FINAL MASTER PROJECT

Degree in Materials Engineering

TEXT CLASSIFICATION OF BIOMATERIALS ABSTRACTS AND INFORMATION EXTRACTION FROM THE 3D PRINTING LITERATURE FOR BIOMEDICAL APPLICATIONS USING MACHINE LEARNING ALGORITHMS

Author: Clarence LEPINE
Director: Maria Pau GINEBRA MOLINS
Co-Director: Osnat HAKIMI
Call: February 2021

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

Resumen

Desde el advenimiento de la era digital, la mayoría de los resultados de investigación se publican en línea, e Internet se ha convertido en la mayor fuente de conocimiento disponible en el mundo. Con el paso de los años, el número de publicaciones ha aumentado progresivamente y la literatura sobre biomateriales no es una excepción. Por este motivo, debido a su gran volumen y a su naturaleza heterogénea, recopilar y sintetizar conocimientos en este campo de forma manual es una tarea agotadora. Las técnicas de minería de textos, como por ejemplo la clasificación de textos, facilitan la organización y extracción de estos datos. De hecho, la clasificación de textos es una de las tareas básicas supervisadas en el procesamiento del lenguaje natural, que permite asignar etiquetas o categorías al texto según su contenido. Estas categorías se determinan a partir de un conjunto de documentos muestra usados para el entrenamiento ("training set"), que se clasifican manualmente de antemano. Para construir un modelo de clasificación, se utiliza un algoritmo de aprendizaje automático. Este algoritmo analiza los datos de entrenamiento para crear un modelo que puede predecir la clase de un nuevo documento procedente del mundo real. En este trabajo, se propone un procedimiento de clasificación de textos basado en resúmenes de artículos, ensayándolo en resúmenes de artículos biomédicos obtenidos de la base de datos Pubmed. El conjunto de artículos usado como entrenamiento se compone de 3224 resúmenes de artículos de biomateriales y el conjunto de análisis se compone de 477 resúmenes de artículos sobre impresión 3D. Los objetivos del proyecto son dos: (1) comparar varios modelos para la clasificación de documentos de biomateriales y (2) obtener una visión general del dominio de la impresión 3D en el ámbito biomédico a través de la clasificación de textos y la extracción de datos de una gran cantidad de publicaciones de investigación. Nuestro clasificador se implementó utilizando el lenguaje de programación Python. Después de probar varios modelos de clasificación, incluidos modelos binarios y multinomiales, logramos lograr 0.92 de precisión y 0.89 de F1-score con Stochastic Gradient Descent (SGD) con el modelo multinomial en comparación con el tratamiento manual. Este modelo se utilizó para clasificar toda la literatura de impresión 3D disponible en Pubmed hasta la fecha, un total de 11.153 artículos. Posteriormente, se utilizaron técnicas de extracción de datos para extraer información sobre la tecnología de impresión 3D para aplicaciones biomédicas. Esta información permitió identificar 4 aspectos relevantes: (1) enfermedades que se pueden tratar con esta tecnología, (2) tejidos u órganos que pueden reemplazarse por implantes impresos en 3D, (3) biomateriales más utilizados y (4) nuevas aplicaciones no directamente relacionadas con prótesis o implantes.

Resum

Des de l’aparició de l’era digital, la majoria dels resultats de la investigació es publiquen en línia, i Internet s’ha convertit en la font de coneixement més gran del món. Amb el pas dels anys, el nombre de publicacions ha augmentat progressivament i la literatura sobre biomaterials no n’és una excepció. Precisament a causa del seu alt volum i la seva naturalesa heterogènia, recopilar i sintetitzar coneixements en aquest camp de forma manual és una tasca esgotadora. Les tècniques de mineria de textos, com per exemple la classificació de textos, faciliten l’organització i l’extracció d’aquestes dades. De fet, la classificació de textos és una de les tasques supervisades bàsiques en el processament del llenguatge natural, que permet assignar etiquetes o categories al text segons el seu contingut. Aquestes categories es determinen a partir d’un conjunt de documents mostra utilitzats com a entrenament (“training set”), que es classifiquen manualment prèviament. Per construir un model de classificació, s’utilitza un algorisme d’aprenentatge automàtic. Aquest algorisme analitza les dades d’entrenament (“training set”) per crear un model que pugui predir la classe d’un nou document procedent del món real. En aquest treball, es proposa un procediment de classificació de textos basat en resums d'articles, assajant-lo en resums d’articles biomèdics obtinguts de la base de dades Pubmed. El conjunt d’entrenament es compon de 3224 resums d’articles de biomaterials, i el conjunt d’estudi es compon de 477 resums d’articles sobre impressió 3D. Els objectius del projecte són dos: (1) comparar diversos models de classificació de documents de biomaterials i (2) obtenir una visió general del domini de la impressió 3D en l’àmbit biomèdic mitjançant la classificació de texts i l’extracció de dades d’un gran nombre de publicacions de recerca. El nostre classificador es va implementar utilitzant el llenguatge de programació Python. Després de provar diversos models de classificació, inclosos els models binaris i multinomials, es va aconseguir un 0,92 de precisió i un 0,89 de la puntuació F1 amb descens de gradient estocàstic (SGD) en comparació amb el tractament manual. Aquest model es va utilitzar per classificar tota la literatura d’impressió en 3D disponible de Pubmed fins a la data, que representa un total de 11,153 articles. Després es van utilitzar tècniques d'extracció de dades per recuperar informació sobre la tecnologia d’impressió 3D per a aplicacions biomèdiques. Després es van utilitzar tècniques d’extracció de dades per recuperar informació sobre la tecnologia d’impressió 3D per a aplicacions biomèdiques. Aquesta informació va permetre identificar 4 aspectes rellevants: (1) malalties que es poden tractar mitjançant aquesta tecnologia, (2) teixits o organs que es poden substituir per implants impresos en 3D, (3) biomaterials més utilitzats i (4) altres aplicacions no relacionades directament amb pròtesis o implants.

Abstract

Since the advent of the digital age, the majority of research findings are published online, and the Internet has become the largest source of knowledge available in the world. As the years passed, the number of publications has steadily increased and the biomaterials literature is no exception. But because of its high volume and heterogeneous nature, collecting and synthesizing knowledge in this field is a gruelling manual task. Text Mining (TM) techniques such as Text Classification (TC) can facilitate the organization and the extraction of these data. Indeed text classification is one of the fundamental supervised tasks in natural language processing that allows for assigning tags or categories to text according to its content. These categories are determined by a set of training documents that were manually classified beforehand. In order to build a classification model, a machine learning algorithm is used. This algorithm analyses the training data to create a model that can predict the class of a new unseen document (that comes from the real world). In this work, a text classification approach based on article abstracts will be proposed and tested with biomedical abstracts retrieved from Pubmed. The training set is composed of 3224 biomaterials abstracts covering a broad range of topics, and the testing set is composed of 477 abstracts about 3D-printing. The objectives of the project are two fold: (1) to compare various models for biomaterials document classification and (2) to get an overview of the 3D-printing domain in the biomedical field through text classification and data extraction (DE) of a large number of research publications. Our classifier was implemented using Python programming language. After testing several classification models, including binary and multinomial models, we manage to achieve 0.92 of accuracy and 0.89 of F1-score with Stochastic Gradient Descent (SGD) with the multinomial model compared to the manual curation. This model was used to classify the entire 3D-printing literature available from Pubmed to date, a total of 11,153 articles. DE techniques were then used to retrieve information about the 3D-printing technology for biomedical applications. These informations permitted to identify 4 main aspects: (1) diseases that can be treated using this technology, (2) tissues or organs that can be replaced by 3D-printed implants, (3) most commonly used biomaterials and (4) new applications which do not deal with prosthetics or implants.

Acknowledgements

First of all, I would like to thank Professor Maria-Pau Ginebra for giving me the opportunity to carry out this project from France. In fact, given the health situation with COVID 19 in Spring 2020, she has to find me another Master Thesis Project that I could do entirely from home.

I would also like to thank Doctor Osnat Hakimi who proposed to me this new project and who introduced me to Python. She was present throughout the all project to answer my slightest doubt and guide me. I am very grateful to her for that.

Finally I thank my colleague José Masache Cevallos who has been a great support throughout this project.

Glossary

AI: Artificial Intelligence

ATC: Automated Text Classification

BOW: Bag of Words

CAD: Computer Aided Design

CAM: Computer Aided Manufacturing

CT: Computed Tomography

CTD: Comparative Toxicogenomics Database

DE: Data Extraction

DEBBIE: Database of Biomaterials and their Biological Effect

hLDA: Hierarchical Latent Dirichlet Allocation

k-NN: k-Nearest-Neighbors

LDA: Latent Dirichlet Allocation

MeSH: Medical Subject Headings

ML : Machine Learning

NB: Naive Bayes

NCBI: National Center for Biotechnology Information

NLM: National Library of Medicine

NLP: Natural Language Processing

PMID: PubMed Identification Number

RefSeqs: Reference Sequences

SGD: Stochastic Gradient Descent

SL: Supervised learning

SNP: Single Nucleotide Polymorphism

TC: Text Classification

TF-IDF: Term Frequency - Inverse Document Frequency

TM: Text Mining

USL: Unsupervised learning

Contents

1 Introduction

Nowadays the field of biomaterials is growing rapidly, and especially the area of 3D-printing and bioprinting. The 3D-printing of biomaterials attracts more and more attention in healthcare and especially in tissue engineering [1]. The majority of the knowledge in this field can be found in scientific articles published on platforms such as Pubmed, Scopus and Web of Science. In October 2020, a search on Pubmed [2] with the query (((3d printing) OR (3d-printing) OR (three dimensional printing) OR (bioprinting)) NOT ((review)[Publication Type])) NOT ((systematic review)[Publication Type]) returns 10,997 articles (from 1968 to 2020). Among them, 10,805 articles were published within the last 10 years (see Figure 1).

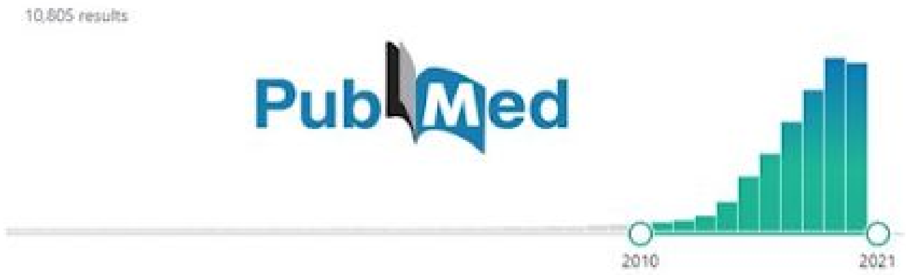


Figure 1 : Distribution of the 3D-printing articles published the last 10 years on Pubmed

But for the moment, there is no open readable data repository and as the number of research publications grow, it is becoming harder and harder to make any exhaustive synthesis or state-of-the-art on the subject. [3] A solution to this problem could be the use of Artificial Intelligence (AI) tools to help and support better data retrieval, extraction and indexing. Natural Language Processing (NLP) is a subfield within AI and Machine Learning (ML) that combines linguistics and computer science to break down language, so it can be analyzed by machines [4]. One of the common tasks in NLP is Text Classification (TC). Text classification is the process of assigning one or more predefined tags or categories to text according to its content [5]. In this paper, we will perform TC of articles' abstracts. We chose to work only with abstracts because analysing full articles has several obstacles, including availability, format conversion and high memory requirement for processing. Moreover, the abstract should reflect reasonably well the main content of the document.

This project is related to the DEBBIE project started in 2018 by a group of researchers from the Universitat Politècnica de Catalunya and the Barcelona Supercomputing Center. DEBBIE uses text classification as part of an automated pipeline for data extraction. The project aims to create an open-access database for biomaterials, implants and medical devices and facilitate a more efficient access to the large literature in the field. As part of the pipeline development, 250 representative biomaterials articles were manually curated from Pubmed [6] . These articles were identified using the following MeSH terms and keywords 'Biomaterials', 'Cell scaffolds', 'Biomedical and dental materials', 'Prostheses and implants', 'Materials testing', 'Tissue engineering', 'Tissue scaffolds', 'Equipment safety' and 'Medical device recalls' in PubMed. They composed the preliminary gold standard set (which contained only biomaterials articles). (The final gold standard set will be used to train a classifier to recognize relevant articles that will then enter in the database about biomaterials). To extend this gold standard set, the PubMed Identification Numbers (PMIDs) of the first 250 articles retrieved were used to train the MedlineRanker classifier[7]. All of the abstracts published in the last 10 years were ranked according to their similarity to the gold standard set. The top 1000 ranked records were manually scanned to remove reviews and added to the biomaterials set. The final biomaterials set contained over 2500 articles. Additionally, a background set has been created to enable classification and comparative analysis of the biomaterials literature against the general literature. This set of non biomaterials articles was created by generating a list of random PMIDs from the Pubmed database containing articles from 1999 to 2018. To make sure that the random set did not contain any biomaterials articles, it was also ranked in the MedlineRanker classifier, and the top 200 ranked records were manually scanned to remove the last biomaterials articles from the list [8].

Since the goal of DEBBIE is to annotate the biomaterials literature, a binary classifier was implemented to identify relevant abstracts. To do so, the SGD classifier from Scikit-learn library has been trained on the gold standard set against the background set. To evaluate the performance of the DEBBIE classifier, a manually constructed test set made of the literature on polydioxanone was used. The trained classifier achieved 0.9011 precision, 0.9623 recall, and 0.9307 F1-score compared to the manual curator (Fuenteslopes et al, unpublished data). But after analysing the articles that were not well classified, it was found that mostly clinical studies were not recognized as biomaterials publications, even if it was the case. Suggesting a bias of the binary classifier to laboratory studies.

This brings us to the objectives of this Master Thesis project which are, first, to test the DEBBIE binary classifier on a new test set regarding 3D-printing to see if we encounter the same results as those obtained with the polydioxanone set (i.e. the classifier considers clinical studies as non relevant even if they talk about biomaterials), then try to implement a multinomial classifier that can identify clinical studies from non biomaterials studies and finally to classify the 3D-printing literature available from Pubmed and extract informations from it.

1.1. Objectives of the project

The objectives of the project are the following:

- Create a new 3D-printing testing set using manual curation from Pubmed in order to test the ability of various classifiers
- Test the binary classifier from DEBBIE project with the 3D-printing set
- Create a multinomial classifier that can classify Pubmed articles in 3 categories: non biomaterials studies, in vivo/in vitro studies and clinical studies
- Test the performance of the classifier using the 3D-printing set
- Classify the entire 3D-printing literature available from Pubmed
- Do text analysis to extract informations about the 3D-printing literature:
    1) Extract the most frequent terms from each category
    2) Semantic analysis using Pubtator annotations
    3) Topic Mining (using Latent Dirichlet Allocation and Hierarchical Latent Dirichlet Allocation)

1.2. Project justification

In this section we are going to explain why text classification is an essential step in data extraction from the scientific literature, how it can be used by researchers to improve access to information and why we should extract information about the 3D-printing of biomaterials in particular.

Nowadays most of the knowledge in the biomaterials field is available online and is presented in a narrative form. As such, it includes many ambiguous terms and typographical errors. Looking at the 3D-printing literature in particular, a lot of terms are used to refer to this technology, such as '3D printing', '3D-printing', 'three dimensional printing', 'bioprinting' or 'additive manufacturing'. Indeed, the scientific literature presents heterogenous use of the terminology of a specific word or concept. Often, authors use different terms to refer to the same concept in their articles. These differences of terminology make it difficult to organize the literature. TM techniques such as TC can facilitate the access to this wealth of information. It is useful for automatically sorting relevant and non-relevant documents, as well as different categories within a group of documents (e.g. clinical studies on patients, in vitro studies in laboratory, and in vivo studies using animal models). Moreover, by combining TC with text analysis, it is possible to extract the most frequently used words in each type of study and discover the keywords that we need to enter in our search bar if we want to find similar articles on the internet. If we take the case of a scientist who wants to start working in the 3D-printing field for biomedical applications, he might begin by looking at the publications available online to get an idea of what has already been done and what still needs to be done. But if the authors use different terminologies from those the scientist is looking for, what can happen is that he might miss some publications of interest just because they did not have the right spelling. In order to get an overview of a domain, it will be useful to have a tool that can handle language variations and that can analyse very quickly a large amount of data. So we can say that because of the proliferation of research publications online and the heterogenous form of the literature, TC is becoming a crucial task to help researchers in their work.

But why should we help researchers extract information about the 3D-printing domain? To answer this question, we will first define what 3D-printing is. 3D-printing is a technology that was invented in the early 80s by Dr Kodama, a Japanese scientist [9]. At that time, the technique was referred to as rapid prototyping because Kodama wanted to create a technique for rapidly fabricating a large amount of prototypes. It is not until the 90s that the term 3D-printing appeared (in the United States) and described a layer by layer process that allows the fabrication of three dimensional products. Typically the process works by adding successive layers of fused materials in a 2D plan. To build 3D-objects, the process relies on Computer-Aided-Design (CAD) [10]: the printer reproduces virtual 3D-models of the object. The term additive manufacturing is often used to describe the 3D-printing technology [10]. This term indicates how 3D-printing is opposed to traditional subtractive manufacturing techniques such as machining, casting, molding and forming. The advantage of this technique is that it does not require any assembly which means that the products are printed in one step. Moreover it allows to manufacture custom products with very complex shapes and to replicate objects with high fidelity. Due to these advantages, 3D-printing aroused growing interest from the 2000s when 3D-printers were democratized. Over the years, biomaterials started to be used in 3D-printing. Biomaterials refer to natural or synthetic materials interacting biologically with human fluids that can be used in the body to replace any organ or tissue without inducing adverse reactions. There are different types of biomaterials depending on their chemical nature: metals, ceramics, polymers and composites. The arrival of biomaterials in the 3D-printing field has enabled the development of many applications in healthcare: from pharmaceutics to the fabrication of implants and organ development [10]. However some of these applications are still under development, that is why more and more articles are published on this field.

It is essential to help research in this field because it can improve people’s lives. Indeed humans can benefit from this technology in many ways: manufacturing of prosthetics, implants, surgical guides, etc. But this could also reduce the mortality due to organs failing. Even today, many people die every year because they have not had access to organ donation. The future progress in organ 3D-printing could eradicate this problem. Researchers are working to develop the printing of functional human organs such as heart, kidney, etc. Moreover this technology can not only help humans but also animals. By printing a prosthetic leg for an animal, we could possibly save his life.

1.3. Vocabulary

The section contains some useful vocabulary related to the project that might facilitate understanding of the report.

Machine learning: An application of AI that allows computers to learn automatically from data without the need for prior programming. [11]

Supervised learning: A type of learning based on examples of input and output. A supervised machine learning algorithm requires training data with predefined outputs to make predictions.[12]

Unsupervised learning: A type of learning that does not require any example of input and output to make predictions. An unsupervised machine learning algorithm does not require training data.[12]

Classifier: A supervised machine learning algorithm that maps the input data to a specific category (also called label).

Classification model: A classification model is an algorithm that can draw some conclusions from the input data (i.e. the training set). In order to predict the class labels or tags for unseen data.

Feature: A feature is a term extracted from the text on which the classification model relies to make some predictions.

Binary classification: Classification task with two possible outputs : a document can either belong to class A or class B. A simple example is the gender classification (Male / Female). A person can either be a male or a female.

Multi-class or multinomial classification: Classification task with more than two classes. In multi-class classification, each document can be assigned to one unique label. You can imagine an animal can be either a rabbit or a horse but not both at the same time.

Multi-label classification: Classification task where each document can be assigned to a set of labels (with one or more classes). A good example concerns the news article. Indeed an article can talk about food, a person and location at the same time.

Training Set: The training set is used to build our model that aims to predict the test data. It is data of which we know the category beforehand.

Validation Set: The training set can be divided into a train set and validation set. Thanks to the results from the validation test, the model can be trained while changing the parameters and the classifiers in order to get the most optimized model.

Testing Set: The test set is used at the end to validate our model and get the performance scores.

1.4. Automated classification

Automated Text Classification (ATC) is the task of automatically classifying documents into a predefined set of classes or categories [5]. ATC helps to organize, structure and categorize large amounts of documents. It is often used to support systematic revues in the medical field. The classification task can be binary (i.e. with 2 classes) or multinomial (i.e. with more than 2 classes) as you can see in Figure 2.

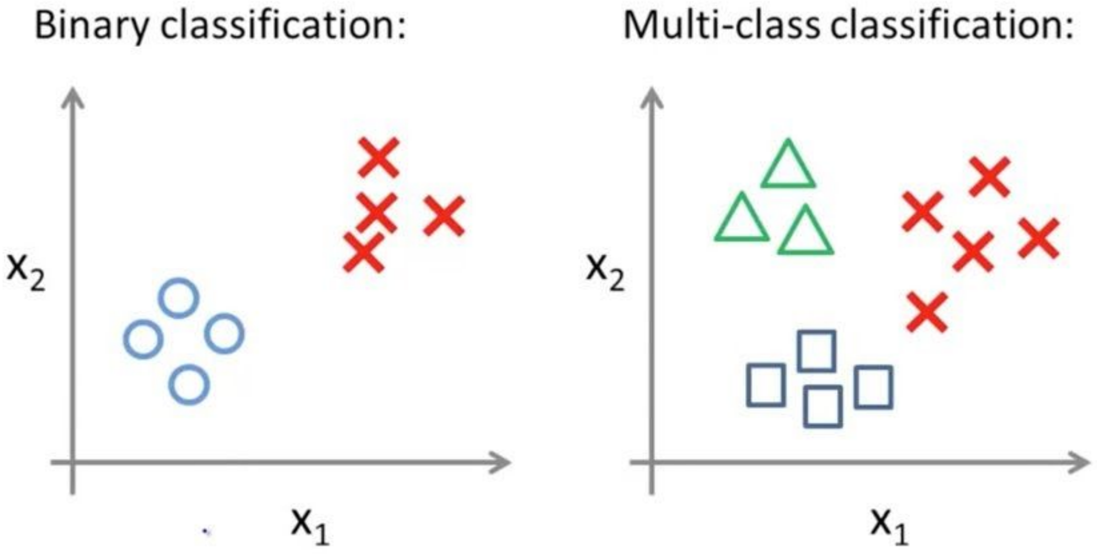


Figure 2 : Difference between binary and multinomial classification

In a binary classification, two outputs are possible and documents can be classified either in the first class or in the second class. A simple example of binary classification is the gender classification. A person can be classified as a male or a female.

