-16.94
93	agassi, coca_cola, hewitt, solzhenitsyn, lynch	-15.57
94	irgun, hezbollah, botham, israel, druze	-7.97
95	firearm, leibniz, crossbow, albanian, albania	-16.58
96	ballarat, roussimoff, greece, rodin, caravaggio	-14.40
97	kipsigis, laos, khmer, steinbeck, bulgaria	-15.02
98	finland, updike, stanislavski, chess, finnish	-14.12
99	basquiat, darwin, emperor, astrology, elfman	-12.49
100	bear, living, john, thomas, robert	-1.52
101	jump, calendar, symbol_definition, unit, month	-8.23
102	fujiwara, imperial_prince, daughter_imperial, minamoto, kammu	-4.37
103	niger_congo, christianity_catholicism, islam_sunni, islam, indo_european	-4.14
104	political_activist, hitomaro, anti_apartheid, bear, south_african	-6.80
105	cichlid, catfish, common_taxonomy, size_remark, tetra	-1.77
106	cellulose, cell_wall, laetrile, hemicellulose, amygdalin	-6.02
107	intelligence, inteligencia, military_intelligence, intelligence_service, october_général	-6.42
108	berber, berber_language, tuareg, khoisan, kabyle	-10.31
109	galicia, lugh, galician, pontevedra, coruña	-8.08
110	cuba, bahama, cuban, bahamian, cubans	-8.23
111	cryptocoryne, echinodorus, aponogeton, sagittaria, bacopa	-9.22
112	megatokyo, piro, miho, casuistry, craig_charles	-11.24
113	macro, virus, dye, leviticus, biological_weapon	-12.22
114	psychology, psychologist, organizational, imam, meta_element	-8.87
115	cocreate, junit, cofounde, algol_ifip, artificial_intelligence	-3.96
116	sayyaf, kidnapping, jolo, hostage, philippine	-2.23
117	hindu, sketch, cast_member, hindus, makran	-9.58
118	bordeaux, feudalism, feudal, vassal, file_bordeaux	-11.19
119	library, cyrus, die, bear, public_library	-9.29
120	aristophane, parabasis, comedy, agathon, acharnian	-2.01

Table A.4: Examples of the first topics with moderately good coherence score found by the HDP model with the weighting scheme IDF in the English corpus.

ID	Top words	U_{mass} score
240	william.laud, orseolo, agatho, issai, birkbeck	-1.22
241	bahrain, bilotti, alphonse, gabriel_narutowicz, pepin_herstal	-1.73
242	bösendorfer, johann.sebastian, alphonsa, kennewick, clipper	-1.94
243	laetare, rudesind, mustafa_barzani, banten, socialist_federal	-1.48
244	molay, anselm_lucca, bromell, sayafi, ngouabi	-0.17
245	anandpur, beauty_queen, pathé, tannous, bodolai	-0.62
246	laker, geochang, greenville, gabelsberger, ehime	-1.09
247	sedre, tkvarcheli, welsh_rugby, millau, kirkham	-1.13
248	mexican_painter, mirbach, goretta, mowry, daniel_morgan	-1.67
249	kyun, myanmar, taung, razi, taungdan	-1.61
250	ludger, hockey_association, gucci, birkman, boerwinkle	-0.67
251	tekakwitha, kateri, emslie, sirhan, grierson	-1.23
252	principalities, runeberg, keren, rosenblat, covian	-0.54
253	alamance, mancroft, bobola, woolman, hollowell	-0.12
254	tashkent, aldobrandesca, kaikan, uzunov, dechko	-0.35
255	ebor, fawkes, waterfall, armidale, guyra	-5.23
256	marto, haselbury, wulfric, jacinta, duke_urbino	-1.14
257	giffords, cockfield, severinus, kadi, giustiniani	-1.07
258	desjardin, rodez, monorail, lebeck, sereny	-0.50
259	völkner, jamahiriya, west_fracia, secretary_interior, plumain	-0.81
260	mullet, amyl_alcohol, methyl, butanolor, methylbutan	-8.45
261	brandenburg, dietrich, ziesar, prince_bishopric, brandenburg_havel	-2.07
262	akre, goodness, friel, poul, familien	-13.08
263	künstlerroman, bildungsroman, halldór_laxness, lanark, tutunamayanlar	-0.68
264	acoustic_wave, stereo, microflare, uvcs, edlen	-0.76

Table A.5: Examples of the very last topics with relatively good coherence score found by the HDP model with the weighting scheme IDF in the English corpus.

A.1.3 PMI

ID	Top words	U_{mass} score
0	system, computer, datum, code, software	-1.27
1	function, theory, number, point, define	-1.13
2	american, bear, english, politician, player	-1.22
3	education, university, student, college, theory	-1.35
4	country, ghana, luxembourg, city, libya	-2.66
5	church, bishop, book, bible, moses	-1.63
6	washington, party, john, spinoza, guatemala	-3.64
7	reaction, metal, mineral, compound, carbon	-1.58
8	hoover, haiti, carson,iefenbaker, conrad	-8.78
9	aston_martin, british, government, year, party	-4.68
10	relation, cuba, country, president, government	-4.56
11	album, film, band, song, bowie	-4.95
12	chiang, jakarta, aristotle, medicine, climate	-5.13
13	disease, treatment, patient, disorder, cause	-3.50
14	rommel, german, germany, morocco, crowley	-4.52
15	batman, game, film, cagney, club	-5.79
16	stalin, party, polk, aclu, soviet	-5.07
17	population, male_female, year_male, female_year, european	-2.05
18	film, dub, lovecraft, frankfurt, music	-5.65
19	hong_kong, island, milk, macau, flag	-2.59
20	lithuania, eisenhower, latvia, church, firearm	-5.44
21	karachi, mdma, minimum_wage, rowling, jedi	-6.11
22	french, army, ribbentrop, france, henry	-3.80
23	martin, hubbard, elijah, story, character	-3.32
24	henry, disraeli, emperor, augustus, dolphin	-7.05

Table A.6: Examples of the very first topics with relatively good coherence score found by the HDP model with the weighting scheme PMI in the English corpus.

