La classification non supervisée et le *topic modeling* en Python

Paul Robert

21/05/2020

Table des matières

Introduction

Dans un article publié sur le site *Debates in the Digital Humanities* intitulé « Do digital humanists need to understand algorithms ? », Benjamin M. Schmidt écrit :

> Past a certain points, humanists certainly do not need to understand the algorithms that produce results they use. [...] But although there are elements to software we can safely ignore, some basic standarts of understanding remain necessary to practicing humanities data analysis as a scholarly activity and not merely a technical one.[11]

Pour obtenir un résultat intéressant et utile, il est nécessaire de comprendre le fonctionnement pratique des algorithmes utilisés et de pouvoir ainsi avoir du recul sur le résultat obtenu. Cela permet d'être critique à l'égard de ses résultats, et de comprendre là où on échoue. Je vais donc à mon tour me pencher sur la pertinence de l'utilisation de la classification non supervisée et du *topic modeling* pour l'étude des comptes-rendus des colloques TEI des années 2012, 2016 et 2019.

Nous pouvons d'ors et déjà étudier l'exemple de Netflix pour anticiper l'idée que l'algorithme de *topic modeling* utilisé pour la classification non supervisée ne pourra pas tout classifier sans erreurs et sans travail humain. En 2006, Netflix lance une plateforme : Netflix Prize[4]. L'idée était de proposer un prix d'un million d'euros pour récompenser la personne capable de trouver un algorithme de prédiction qui filtrerait les recommandations de film pour chaque personne. L'algorithme devrait étiqueter chacun des films et des séries de la plateforme dans les moindres détails pour ainsi offrir une expérience personnalisée à tous les utilisateurs selon leurs centres d'intérêts. En 2009, le site ferme. Il fut impossible de développer un tel algorithme, et Netflix continue d'étiqueter ses contenus par des utilisateurs sans automatisation. La classification non supervisée n'offre donc que difficilement un résultat absolument probant, si nous souhaitons obtenir un résultat prêt à l'emploi et digne du travail d'un humain.

Dans un entretien avec Giuseppe Granieri, Ted Striphas avance l'idée que nous vivons dans une « Algorithmic culture[6] », dans le fantasme d'un algorithme objectif qui résolverait tous les problèmes de manière automatique et insensible. Cependant, tout comme nous pouvons le voir dans une série de clichés du photographe Michael Wolf[13], même si des objets tels que des jouets sont créés par des machines et des algorithmes industriels, il faut toujours le travail minutieux d'être humains pour rendre le produit fini. Il en va de même pour la classification non supervisée : le calcul est effectué par la machine, mais le paramétrage et l'analyse du résultat restent le fruit du travail d'un être humain. Et c'est du paramétrage autant que de l'analyse critique finale que peuvent provenir les erreurs. Nous allons donc critiquer les différentes étapes de la préparation du corpus, puis les configurations de l'algorithme et enfin voir comment il est possible de comprendre ses erreurs en étudiant le résultat proposé par l'algorithme.

Chapitre 1

L'importance du preprocessing dans la classification non supervisée

1.1 Comment nettoyer son corpus ?

Pour obtenir un résultat sensé et de valeur, il faut préparer son corpus. Cette étape est automatisable via différents algorithmes [1] de traitement naturel du langage (NLP). Ces algorithmes, basés sur l'intelligence artificielle, « comprennent » le langage humain, et peuvent ainsi le modifier. Cette étape préparatoire est obligatoire pour que l'ordinateur puisse compter deux conjugaisons différentes d'un même verbe comme un seul et même mot. Elle permet aussi d'harmoniser et de nettoyer les textes d'un corpus.

Cependant, le choix de cette étape est humain. Doit-on garder tout le texte neuf ? Les mots grammaticaux pollueraient l'analyse à cause de leur omniprésence et de leur manque de poids sémantique. Doit-on uniquement retirer les *stop words* ? Quels mots sont des *stop words* à retirer ? Et doit-on lemmatiser les mots, c'est à dire ramener tous les mots à une forme canonique, le lemme ? Et quel lemme prendre ? Ne vaudrait-il pas mieux prendre un stemme, c'est à dire la racine tronquée de chaque mot ? Il serait même possible de ne garder que les noms communs de chaque texte du corpus, car ce sont eux qui portent véritablement le thème d'un texte.

J'ai personnellement opté pour une lemmatisation, car les lemmes sont peu contestés en anglais, et mon corpus est trop petit pour me permettre de

1. J'ai utilisé SpaCy pour lemmatiser et NLTK via PorterStemmer pour stemmer mon corpus.

le rogner.