In a multinomial classification or multi-class classification, there are necessarily more than 2 outputs. For example, we can imagine an animal which can be either a horse, a fish, a rabbit or a bird.

As you can see in binary and multinomial classification, an object can be assigned to one unique class. But another type of classification exists where an object can be classified into more than one category. This task is called multi-label classification. A good example of multi-label classification is the classification of news articles. In a newspaper, we can find articles that talk about food, a person and a location at the same time. In this project, we applied binary and multinomial classification.

ATC is performed using a Machine Learning (ML) approach. ML is an application of Artificial Intelligence that allows computers to learn automatically from data without the need for prior programming [11]. We can distinguish 2 types of ML: the Supervised Learning (SL) and the Unsupervised Learning (USL). SLA is a type of learning based on examples of input and output. In SL, the algorithm requires training data with predefined outputs to make predictions on new data. USL, on the other hand, does not require any example of input and output to make predictions. So in USL, the algorithm does not require training data. [12] ATC relies on supervised ML algorithms also called classification models to classify documents into categories. The categories of unclassified documents are predicted by the classification model using a corpus of manually pre-classified documents (i.e. the training set). The construction of a classification also requires a testing set which corresponds to pre-classified documents which will serve to evaluate the performance of the model. In brief, a classification model learns from a training set how to perform classification and is then tested using a separate, testing set to verify how accurate its predictions are. Once its results are satisfactory, new data can be fed into the algorithm to be classified (new data that has not been pre-classified beforehand).

1.5. Classification models

There are many classification models available for machine learning and each model has its own strengths and weaknesses. But before choosing one of them, it is essential to look at the data (What kind of data is it ? In what format ? Do we have a balanced dataset or not ?) and the type of problem we need to solve (Are we doing a binary classification, a multi-class classification, or a multi-label classification?). Depending on the answers to these questions, we will be able to choose the most suitable model for our project. In practise, we will not only work with one classification model but we will compare the performances of a few different ones as well as their computational efficiency. Because even if some models have very good performances they can take a lot of time to compile and that is not something we need.

In this project, we performed a binary classification and a multinomial (or multi-class) classification. For the binary classification, we will compare the performances of two classifiers: the Stochastic Gradient Descent (SGD) classifier used in the DEBBIE project and the Random Forest classifier that usually performs well on text classification. Then, for our multi-class classification, we compared 4 classifiers: Stochastic Gradient Descent (SGD), Random Forest, Multinomial Naive Bayes (NB) and k-Nearest-Neighbors (k-NN).

1.5.1 Multinomial Naive Bayes

Naive Bayes classifiers are linear classifiers that are known for being quite simple but very efficient for text classification. They use a probabilistic model based on Bayes' Theorem which describes the probability of a feature, based on previous observations. The adjective 'naive' comes from the assumption that the features in the dataset are mutually independent. In other words, it means that each word of a sentence is independent of the other ones. This is rarely true in practice. But in general NB classifiers still perform very well under this unrealistic assumption. Moreover, this allows the classifier to work on very small data which can be quite useful. NB classifiers offer other advantages: they are easy to implement, fast, robust, and pretty accurate. So they are used in many applications such as spam filtering in emails. [13]

As mentioned above, Bayes' Theorem is at the heart of the whole concept of NB classification and it can be written as follows:

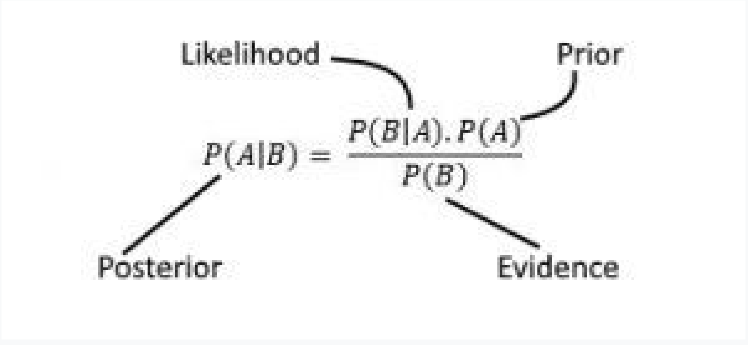


Figure 3 : Bayes' Theorem

where:

- A and B are two independent events. (In text classification, events represent either a document or a category.)
- P(A) is the prior probability of A occurring before the evidence is taken into account. It is like calculating the probability of an outcome based on the current knowledge but before an experiment has been performed.
- P(B) is the evidence probability of B occurring where B represents new data that is collected.

- P(B|A) is the likelihood probability of B occurring, given A. The likelihood quantifies to what extent the evidence B supports the proposition A.
- P(A|B) is the posterior probability of A occurring, given that B has occurred. It is the revised probability of A occurring after taking new data in consideration (i.e. after taking the evidence into account).

NB: P(A|B) and P(B|A) are conditional probabilities and P(A) and P(B) are marginal probabilities (because they are independent).

In brief the Bayes' Theorem relies on incorporating prior probability distributions in order to generate posterior probabilities. The posterior probability is calculated by updating the prior probability thanks to Bayes' Theorem. So we can revise existing predictions or update probabilities given new or additional information.

When we are doing text classification with this formula A represents a given class or category and B represents the data/document we are working on.

To understand how Multinomial NB makes some predictions on the classes, we will now apply the Bayes' Theorem to a binary classification problem with two classes C1 and C2 and a document D part of the training set.

$$P(C1|D) = P(D|C1) * P(C1) / P(D) \quad (1)$$

$$P(C2|D) = P(D|C2) * P(C2) / P(D) \quad (2)$$

As you can see the denominator remains the same in the calculus of the posterior probability for both classes (equations (1) and (2)). So we will not have to calculate it in practise.

This means that the category/class predicted for the document D is based on the highest probability given by arg max P(C|D) = arg max P(C) P(D|C), with C={C1,C2}.

So the document D will be assigned to class C1 if P(D|C1)*P(C1) > P(D|C2)*P(C2) and will be assigned to class C2, if P(D|C2)*P(C2) > P(D|C1)*P(C1).

1.5.2 Stochastic Gradient Descent

Originally Stochastic Gradient Descent (SGD) is the name of an optimization method and not a machine learning algorithm/model. But in the Scikit-learn library [14], we can find a model called SGDClassifier which can be a bit confusing. Indeed SGDClassifier is a linear classification model (Linear SVM or Logistic Regression for instance) that is optimized with the Stochastic Gradient Descent method. To understand what SGD is, we will first talk about Gradient Descent in general. Gradient Descent is an algorithm able to minimize functions. It means that when we give him a function defined by a set of parameters, Gradient Descent begins with an initial set of parameter values and makes iteration to move toward a set of parameter values that find the minimal point for the function (see Figure 4).

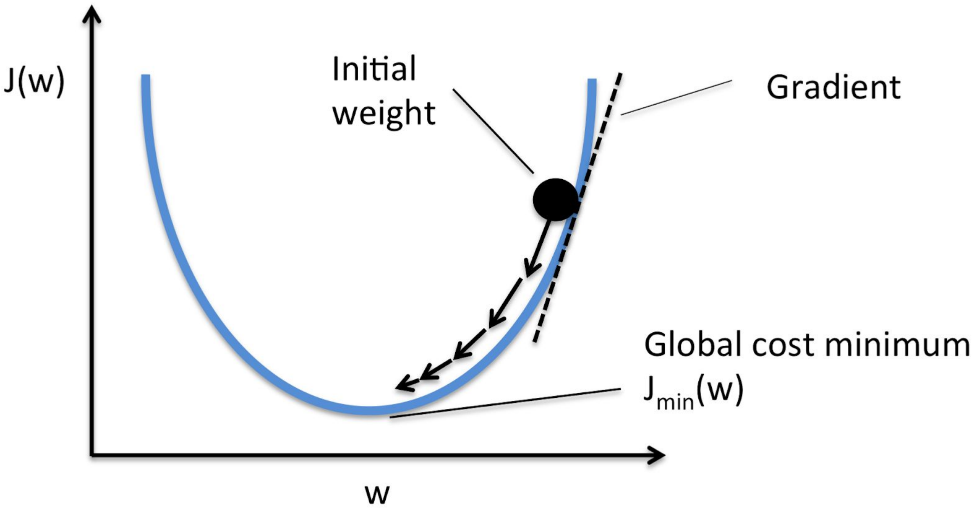


Figure 4 : Principle of operation of Gradient Descent algorithm (J is the function that we want to minimize also called the loss function and w is the weight parameter)

This minimization process involves calculating partial derivatives to get the gradient of the loss function with respect to the weights and find the line that the most approach the minima. A main disadvantage of Gradient Descent (GD) is that the algorithm can be slow while working with very large datasets. One iteration of the GD algorithm requires a prediction of each instance in the training dataset, it can take a long time when there are a lot of instances. That is why programmers usually use a variant of the algorithm known as Stochastic Gradient Descent (SGD) to make their model learn a lot faster. The main difference between SGD and GD is that in Gradient Descent, the whole training set is considered before taking one parameters update whereas in Stochastic Gradient Descent only one random data point or instance is considered for each parameters update step, cycling over the training set. [15]

1.5.3 Random Forest

Random Forest is an ensemble method which means it uses the average results from several decision trees to produce more accurate predictions than using a single decision tree [16]. A decision tree is a tree-like chart constituted of decision nodes and leaves. The leaves correspond to the final outcomes (or decisions) and the decision nodes are the data that is examined. The principle of the ensemble methods is that a group of weak learners (that perform the classification task alone) can come together to form a strong learner (with higher performance). [17] To perform ensemble decision trees two techniques can be used: bagging and boosting. Here we will only explain what bagging is because the boosting technique does not apply to Random Forest.

Bagging is used when we want to reduce the variance of a decision tree. The idea is to create several subsets of data from the training set chosen randomly. Each subset of data will be used to train its own decision tree. At the end, we obtain an ensemble of models with their own predictions. Then we use the average of all the predictions from different trees to get our final prediction. This is more robust than using a single decision tree. [17]

Random Forest does not only use the bagging technique but it goes a little further. Indeed it takes one extra step where in addition to taking a random subset of training data, it also takes random selection of features rather than using all features to grow trees. So at the end, there are many random trees which are called Random Forest.

The operation principle of Random Forest is described in Figure 5. As we can see, the final prediction is given based on the aggregation of predictions from all of the decision trees.

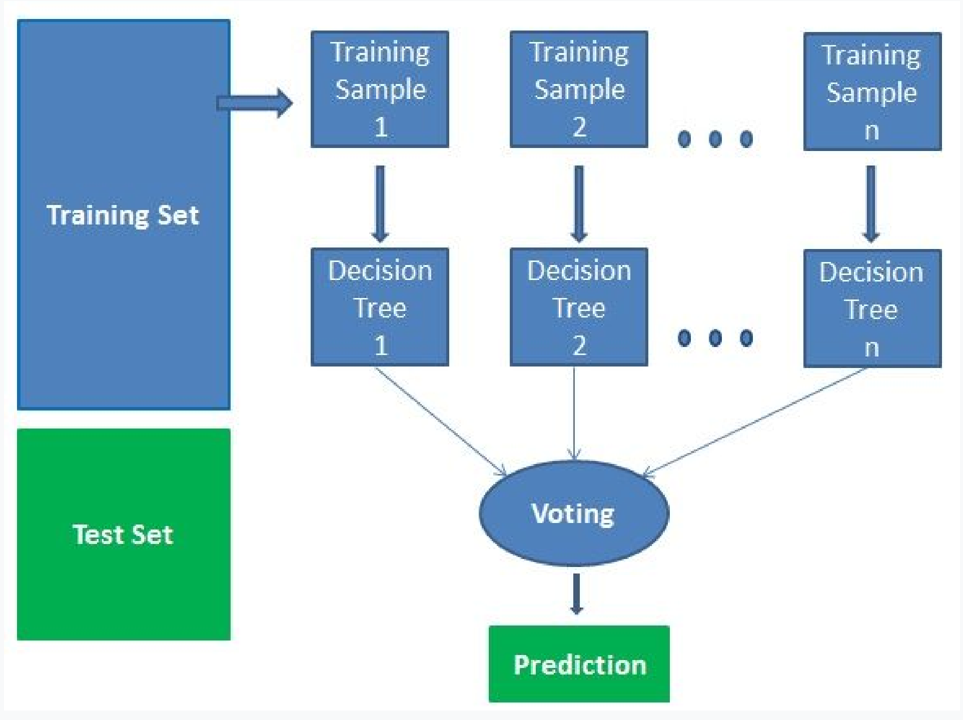


Figure 5 : Principle of operation of Random Forest machine learning algorithm

Random Forest has many advantages: first it does not require a lot of text preprocessing, then it handles missing values and maintains the accuracy for missing data. But the technique also has some drawbacks : since final prediction is based on the mean predictions from subset trees, it won't give precise values for the regression model.

1.5.4 K-Nearest-Neighbors (k-NN)

k-NN is a simple supervised machine learning algorithm which is mostly used for classification. The algorithm is based on a distance function for pairs of observations, such as the Euclidean distance (see Figure 6) [18]. It means that it classifies documents as data points based on how their neighbours are classified. The letter “k” in k-NN represents the number of neighbours which is an important parameter of the model. Thanks to its effectiveness and its easy implementation properties, the method can be used for many applications. However the model still has some drawbacks: the classification time is long and it is quite difficult to find the optimal value of k. Indeed the best choice of k depends on the data we are using. Usually choosing higher values of k reduces the effect of noise on the classification but it can make limits between classes less distinct.

The k-NN algorithm works in three steps. First he calculates the Euclidean distance between any two points from the dataset. Then he finds the nearest neighbours by ranking points by increasing distance. Finally he votes on a predicted class label based on the classes of the k nearest neighbours.

The Euclidean distance can be calculated equation (3):

$$d(p,q) = d(q,p) = \sqrt{(x-a)^2 + (y-b)^2} \qquad (3)$$

where p and q are two points representing documents in the plane and d(p,q) is the Euclidean distance between the two points of coordinates p=(x, y) and q=(a, b) .

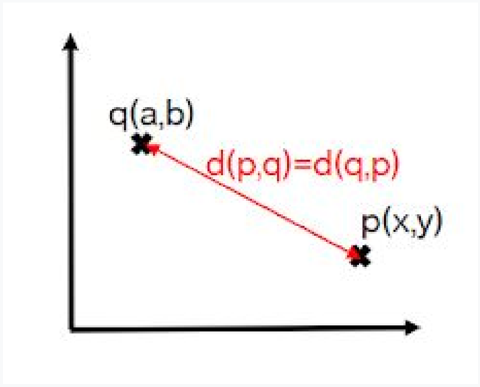


Figure 6 : Schematic representation of the Euclidean distance between two points p and q

In this case, the smallest the value of d(p,q) is, the most likely the documents belong to the same class.

1.6. Text analysis

Text analysis or text mining is the process of deriving meaningful information from text. The process usually involves translating a large amount of unstructured text into easy-to-manage and easy-to-interpret data (semi-structured or structured data). Just like TC, text analysis is an application of NLP which allows for extracting keywords, topics or sentiments from the text. To do so, text analysis may rely on simple statistics or ML algorithms. [19],[20]

The text mining workflow (see Figure 7) can be divided into the following steps:

1) Data retrieval
2) Text preprocessing
3) Text segmentation
4) Data extraction
5) Visualization

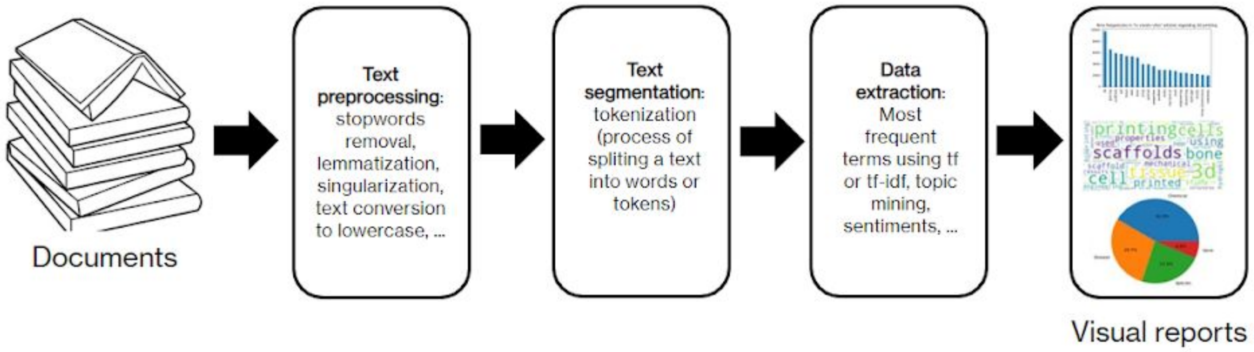

Documents
Text preprocessing: stopwords removal, lemmatization, singularization, text conversion to lowercase, ...
Text segmentation: tokenization (process of spliting a text into words or tokens)
Data extraction: Most frequent terms using tf or tf-idf, topic mining, sentiments, ...
printing cells scaffolds bone cell tissue 3d printed using properties
Visual reports

Figure 7 : Representation of the text mining or text analysis workflow

As you can see, text analysis techniques can be combined with data visualization tools to get the key insights from documents.

1.7. Text analysis tools

There are many text analysis tools available throughout the internet such as MetaMap,cTAKES or GATES. In this section, we review the tools that were used during this project.

1.7.1 Most frequent terms extraction using tf and tf-idf

In TM and NLP, an important question is how we can quantify what a document is about. To do that, we can start by looking at the words that compose a document and extract the terms which are the most important. Indeed two measures are available to know the importance of a word inside a document or a corpus: the term frequency (tf) and the term frequency-inverse document frequency (tf-idf). The tf of a word represents how many times the word occurs in a document and is one of the most basic ways to find the most important words in a document. But when we have a collection of documents, another approach is to calculate inverse document frequency (idf). The idf approach helps measure if a word is common or rare across all documents. This means that if a word appears in all the documents, it is less 'unique' to the document we are looking at and has a lower inverse document frequency. The idf can be multiplicated by the term frequency in order to access the tf-idf value which is basically the frequency of a term in a document adjusted for how rarely it is used in the collection.

Here are the formulas to obtain the tf-idf values:

Term Frequency (tf) = (Number of occurrences of a word) / (Total words in the document)

Inverse Document Frequency (idf) = Log ((Total number of documents) / (Number of documents containing the word))

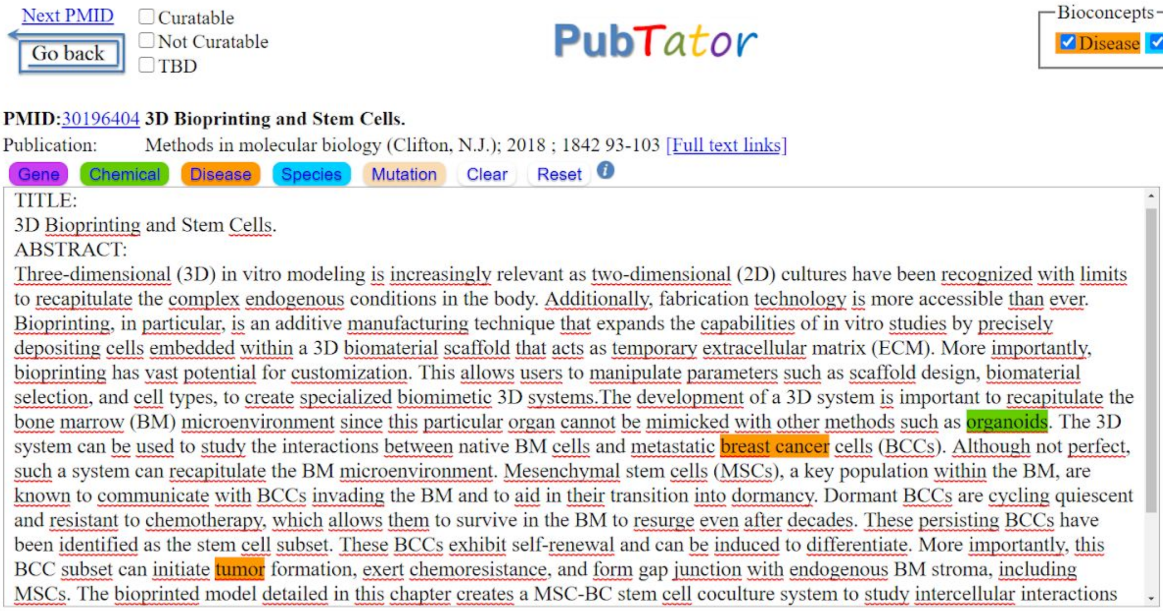
tf-idf(word) = tf(word) * idf(word)

But before calculating the most frequent terms of the documents, we have to do some text preprocessing just like for text classification. In each text document there are some words that appear often but do not carry much information. It is the case with stopwords such as: “is”, “the”, “it”, “of”, etc. For some applications, it is necessary to remove such words from the text in order to keep the most meaningful terms for the analysis.

Once stopwords are removed, we can easily calculate the tf and the tf-idf of each word and plot the results using libraries such as matplotlib and wordcloud.