ID	Top words	U_{mass} score
60	star, dalek, galaxy, ballarat, constellation	-6.26
61	iraq, cardiff, berlin, iran, buddha	-8.54
62	hitchcock, cannon, century, diocletian, monet	-7.38
63	ashoka, coin, ltte, court, philby	-11.54
64	music, aaliyah, enron, advertising, kansas	-8.62
65	comet, cornwall, joyce, capone, nansen	-9.68
66	jersey, intel, bicycle, galen, lombard	-11.49
67	city, denver, kansas_city, montana, douglass	-5.61
68	island, hemingway, insurance, hawaii, parton	-10.08
69	helium, hayek, neumann, aluminium, acid	-8.91
70	blue, alfred, guam, haydn, music	-9.62
71	apollo, mission, crew, armenia, darwin	-2.53
72	hamas, hezbollah, japan, greece, lebanon	-4.49
73	film, mirror, botham, star, milton	-9.70
74	word, language, japanese, english, image	-5.58
75	nietzsche, schwarzenegger, sartre, hadrian, galactus	-13.88
76	einstein, carnegie, painting, berkeley, ghost	-4.93
77	gandhi, food, demille, marx, cuisine	-9.19
78	game, kosovo, relation, king, albanian	-6.18
79	lewis, chaplin, hungary, chaucer, kepler	-7.82
80	sinatra, door, shaw, munich, aphrodite	-10.43
81	basque, cyprus, bangladesh, country, baptism	-4.31
82	liverpool, austin, city, qaeda, israel	-6.28
83	band, presley, album, jazz, metallica	-8.97
84	ainu, burke, kazakhstan, coca_cola, moose	-9.60
85	black, green, honda, matilda, adelaide	-5.06
86	mars, earth, moon, cobain, athena	-8.09
87	black_hole, film, jesuit, hewitt, carver	-8.07
88	address, network, element, ithaca, mieszko	-8.21
89	luke, literature, kant, borges, lister	-7.26
90	dodo, bone, andorra, meat, methodist	-7.34

Table A.7: Examples of topics around ID 75 with very low coherence score found by the HDP model with the weighting scheme PMI in the English corpus.

ID	Top words	U_{mass} score
170	drayson, crispus_attuck, patsy_cline, joseph_stalin, piran	-10.92
171	alfred_nobel, jople, thomas_merton, birt, nyköpe	-2.16
172	kargil, mawhinney, ulises_heureaux, thornycroft, guelmim	-1.91
173	mezouara, dolet, thuringia, kruševó, burghersh	-2.10
174	author_translator, carole_lombard, astronomer_cartographer, bogle, schöner	-1.22
175	nemtsov, majuba, dominican_republic, hazlehurst, neuberger	-1.98
176	assisi, wotruba, brian_boru, æthelre_unready, childebert	-1.93
177	boste, crook, vicente, maracaibo, hwan	-1.68
178	lewis, television_series, temperance, harold, opler	-3.20
179	kroonstad, hennenman, žižić, topsy, maktoum	-1.39
180	eames, connectu, winklevoss, ninoy, vimeiro	-0.73
181	beriev, barletta, catherine_howard, faiz, kurup	-0.57
182	ajax, digital_library, london_william, available_perseus, heinemann_online	-7.83
183	sailor, sears, sufi, attack_pearl, leeroy	-1.64
184	anterus, chavara, nimr, qasem_soleimani, бага	-0.51
185	mostaganem, caber, sidi, province, titanite	-12.96
186	abingdon, aleni, giulio_aleni, brescia, bresciana	-1.17
187	avahi, woolly_lemur, sifaka_propithecus, indri, indriidae	-1.29
188	tigin, bilge, xinjiang, qapagan, kaghan	-1.40
189	perceval, cadbury, huepe, brewton, gebrhiwet	-0.54

Table A.8: Examples of the very last topics with relatively high coherence score found by the HDP model with the weighting scheme PMI in the English corpus.

A.2 French corpus

A.2.1 ONE

ID	Top words	U_{mass} score
0	français, américain, homme_politique, acteur_américain, saint	-0.91
1	juin, septembre, août, france, juillet	-0.53
2	utiliser, exemple, forme, élément, produire	-1.12
3	utiliser, fichier, logiciel, système, version	-1.18
4	commune, ancien, occupation_sol, note_référence, français	-0.88
5	commune, saint, ancien, jean, breton	-1.04
6	français, maire, france, catégorie_député, catégorie	-1.07
7	roman, œuvre, homme, monde, catégorie	-0.71
8	nombre, fonction, ensemble, exemple, définir	-0.86
9	film, catégorie, roman, série, américain	-0.62
10	romain, empire, empereur, rome, grand	-0.84

Table A.9: Examples of the very first topics with very good coherence score found by the HDP model with the weighting scheme ONE in the French corpus.