Cette étape fragilise la pratique de la classification non supervisée en tant que méthode scientifique incontestable, car il n'existe aucun consensus universitaire sur ses pratiques orthodoxes. Néanmoins, elle reste obligatoire, et la suppression des *stop words* reste le minimum.

1.2 Quelle valeur donner aux mots du corpus ?

Tout d'abord, il faut savoir comment définir le texte. Est-ce le texte en lui-même sans retouches ? Doit-on y ajouter le titre, qui est généralement la partie du livre la plus travaillée par l'auteur ? Doit-on ajouter, dans le cadre d'un travail scientifique, les notes de bas de page, souvent lieu de commentaires pertinents ? La bibliographie doit-elle être comptée dans l'ouvrage, car une bibliographie définit et caractérise un ouvrage de recherche dans un certain sens ?

J'ai personnellement choisi d'ajouter uniquement le titre, car son importance est essentielle à mes yeux pour comprendre une œuvre. Mais c'est un choix purement personnel très probablement peu partagé par la communauté des chercheurs et des ingénieurs d'étude. Ici encore, ce choix humain et subjectif oriente le résultat final, notamment à cause du fonctionnement de l'algorithme.

La valeur des mots est donnée par leur fréquence dans le corpus, ou plus précisément par la valeur de la fréquence pondérée du TF-IDF [2]. Cette méthode permet de donner un poids plus important aux termes les moins fréquents dans tout le corpus. En effet, le discriminant qui permet de distinguer un texte des autres textes du corpus est la fréquence de certains mots rares. La répartition et la fréquence des mots dans un texte suit la loi de Zipf selon laquelle quelques mots sont utilisés très fréquemment par tous les textes, mais que de nombreux mots sont utilisés uniquement par un très petit nombre de textes. Et c'est la présence de ces mots qui permet donc de rapprocher deux textes.

2. C'est une valeur calculée à partir de la fréquence brute du terme dans le texte divisé par le nombre de textes où ce mot apparaît. Ainsi, le mot a une grande valeur s'il est répété très souvent dans un texte, et qu'il n'apparaît que dans ce texte.

1.3 Quel sens peut prendre un mot dans le cadre du *topic modeling* ?

Le *topic modeling* consiste à caractériser un texte par sa représentativité d'un nombre prédéfini de thèmes [3], valeur calculée à partir des mots qui composent le texte. Le mot est donc l'unité fondamentale et essentielle en *topic modeling*, et son identité doit être le coeur de chaque étude ainsi que l'écrit Benjamin M. Schmidt :

> Humanists need to ground the analysis of topic models in the words they are built from.[12]

Ce mode de calcul en apparence objectif est en réalité trompeur. Tout d'abord, la valeur sémantique d'un mot diffère au cours des époques, entre le genre et l'âge de chaque auteur, et d'une multitude d'autres facteurs. Mon corpus étant le fruit d'une multitude d'auteurs différents très souvent anglophones non natifs, le choix d'un mot ne reflète donc pas vraiment une valeur sémantique globale. Par ailleurs, aucun mot ne reflète parfaitement une idée. Le choix d'un thème est donc partiellement aléatoire.

Le plus gros facteur ayant créé des erreurs dans mon analyse reste la petite taille de mon corpus. Selon Benjamin M. Schmidt :

> Statistical noise may overwhelm any signal for a smaller corpus of merely a few thousand documents.[12]

Or, mon corpus ne dépasse jamais les 70 textes [4]. En parallèle, Benjamin M. Schmidt revendique le besoin d'au moins un milliard de mots dans le corpus pour que les mots les plus rares prennent leur vraie valeur[12], conformément à la loi de Zipf. Mes corpus, même additionnés, sont bien loin d'une telle valeur. Il n'y a donc pas assez de documents par corpus pour avoir une valeur de TF-IDF vraiment représentative, et la trop petite taille de chaque document empêche d'avoir une valeur d'appartenance à chaque thème fiable. Il en découle qu'à chaque lancement de l'algorithme, les documents appartiennent à un thème différent. La classification non supervisée par *topic modeling* n'est donc pas adaptée aux petits corpus.

3. Voir la définition de la LDA en 2.1.
4. Le corpus de l'année 2016, le plus imposant, contient 63 documents.

Chapitre 2

Le fonctionnement de l'algorithme et l'importance de son paramétrage

2.1 L'objectif initial de la création des algorithmes de *topic modeling*

Lorsque David Blei a créé l'algorithme de LDA pour le *topic modeling* au début des années 2000, il a produit cette algorithme pour de l'« information retrieval[2] ». La LDA devait servir initialement à étiqueter des masses documentaires afin de les trier avec des mots-clés. Le sens intrinsèque de chaque thème ou *topic* n'avait pas grande importance, du moment que le groupe soit cohérent sur une base de logique thématique.