1.7.2 PubTator annotations

PubTator Central is a tool which allows us to do semantic analysis. It is a web-service from NCBI that enables researchers to get annotated PubMed abstracts. The tool works by identifying terms belonging to specific categories that are built into PubTator. These categories are called Bioconcepts [21]. Visually, the annotations are displayed in the abstract using different colors: one for each bioconcept (see Figure 8).



PMID:30196404 **3D Bioprinting and Stem Cells.**

Publication: Methods in molecular biology (Clifton, N.J.); 2018 ; 1842 93-103 [Full text links]

TITLE:
3D Bioprinting and Stem Cells.
ABSTRACT:
Three-dimensional (3D) in vitro modeling is increasingly relevant as two-dimensional (2D) cultures have been recognized with limits to recapitulate the complex endogenous conditions in the body. Additionally, fabrication technology is more accessible than ever. Bioprinting, in particular, is an additive manufacturing technique that expands the capabilities of in vitro studies by precisely depositing cells embedded within a 3D biomaterial scaffold that acts as temporary extracellular matrix (ECM). More importantly, bioprinting has vast potential for customization. This allows users to manipulate parameters such as scaffold design, biomaterial selection, and cell types, to create specialized biomimetic 3D systems.The development of a 3D system is important to recapitulate the bone marrow (BM) microenvironment since this particular organ cannot be mimicked with other methods such as organoids. The 3D system can be used to study the interactions between native BM cells and metastatic breast cancer cells (BCCs). Although not perfect, such a system can recapitulate the BM microenvironment. Mesenchymal stem cells (MSCs), a key population within the BM, are known to communicate with BCCs invading the BM and to aid in their transition into dormancy. Dormant BCCs are cycling quiescent and resistant to chemotherapy, which allows them to survive in the BM to resurge even after decades. These persisting BCCs have been identified as the stem cell subset. These BCCs exhibit self-renewal and can be induced to differentiate. More importantly, this BCC subset can initiate tumor formation, exert chemoresistance, and form gap junction with endogenous BM stroma, including MSCs. The bioprinted model detailed in this chapter creates a MSC-BC stem cell coculture system to study intercellular interactions

Figure 8 : Example of annotations made by PubTator on a particular article from the 3D-printing literature (Caitlyn A. Moore, Niloy N. Shah, Caroline P. Smith, and Pranela Rameshwar. *3D Bioprinting and Stem Cells*. Available from: https://pubmed.ncbi.nlm.nih.gov/30196404/)

The working principle of PubTator is presented above. As you can see five bioconcepts are available:

- Gene
- Disease

- Chemical
- Mutation
- Species

The “Gene” bioconcept is built from the gene database in NCBI which includes nomenclature, Reference Sequences (RefSeqs), maps, pathways, variations, phenotypes, and links to genome-, phenotype-, and locus-specific resources worldwide. [22]

The “Disease” bioconcept uses the Comparative Toxicogenomics database (CTD) which includes chemical–gene/protein interactions, chemical–disease, chemical-phenotype and gene–disease relationships. [23]

The “Chemical” bioconcept utilizes the Medical Subject Headings (MeSH) thesaurus which is a controlled and hierarchically-organized vocabulary produced by the United States National Library of Medicine (NLM). [24]

The “Mutations” bioconcept is based on the Single Nucleotide Polymorphism database (SNP) in NCBI which contains information about nucleotide variations, insertions and deletions, and genomic and RefSeq mapping for both common variations and clinical mutations. [25]

The “Species” bioconcept uses the taxonomy database in NCBI which is a classification of approximately 10% of the described species of life on the planet. [26]

PubTator is a very useful tool because it allows for extracting relatively accurate, high quality information already organized in predefined categories (biocepts). []

1.7.3 Topic mining/ topic modeling

The concept of topic mining relies on the assumption that each document can be represented by a list of topics [27]. Topic modeling or topic mining techniques aim to retrieve information from documents and to represent it in the form of themes or topics. These techniques are able to discover hidden topics in a collection of documents and depending on the technique used, the relative importance of each topic can also be found. Topic models are a great tool to help organize and offer insights on a large amount of unstructured data. In this part, we will describe the two topic models that were used during the project.

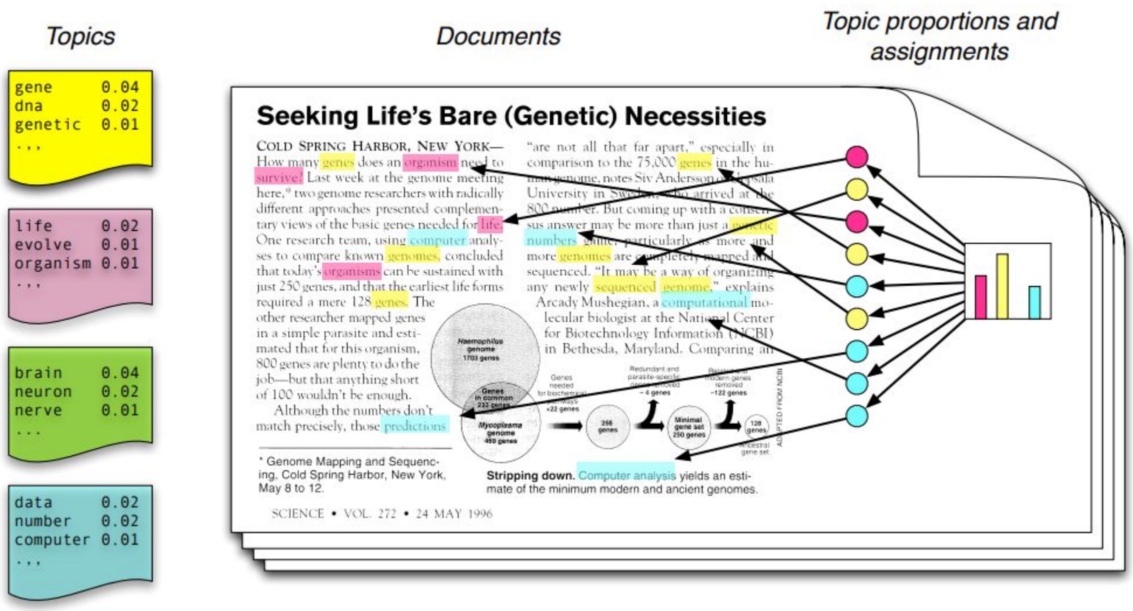
1.7.3.1 Latent Dirichlet Allocation model

Latent Dirichlet Allocation (LDA) is one of the most popular topic modeling algorithms that allows mining latent topics from a collection of documents. Just like any topic model, LDA is an unsupervised algorithm. This means that we do not need to give it training data that contains predefined topics to let it learn how to process (in contrast to text classification models). We simply have to provide a collection to the algorithm and it figures out on its own how to get topics from it. Although this is convenient, because we do not have to go through the effort of hand-labeling a set of data, it also means that we cannot guarantee the outputs of the topic model or evaluate its performances. Another thing to take into account is that topic models require an adjustment of the parameters to achieve optimal results, which is time consuming and can be computationally expensive with large collections. Sometimes the algorithm is also mentioned as a “fuzzy clustering” algorithm. The term "clustering" comes from the fact that the model performs the classification on its own without the programmer specifying the features and the term "fuzzy" is here because the groups of words creating topics can overlap.

Now let’s explain a little more how LDA generates topics from the text. LDA starts by attributing a random topic to each word in the corpus of documents. Then it calculates the document to topic count i.e. the number of words corresponding to each topic inside each document. At the same time, the algorithm also counts the times a word is associated with each topic (i.e. the topic to words count) because, as mentioned above, a word can appear in several topics. After this random initialization, we need to know the words that are significant to each topic. So the algorithm has to reassign the topic of each word. To do this, it relies on probability calculations such as the prior, and the likelihood. [28]

Figure 9 shows the principle of operation of LDA. As you can see, for each document of the corpus, words are assigned to one of the 4 topics available on the left (in the yellow, pink, green and blue panels). Each panel contains the top words associated with the topic but we notice that the algorithm does not name the topics for us. Indeed the output of a topic model such as LDA are 2 things: first a list of words associated with each topic with high probability and then an assignment of each document to topics.

But the absence of titles for each topic is not the only problem with LDA. Indeed the main drawback of the model is that the algorithm does not tell us how many topics a given corpus has. Instead it is the programmer that needs to choose this number beforehand and set it as a parameter inside the topic modeling algorithm.



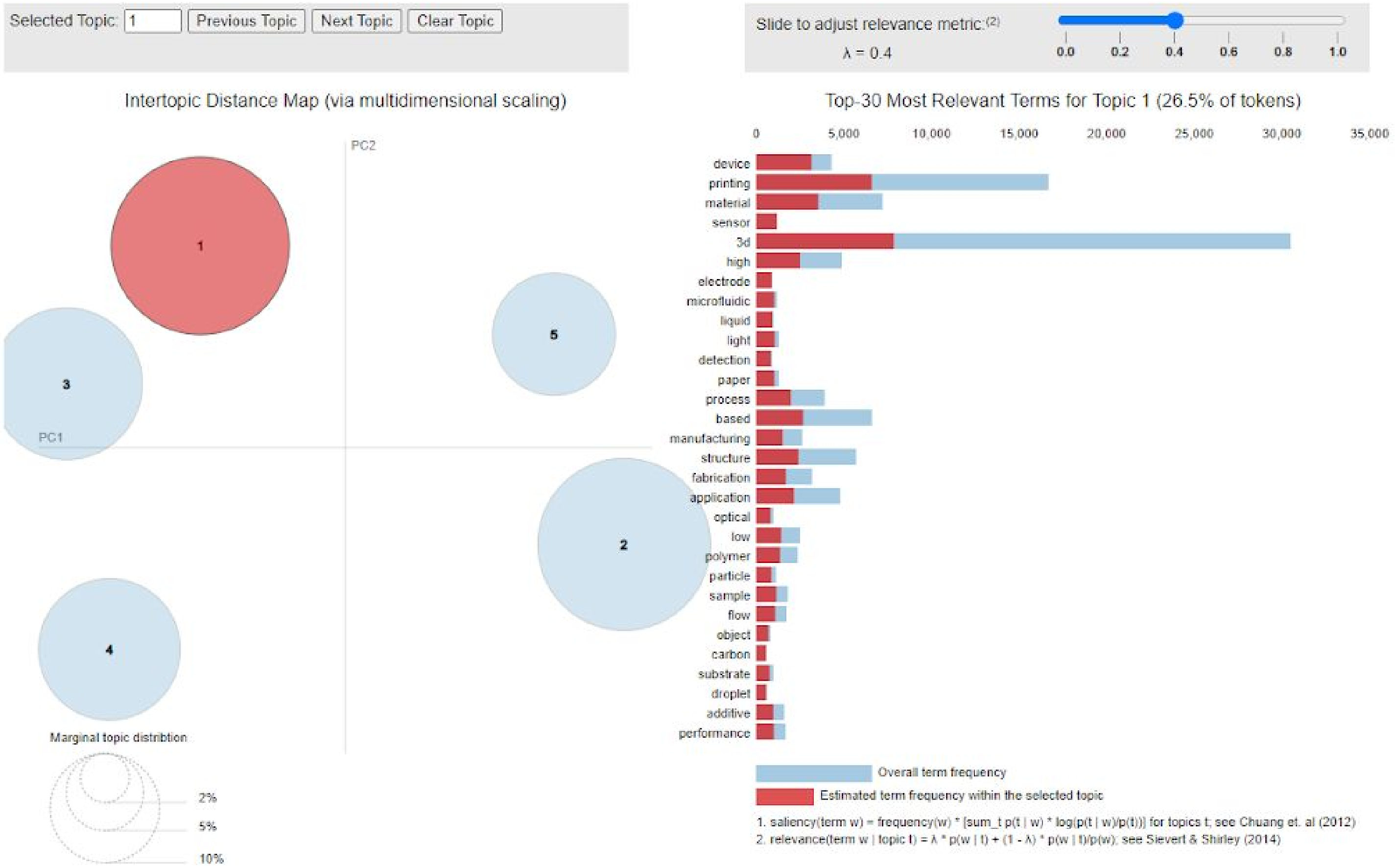
- Each **topic** is a distribution over words
- Each **document** is a mixture of corpus-wide topics
- Each **word** is drawn from one of those topics

Figure 9 : Principle of operation of the LDA model

The code used to perform LDA topic modeling was implemented by Shashank Kapadia and is available from GitHub account [29]. GitHub is a web-based platform that allows for collaboration between programmers. Indeed it is the largest open source repository in the world where developers can store their code and share it with the community. We chose to work with this code because it worked really well for visualizing the results of the topic model. In fact, in the code, a popular visualization package was used called pyLDAvis which is designed to help with:

1. Better understanding and interpreting individual topics, and
2. Better understanding the relationships between the topics.

As you can see in Figure 10, the results are represented in an interactive webpage where we can manually select a topic to view its top 30 most frequent and/or relevant terms in a bar chart. The bars represent the total frequency of the word across the entire collection of documents. We can adjust the results for each topic using different values of the λ parameter (i.e. the relevance parameter). If we choose lambda values close to 0, it will highlight potentially rare but more exclusive terms for the selected topic. If we choose higher lambda values (i.e. closer to 1), it will highlight more frequently occurring terms in the document that might not be exclusive to the topic. Adjusting the λ parameter can help us assign a proper title to each topic, so it is up to us to find the most adapted value for our data. In this project, we could easily interpret the topics using a parameter of 0.4 so we used this value for both topics. Then if we look on the left of the webpage, we can see the Intertopic Distance Plot map which is a visualization of the topics in a two-dimensional space. In this map, each topic is represented by a circle and the area of the circle is proportional to the amount of words that belong to the topic across the corpus. The representation helps to discover how topics are related to each other: the topics that are closer in the map have more words in common.



Selected Topic: 1

Slide to adjust relevance metric:(2) λ = 0.4

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

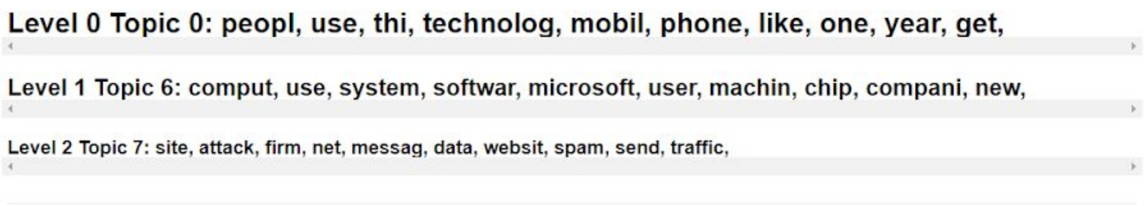
Figure 10 : Overview of the results that can be obtained using LDA model and pyLDAvis package [30]

1.7.3.2 Hierarchical Latent Dirichlet Allocation model

Hierarchical Latent Dirichlet Allocation (hLDA) is a generative statistical model that allows extracting hidden topics from a large amount of data and organizes them hierarchically. It is an upgraded version of the LDA algorithm which can determine how many topics a collection of documents contains. The model was developed in 2003 by a group of American researchers in Computer Science: David M. Blei, Andrew y. Ng, and Michael I. Jordan [31]. hLDA works by sweeping through documents and identifying patterns of word usage, and then clustering those words into topics. After the algorithm performs this clustering, it determines what documents contain what topics and then groups documents that contain similar topics.

This hLDA algorithm works by generating a tree made from nodes and branches where each branch represents a topic. This tree can be seen as a family tree where the persons located at the bottom levels inherit from the ancestors at the upper levels. The model is based on a non parametric prior probability called the nested Chinese Restaurant Process (CRP) which allows for arbitrarily large branching factors and readily accommodates growing data collections [32],[33]. The hLDA model combines this prior with a likelihood probability that is based on a hierarchical variant of the LDA model. (see the section 2.4.1 about Multinomial NB and Bayes' Theorem for the definitions of the prior and the likelihood) One of the main requirements in this model is to specify the depth of the tree (or number of levels) through which it will iteratively look for subtopics.

As mentioned earlier, hLDA is based on a hierarchical variant of LDA so the main difference between the models results in the hierarchization of the topics. With hDLA, there are different levels of topics: the topics of higher level are more general while the topics of lower level are more specific. If we take a look at the following example from Joe Wandy's GitHub account (see Figure 11), we can see the level associated with each topic [34]. Here the bigger the topic appears, the higher its level is. In this example, topics were generated from BBC news articles and we found out that the topic "technolog" (i.e. technology) is more general than the topic "softwar" (i.e. software) in the BBC articles but the topic "spam" is more specific than the topic "softwar".



Level 0 Topic 0: peopl, use, thi, technolog, mobil, phone, like, one, year, get,

Level 1 Topic 6: comput, use, system, softwar, microsoft, user, machin, chip, compani, new,

Level 2 Topic 7: site, attack, firm, net, messag, data, websit, spam, send, traffic,

Figure 11 : Representation of the hierarchy between topics in hLDA model

2 Methodology

2.1. Python programming language

This project was implemented on PyCharm Community which is an Integrated Development Environment (IDE) specially designed for Python. Python is a powerful general-purpose programming language. It is used in many applications, including web development, data science, and creating software prototypes. We decided to use Python instead of other programming languages because it offers a lot of advantages. First, it is the language of choice for data analysis and machine learning and it has many libraries to this end. Then, it is easy to learn: I did not know anything about Python before starting this project and I managed to write my first lines of code after only one month of studying. Moreover, its syntax is quite simple and the code is very readable which contributes to the ease of learning. It allows to write programs in fewer lines of code than most of the programming languages. Finally it is very popular so it is easy to find code or help throughout the internet.

2.2. Creation of the datasets

2.2.1 Creation of the 3D-printing test set

To create the testing set about 3D-printing, 477 articles were manually curated from Pubmed by scanning the abstracts, keywords and titles. The articles were found using the following Pubmed query: (((3d printing) OR (3d-printing) OR (three dimensional printing) OR (bioprinting)) NOT ((review)[Publication Type])) NOT ((systematic review)[Publication Type]). This query was designed to exclude reviews from the selection because they typically cover a large range of studies. Indeed reviews usually contain a mix of in vivo, in vitro and clinical studies making them hard to associate with one class. Records with no abstracts and non-english records were also excluded from the testing set. A special effort was made to create a set as representative to the literature as possible. So articles from each year between 2007 and 2020 were retrieved with a number of articles selected per year proportional to the number of publications available for that year. At the end, the 477 articles retrieved were manually classified into one of the following categories: in vivo studies, in vitro studies, clinical studies, studies that combine in vivo and in vitro experiments and studies that do not belong to any of the previous classes (also called non-biomaterial studies).(It is important to mention that the in vivo, the in vitro and the mixed in vivo and in vitro studies were then combined into one unique category so-called in vivo/in vitro studies.) This testing set will be first used to test the binary classifier from DEBBIE to see how it performs in the field of 3D-printing. Then it will be used to evaluate the performance of multinomial classifiers.

2.2.2 Extension of the DEBBIE training set

To build a multinomial text classifier, it is essential to have a balanced training set. In our case we wanted to classify articles into 3 classes (or categories): in vivo/in vitro studies, clinical studies and non biomaterials studies. It advised to have a similar number of abstracts for each class [35]. At the beginning of the project, we already had access to the gold standard set and the background set from the DEBBIE project. However, the number of clinical articles PMIDs was very small compared to the other classes, with only 106 clinical abstracts which is not sufficient to train a classification model. Thus it was necessary to extend the clinical studies set. To do so, the PMIDs of the 106 clinical abstracts from DEBBIE gold standard set were used to train the MedlineRanker classifier in combination with the Mesh terms ‘clinical trial’ and ‘scaffold’ in order to rank the 1,000 abstracts most similar to the set. MedlineRanker is an open access webserver from the University of Mainz offering a Naïve Bayes classification algorithm to directly rank PubMed abstracts [7]. The top 1000 ranked records, which are those considered to be most related to the topic of interest, were manually scanned to remove reviews, and the relevant ones were added to the clinical set. At the end, the training set for each category had around 1,000 abstracts and the complete training set contained a total of 3224 abstracts.

2.3. Building a text classifier

In this section, we will describe the steps that had been followed to build a text classifier:

1) Data retrieval
2) Text preprocessing
3) Labeling/indexing
4) Text conversion into vectors and feature selection
5) Training and testing sets
6) Model training
7) Adjustment of the model parameters (with cross validation)
8) Predictions on the testing set
9) Evaluation of the model performance

These steps can be summarized in Figure 12.

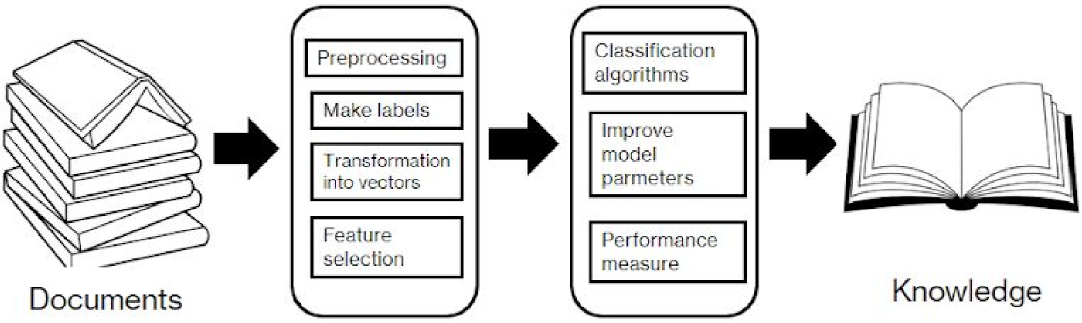


Figure 12 : Text classification process

2.3.1 Data Retrieval

This is the first step of the classification process in which we are collecting our documents of interest. For this project, we aim to classify scientific articles from Pubmed. To do so, a tool called Ebot from the National Center for Biotechnology Information (NCBI) was used [36]. Ebot facilitates data retrieval from Pubmed such as article metrics and abstracts. Ebot works by taking a list of PMIDs (in text file format) to generate a perl code. Then this code can be run on the

Windows Interpreter to retrieve a text file from MEDLINE with all of the abstracts one after the other.