APPENDIX A. SUPPLEMENTARY EXAMPLES OF TOPICS FOUND BY THE HDP MODEL61

ID	Top words	U_{mass} score
90	année, catégorie, roman, groupe, nouveau	-0.96
91	saint, commune, rochelle, venir, église	-2.91
92	ville, royaume, franc, maastricht, fils	-1.86
93	département, région, ville, nord, fichier	-2.01
94	président, américain, venise, truman, nouveau	-2.62
95	ville, fichier, saint_quentin, toronto, ligne	-1.77
96	œuvre, pays, coupe_monde, france, caravelle	-2.12
97	estonien, pays, cancer, estonie, céline	-2.07
98	langue, suisse, saint, sénégal, jean_baptiste	-4.41
99	école_national, public, pari, école_supérieur, ingénieur	-2.37
100	langue, groupe, européen, langue_indo, langue_austronésien	-1.95
101	scénario, dessin, titre, français, liste	-1.63
102	bavièr, rhénanie_nord, bad_wurtemberg, westphali, hesse	-2.19
103	harpalu, genre, catégorie, espèce, acarien	-8.20
104	kerlaz, josé, hidalgo, juan, manuel	-5.87
105	sanskrit, grec, kâlo_pays, persan, romani	-2.24
106	jean, françois, pierre, louis, charles	-1.28
107	agglomération, population_urbain, guangdong, urbain_agglomération, jiangsu	-2.22
108	sony, encoche, corporation, æquo, festival	-8.58
109	poète, espéranto, langue, william, littérature	-3.77
110	pizza, romaniser, chiffre_nom, italien, zero	-7.12
111	danseur_chorégraphe, danseur, danseus_chorégraphe, danseus, danseur_chorégraph	-2.45
112	nicholson, meilleur_acteur, acteur, jack_nicholson, marin	-5.95
113	jus, dumort, airy_shaw, agardh, lindl	-6.78
114	copte, tell, sobek, mongol, póli	-11.67
115	pterostichus, pterostichu, casey_pterostichus, casey_pterostichu, danse	-12.05
116	numéro, part, partie, écrire_jos, tome	-3.93
117	page, biologie, file, groupe_bayard, documentation	-11.29
118	noël, angelina, sima, lapidu, christine	-10.26
119	footballeur_international, joueur_professionnel, taksin, aude, slovaque	-5.60
120	neurone, bihar, axone, dendrite, viessoix	-11.44

Table A.10: Examples of topics with ID close to 100 found by the HDP model with the weighting scheme ONE in the French corpus.

ID	Top words	U_{mass} score
185	douane, duarte, juan_pablo, victor_marseille, blinn	-8.32
186	schooneveld, aljubarrota, nocera, faustino, sonde_pioneer	-7.74
187	abba, junte, congar, dramaturge_irlandais, outlaw	-6.95
188	fauguernon, cfm, faguernon, regnobert, john_tyler	-12.96
189	autun, saalfeld, aléria, exarqu, ugene	-7.00
190	saint_rémi, automobile_endurance, steeg, typhain, coderr	-5.44
191	saint_évarzec, moustoir, saint_evarzec, troyalac, varzécoi	-8.40
192	franz_papen, walter_raleigh, byzance, norodom_sihamoni, pester	-5.78
193	puits, diogèn, frédou, nizâr, sunqur	-9.00
194	zempla, affaire_corruption, lutte_désertification, murphy_acteur, frankfurt	-3.20
195	code_justinien, zamenhof, déclaration_droit, calvair, pencher	-4.10
196	laïka, chien, chienne, cabine, taux	-2.88
197	alamut, irty, saule, arensnoughi, rosan	-13.52
198	kentucky, grêle, möhne, izaki, hucleux	-0.79
199	soury, ashmor, bisseuil, guguss	-0.31
200	agnietenberg, casi, pulad, octonville, rambulo	-7.27
201	bryne, warmond, decollato, fontico, weidhausen	-0.20
202	tsuruhim, kanesada, ostroma, hjorten, zeimoto	-0.55
203	geilon, nomismater, tatzatè, adalgis, süntelgebirge	-1.37
204	hinba, cill, revatidvipa, mangalesha, luleburgaz	-5.51

Table A.11: Examples of the very last topics with relatively high coherence score found by the HDP model with the weighting scheme ONE in the French corpus.

A.2.2 IDF

ID	Top words	U_{mass} score
0	commune, occupation_sol, nombre_jour, église_saint, type_climat	-0.88
1	page_concerne, julien_événement, août, début_règne, juin	-0.98
2	logiciel, version, serveur, langage, utiliser	-1.21
3	politique, parti, état, gouvernement, socialisme	-1.33
4	catégorie_député, maire, république_catégorie, assemblée_national, député	-1.03
5	nietzsche, œuvre, freud, pétain, droit	-1.16
6	churchill, science, langue, durkheim, estonien	-1.69
7	molécule, atome, étoile, température, métal	-1.72
8	entreprise, microsoft, système, société, sncf	-1.36
9	chilpéric, peine_mort, macédonien, pays, macédoine	-3.91
10	ville, département, lille, laval, grenobl	-3.38
11	senna, prost, grand_prix, alain_prost, pilote	-5.89
12	strasbourg, ville, quartier, york, bruxelle	-2.05
13	fonction, nombre, théorème, ensemble, définir	-1.80
14	vampire, gouvernement, pays, irak, politique	-1.85
15	alenço, cherbourg, ville, chartre, région	-4.26
16	août, septembre, juin, juillet, avril	-0.96
17	trou_noir, électron, particule, masse, vitesse	-1.85
18	pays, hong_kong, population, corée_nord, port	-2.14
19	pesticide, caféine, sega, produit, dreamcast	-6.18
20	manga, platon, lion, socrate, gosciny	-3.42
21	allemand, guerre, juif, britannique, stalingrad	-1.13
22	toulon, alger, versaille, kerlouan, pont_aven	-3.23
23	heidegger, lannion, locquirec, landévennec, commune	-8.53
24	metz, ville, colmar, troye, melun	-5.75

Table A.12: Examples of the very first topics with very high coherence score found by the HDP model with the weighting scheme IDF in the French corpus.