La LDA, ou Allocation de Dirichlet latente, est un modèle probabiliste utilisé en *topic modeling* partant du postulat que chaque texte est un petit ensemble fini de thèmes prédéfinis, ainsi que le dit Matthew Jockers :

> We assume that documents are constructed out of some finite set of available topics.[8]

Ainsi, l'algorithme analyse chaque mot comme ayant une probabilité plus ou moins grande d'appartenir à chacun des thèmes. C'est la somme de proportion des thèmes de chacun de ces mots qui définit la proportion des thèmes de chaque document. La LDA se base sur la distribution d'un mot dans un texte pour retrouver les thèmes : ce n'est pas une propriété qui identifie un mot, ce n'est donc pas surprenant que la LDA soit utilisée également pour classer des images[5] ou de la musique[7]. Ce manque de prise en compte de la

particularité de la sémantique des mots et de son évolution dans l'algorithme est une source d'erreur.

L'autre algorithme utilisé dans mon étude est l'algorithme de Gensim, fonctionnant sur une base différente. Gensim[10] est la contraction de deux termes : « Generate similarity ». Chaque terme est vectorisé, c'est à dire représenté dans un plan dans l'espace [1]. La valeur du vecteur est fondée sur sa fréquence. Pour regrouper les mots puis les documents dans des clusters, on utilise le modèle de partitionnement des k-moyennes [2] : des points, les centroids, sont aléatoirement [3] disposés dans le plan, et les points les plus proches de chaque centroid sont regroupés dans un cluster. Les clusters permettent ainsi de réunir les documents composés de ces mots grâce à la similarité de fréquence d'occurrence de certains termes. La part d'aléatoire, amplifiée à cause de la trop petite taille de mon corpus, finit par rendre les résultats donnés par l'algorithme de Gensim inintelligibles et surtout inconstants. Le résultat est lui-même aléatoire, en définitive.

J'utilise ces deux algorithmes : l'algorithme de Gensim est le plus simple et le plus rapide à utiliser, cependant l'algorithme de LDA Mallet est plus complet et plus pertinent [4].

2.2 Les axiomes de la LDA

La LDA étant l'algorithme qui m'a offert les résultats les plus concluants, je vais l'étudier un peu plus en détail.

En *topic modeling*, deux hypothèses fondamentales guident la justesse des résultats. La première : un thème doit être cohérent. Chaque topic est un set de mots qui sont sensés avoir du sens entre eux. Or, il apparaît que mes thèmes, outre le fait qu'ils changent presque à chaque fois, ont des mots qui n'ont aucun sens les uns envers les autres. Et il semble que la LDA ne donne presque jamais des thèmes plein de sens. Benjamin Schmidt écrit :

> When humanists examine the output from Mallet [...], they need to be aware of the ways that topics may not be as coherent as they assume.[12]

La cohérence d'un thème restera donc, au mieux, vague.

1. Via l'algorithme Word2Vec développé par Google et implémenté dans Gensim qui représente chaque mot par un vecteur unique dans un plan en 2D.
2. Appelé *K-Means* en anglais.
3. On peut choisir un placement aléatoire, comme je l'ai fait, ou un placement déterminé, voire une trajectoire prédéfini à partir d'un placement aléatoire.
4. Il semble aussi que l'algorithme de LDA Mallet est le plus utilisé dans les humanités numériques : *MALLET (the most widely used topic-modeling tool)[12]*

La deuxième hypothèse fondamentale est la stabilité d'un *topic*. Si un thème est présent dans deux documents, c'est que ces livres partagent un concept, un champ ou un discours commun qui a un sens. Or, la trop petite taille des documents de mon corpus ne permet pas de développer un discours commun et cohérent, si ce n'est celui du thème général, la TEI. Il est ainsi logique qu'il y ait régulièrement un thème regroupant la majorité des textes, et plusieurs thèmes regroupant guère plus de deux documents.