2.3.2 Text preprocessing

After retrieving the data, it is important to do text preprocessing. Text preprocessing is the task to bring a text into a form that is predictable and analyzable by the machine learning algorithm. It enables you to harmonize documents before text classification. In this step, we get rid of all of the non relevant information for the classification such as the author's name, the date of publication, the PMID, etc. Then, we can remove stopwords. Stopwords correspond to words thare are commonly used in a given language. In English, they can be short words such as “a”, “is”, “and”, etc that appear frequently in documents but are not essential for their understanding. It is also possible to lower the text to remove any capital letter. Text preprocessing is an essential step before the classification because it will help the classifier to perform faster. [37]

Other steps that are usually taken during text preprocessing are:

Tokenization: A document is treated as a string, and then partitioned into a list of tokens (tokens are typically single words).

Stemming: Applying the stemming algorithm that converts different word forms into a similar canonical form. This step is the process of converting tokens to their root form, e.g. connection to connect, caresses to caress, superficies to superfici, etc. The problem with this technique is that it can generate words that do not exist.

Lemmatization: The process of reducing tokens to their ‘lemma’, eg: the word ‘wolves’ decomes ‘wolf’ and the word ‘talked’ becomes ‘talk’. The advantage of using lemmatization is that it provides only existing words. [37]

Finally depending on the model that will be implemented, we have to change the format of our data (or structure it). Several options are available when we are doing text classification. For example, it is possible work with multiple text files as it will be the case for the binary classification. But it is also possible to put all documents in a dataframe and then convert it into a csv file with all of the data in it. This method was chosen for the multinomial classification.

2.3.3 Labeling/indexing

An essential step in the text classification process is labelling/indexing. Labelling consists of assigning a tag or a label to each document of the training set to provide an example for the algorithm to learn. Indeed machine learning text classification algorithms learn to make classifications based on past observations instead of relying on manually crafted rules. Using pre-labeled documents as the training set allows the machine learning algorithms to learn how the different parts of text are associated together, and that a specific output (i.e. label or tag) is expected for a specific input (i.e. document). A “tag” or “label” is the predetermined classification or category that any given text could fall into. Labelling is not only necessary for the training set but also for the testing set. If we want to know how the classifier performed on a particular testing set, we must have an element of comparison and that is exactly the role of the testing set labels. When we evaluate the model performance, we simply compare the labels generated by the model to the labels that we predefined at the beginning. [38]

2.3.4 Text conversion into numerical features (vectors)

To reduce the complexity of the documents and make them easier to handle for the machine learning algorithm, the documents have to be transformed from the full text version to feature vectors. The first step is to convert the text into a numerical form using the bag of words (BOW) method. The BOW method creates a set of vectors containing the count of word occurrences in the document. But this representation scheme has some limitations. When we are using the BOW representation, we lose the correlation with adjacent words and the semantic relationship that exists among the terms in a document. So to overcome these problems, term weighting methods must be used to assign appropriate weights to the terms. Usually what programmers do is that they convert BOW vectors into TF-IDF vectors. TF-IDF is a score that represents the relative importance of a term/token in the document and in the entire dataset. So at the end, we obtain vectors that contain information on the more important words and the less important ones as well. It is possible to extract part of these vectors in order to get features for the machine learning algorithm.[37],[38]

2.3.5 Training and testing sets

As mentioned before, to do the classification, a classifier should be trained using a training dataset in which the category of each document is known in advance. To obtain their training set, programmers usually split their original dataset using Python (see Figure 13). In the Scikit-learn library, there is a function able to shuffle a dataset and then divide it into two subsets. One subset is directly used to build the classification model and the other is set apart for future performance evaluation. The name of this function is train.test.split() [39]. But before using it, we must check if our dataset is large enough because it takes a large number of documents to train a text classifier and if we have a small dataset, it will not perform well. Then it is also important to make sure that the testing set is representative of the all dataset i.e. we have to choose a testing set that has the same characteristics as the training set. When all of these conditions are respected, we can choose a split ratio. Most of the time, 70% of the dataset is used for the training set and 30% for the testing set. In this project, we did not use the train.test.split() function because our testing set is composed of manually curated articles regarding 3D-printing but we did make sure to have a good ratio of testing data/training data and a testing set as representative as possible.

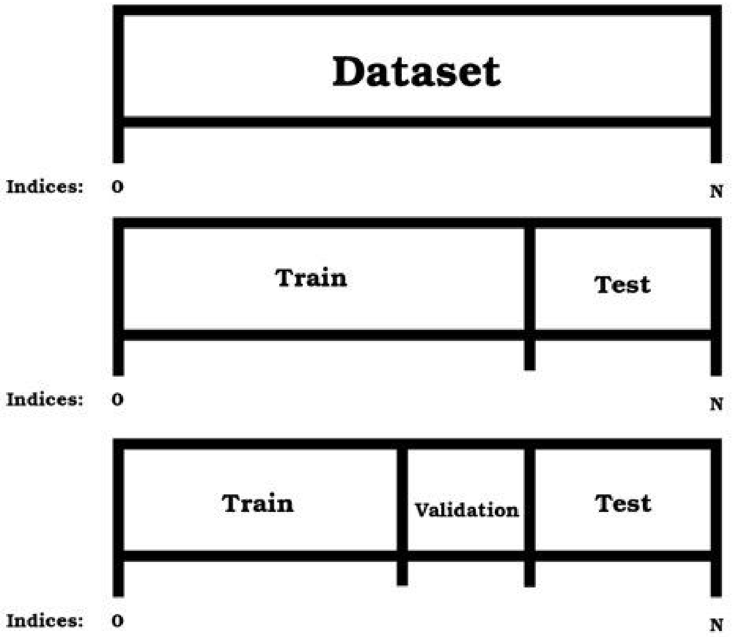


Figure 13 : Representation of the training set and the testing set

2.3.6 Model training

Once we have our training set, the next step in the classification process is to train our classifier. This is done by feeding the machine learning algorithm with training data that consists of pairs of features (vectors for each document obtained with BOW and TF-IDF techniques) and tags/labels (see Figure 14). The most important thing while training a classification model is to make sure that any documents from the testing set are also present in the training set because this will cause what is called 'overfitting'. According to link.springer.com, overfitting corresponds to situations where the model "describes features that arise from noise or variance in the data". When we have that kind of problem, we can get high performance scores such as accuracy=0.99, which is quite unlikely. Cross validation can be used as a preventive measure against overfitting.

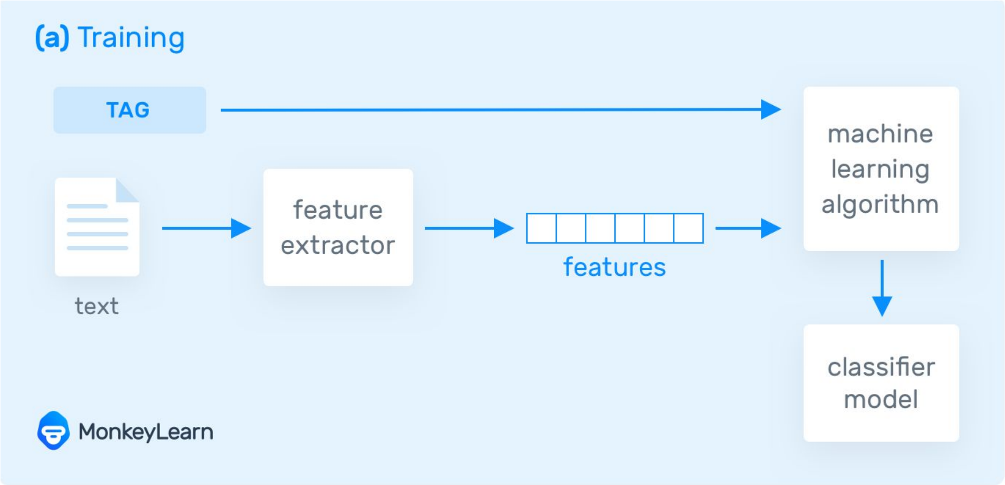


Figure 14 : Training step in text classification

2.3.7 Adjustment of the model parameters (with k-fold cross validation)

Any model is defined by many model parameters. If we want our classifier to perform well, it is essential to find the best model parameters. Cross validation is an iterative method for evaluating the performance of a model built with a given set of parameters. With cross validation, we can fit and evaluate a model with various sets of parameters to find out the more optimized parameters. K-Fold is an example of cross validation technique where a given dataset is split into K numbers of folds (see Figure 15) [40]. In K-fold cross validation, each fold is used as a testing set at some point. In the K=10 scenario which is the most commun, the data set is split into 10 folds. To understand the process, we will consider an example with only 5 folds just like in Figure 15. During the first iteration, the "Split 1" configuration is used so the documents in Fold 1 are kept for testing the model and the documents in Folds 2 to 5 are used to train the model. Then, in the second iteration, the "Split 2" configuration is used so the documents in Fold 2 are kept for the testing step while the rest of the documents (in Folds 1,3,4,5) are used as the training set. This process is repeated until each of the 5 folds have been used as the testing set. This method is interesting because it allows us to adjust model parameters with only the training data to avoid problems such as overfitting.

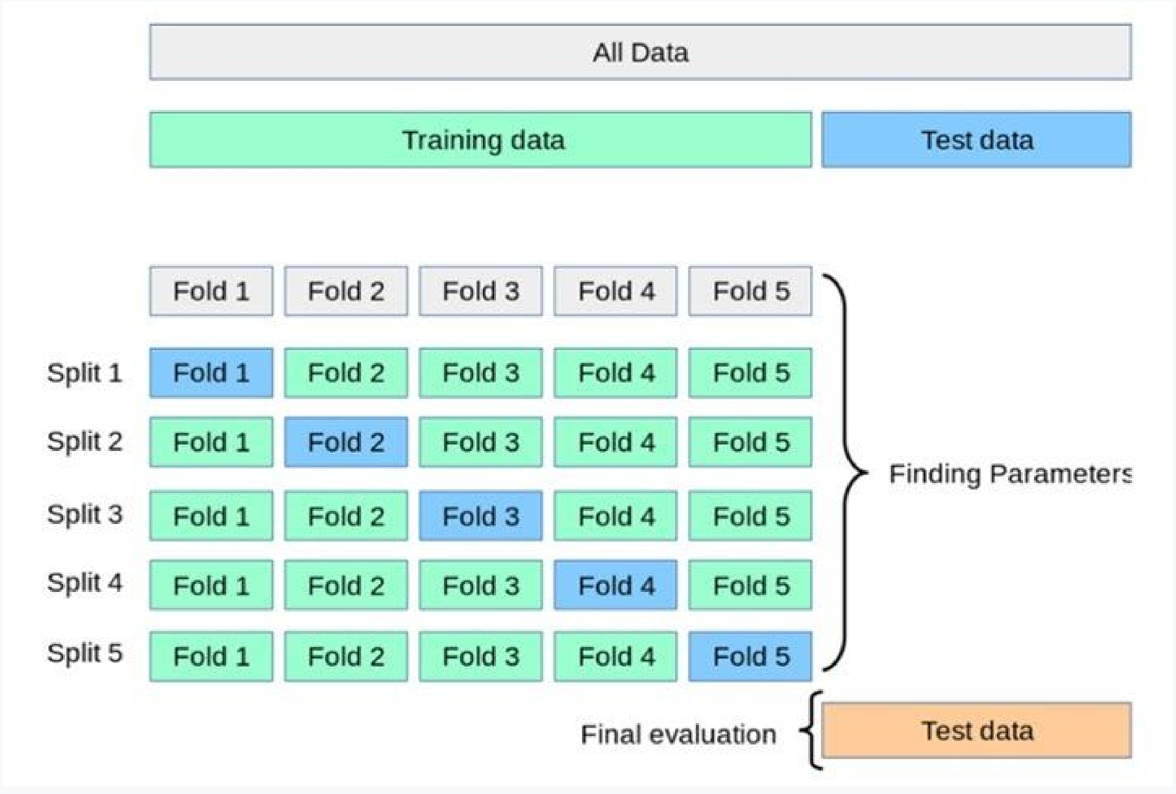


Figure 15 : Principle of k-fold cross validation

2.3.8 Predictions on the testing set

After training the classification model and adjusting its parameters, we can start to make some predictions on the testing set (see Figure 16). To get these predictions, we can use the predict() function available from Scikit-learn library [41]. This function returns an array of numbers that represent predicted labels/categories. For example, in binary classification, after using the predict() function we obtained an array of zeros and ones that represented our two categories (non biomaterials for the ones and biomaterials for the zeros).

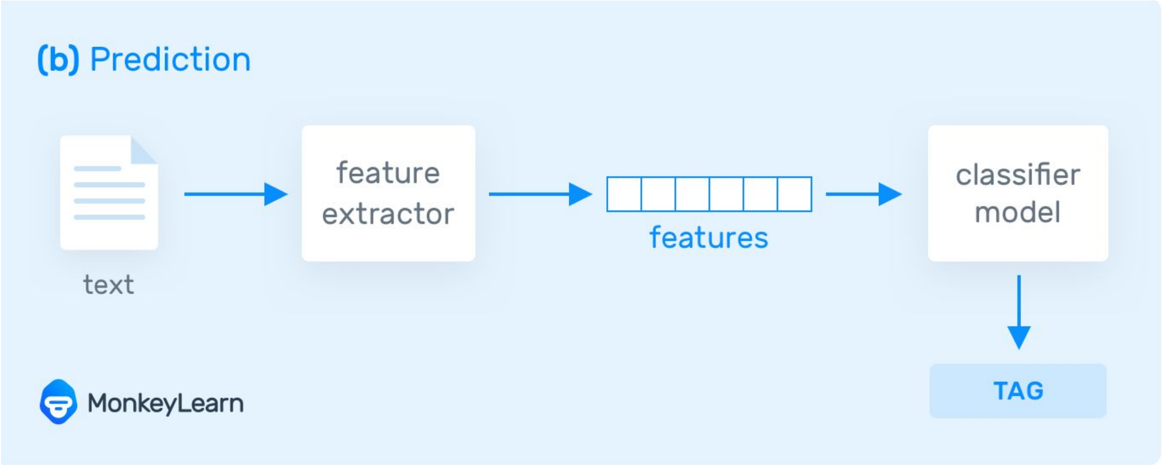


Figure 16 : Testing step in text classification

2.3.9 Evaluating model performance

This is the last step in the classification process where the effectiveness of the classifier is evaluated i.e its capacity to take the right categorization decisions. To do so, performance scores are generally calculated such as accuracy, precision, recall and F1-score. Their definition and formula are given below:

Accuracy : is a ratio of accurate predictions to the total number of predictions. Accuracy is the most intuitive performance measure.

Accuracy = (tp + tn) / total population where tp is the number of true positives and tn the number of false positives.

Precision : is a score that evaluates the ability of the classifier not to label as positive a document that is negative. The closest to 1 its value is, the best the classifier performs. [42]

Precision = tp / (tp + fp) where tp is the number of true positives and fn the number of false positives. [43]

Recall : is a score that evaluates the ability of the classifier to find all the positive documents. NB: The worst value for recall is 0. [44]

Recall = tp / (tp + fn) where tp is the number of true positives and fn the number of false negatives. [43]

F1-score : is a weighted average of the precision and recall. Just like precision and recall F1 score reaches its best value at 1 and its worst value at 0. [45]

F1 = 2 * (precision * recall) / (precision + recall) [43]

2.4. Text analysis

Using text analysis, we analyzed the classified text documents (i.e. the results from the multinomial text classification) in order to get insights about their content. For each of the three groups of documents obtained, we calculated the most frequent words using the metrics tf and tf-idf. The process started by removing all of the English stopwords from the text and converting text into tokens (i.e. the text was split into tokens) using Natural Language Toolkit (NLTK) library [46]. Tf and tf-idf of words were then calculated using tools from the Scikit-learn library. The matplotlib library and the wordcloud library were used for the visualization of the results.

Pubtator annotations

PubTator can annotate publications from a list of PMIDs or directly from a Pubmed query. So in order to extract concepts from the text, the Pubmed query about 3D-printing was used and put it into the web-service. The results were then downloaded in the pubtator format and converted into a text file. This file was opened on Excel to remove manually the information that we did not need (i.e. the title, the abstract and the MesH terms). After removing the irrelevant content from the document, we imported the final text as a dataframe into our Python code to extract information on each bioconcept. The process used here included preprocessing the text (remove English stopwords), splitting the text into tokens (words) and finally calculating the tf and tf-idf of words for each bioconcept and plotting the results.

Topic mining

Finally, we used two topic mining techniques: LDA and hLDA to extract topics or themes from the 3D-printing literature from Pubmed.

LDA topic modeling was performed on the 3D-printing corpus (i.e. more than 11,000 abstracts from Pubmed) using a code from Shashank Kapadia available in his GitHub account [29]. The process included the following steps: text preprocessing (with English stopwords removal), text conversion into tokens/words, words conversion to their lemma (lemmatization) and term frequency calculation. To make the algorithm work, we had to set the parameters. So we set the number of topics to 5.

Then, the hLDA model was applied to the same corpus to find out how many topics they are in the 3D-printing literature. hLDA was performed using a code [34] from Joe Wandy, a data scientist Glasgow Polyomics. As part of the process, English stopwords and words with less than 3 letters were removed before term frequencies were calculated for each remaining word in the vocabulary. 200 iterations and 3 levels of hierarchy were used to generate a list of the most probable six words per topic and the number of abstracts belonging to each topic.

2.5. Project sharing & reproducibility

Since the main purpose of this project is to perform biomaterials data mining and to participate in the creation of a database of biomaterials, it was key to openly share the work that was done during this project. To do so, a GitHub repository was created within the DEBBIE Project where we uploaded all of the code developed in this project, with additional information to facilitate the understanding. The datasets used for TC are also available from this repository. You can find them in the form of text files with a list of PMIDs inside. This repository also contains a README file that explains what the project is about and what you can find in each file. You can access the repository from the following link: https://github.com/ProjectDebbie/Multinomial_classifier. It is a completely open access repository, where anyone can download the resources i.e. the codes in Python as well as the datasets and use it for their own research. This way, we make sure that the project is reproducible and that it can continue to serve in the future.

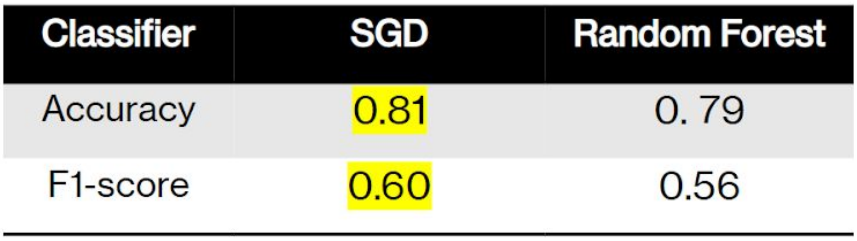
3 Results and discussion

3.1. Text Classification

3.1.1 Binary text classification

The binary classification (biomaterials vs non biomaterials) was made with Stochastic Gradient Descent SGD and Random Forest was chosen as a comparison classifier because it is wide acknowledged as a good classification model when dealing with text [16]. The models were trained using 2351 articles from the DEBBIE gold standard and background datasets and the performance scores of the models were evaluated using a testing set made from publications about 3D-printing. The testing set consisted of 477 articles manually curated from the 3D-printed literature on Pubmed. This set was composed of 407 biomaterials articles (in vivo, in vitro and clinical studies) and 73 non biomaterials articles. The results obtained are presented in Table 1 with the best results highlighted in yellow.

Table 1 : Performance results of trained binary classifiers



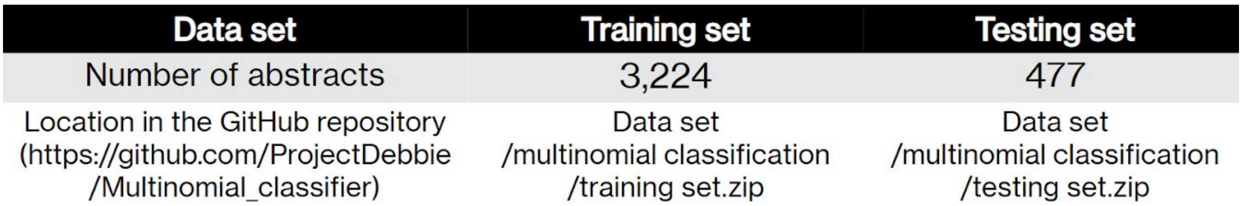
Classifier	SGD	Random Forest
Accuracy	0.81	0. 79
F1-score	0.60	0.56

The results revealed that the SGD classifier performed the best on the binary classification with 0.81 of accuracy and 0.60 of F1-score. Random Forest, on the other hand, obtained 0.79 of accuracy and 0.56 of F1-score. But to understand why we did not get even better results, we decided to look at the articles that were not well classified. With SGD classifier, we discovered that among the 18% of the articles that were not well classified, 69% were clinical. This means that the binary classifier does not perform well on clinical studies. So the vocabulary used in clinical articles differ from the vocabulary of biomaterials articles. This suggests that the researchers that write about clinical studies are not the same as those who write about in vivo/in vitro studies because they do not use the same terms in their publications making the classification process harder for our algorithm. This confirms the results obtained with the polydioxanone test set (in the framework of the project DEBBIE, Fuenteslopez et al, unpublished). The binary classification performance is compromised when mixing clinical and laboratory study abstracts. To successfully classify all articles, we need to find a new approach that can deal with the clinical literature. In the following section, we will see the results obtained after performing a multinomial classification on the same testing set and we will discover if this method handles better the clinical abstracts.