APPENDIX A. SUPPLEMENTARY EXAMPLES OF TOPICS FOUND BY THE HDP MODEL64

ID	Top words	U_{mass} score
140	bri, brix, américain, acteur_américain, brui	-9.04
141	mitsubishi, japonais, japon, shōgakukan, shōgakukan_scénario	-10.93
142	bit, cryptographie, scsi, valeur_hachage, message	-8.25
143	malgache, amharique, folo, puluh, malais	-10.82
144	carte, homme_politique, américain, canadien, joseph	-6.55
145	appariement, nucléotide, arnm, messenger, ribosome	-1.56
146	atari, captain_tsubasa, captain, tsubasa, page_tome	-9.96
147	mespaul, Frédéric_dard, same, antonio, saint_catherine	-11.51
148	ouganda, assouan, amin_dada, victorien, fleuve	-3.37
149	acteur_américain, acteur_français, homme_politique, champion_olympique, français	-2.22
150	yonne, auxerr, bourgogne, auxerre, bernard_borderie	-3.74
151	président_république, ministre, président, gouverneur_général, saint_géraud	-6.34
152	population_urbain, urbain_agglomération, guangdong, agglomération, jiangsu	-1.92
153	ger, territoire_ciuita, lannemezan, ciuita, auch	-5.12
154	william, john, richard, robert	-9.11
155	sultan_sultanat, principauté, royaum, sultanat, sultan	-2.49
156	allemand, vert, écologiste, bundestag, jürgen_trittin	-9.15
157	étoile_filant, variable_irrégulier, aquaride, faible, juillet	-3.59
158	canton, gavray, mesnil, octeville, canisy	-4.83
159	acteur_américain, homme_politique, américain, québécois, footballeur_international	-5.22
160	linux, fichier, affiche, système, commande	-5.76
161	python, module, yield, thury, harcourt	-6.74
162	mélodie, dame_eboshi, princesse_mononoker, ashitaka, ashitaker	-11.36
163	daniel_prévost, blet, prévost, mettre_scène, garage_gaudin	-5.69
164	immigration, immigré, migrant, migration, pascal_blanchard	-6.45
165	bouche_rhône, mana, marseille, marcel_maus, provence	-7.79
166	alfred_musset, musset, george_sand, badine_amour, caprice_marianne	-4.66
167	footballeur_français, américain, veille_noël, français, homme_politique	-4.89
168	delhi, punaise, infra_ordre, inde, delhi_delhi	-6.21
169	zappa, frank_zappa, michel_jeury, fleuve_noir, anticipation	-8.93
170	samouraï_époque, kett, août, cardinal_italien, saadien_mohammed	-2.89

Table A.13: Examples of topics with ID close to 150 found by the HDP model with the weighting scheme IDF in the French corpus.

APPENDIX A. SUPPLEMENTARY EXAMPLES OF TOPICS FOUND BY THE HDP MODEL65

ID	Top words	U_{mass} score
365	volt, tension_alternatif, maire_pfastatt, tension, haut_rhin	-9.17
366	dollfuss, lerroux, nsdap, autriche_chancelier, incendie_reichstag	-1.13
367	bacilly, fief, rousselière, chantore, ernault	-11.54
368	gaghik, badi, alitigin, hammad, edmond	-12.39
369	carcagny, montargi, pierre_carcagny, cachemire, bulan	-15.32
370	slammer, bruchevill, correctif, routeur, server	-5.59
371	urer, urée, uré, azoté, mélamine	-1.70
372	beaucoudray, brouain, freul, magny_freule, wingle	-14.86
373	fauguernon, cfm, faguernon, guernon, regnobot	-0.46
374	cthulhu, lovecraft, lyeh, august_derleth, mythe_cthulhu	-10.34
375	crépon, moutier_cingler, crepon, cingler, wiślica	-14.59
376	mézeret, saint_vigor, fresvill, fresville, vigor	-9.10
377	audrieu, bény_bocage, haise, foucardière	-9.91
378	audiberti, bryen, giroud, jacquelin, hobereaute	-0.54
379	ismaël, noaille, géronne, palmare, morcellement	-6.88
380	mogholistan, barletta, shimazu, luchino_visconti, maniériste_graveur	-4.16
381	manfred_sicile, batu, ezzelino, phénoménal, saule	-13.33
382	herqueville, herquevill, herqu, brumer, hergue	-0.45
383	mallouer, malloué, ortair, nantier, burnside	-12.54
384	inishbofin, rechru, centula, kuffenstein, anousan	-12.02
385	ruyter, jouravno, bosse, vainqueur, amiral	-3.33
386	yonen, maslama, buzz, khanzim, bhavabhuti	-5.35
387	saichō, kūkai, nacoleia, claudiopoli, mesembria	-14.45
388	liao, kharpout, suleiman, aguda, kitan	-6.08
389	entourer_ange, pachymère, smrtludvíkai, zague, congréation	-9.75
390	kant, gregorii, decretale, altenesch, stedinger	-8.53
391	bayazid, zizim, veli, bahmanî, zaharer	-0.75
392	simihel, tubua, gegncésiu, hingan, moïountchour	-9.81
393	refrancore, perthari, forino, saburrus, calor	-9.60
394	hajjam, reginar, arciat, meknassa, minhal	-3.11
395	arciciu, mûsa, mafjar, khirbat, qousayr	-2.93
396	chvarno, zahiriyah, lizong, lewer, merton_college	-3.38

Table A.14: Examples of the very last topics found by the HDP model with the weighting scheme IDF in the French corpus.