Mes thèmes ne sont donc ni cohérents ni stables, donc en apparence mon travail est un échec. Cependant, ainsi que l'écrit Benjamin Schmidt :

> New ways of reading the composition of topics are necessary.[12]

J'ai remarqué un résultat particulièrement pertinent à lire hors de la composition des topics : en calculant le nombre de thèmes optimaux, j'ai remarqué que cet optimum était identique au nombre de thèmes choisis par l'organisme de la TEI Conference. Et c'est d'autant plus intéressant qu'après avoir contacté une participante, l'écriture du *paper* n'est pas subordonnée à un thème. Il y a donc un lien entre certains textes, bien que ce lien ne soit pas donné par la machine et l'étude des mots composants chaque thème. La suite de l'étude serait donc à faire hors du cadre du travail de l'algorithme.

Chapitre 3

Comprendre ses erreurs en analysant les résultats

> « A poorly supervised machine learning algorithm is like a bad research assistant. It might produce some unexpected constellations that show flickers of deeper truths ; but it will also produce tedious, inexplicable, or misleading results. » [12]

3.1 Le piège de l'analyse des *Top 10 Keywords*

L'algorithme de LDA propose de donner les 10 mots-clés les plus représentatifs de chaque *topic* afin de trouver le sens d'un thème. Les comprendre relève de l'ésotérisme ou de la cléromancie : sachant qu'un thème, parfois complexe ou précis, ne peut être recoupé par un seul mot, ce n'est pas non plus en 10 mots que cela sera possible. Car leur faible nombre peut ne décrire qu'une partie du thème, et occulter l'intégralité de sa définition [1]. Il faut donc étudier l'intégralité des mots de chaque thème pour le comprendre.

1. Benjamin Schmidt a montré par l'analyse de registres de marine de pêche à la baleine du XIXème siècle que les 10 lieux les plus mentionnés de chaque trajet de pêche ne représentent absolument pas le trajet. En effet, les lieux les plus cités sont les lieux de mouillage et les lieux de vente, où le bateau reste le plus longtemps. Il en découle ainsi l'impression fausse que les bâteaux pêchaient devant les ports.

3.2 Le *topic modeling* doit répondre à un but précis

Pour encadrer l'utilisation des algorithmes dans la littérature et l'histoire, Stephen Ramsay évoque le besoin d'un « Algorithm criticism[9] ». Selon lui, l'utilisation de transformations algorithmiques permet de déployer de nouveaux points de vue et de nouveaux axes de recherche en littérature et en histoire. Cependant, il faut que l'utilisation ait un but, qu'il y ait un objectif à comparer. En l'occurence, mon objectif initial était de savoir si le découpage thématique des organisateurs des conférences répondait à un besoin pratique, avoir autant d'intervenants dans chaque demi-journée, ou s'il y a réellement des thématiques profondes qui relient ces conférences. Et mon analyse a terminé sur la conclusion qu'il y a bel et bien des thématiques réelles, et que les regroupements sont fondés. Cependant, il faut produire une autre analyse pour savoir si ces thématiques sont propres à l'univers TEI, ou à certaines universités et certains centres de recherche en dépit du sujet traité par l'intervenant. Le *topic modeling* ne permet pas de répondre à toutes les questions, mais seulement à celle que j'ai posé.

3.3 Quel sens donner aux mots obtenus ?

Il faut enfin étudier avec précision l'ensemble des mots qui composent un topic. Benjamin Schmidt écrit :

> Humanists using topic models need to be extensively and creatively checking the individuals words that constitute their topics to see how grounded their inferences are.[12]

Ce n'est cependant pas la quantité de mots relevants du thème qui importe, mais leur sens commun. Il ne faut pas hésiter à voir des thèmes minuscules comprenant à peine plus de deux documents dans un corpus, ce qui est par exemple le cas dans mon corpus de l'année 2016, et séparer un thème en deux pour qu'il soit homogène en dépit de sa taille.

Cependant, la trop petite taille des documents de mon corpus empêche de réunir assez de mots homogènes et de développer un véritable champ sémantique propre à un thème. Par ailleurs, des notes résumant une intervention orale restent difficilement utilisables, car elles n'ont pas pour but d'être redondantes les unes envers les autres, ni de réunir plusieurs thématiques, notamment car ce sont de courtes interventions.