3.1.2 Multinomial text classification

Four models were used for the multinomial classification: Stochastic Gradient Descent (SGD) and Random Forest that have already been tested for the binary classification and two other classifiers which are Multinomial Naive Bayes (NB) and k-Nearest-Neighbors (k-NN). As you can see in Table 2, the training was composed of 3224 articles from the extended DEBBIE training set and the testing set was composed of our 477 articles regarding 3D-printing. The relative size of each class inside the training set and the testing set can be found in Figure 17 and 18. It will be interesting to compare our 3D-printing testing set to the 3D-printing literature to see if we managed to approach the real distribution.

Table 2 : Location of the data sets used for the multinomial classification, in the Github repository and representation of the number of abstracts in each set



Data set	Training set	Testing set
Number of abstracts	3,224	477
Location in the GitHub repository (https://github.com/ProjectDebbie /Multinomial_classifier)	Data set /multinomial classification /training set.zip	Data set /multinomial classification /testing set.zip

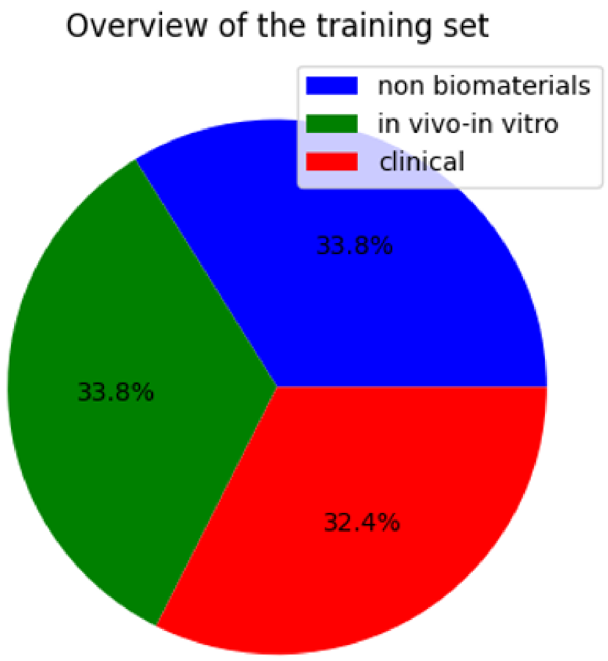


Figure 17 : Representation of the training set for the multinomial classification

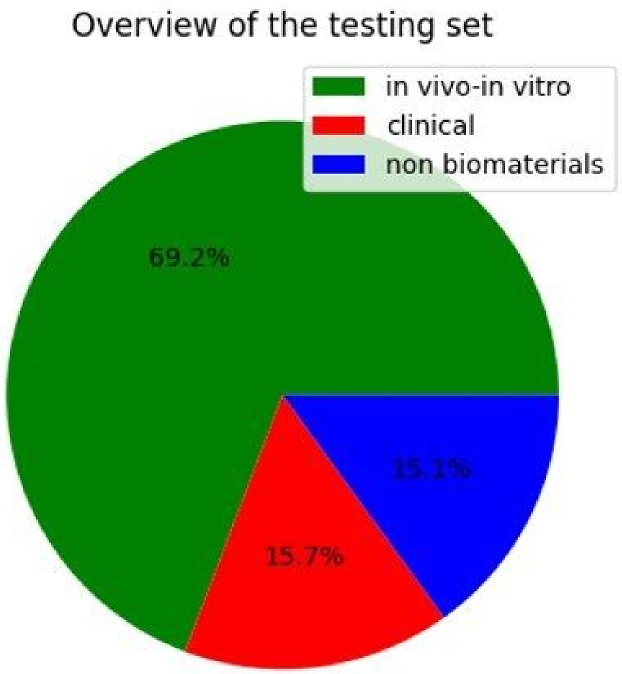
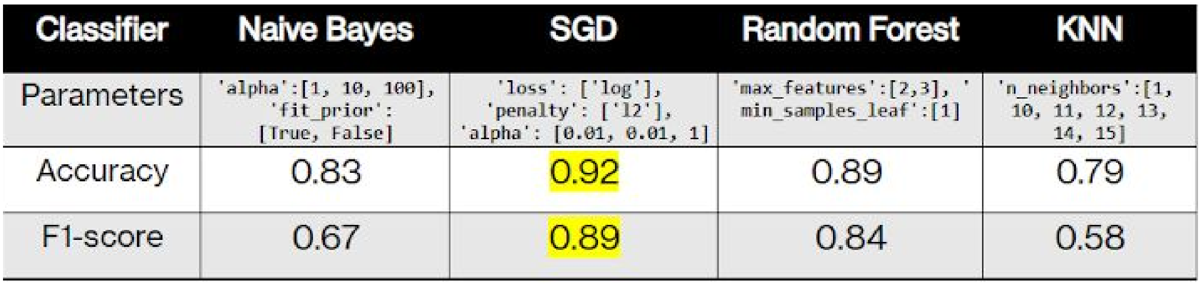


Figure 18 : Representation of the testing set for the multinomial classification

As you can see on Figure 17, the training set was balanced before implementing the multinomial classifier to avoid overfitting. When the implementation was complete, we took some time to adjust the model parameters in order to get the best results possible for each classification model. At the end, here are the model parameters that we selected and the performances scores associated (see Table 3):

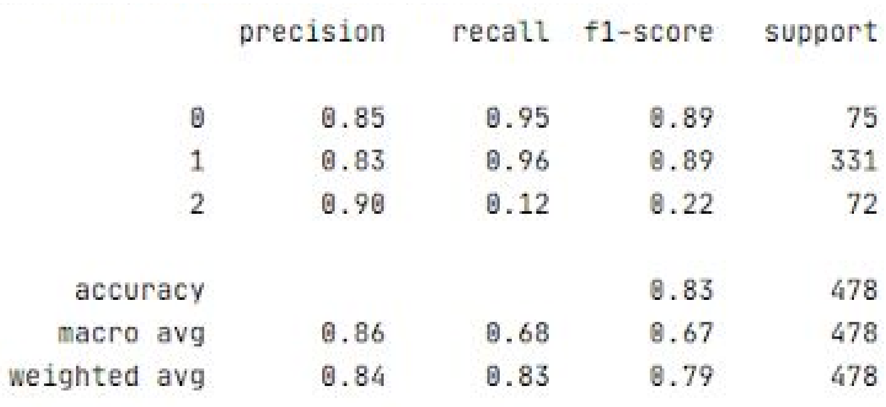
Table 3 : Performance results of trained multinomial classifiers obtained after adjusting the model parameters of each model



Classifier	Naive Bayes	SGD	Random Forest	KNN
Parameters	'alpha':[1, 10, 100], 'fit_prior': [True, False]	'loss': ['log'], 'penalty': ['l2'], 'alpha': [0.01, 0.01, 1]	'max_features':[2,3], ' min_samples_leaf':[1]	'n_neighbors':[1, 10, 11, 12, 13, 14, 15]
Accuracy	0.83	0.92	0.89	0.79
F1-score	0.67	0.89	0.84	0.58

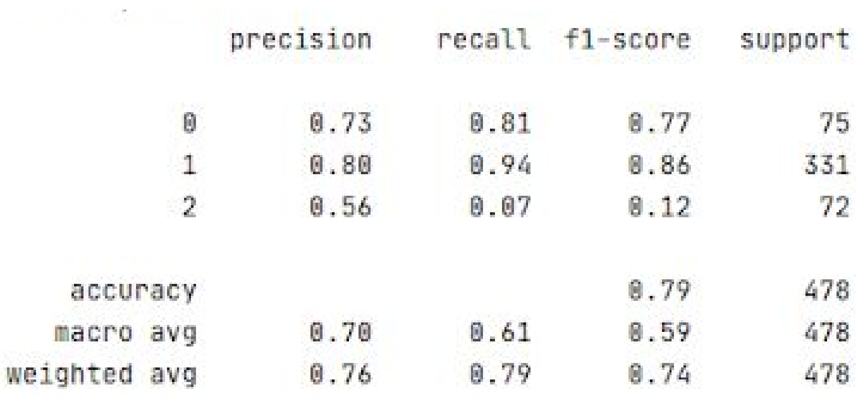
After calculating the performance scores of each classification model, we found out that SGD had the best results in terms of accuracy and F1-score with 0.92 and 0.89 respectively, quickly followed by Random Forest with 0.89 for the accuracy score and 0.84 for the F1-score (see Table 3). These results confirm our choice of models for the binary classification. SGD and Random Forest perform really well on text classification. On the other hand, Multinomial NB and k-NN have overall less good results than SGD and Random Forest. But they still presented acceptable accuracy scores for text classification with respectively 0.83 and 0.79 accuracy. As a remainder, we obtained 0.81 of accuracy with SGD and 0.79 of accuracy with Random Forest while performing the binary classification. Thus, in the case presented here, the multinomial or a multi-class classification performed better than the binary classification. When dealing with the same 3D-printing testing set, we obtained higher performance scores with the multinomial classifier.

Going back to NB and k-NN results, we observe that the main issue with their performances lies in their F1-score which are really low compared to SGD and Random Forest models with 0.67 for NB and 0.58 for k-NN. If we take a look at the confusion matrix from these 2 models (see Figure 19 and 20), we have the detail of the F1-score associated with each category of articles (0 for clinical studies, 1 for in vivo/in vitro studies and 2 for non biomaterials studies). (The confusion matrix is a metric from the Scikit-learn library that allows visualization of all performance scores of an algorithm inside a table). These results show that the F1-score is particularly low for the non biomaterials category (class 2). With NB, we got 0.22 for the non biomaterials category and with k-NN, we got 0.12. This means that both of these models are not able to classify articles that are non biomaterials. Instead they identified them as part of the in vivo/in vitro category. A possible reason why k-NN did not perform well with the non biomaterials category could rely on the principle of operation of the model. As mentioned earlier, k-NN classifies documents based on the Euclidean distance. But because the non biomaterials literature is very broad, evaluating the distance between documents might not be the best approach to classify documents from this category.



```
              precision    recall  f1-score   support

           0       0.85      0.95      0.89        75
           1       0.83      0.96      0.89       331
           2       0.90      0.12      0.22        72

    accuracy                           0.83       478
   macro avg       0.86      0.68      0.67       478
weighted avg       0.84      0.83      0.79       478
```

Figure 19 : Confusion matrix of NB model

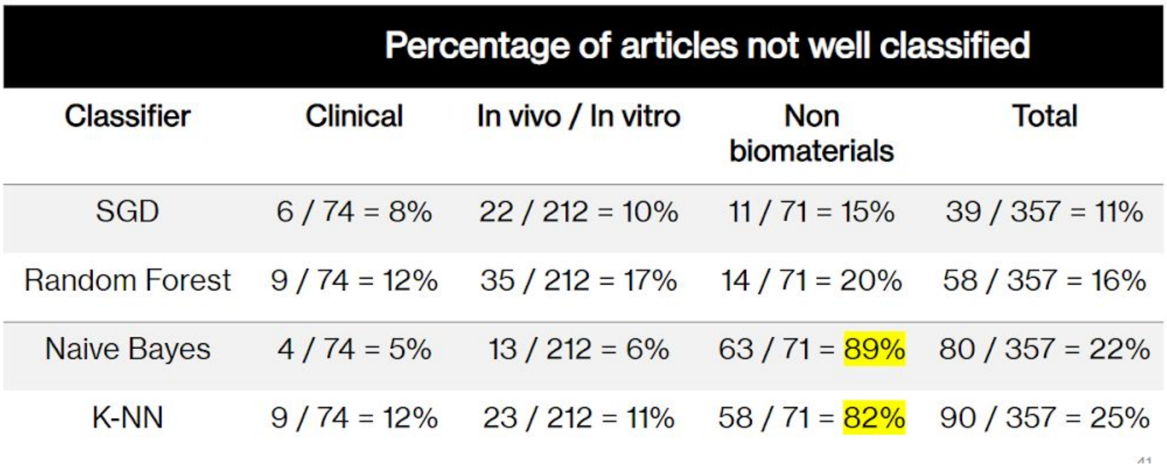


```
              precision    recall  f1-score   support

           0       0.73      0.81      0.77        75
           1       0.80      0.94      0.86       331
           2       0.56      0.07      0.12        72

    accuracy                           0.79       478
   macro avg       0.70      0.61      0.59       478
weighted avg       0.76      0.79      0.74       478
```

Figure 20 : Confusion matrix of k-NN model

A more complete analysis was done to identify the articles that were not well categorized for each classification model (see Table 4) and see if both of the models had problems with the non biomaterials category or not. The results showed that only Multinomial NB and k-NN were not able to identify articles inside the non biomaterials category. Indeed they were not performing as well as the others because of this one particular class. Moreover we noticed that if the models did not have issues with this category they would have shown better results than NB and Random Forest. If we look at the percentage of articles not well classified for the clinical and the in vivo/in vitro categories, we can see that these percentages are really low especially for NB with only 5% of clinical articles not well classified and 6% in vivo/in vitro articles not well classified. Multinomial NB performed even better than SGD on biomaterials articles. So to take advantage of these really good results we could think of a two-steps classification process which will first perform a binary classification distinguishing biomaterials abstracts from non biomaterials abstracts and then perform a second classification on the relevant abstracts (i.e. biomaterials) to identify clinical studies and in vivo/in vitro studies. This process would potentially solve the problems we encountered with NB and we could access overall better classification results.

Table 4 : Distribution of the articles that were not well classified for each class and each classification model



Percentage of articles not well classified

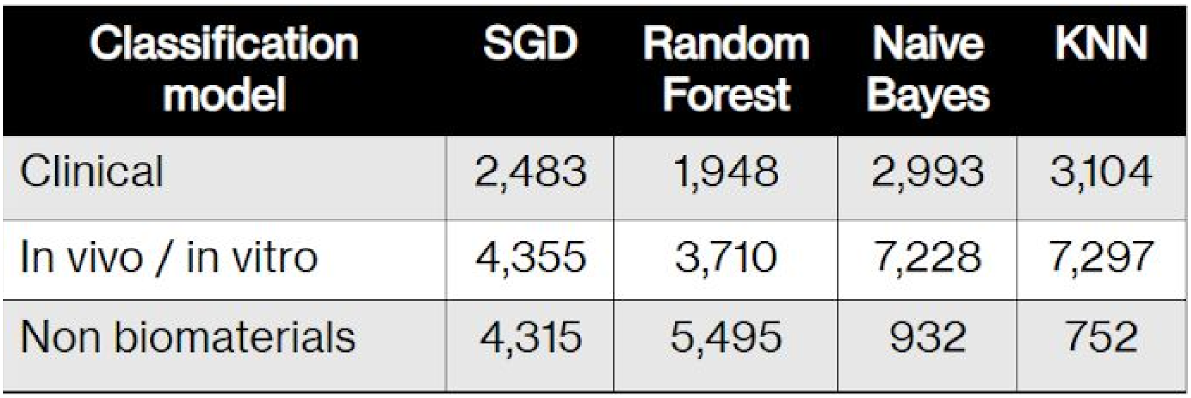
Classifier	Clinical	In vivo / In vitro	Non biomaterials	Total
SGD	6 / 74 = 8%	22 / 212 = 10%	11 / 71 = 15%	39 / 357 = 11%
Random Forest	9 / 74 = 12%	35 / 212 = 17%	14 / 71 = 20%	58 / 357 = 16%
Naive Bayes	4 / 74 = 5%	13 / 212 = 6%	63 / 71 = 89%	80 / 357 = 22%
K-NN	9 / 74 = 12%	23 / 212 = 11%	58 / 71 = 82%	90 / 357 = 25%

To conclude this part, we can say we managed to improve the classification of relative clinical articles by using a multinomial classification model. Indeed for all the tested models, clinical articles were relatively well classified. The best results were obtained using SGDClassifier from Scikit-learn library. We manage to get 0.92 of accuracy and 0.89 of F1-score with this model.

3.1.3 Classification on the 3D-printing literature using the multinomial classifier

After implementing the multinomial classifier and evaluating its performances on the 3D-printing testing set, it was time to use our classifier properly and classify the entire 3D-printing literature available from Pubmed. The relevant articles were selected using the same Pubmed query as the one used to create our 3D-printing testing set but this we were interested in all results. On the 16th of December 2020, the query (((3D-printing) OR (3d-printing) OR (three dimensional printing) OR (bioprinting)) NOT ((review)[Publication Type])) NOT ((systematic review)[Publication Type]) returned 11,786 articles regarding 3D-printing. From these 11,786 articles, 11,153 abstracts were successfully retrieved using the Ebot tool. These abstracts were then fed into the multinomial classifier. The results obtained are represented in Table 5. Here we classified the 3D-printing literature with our 4 classification models but in practise, we will only use the results from SGD because it is the most accurate model. We do not want to draw some conclusions based on predictions that are less accurate.

Table 5 : Classification results of the 3D-printing literature



Classification model	SGD	Random Forest	Naive Bayes	KNN
Clinical	2,483	1,948	2,993	3,104
In vivo / in vitro	4,355	3,710	7,228	7,297
Non biomaterials	4,315	5,495	932	752

For the reason mentioned above and to be more succinct, we are only commenting on the results from SGD in this part. In total, 2,483 articles were identified as clinical, 4,355 as in vivo/in vitro and 4,315 as non biomaterials by the model. We observe that 22% of the studies were considered as clinical by the classifier, 49% as in vivo/in vitro and 39% as non biomaterials. These results tell us that clinical studies are less represented than in vivo or in vitro studies. This is not surprising because clinical studies rely on human trials and, as the field of 3D-printing is relatively new, it is expected not to have experienced much on humans yet. Clinical studies are very expensive and require long regulatory processes, so only few laboratory-tested implants reach the clinic. If we compare these results to the distribution inside the 3D-printing testing set (see Figure 18), we observe that the non biomaterials class is more important in reality. The reason why we do not have as many non biomaterials articles inside the testing is because we encountered more difficulties to identify these kinds of articles. Indeed when we were unsure about the class of an article, we simply skipped it in order to have as little biais as possible. As the non biomaterials literature is very vast, it is not surprising to have had more difficulties with it .

3.2. Text analysis for data extraction

To extract information about the 3D-printing literature from Pubmed, we analysed more than 11,000 abstracts from the platform. To reach this goal, three approaches were used : first the text itself was analyzed to extract the most common terms for each category of studies, then we analysed the annotations from Pubtator and finally we used topic mining/modeling techniques.

3.2.1 Bag of words analysis

To see the dominant words in each category of studies, we started by calculating the term frequency of each word in the text and plotted the 20 most common terms of the category (see Figure 21).

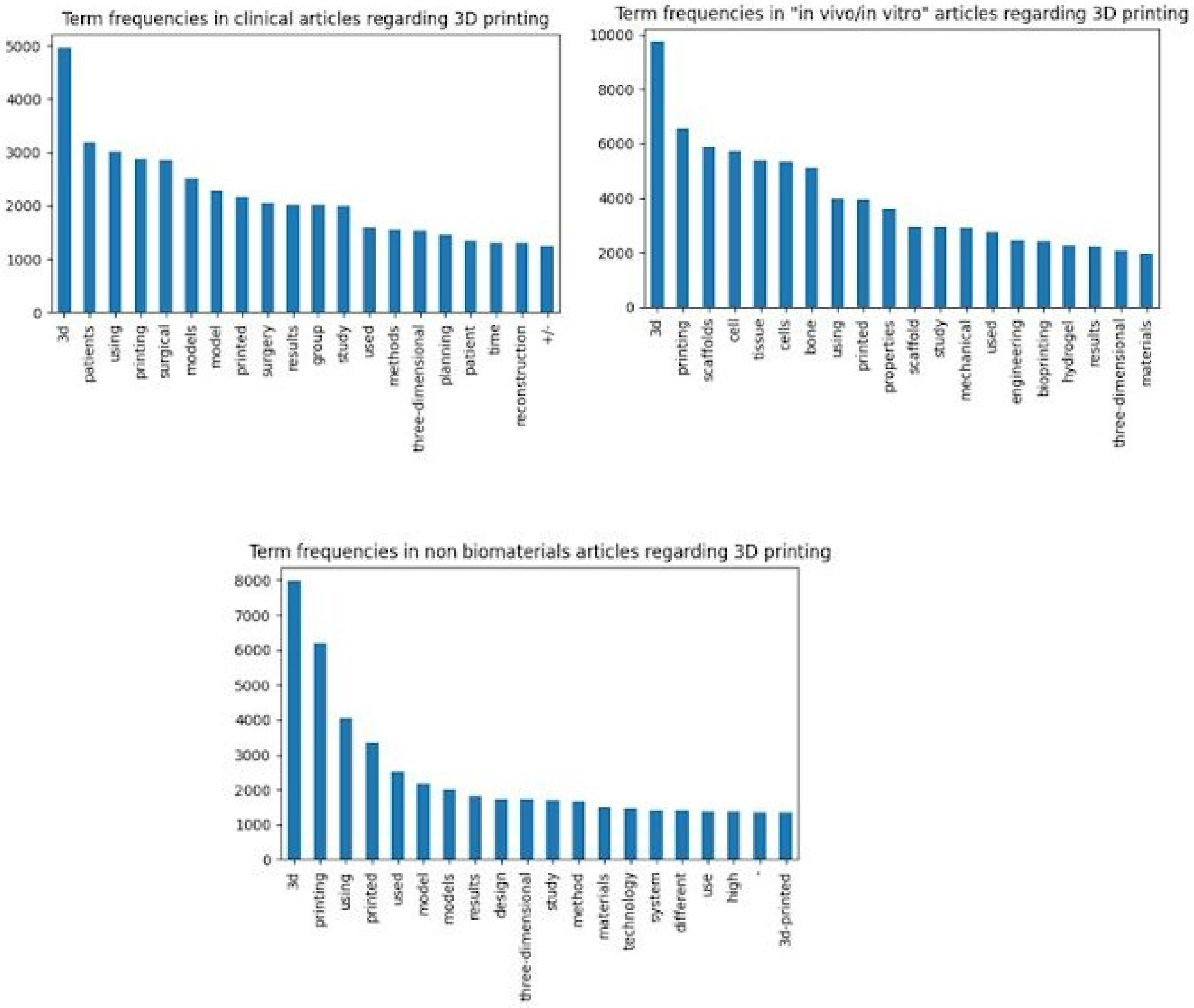


Figure 21 : Representation of the 20 most frequent terms in each category using tf

For each category, we found words like “3d” , “printing”, “printed”, “3d-printed” or “three-dimensional” in the top 20 that do not refer to a particular category of articles but to the general topic of the literature which is 3D-printing. But if we look at the other terms beside these, i.e. those who appear in only one category, we can see some terms representative of each category. For example, in clinical studies authors frequently use words such as “patients”, “surgical”, “group”, “reconstruction”, “time” or “surgery” which are more characteristic of the clinical literature. On the other side, if we analyse the 20 most common terms of the non biomaterials literature we encounter terms like “design”, “method”, “technology” or “system” which do not appear as much in other categories. Finally words like ‘scaffolds”, “cell”, “tissue”, “hydrogel” and “bioprinting” are more frequent in in vivo and in vitro articles.