A.2.3 PMI

ID	Top words	U_{mass} score
0	français, américain, homme_politique, acteur_américain, canadien	-0.98
1	juin, août, septembre, juillet, avril	-0.88
2	logiciel, utiliser, version, donnée, système	-1.39
3	catégorie_député, maire, république_catégorie, député, législature_république	-0.80
4	commune, saint, manoir, bourg, paroisse	-1.63
5	commune, occupation_sol, nombre_jour, église_saint, type_climat	-0.89
6	roman, mozart, jules_verne, arsene_lupin, œuvre	-2.62
7	maurra, grèce, grec, concile, politique	-2.56
8	fonction, nombre, ensemble, groupe, définir	-1.39
9	étoile, planète, constellation, soleil, trou_noir	-1.97
10	astrologie, génocide, roman, éthanol, king	-7.28
11	social, femme, intelligence_artificiel, genre, durkheim	-3.46
12	churchill, pétain, wikipédia, sega, dreamcast	-5.13
13	œuvre, violence, droit_auteur, otan, mystique	-4.70
14	pays, langue, mexique, stalingrad, volapük	-3.19
15	empire, état, mandela, nelson_mandela, france	-2.39
16	nietzsche, descartes, heidegger, platon, culture	-3.90
17	pape, édouard, france, léopold, français	-3.70
18	rousseau, socialisme, socialiste, politique, hadith	-5.48
19	tibet, tibétain, inde, roosevelt, corée_nord	-2.74
20	florida, pays, état_uni, chine, état	-3.20
21	béarn, béarnais, saint_quentin, créteil, département	-7.55
22	espèce, pesticide, virus, plante, produit	-3.53
23	truman, peine_mort, françois_mitterrand, parti, décembre	-1.52
24	film, hitchcock, molière, sarde, gaulois	-6.56

Table A.15: Examples of the very first topics with relatively good coherence score found by the HDP model with the weighting scheme PMI in the French corpus.

APPENDIX A. SUPPLEMENTARY EXAMPLES OF TOPICS FOUND BY THE HDP MODEL67

ID	Top words	U_{mass} score
379	coulouvray_boisbenâtr, bois_benastre, coulouvray, boisbenâtre, benatre	-10.76
380	albon, adon, cleef, pakistan_oriental, harmufa	-8.44
381	ruiz, margera, lucka, prévéza, nolwenn	-0.81
382	vacquerie, economie, jane, interest, laps	-10.00
383	feuilleie, ballmer, bill_gate, carantanie, posavski	-15.93
384	alban, verulamium, samosate, prophète, waldir	-1.12
385	vincer_ferrier, burgo, germinal, peïpous, colin_powell	-9.29
386	mccarthy, artificial, intelligence_artificiel, reasoning, shapur	-9.07
387	arzano, plouay, maire_plouay, bizien, hennebont	-9.63
388	sobek, grand_celland, celland, xavier_roux, roux	-14.86
389	sèvres, aubier, nueil, député_sèvres, dominiqu_pailler	-6.94
390	aubierge, severinsen, cutugno, jovenel, aldebert	-1.10
391	serveur_http, covenanter, conder, httpd, robert_blake	-12.93
392	fongicide, cher, fromion, policy, monteille	-12.64
393	turbo_pascal, borland, pascal, compilateur, turbo	-12.94
394	saint_honoré, civette, mieszko, giurgola, romaldo	-0.77
395	lannédern, tréouergat, edern, saint_edern, gouescat	-7.64
396	spectrum, sinclair_research, sinclair, hasteinn, valentin	-14.96
397	aisne_rang, aisne, chauny, densité, hamazaki	-3.11
398	oliver_cromwell, scheveningen, leuenberger, province_uni, paysan	-2.64
399	ballade, conseil, autruy, offenser, appostr	-11.47
400	couvain, libra, nendo, wahan, bretel	-14.93
401	diessenhofen, philippsburg, danzig, baïdou, baldovinetti	-15.06
402	hyperglycémie, cassano, marbella, sangyé_gyatso, tesser	-8.65
403	calogero, arigi, vencer_delerm, alain_bashung, johnny_hallyday	-3.94
404	hunyadi, yassin, vlad_empaleur, ladislav, ziper	-12.76
405	monbazillac, bergerac, dordogne, malfourat, fonvieille	-1.29
406	rastatt, ouverture_conférence, baden, hoffnung, demotiker	-0.90
407	aistolf, guifei, lushan, marcellae, concubin	-8.35
408	badefol, robaut, enköping, prapanca, sinjbar	-3.21
409	igraïne, sigismer, domiciu, pendragon, uther_pendragon	-3.05

Table A.16: Examples of the very last topics found by the HDP model with the weighting scheme PMI in the French corpus.