Conclusion

Les algorithmes de *clustering* et de *topic modeling* ont connu leur heure de gloire avec le développement du « Distant reading » de Franco Moretti, c'est à dire l'analyse d'immenses corpus littéraires pour relever des évolutions ou des regroupements sous-jaçants qui sont difficilement identifiables par un être humain. Cependant, force a été de constater que les algorithmes ne pouvaient donner la réponse à tout, et que non seulement la particularité du mot devait être pris en compte, mais qu'il était également compliqué de voir où l'algorithme avait échoué. En effet, celui-ci donnera toujours un résultat. Personnellement, je m'en suis rendu compte lorsque les topics changeaient systématiquement à chaque lancement de l'algorithme. A partir de là, il me fut plus intéressant d'étudier les causes mon erreur que mon résultat propre. Et la raison principale en est la trop petite taille, aussi bien en nombre de documents par corpus qu'en nombre de mots par documents. Ainsi, la part d'aléatoire sensée équilibrer l'algorithme devient prépondérante, et mes *topics* deviennent simplement aléatoires. Néanmoins, un résultat intéressant a émergé : le nombre de thèmes créé par les organisateurs des conférences TEI est cohérent. L'algorithme ne peut cependant dire pourquoi.

Néanmoins, il faudrait peut-être retenir que le plus grand apport des algorithmes de *topic modeling* n'est pas la révélation d'axes de lecture sur des immenses corpus littéraires, mais plutôt les nombreux débats qui naissent autour de chaque résultat, tel que celui entre Annie Swafford et Matt Jockers à l'occasion de la création du package Syuzhet. Il émerge par la critique l'idée de développer des pratiques communes, qui manque par exemple ici pour le *preprocessing*, et ces débats permettent de créer ces normes qui feront des Humanités Numériques une science définie par un ensemble de pratiques reconnues et admises par ses membres.

Bibliographie

[1] Benjamin Bengfort, Rebecca Bilbro et Tony Ojeda. *Applied text analysis with Python : enabling language-aware data products with machine learning*. First edition. OCLC : ocn962257016. Sebastopol, CA : O'Reilly Media, Inc, 2018. isbn : 978-1-4919-6304-3.

[2] David M Blei, Andrew Nq et Michael I Jordan. "Latent Dirichlet allocation, Journal of Machine Learning Research". en-US. In : (2003), p. 993-1022.

[3] Ian Bogost. *The Cathedral of Computation*. en-US. Library Catalog : www.theatlantic.com Section : Technology. Jan. 2015. url : https://www.theatlantic.com/technology/archive/2015/01/the-cathedral-of-computation/384300/ (visité le 21/05/2020).

[4] Netflix Company. *Netflix Prize : Home*. url : https://www.netflixprize.com/ (visité le 21/05/2020).

[5] Pradheep K Elango et Karthik Jayaraman. "Clustering Images Using the Latent Dirichlet Allocation Model". en. In : (Décembre 2005), p. 18.

[6] Giuseppe Granieri. *Algorithmic culture. Culture now has two audiences : people and machines*. en. Library Catalog : medium.com. Mai 2014. url : https://medium.com/futurists-views/algorithmic-culture-culture-now-has-two-audiences-people-and-machines-2bdaa404f643 (visité le 21/05/2020).

[7] Diane J Hu et Lawrence K Saul. "A Probabilistic Topic Model for Music Analysis". en. In : (), p. 4.

[8] Matthew Jockers. *ż The LDA Buffet is Now Open ; or, Latent Dirichlet Allocation for English Majors Matthew L. Jockers*. en-US. Library Catalog : www.matthewjockers.net. url : http://www.matthewjockers.net/2011/09/29/the-lda-buffet-is-now-open-or-latent-dirichlet-allocation-for-english-majors/ (visité le 22/05/2020).

[9] Stephen RAMSAY. *Reading machines : toward an algorithmic criticism*. Topics in the digital humanities. OCLC : ocn708761605. Urbana : University of Illinois Press, 2011. ISBN : 978-0-252-03641-5 978-0-252-07820-0.

[10] Radim EHEK et Petr SOJKA. "Software Framework for Topic Modelling with Large Corpora". English. In : *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. http://is.muni.cz/publication/884893/en. Valletta, Malta : ELRA, mai 2010, p. 45-50.

[11] Benjamin SCHMIDT. *48. Do Digital Humanists Need to Understand Algorithms ?* en-US. Library Catalog : dhdebates.gc.cuny.edu. URL : https://dhdebates.gc.cuny.edu/read/untitled/section/557c453b-4abb-48ce-8c38-a77e24d3f0bd (visité le 21/05/2020).

[12] Benjamin SCHMIDT. *ż Words Alone : Dismantling Topic Models in the Humanities Journal of Digital Humanities*. 2012. URL : http://journalofdigitalhumanities.org/2-1/words-alone-by-benjamin-m-schmidt/ (visité le 21/05/2020).

[13] Michael WOLF. *MICHAEL WOLF PHOTOGRAPHY*. URL : http://photomichaelwolf.com/#the-real-toy-story-factories/8 (visité le 21/05/2020).