But in order to get a different perspective from the results and an idea of the terms that distinguish the topics from each other, we decided to calculate the tf-idf values of each word in the text and we plotted the top 20 for each category. The results are presented in Figure 22.

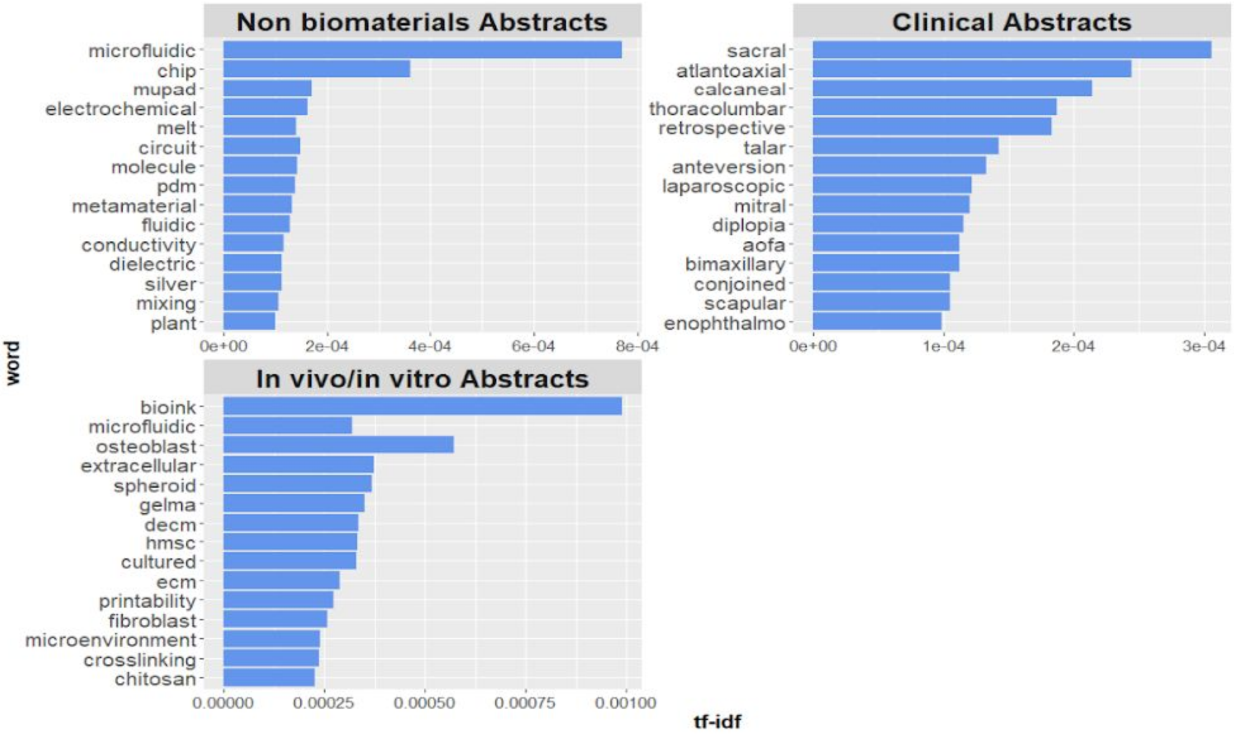


Figure 22 : Representation of the 15 most representative terms for each category using tf-idf

After retrieving the most frequent terms of each category using tf, we decided to check the most important words regarding their tf-idf scores. The results are presented in the Figure .

If you look at the results from the in vivo/in vitro category, we observe the appearance of new words compared to the tf result such as “bioink”, “osteoblast”, “gelma”, “decm”, “ecm”, “hmsc”, "chitosan" and "fibroblast". The terms "gelma" and "chitosan" correspond to hydrogels that can enter in the composition of the bioink. The term "gelma" (for GelMA) refers to gelatin methacrylate which is a gelatin-based hydrogel. The terms "osteoblast", "hmsc" and "fibroblast" refer to cells that have an important role in bone/tissue regeneration. The term "hmsc" (for hMSC) corresponds to the human mesenchymal stem cells. In experimental studies, these types of cells are commonly used for making and repairing skeletal tissues, such as bone and cartilage because they can differentiate into mesenchymal cells including osteoblasts. NB: The fibroblasts can be encapsulated in gelatin-based hydrogels like GelMA to create a bioink. The terms "ecm" and "decm" also present in the in vivo/ in vitro category refers to extracellular matrix and decellularized extracellular matrix. The extracellular matrix is a structure that acts like an environment for the cells where they can migrate, growth and differentiate, and some of its components may also be used for the formulation of bioinks.

Then if we analyse the results from the clinical category, there are new terms in the list such as "sacral", "atlantoaxial", "calcaneal", "thoracolumbar", "talar", "scapular". These terms all refer to parts of the body that are being replaced using 3D-printed implants. As you can see in the results, the sacral reconstruction seems to be the more represented intervention in the clinical articles. The terms "diplopia" and "mitral", on the other hand, correspond to diseases which are tackled using 3D-printing technology. The term "mitral" can be associated with mitral valve disease which is a type of heart disease and the term "diplopia" is a vision problem.

Finally by looking at the non biomaterials category, we found out terms like "chip", "fluidic", and "microfluidic" which correspond to articles focused on 3D-printing of microfluidic devices. A microfluidic device is an instrument that uses very small amounts of fluid on a microchip to perform certain laboratory tests, that is why they are called "Lap-on-a-Chip". In healthcare applications, it may use body fluids or solutions containing cells to diagnose diseases like cancer. The fabrication of microfluidic devices is one the main applications of the 3D printing technology in the biomedical domain [48]. Then the term "metamaterial" that also appears in the non biomaterials category might refer to articles that deal with 3D-printing of metamaterials for biomedical applications. Metamaterials are artificially engineered materials which derive their properties from their structure and not the material they are made of. Due to their unique geometry, metamaterials present unique properties that are not possible with conventional materials such as particular electromagnetic properties. Some of them have the ability to control and mold the flow of electromagnetic waves. This is why these materials are gaining more and more interest nowadays, especially in the biomedical field where they could contribute to the development of new diagnostic devices. [49]

3.2.2 Data extraction using Pubtator annotations

We retrieved the annotations of the 3D-printing literature made by PubTator and analysed them using Python. As mentioned earlier, PubTator annotated articles based on 5 bioconcepts: (“Chemical”, “Gene”, “Disease”, “Species” and “Mutation”) but depending on the data we are working on, we might not find each bioconcept in our results. In our case, we discovered that none of our articles contain terms that belong to the “Mutation” bioconcept. So only 4 four categories or bioconcepts were represented in our set.

Figure 23 represents the percentage of each bioconcept in total annotations i.e. we count each type a bioconcept appears in the annotations without taking into account the article associated with it. So if an article contains 5 terms associated with a bioconcept, we count this bioconcept 5 times. Considering this, we can say that 4 categories are represented inside the 3D-printing literature: “Chemical”, “Gene”, “Disease” and “Species” ; the most abundant is the “Chemical” category with 24,650 items. Then we have the “Disease” category with 17,094 items, quickly followed by the “Species” category which contains 13,895 items. Finally the “Gene” category has the weakest representation with only 3,901 items.

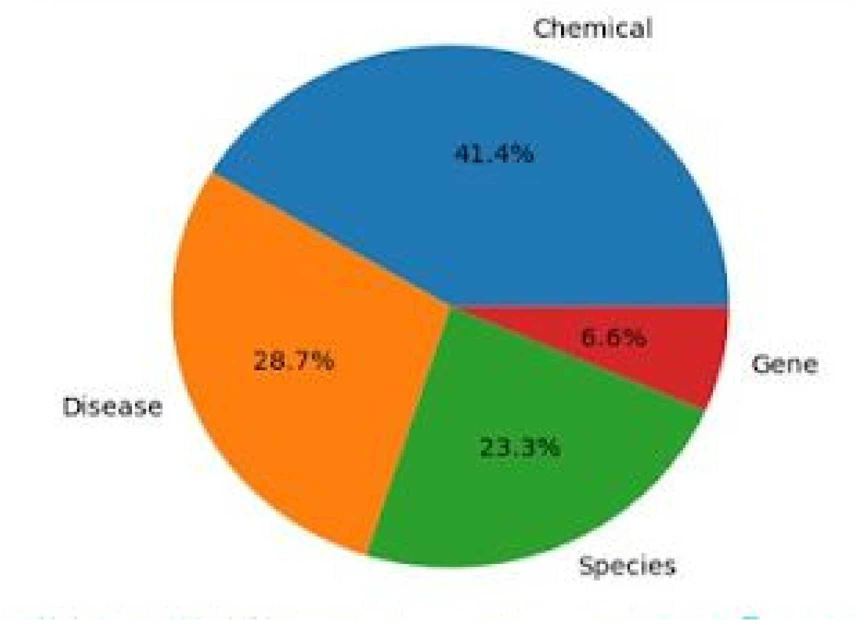


Figure 23 : Percentage of bioconcepts/categories from Pubtator

To start the semantic analysis of the 3D-printing literature from Pubmed, we decided to look at the 20 most frequent terms in all PubTator annotations independently of the category. The results are presented in Figure 24. In this diagram, the top 3 most common terms are “patient”, “patients” and “human”. Even if one might think that these terms should appear only in clinical articles and not the entire literature, it is not surprising to find them in the top 3 of the most frequent terms in all the annotations. Indeed many articles about 3D printing for healthcare use the expression ”patient-specific” to refer to the application of the 3D-printing technology. Moreover, the authors from clinical studies sometimes do not use the term “patient” at all in their publications but prefer using terms like “participants” , “group” and “person” to refer to their subjects. That is why the term “patient” can be representative of all kinds of articles. In the same vein, the term “human” frequently appears in the literature to designate the origins of the cells used and not necessarily the subject of the clinical trial.

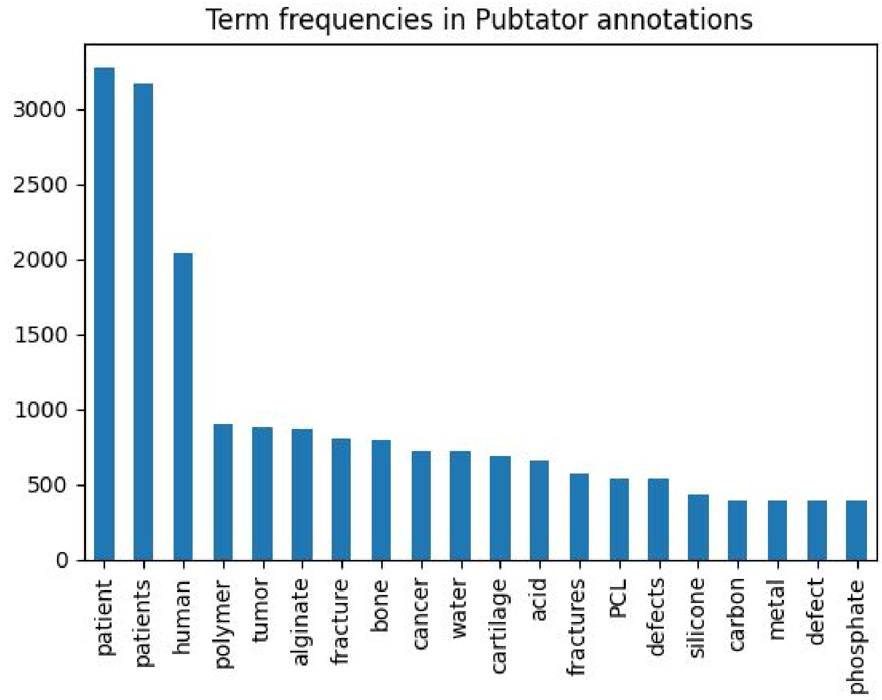


Figure 24 : Representation of the 20 most frequent terms in the PubTator annotations using tf

To get results more easily readable, we represented the most frequent terms in a word cloud using the wordcloud package (see Figure 25). In this representation, the bigger the word is, the more frequently it is used. The most frequent words in the text are the most visible. Here we can say that the terms “patient”, “patients” and “human” are the most represented in the PubTator annotations. In the next paragraph, we will see that these terms correspond to the “Species” bioconcept.

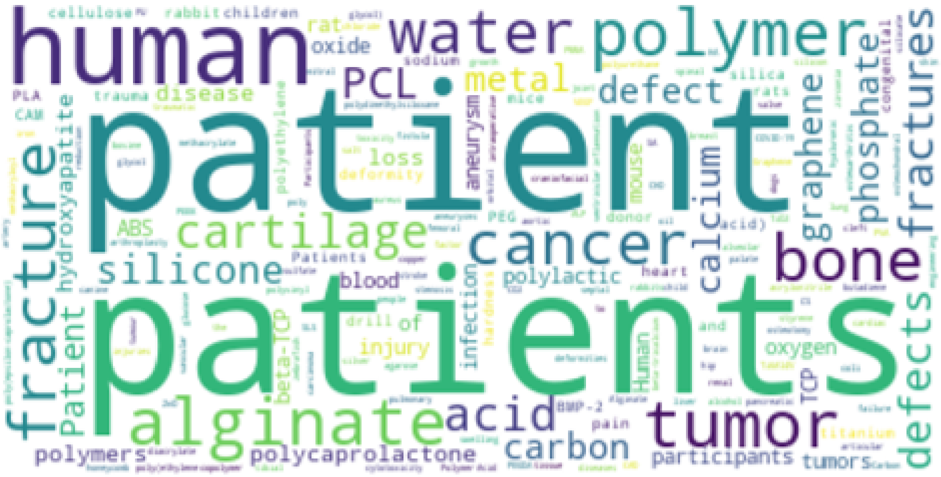


Figure 25 : Word cloud representation of the 20 most frequent terms in the PubTator annotations

In a second step, we decided to analyze the annotations from each bioconcept separately. The results are shown in Figure 26, which represents the 20 most frequent annotated terms in each category.

For the “Chemical” bioconcept, we observe that the terms such as “alginate”, “polycaprolactone”, “PCL”, “graphene”, “silicone”, “phosphate”, “hydroxyapatite”, “carbon”, “calcium”, etc. These terms all refer to elements that can enter in the formulation of the ink for the 3D-printer. So they correspond to the raw material that is used to create 3D-printed scaffolds. We can note that the term “alginate” is highly represented, with a number of occurrences in the abstracts superior than 800 times. Alginate is a biomaterial that is commonly used to make hydrogels. The alginate hydrogels have a similar structure to living tissue allowing for a wide range of applications in tissue/bone regeneration. NB: The term “PCL” which is also in the list corresponds to the abbreviation of polycaprolactone, a biocompatible polymer.

Then if we look at the “Species” bioconcept, we can find terms such as “patients”, “patient”, and “participants” in the top 20 of the most frequent terms. These terms might refer to the patients from clinical studies who had a surgery such as the implantation of a 3D-printed bone for instance or to the application of the 3D-printing technology in other types of articles (3D-printing for patient-specific application). Another term that could also refer to the patients from clinical studies is the word “children”. This term is quite interesting because it indicates that the 3D-printing technology has already been tested on kids. The terms “human” (or “Human”), on the other hand, might not refer to the patients themselves but more to the origin of the cells that have been used to make the 3D-scaffolds. Then if we take a look at the other terms in the “Species” category, we can find the most common animals used in laboratories for in vivo/in vitro studies. These animals can refer to the animal models of experimentation presented in the in vivo studies (for example bone replacement in a rabbit) and to the origin of the cells used in in vitro studies (i.e. the donor). Between those, we have members from the rodent family such as rabbits, mice and rats, and members of the canines such as dogs.

In the “Disease” category, we can find terms like “tumor”, “cancer”, “trauma”, “fracture”, “injury”, “aneurysm” and “loss” which correspond to diseases 3D-printing may be aiming to address. Indeed with the 3D-printing technology, we can create patient-specific implants whose purpose is to replace a functional part of the body that has been previously damaged or injured. The 3D-printing implant needs to be accepted by the body, while allowing it to perform the same function as the original part. But because the technology is relatively new, there may be some complications after the implantation as evidenced by the following terms: “defects”, “infection”, “pain”. (NB: the term “pain” could also refer to no pain reported after the surgery.) So the observation of these results from the “Disease” bioconcept permitted us to discover 3 specific cases where the 3D-printing technology can be applied in order to recreate part of a tissue or a bone. From what we can see, the 3D-printing technology can help patients that suffer from cancer (or from a malignant tumor), patients that suffer from an arterial disease such as aortic aneurysm or patients who had a major accident resulting in trauma, fractures, etc.

Finally when we analysed the results from the “Gene” bioconcept, we noticed that some annotated terms do not correspond to the category. It is the case for the terms “PCL”, “mum”, “aid”, “endothelial”, “vascular”, “bone”, “CAM” for instance. But there are still other terms that were correctly labeled in this category. The relevant annotations that we found in the “Gene” category are: “BMP-2”, “VEGF”, “ALP”, “OCN”, “osteocalcin”. These terms refer to genes and proteins that can be found in the human body and are responsible for the bone and the tissue regeneration. NB: The term “OCN” is the abbreviation for osteocalcin.

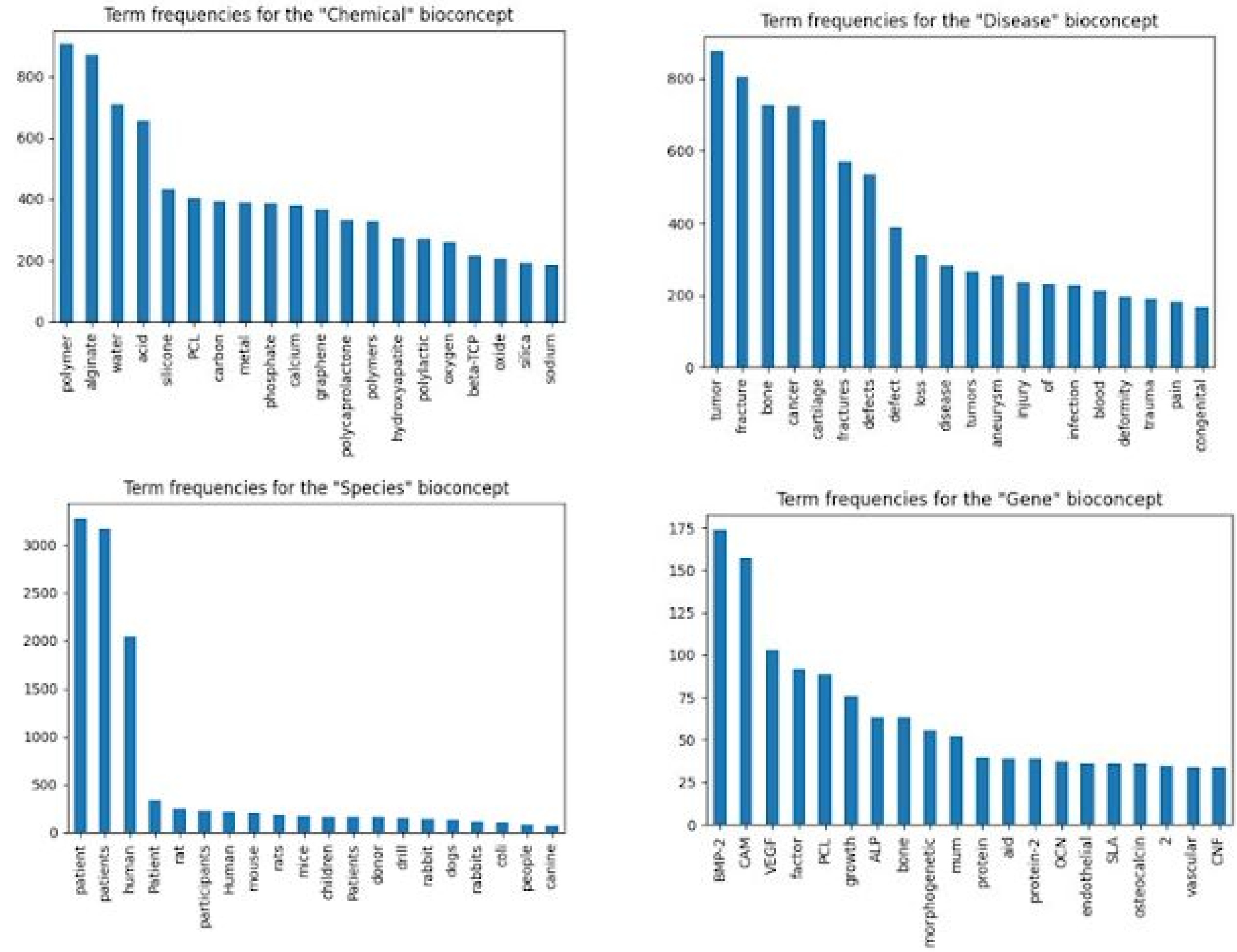


Figure 26 : Representation of the 20 most frequent terms in the annotations of each biocept using tf

After observing the most frequent terms from each bioconcept, we decided to calculate the tf-idf values of each word to know the most important words of each bioconcept compared to the total annotations. The results are presented in Figure 27.