Appendix B

Supplementary examples of topics found by the nHDP model

B.1 English corpus

Level 1		Level 2		Level 3	
ID	Top words	ID	Top words	ID	Top words
9	force, army, state, country, year	117	dostum, johnston, doubleday, force, afghanistan	876	doubleday, baseball, abner_doubleday, game, corps
				877	dostum, afghanistan, taliban, northern, afghan
		115	military, army, personnel, force, south_african	731	army, mercenary, state, soldier, stand
				732	azerbaijan, azerbaijani, military, azeri, baku
				733	military, armenia, armenian, russian, defense
				734	antarctica, antarctic_treaty, antarctic, claim, country
		114	tank, vehicle, german, design, mount	655	tank_destroyer, anti_tank, casemate, superstructure, jagdpanzer
				656	vehicle, tank, armour, carry, armoured
				657	spaag, anti_aircraft, gun, mount, aircraft
				658	assault, assault_gun, tank, role, howitzer

Table B.1: Sub-tree of topics about Military forces found with nHDP in the English corpus.

APPENDIX B. SUPPLEMENTARY EXAMPLES OF TOPICS FOUND BY THE NHDP MODEL70

Level 1		Level 2		Level 3	
ID	Top words	ID	Top words	ID	Top words
10	apollo, time, moon, mission, flight	120	apollo, mission, crew, astronaut, moon	435	crew, spacecraft, borman, earth, flight
				436	armstrong, eagle, july, aldrin, columbia
				437	apollo, moon, shepard, astronaut, roosa
				438	scott, worden, falcon, astronaut, moon
				439	apollo, lovell, swigert, oxygen_tank, tank
		124	aircraft, power, type, wing, fire	705	fire, atmosphere, cabin, grissom, spacecraft
				706	armour, plate, century, world, protection
				707	balloon, design, flight, include, rocket
		122	winter, arctic, summer, mark, cold	560	aardwolf, termite, territory, hyena, family
				561	arctic, arctic_fox, population, fox, lemming
				562	aardvark, burrow, animal, long, dig
				563	antarctica, station, vehicle, snow, length
				564	amphibian, frog, salamander, water, specie
		127	young, apollo, duke, crater, earth	920	zone, ship, blight, slow, novel
				921	apollo, eisele, nasa, crew, stafford
				922	apollo, experiment, mission, crew, landing_site

Table B.2: Sub-tree of topics about Astronomy found with nHDP in the English corpus.

B.2 French corpus

Level 1		Level 2		Level 3	
ID	Top words	ID	Top words	ID	Top words
4	département, france, région, commune, ville	60	département, marseille, bouche_rhône, provence, autoroute	266	montpellier, nice, hérault, alpe_maritime, canne
				267	charente, allier, sèvres, cognac, bourbonnais
				268	isère, nîme, gard, festival, isérois
		65	rhin, haut_marne, strasbourg, château, territoire	485	haut_rhin, département, haut, alsace, rhin
				486	charente_maritime, rochelle, saint, rochefort, oléron
				487	haut_marne, département, chaumont, saint_dizier, marne
				488	france, distribution, direction, électricité, réseau_distribution
				489	région, champagne_ardenn, champagne, aube, marne
		68	pari, france, haut_seine, région, essonne	695	haut_seine, essonne, nanterre, département, défense
				696	france, région, pari, régional, grand

Table B.3: Topics sub-trees found by nHDP in the French corpus.

APPENDIX B. SUPPLEMENTARY EXAMPLES OF TOPICS FOUND BY THE NHDP MODEL72

Level 1		Level 2		Level 3	
ID	Top words	ID	Top words	ID	Top words
8	japonais, kendo, japon, sabre, shinai	101	pratique, école, japonais, enseignement, technique	417	technique, travail, goshindo, exercice, pratiquant
				415	sabre, iaidō, koryū, kata, katana
				416	kamikaze, pilote, appareil, attaque, navire
				418	archer, flèche, kyūdō, kyudo, cible
		102	épée, long, utiliser, arme, bois	472	daishō, samouraï, épée, tantō, port
				470	baguette, chinois, utiliser, bambou, jetable
				471	bokken, travail, employer, koryu, aïkido
				473	arme, cavalier, arme.blanc, vignette.redresse, trancher
				474	instrument, instrument.musique, percussion, musique, musée
		105	daitōryū, déterminer, enseigner, martial, enseignement	680	aïkibudo, ushiro, uchi, arrière, pied
				681	hakkō, jutsu, technique, école, okuyama
				681	hakkō, jutsu, technique, école, okuyama

Table B.4: Sub-tree of topics about Japanese martial arts found by nHDP in the French corpus.

Appendix C

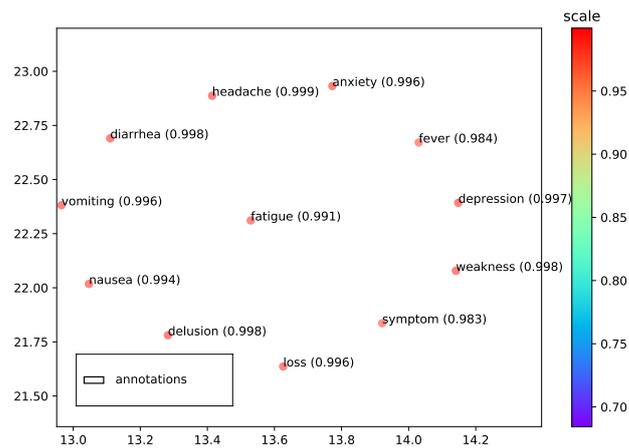
Hearst Patterns used for hypernyms extraction

Hearst Patterns (X is-a Y relations)
NP_Y such as NP_X, ...
such NP_Y as NP_X, ...
NP_X (and or) (some any) other NP_Y
NP_Y include NP_X, ...
NP_Y especially NP_X, ...
NP_X, ... (is are) (a an) NP_Y
NP_Y like NP_X, ...
NP_X, ... like other NP_Y
NP_X, ... (and or) one of (the these those) NP_Y
(example instance) of NP_Y be NP_X, ...
NP_X, ... be (a an)? (example instance) of NP_Y
NP_Y, for (example instance) NP_X, ...
NP_Y, (mainly mostly notably particularly principally in particular) NP_X, ...
NP_Y, (i.e. e.g.) NP_X, ...
NP_X, (and or) (a) kind of NP_Y
NP_X, (and or) (a) form of NP_Y
NP_X, ... which (look sound) like NP_Y
NP_Y, ... which be similar to NP_X
NP_Y type NP_X, ...
NP_X (and or) NP_Y type
NP_Y whether NP_X, ...
compare NP_X, ... with NP_Y
NP_Y among -PRON- NP_X, ...
!(such) NP_X, ... as NP_Y
NP_X, ... (and or) sort of NP_Y
NP_Y which may include NP_X, ...

Table C.1: Hearst Patterns used to extract pairs of hypernyms from the lemmatized text.

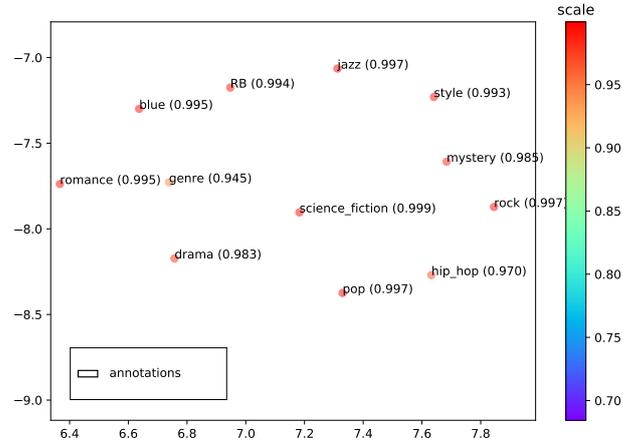
Appendix D

Additional examples of clusters of word embeddings

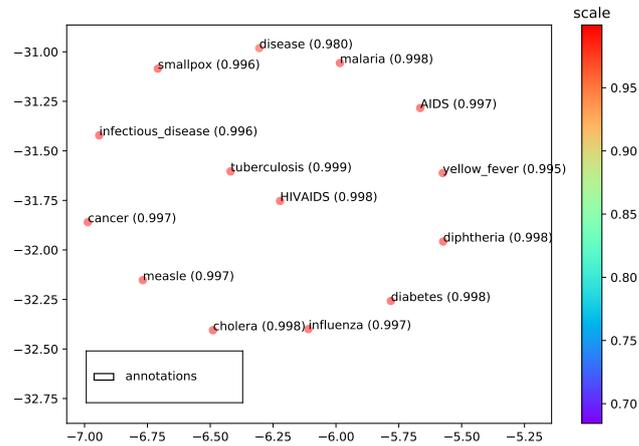


(a) Cluster of words about *fatigue symptoms*.

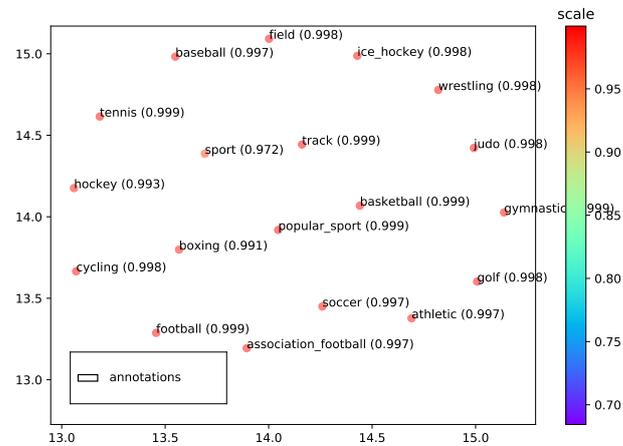
Figure D.1: Examples of clusters of the Poincaré word embeddings found in the *T-SNE* representation.



(b) Cluster of words about *music and movie genres*.

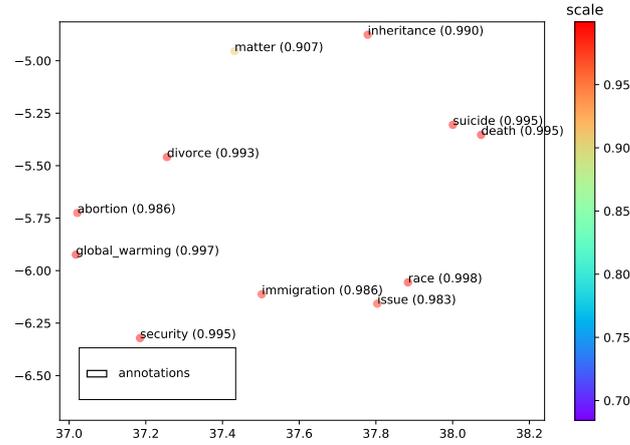


(c) Cluster of words about *diseases*.