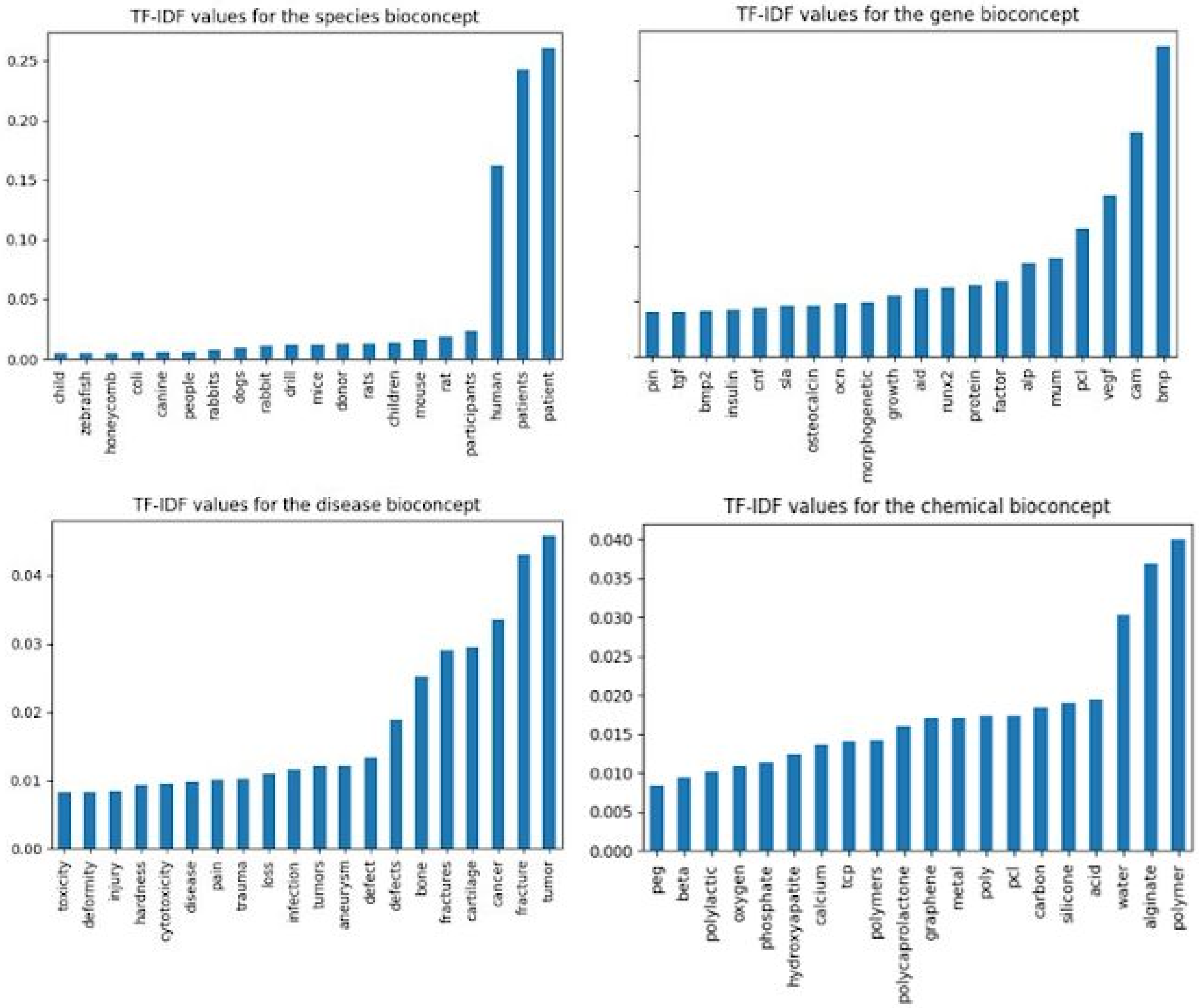


Figure 27 : Representation of the 20 most representative terms in the annotations of each bioconcept using tf-idf

If we compare the top 20 most frequent terms from each bioconcept to the 20 most common terms in terms of tf-idf values, we can find that the order of the terms in the horizontal axis varies a little bit and some new terms that appear for each biocept. For example, for the Species bioconcept, we observe the appearances of 2 new terms: “honeycomb” and “zebrafish”. So another animal model was discovered by using the tf-idf values which is the zebrafish. The term “honeycomb” is not relevant to this bioconcept and might refer to a type of structure that was 3D-printed or to the structure of an element (such as the graphene). For the Disease bioconcept, we discovered 2 new terms thanks to the tf-idf calculations: “cytotoxicity” and “toxicity”. These terms refer to the main issue with 3D-printing for medical applications which is the toxicity of the products. There is research to evaluate the potential of 3D-printed products to be toxic for the human body and especially for the cells. So in this case the new terms that we found do not refer to a specific disease that we want to cure with the 3D-printing technology but more to an essential parameter that we need to take into account while fabricating 3D-printed implants for medical purposes. Now if we look at the “Chemical” bioconcept, we found new materials that can be used to make 3D-printed scaffolds such as “tcp” which stands for tricalcium phosphate and “peg” which stands for polyethylene glycol. Finally for the “Gene” bioconcept, we did not find more relevant information using tf-idf values. In fact, it was quite hard to analyse this bioconcept because there were a lot of ambiguous terms in it (i.e. terms with 2 or more meanings) and knowing if the terms refers to a gene or not requires context, i.e. a complex NLP procedure than lexical annotation.

3.2.3 Data extraction using topic mining/ topic modeling

3.2.3.1 LDA model

Here, topic discovery was used to extract information about the 3D-printing literature available from Pubmed. But before analysing these topics, it is important to note that topic modeling algorithms such as LDA or hLDA create sets of words that the algorithms think are related based on patterns in the corpus. But what the algorithms can not do is to explain the topics’ meaning. In fact, it is the programmer himself that needs to do the interpretation part and that can be difficult. Moreover we have to take into account that documents assigned to a particular topic do not necessarily contain all of the words included in that topic.

Figure 28 represents the Intertopic Distance Map obtained with the pyLDAvis visualization package. In this map, the topics are represented by circles in a 2D plane (PC1,PC2). The relative position of the circles in the map is determined by an algorithm which evaluates the divergence between topics and then projects the points in the plan to facilitate the user's understanding. Typically the closest the topics are in the map, the more words they have in common.

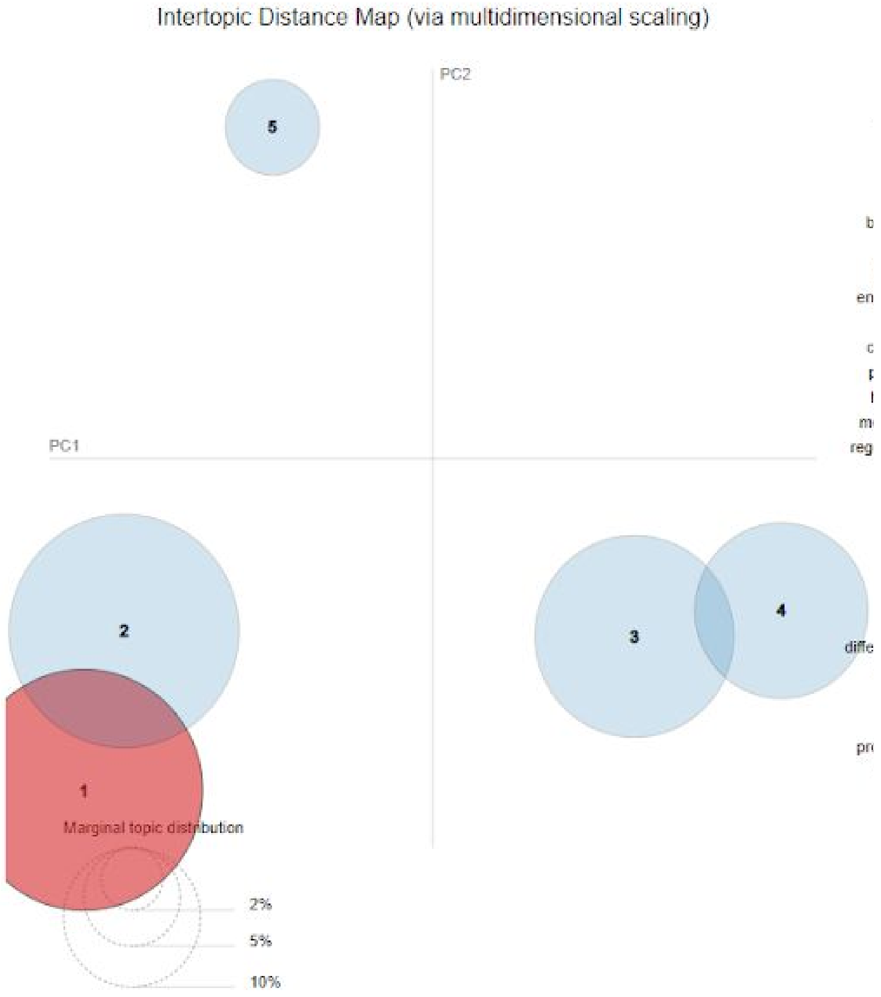


Figure 28 : Intertopic Distance Map from LDA model

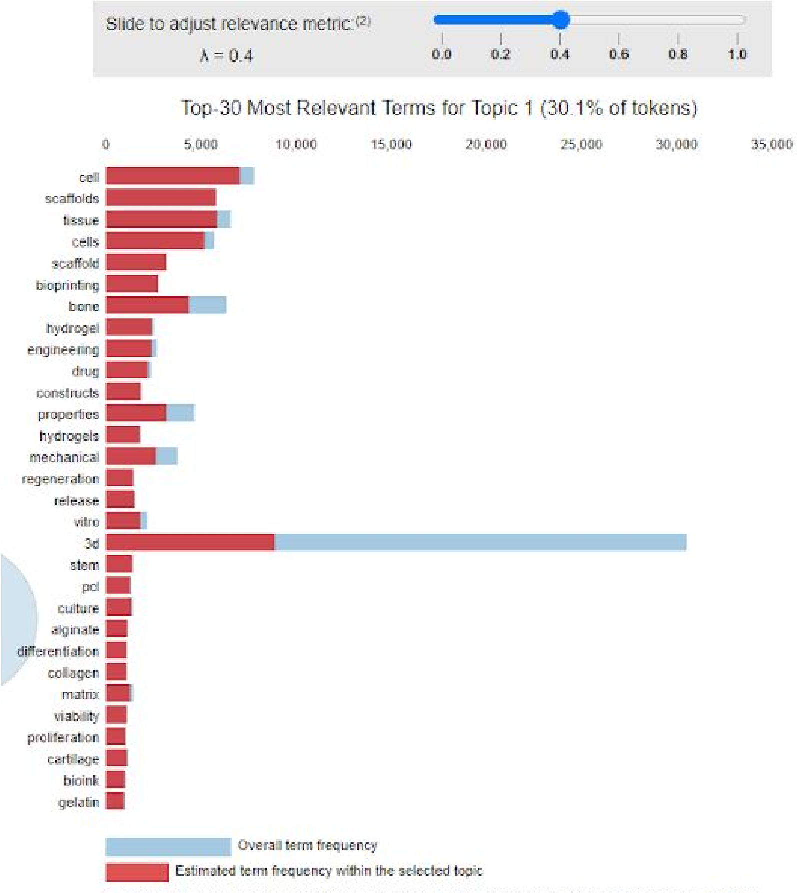


Figure 29 : topic 1

As you can see in Figure 29, there are a lot of terms in topic 1 that we already found in the in vivo/in vitro category after doing text analysis. It is the case with terms like “tissue”, “cell”, “scaffold”, “bone”, “hydrogel” and so on. The appearance of the terms “vitro”, “viability”, “culture”, “proliferation” confirms that topic 1 contains a lot of articles from the in vivo/in vitro category. Indeed during the biopring process, different approaches are possible. One of them is to start by performing a biopsy to isolate cells from the patient (or a donor). Then, harvested cells are cultured in order to get a bigger number of cells: it is the expansion or proliferation step. Afterwards, these cells are mixed with polymers (and functional peptides) to create a bioink that will be fed into a 3D-printer. This leads to a 3D-printed product that can be implanted inside the patient’s body. NB: The terms ”gelatin”, “alginate” and “pcl” (for polycaprolactone) that are present in the list refers to different polymers that can be used in the formulation of the bioink. Topic 1 could be entitled tissue engineering.

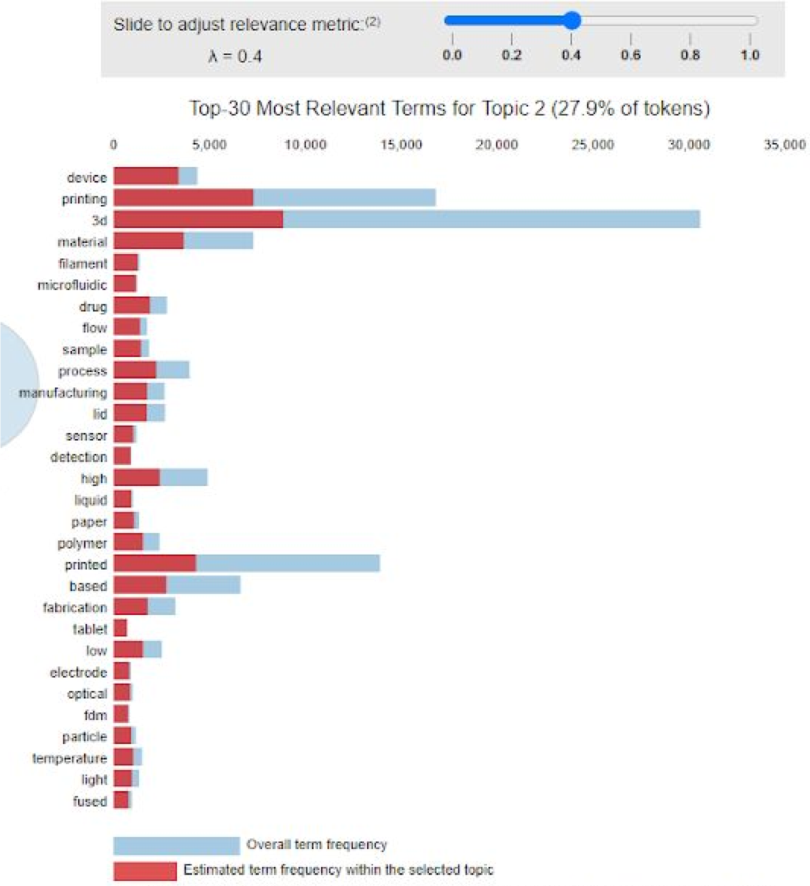


Figure 30 : topic 2

The topic 2 (see Figure 30) may gather articles dealing with two common applications of the 3D-printing technology in healthcare. The terms “drug”, “microfluidic”, “tablet” refer to articles where drug delivery systems were developed using 3D-printing and tested with maybe microfluidic testing devices [50], [51]. These 3D-printed systems are of high interest because they enable tailor-made formulation with customizable size and shape, and release rate. Indeed 3D-printing has also been used in the past few years to design personalized medication for patients that suffer from diseases such as cancer so the technology has a great potential in pharmacology [52]. The terms “electronic”, “detection”, “device”, “sensor” and “electrodes'', on the other hand, refer to articles focused on the fabrication of electronic sensors for biomedical applications such as toxin detection, heart rate detection, etc which can facilitate medical diagnostics [53,[54]]. The term “temperature” might also refer to this application because it is a common issue with that technology: the sensors are usually affected by the temperature which can be problematic when working in physiological conditions.

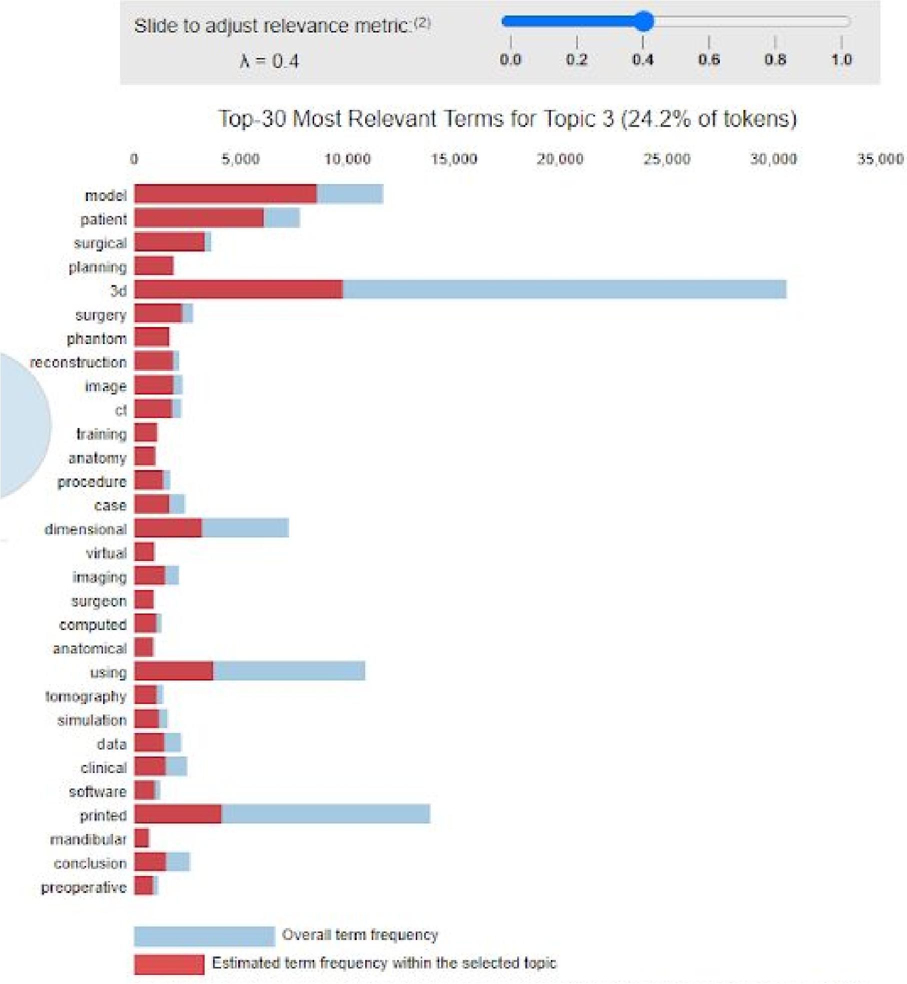


Figure 31 : topic 3

If we observe the terms that define topic 3 in Figure 31, we can see the importance of the Computer Aided Design (CAD) and the Computer Aided Manufacturing (CAM) in the 3D-printing technology. The terms “virtual”, “imaging”, “computed”, “ct”, “image”, “tomography”, “simulation”, “3d” and “software” refer to the same process of fabrication which involves 3D-imaging and CAD to make 3D-products. The process starts by taking a Computed Tomography (CT) scan of part of the body that needs to be replaced. These 3D-images will then serve to generate a 3D-virtual model of the reconstructed tissue or bone using CAD software. Afterwards the 3D-model will be sent to the 3D-printer that can fabricate the final product. It is the CAM step of the process. Finally there is the implantation part or the surgery where we replaced the damaged tissue or bone by the 3D-implant. But this technique does not necessarily serve to fabricate 3D-printed implants, it is also used to create 3D-printed surgical guides that help the surgeon during the operation and reduce drastically the operation time. These guides facilitate the placement of the implant during surgery. [55]

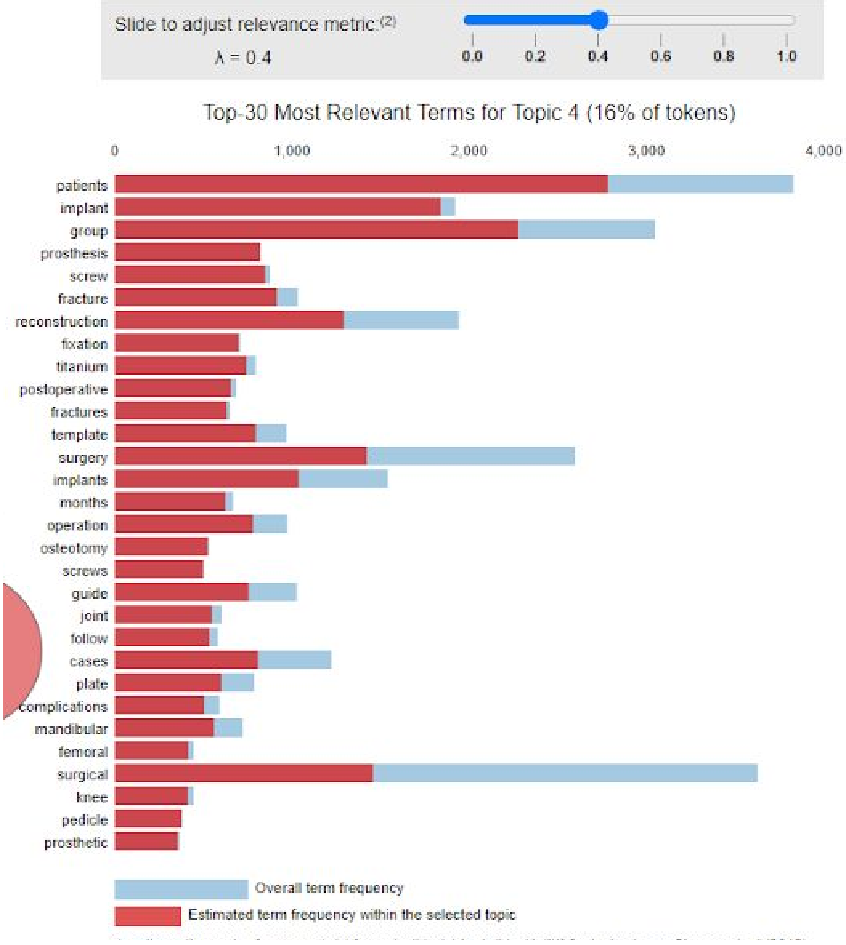


Figure 32 : topic 4

If we look at topic 4 (see Figure 32), we can find terms “patient”, “group” ,”surgery”, “surgical”, “operation”, “postoperative”, “osteotomy”, “prosthesis”, “implants”, etc. (NB: According to Merriam-Webster, osteotomy is a surgical operation in which a bone is divided or a piece of bone is excised as to correct a deformity.) So it seems that topic 4 refers to the clinical studies from the 3D-printing literature. The terms “mandibular”, “femoral”, “knee” indicate the principal locations where 3D-printed products are implanted in the human body or the main parts of the body that are being replaced by a 3D-printed product. Here the 3D-printing technology permits the "reconstruction" of functional units in humans. If we decrease the relevance parameter, we can find terms that refer to other parts of the body such as "acetabular", "hip", "pelvic" and "tibial" which means that we can print almost any bone of the body.