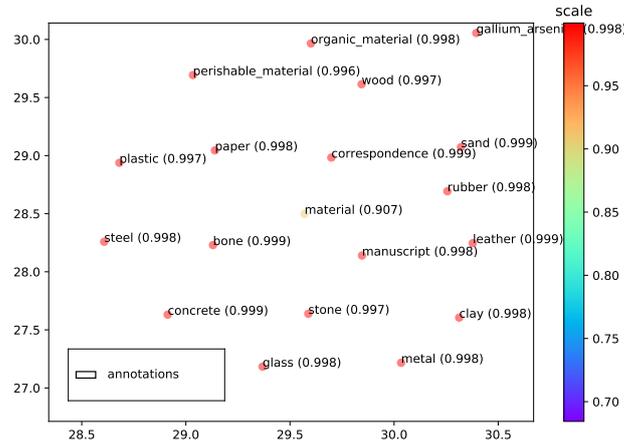


(d) Cluster of words about *sports*.

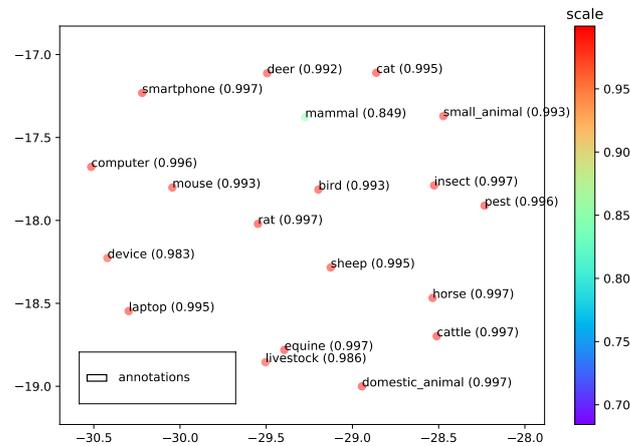
Figure D.1: Examples of clusters of the Poincaré word embeddings found in the T -SNE representation.



(e) Cluster of words about *social matters*.



(f) Cluster of words about *materials*.



(g) Clusters of words about *mammals* and *electronic devices*. The embedding of the word *mouse* is half-way between those two clusters.

Figure D.1: Examples of clusters of the Poincaré word embeddings found in the *T-SNE* representation.

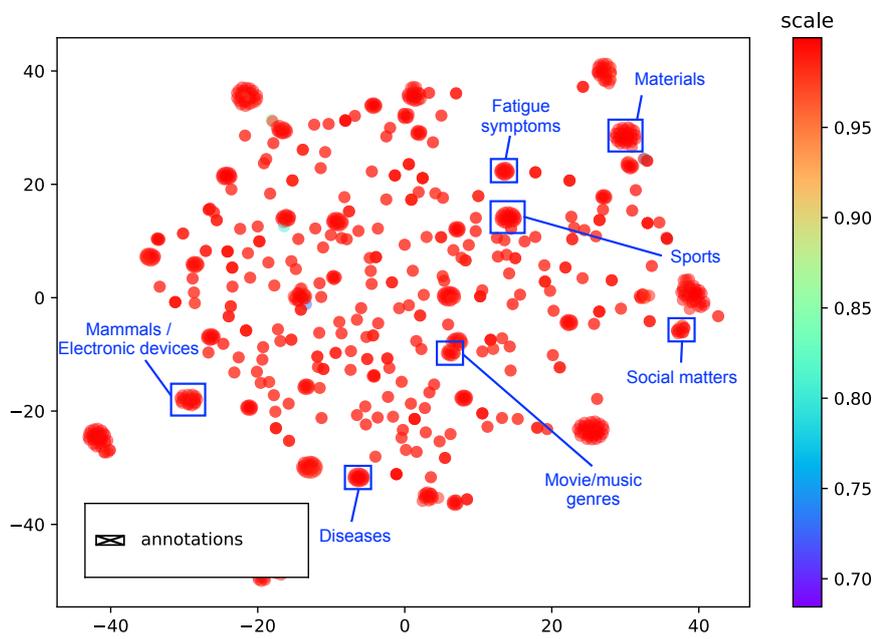


Figure D.2: Clusters exemplified in Figure D.1

Acronyms

BOW Bag of Words. 18

CBOW Continuous Bag of Words. 5

CRF Chinese Restaurant Franchise. 15

CRP Chinese Restaurant Process. 10

DIH Distributional Inclusion Hypothesis. 21

DP Dirichlet Process. 2

ELBO Evidence Lower Bound. 14

HDP Hierarchical Dirichlet Process. 1

hLDA hierarchical Latent Dirichlet Allocation. 2

IDF Inverse Document Frequency. 19

KL Kullback-Leibler. 14

LDA Latent Dirichlet Allocation. 1

MAP Mean Average Precision. 26

MCMC Markov Chain Monte Carlo. 13

nCRP nested Chinese Restaurant Process. 2

nHDP nested Hierarchical Dirichlet Process. 1

NLP Natural Language Processing. 1

NN Neural Network. 5

NPMI Normalized Point-wise Mutual Information. 22

PMI Point-wise Mutual Information. 20

POS Part-of-Speech. 18

RSGD Riemannian Stochastic Gradient Descent. 6

SG Skip-gram. 5

SVI Stochastic Variational Inference. 16

T-SNE T-Distributed Stochastic Neighbor Embedding. 43