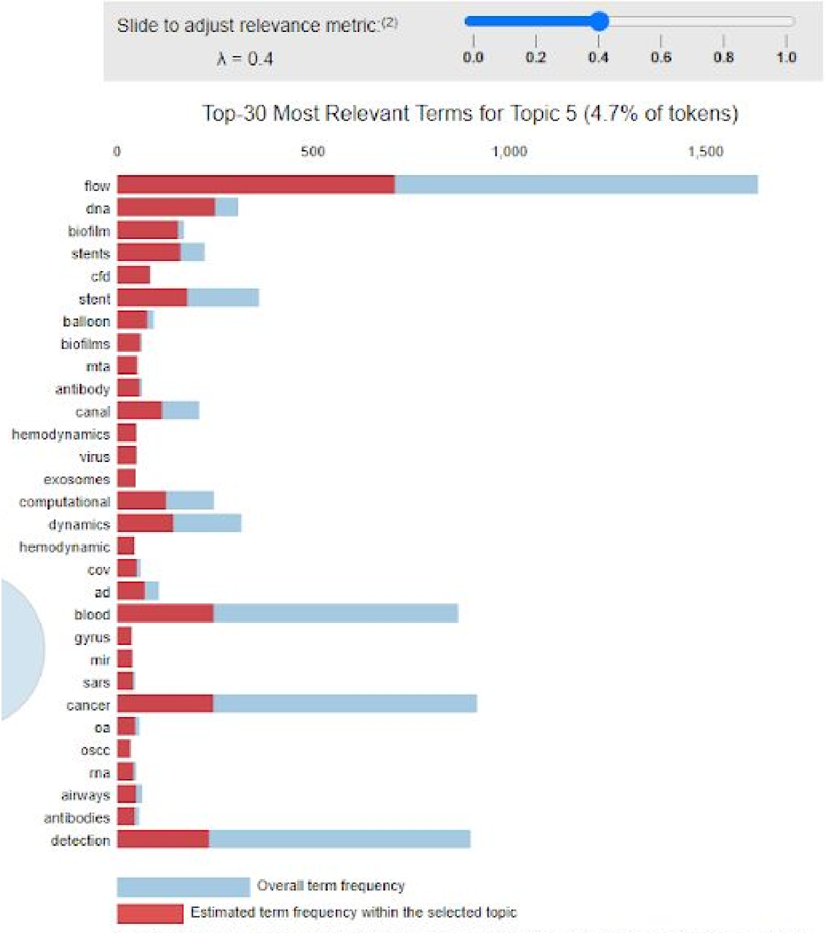


Figure 33 : topic 5

In Figure 3 which represents topic 5, we find terms such as "sars", "cov", "virus, "airways" which refer to a severe acute respiratory syndrome coronavirus. This topic is relatively isolated in the intertopic distance map (see Figure 28) and not as big as the others because articles in this topic are all recent. You will not find publications made before 2020 and the arrival of COVID-19 so this is an emerging subject, distinct.

But topic 5 does not only deal with COVID-19, there are also terms corresponding to cardiovascular applications of the 3D-printing technologies. It is the case with terms like “stents'', “hemodynamic”, “blood”, etc.

So to conclude this part we can say that the LDA model permitted us to discover the dominant topics in the 3D-printing literature. The topics could be centered on applications of the technology (sensors, surgical guides, drug delivery systems), technical tools (CAD), or disease context (SARS).

3.2.3.2 hLDA model

In this section, we will present the results that we obtained using the hLDA model. But before analysing the results, it is important to mention that hLDA sometimes offer more coherent topics because it is hierarchical. In total, 46 topics were discovered in the corpus, with six words representing each topic (see Figure 35). The highest topic in the hierarchy was represented by the terms “model, using, patient, printed, method, surgical" and contained over 5000 abstracts. As we can see, these terms describe the 3D-printing technology in general. They are not specific. The second and third largest topics, with over 2000 abstracts each, were generated by the following subsets of words: "scaffold, tissue, cell, bone, study, stem" and "cell, hydrogel, bioprinting, tissue, bioink, construct". We observe that both of these topics were related to cells and tissues. These two topics seem to represent the in vivo/in vitro category, containing similar terms as topic 1 from the LDA model.

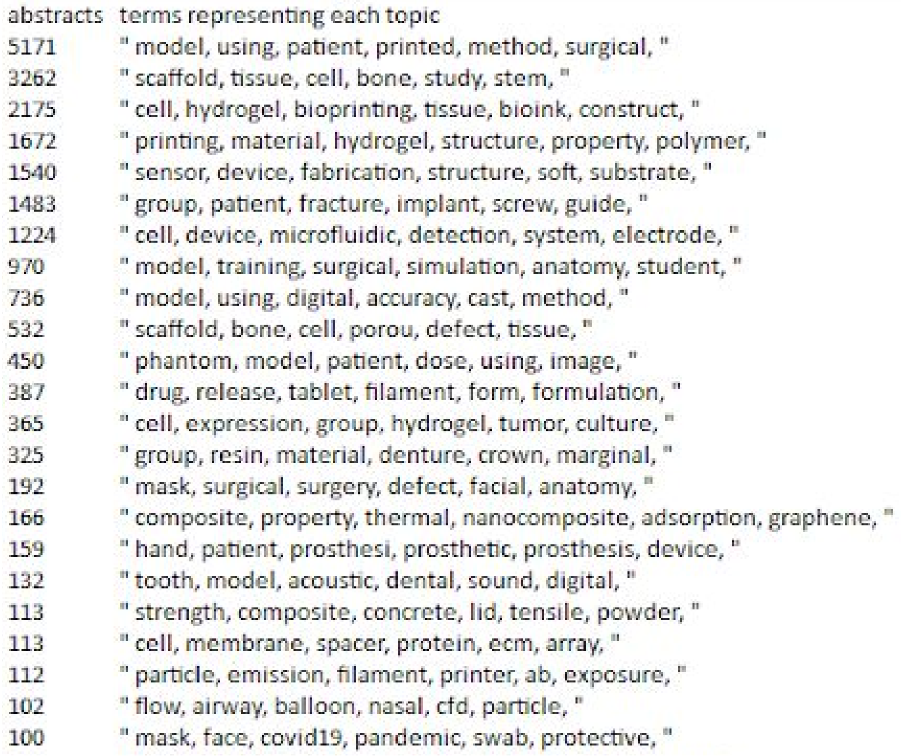
Now if we observe the topics with more than 100 abstracts (see Figure 34), we can find some terms such as "resin”, “denture”, “crown”, "tooth”, “dental”. These key words refer to one of the main applications of the 3D-printing technology in the medical field which is dentistry. Indeed this technology enables dentists to manufacture crowns very quickly compared to the traditional techniques (such as injection molding). With 3D-printing, we create a 3D-model directly from the scan of the patient teeth using CAD. Then the CAD model is fabricated using a 3D-printer and we get our final implant that is specific to the patient. [56]

In the same vein, we can find other applications of the technology by looking at the different terms in the topics. In several topics for example, there were terms like "sensor, device, fabrication, structure, soft, substrate", "cell, device, microfluidic, detection, system, electrode" or "drug, release, tablet, filament, form, formulation" which refer to 3D-printed drug delivery systems and 3D-printed electronic sensors that can detect physiological parameters.

Then if we look at the following topic "model, training, surgical, simulation, anatomy, student", we discover how the 3D-printing technology can be used as an education tool to help medical students learn more about the human anatomy. With 3D-printing, we can recreate accurate 3D-models of the human organs on which students can practise their skills. [55],[57] These 3D-printed models will not only help the medical students but also the surgeons who prepare before an intervention. [58] We can also mention that this technology could limit the use of cadavers in training of medical students which presents ethical issues at the moment.

Finally by looking at the results, we can also see how the current context of pandemic affected the 3D-printing literature. In total, 100 abstracts were linked to the topic "mask, face, covid19, pandemic, swab, protective". Indeed, in March 2020, teams of researchers started to develop 3D-printed face masks to address the lack of surgical masks. Their goal was to produce effective reusable face masks with global availability. This proves how promising the 3D-printing technology is and how it can be applied to everyone. [59]

NB: the two last topics that we mentioned give us an idea of the ‘non biomaterials’ studies with 3D-printing in Pubmed.



abstracts	terms representing each topic
5171	" model, using, patient, printed, method, surgical, "
3262	" scaffold, tissue, cell, bone, study, stem, "
2175	" cell, hydrogel, bioprinting, tissue, bioink, construct, "
1672	" printing, material, hydrogel, structure, property, polymer, "
1540	" sensor, device, fabrication, structure, soft, substrate, "
1483	" group, patient, fracture, implant, screw, guide, "
1224	" cell, device, microfluidic, detection, system, electrode, "
970	" model, training, surgical, simulation, anatomy, student, "
736	" model, using, digital, accuracy, cast, method, "
532	" scaffold, bone, cell, porou, defect, tissue, "
450	" phantom, model, patient, dose, using, image, "
387	" drug, release, tablet, filament, form, formulation, "
365	" cell, expression, group, hydrogel, tumor, culture, "
325	" group, resin, material, denture, crown, marginal, "
192	" mask, surgical, surgery, defect, facial, anatomy, "
166	" composite, property, thermal, nanocomposite, adsorption, graphene, "
159	" hand, patient, prosthesi, prosthetic, prosthesis, device, "
132	" tooth, model, acoustic, dental, sound, digital, "
113	" strength, composite, concrete, lid, tensile, powder, "
113	" cell, membrane, spacer, protein, ecm, array, "
112	" particle, emission, filament, printer, ab, exposure, "
102	" flow, airway, balloon, nasal, cfd, particle, "
100	" mask, face, covid19, pandemic, swab, protective, "

Figure 34 : List of the topics in the 3D-printing literature with more than 100 abstracts

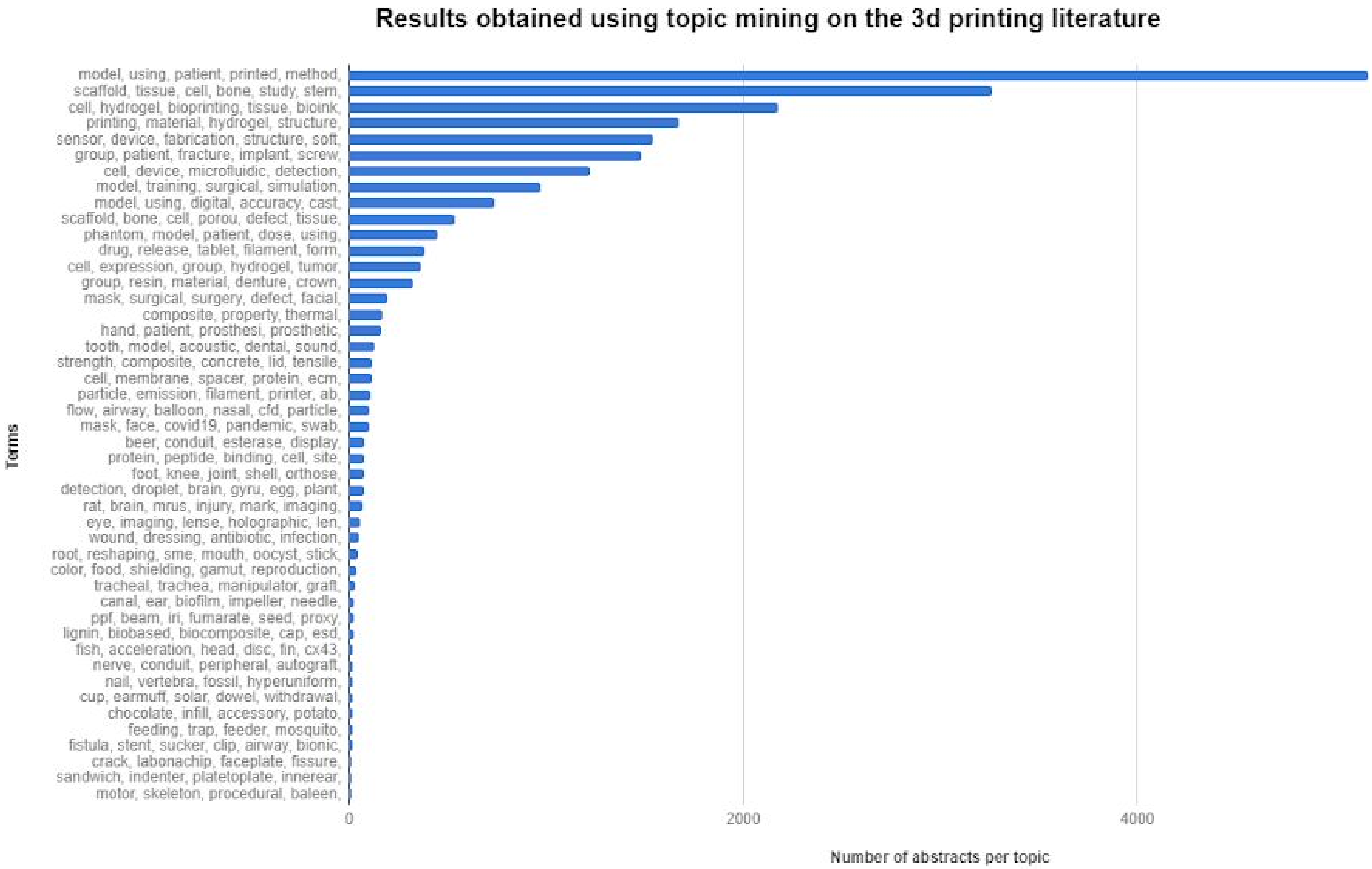


Figure 35 : Representation of the 46 topics obtained with the hLDA model

So to conclude with the text analysis and data extraction part, we can say that we followed different approaches. We started by extracting the most frequent terms of each category using tf and tf-idf, then we analyzed PubTator annotations of the entire 3D-printing corpus using tf and tf-idf again and we finished with topic mining techniques such as LDA and hLDA to discover latent topics in the 3D-printing literature from Pubmed. These three approaches have all their advantages and their drawbacks but each of them was useful for us whether it confirmed the findings from another approach or brought new information.

The first approach permitted us to extract information directly from the results of the text classification. With this technique, we discovered the more relevant terms for each type of study. We found out that the in vivo/in vitro articles were dealing a lot with components of the bioink such hydrogels, extracellular matrix and cells. Between those we can tell that the osteoblasts are the cells that are mostly used in preclinical studies and the gelatin-based hydrogels are the most common (such as GelMA). The articles from the clinical category, on the other hand, contain a lot of information on the parts of the body that are being replaced by 3D-printed implants in clinical trials. We found out that the sacrum replacement was the most mentioned intervention. However this category also contained diseases that are being cured using the 3D-printing technology such as diplopia and mitral valve disease. The analysis of the non biomaterials category revealed the rising importance of metamaterials in the 3D-printing sector. It also showed the application of the technology for producing microfluidic devices.

Then PubTator permitted us to identify bioconcepts in the 3D-printing literature. The “Disease” bioconcept gave us information on the diseases that can be cured using this technology such as cancer or aortic aneurysms (an arterial disease). It can also be applied to patients who had a major accident resulting in trauma, fractures, etc. In the “Disease” bioconcept, we also discovered “infection” which could correspond to a disease that can be induced by an implant or which could refer to a strategy to avoid infection, like antibacterial coating. Then the “Chemical” bioconcept emphasized the materials that can be used to make 3D-printed products for medical applications. Among those materials such as PCL, PEG, alginate and phosphate calcium were found which often serve as raw materials for scaffolds’ 3D-printing. The “Species” bioconcept showed the principal animal models that are used in laboratories like rabbits, mice, rats, dogs, and fishes. As you can see, PubTator annotations provided us a large amount of information on the 3D-printing literature. However PubTator we noticed that it did not always give us accurate results. We identified non relevant terms in the bioconcepts. For example, we encountered the term “CAM” in the Gene bioconcept which corresponds to Computer Aided Manufacturing. Similarly the term “PCL” appeared in this bioconcept which refers to the polycaprolactone polymer. In this case, the Chemical bioconcept was more accurate. So a manual validation is necessary while using this tool.

Finally with topic modeling, 46 topics were discovered in the 3D-printing corpus using the hLDA model and 5 using the LDA model. We observed an overlap between these two techniques, with topics extracted with the LDA model also present in the hLDA results. Indeed these two models allowed us to retrieve information about the applications of the 3D-printing technology in the medical field: fabrication of sensors, drug delivery systems, dental prosthetics, surgical guides, etc. Overall, hLDA provided more information.

Here is a summary of the principal findings obtained during this project:

- Clinical studies are less represented in the 3D-printing literature from Pubmed
- We can find terms like “patients” and “human” in all kinds of studies. Authors usually use the word “patient” to refer to the application of their 3D-printing product.
- The main applications of the 3D-printing technology in healthcare are: the fabrication of prosthetics and implants, the production of devices for medical diagnostics (such as biosensors or Lab-on-the-Chip), the fabrication of drug delivery systems, and the creation of surgical guides and dentistry.
- Among ‘non biomaterials’ studies, 3D-printed models are very useful to help medical students learn about anatomy and to prepare surgeons before an intervention.
- Between the parts of the body that are being replaced using 3D-printed products according to the Pubmed literature, we can cite: sacrum, atlantoaxial joint, calcaneus, thoracolumbar vertebrae, talus, scapula, mandible, femur, knee, acetabulum, hip, pelvis, teeth and tibia.
- In in vivo/in vitro articles, the most mentioned intervention is the sacrum replacement.
- A total of 5 diseases were identified in the 3D-printing corpus including cancer, aortic aneurysm (an arterial disease), mitral valve disease, diplopia and infection (+ covid 19)
- Between the numerous biomaterials that can be used to manufacture 3D-printing products for medical applications alginate, GelMA, PCL, PEG and calcium phosphate seem to be the most mentioned.
- Most of the animal models used in in vitro/in vivo studies include rabbits, mice, rats, and dogs.
- New materials are constantly being tested in 3D-printing such as metamaterials which seem to be really promising.

3.3. Methodological considerations and project limitations

First of all, we are going to talk about the text classification and how we could have introduced bias in the results. As already mentioned in the previous parts of the report, we had to go through manual curation from Pubmed to create our 3D-printing testing set. We decided to use the abstracts because not all the articles are freely available online. But even if several articles were really clear about their content (just by looking at the title, you could learn whether an in vitro study or a clinical study was performed), sometimes it was not so easy to understand what the authors did during their study based on the abstract. There were some cases where the abstract was very vague and where the title or the MeSH terms/keywords did not give us more information. This might have led to errors in our manual classification resulting in bias in the supervised text classification. Moreover we also used data from the DEBBIE datasets which were originally created using manual curation. Because only two human curators participated in the selection of the publications, a biais could have been introduced. Similarly the size of the training set was limited by the burden of manual curation. Indeed for the multinomial classification the training is relatively small compared to other training sets that can be found throughout the internet : there are around 1,000 abstracts in each category but given the size of the literature (>11,000 abstracts), this size is still reasonable.

For the multinomial classification, we decided to work with 3 classes: non biomaterials studies, clinical studies and in vivo/in vitro studies. We choose to put the in vivo and the in vitro studies together because when we were creating our 3D-printing testing set by manually looking at articles from Pubmed, we found that a lot of studies combined in vitro and in vivo experiments. For this reason we decided to make one class with all of the articles that were either in vivo, in vitro or both at the same time, i.e. gathering all types of pre-clinical studies in one category. But what if we took them separately ? Future work on this project should implement a multi-label classifier that can attribute more than one label to an abstract to see if it makes any difference to distinguish in vivo studies from in vitro studies or from studies that are both in vivo and in vitro. For example, this classifier could have started by telling whether an abstract refers to biomaterials or not and then specifying the type of study (in vitro, in vivo, clinical or in vitro + in vivo at the same time) in case of biomaterials abstracts.

Finally with topic modeling, we encountered some limitations due to computational power. We were not able to extract more than five topics using the LDA model because more processing power was required for the task.

Conclusion

In this paper, a text classification model based on abstracts has been proposed and tested on 3D-printing articles retrieved from Pubmed. This is a relatively new approach and there are not many studies on the subject in the literature. The initial results that we obtained with the multinomial classification are satisfactory and permitted us to classify the entire 3D-printing literature available from Pubmed. We have been testing four classification models to build a multinomial classifier: SGD, Random Forest, Multinomial NB and k-NN. With SGD, we managed to get 0.92 of accuracy and 0.89 of F1-score (which are pretty good results for automatic text classification). These results can be explained by several reasons: first the abstract/summary of a text covers the main ideas of the whole text so it can be used to identify the topic and the kind of study that was made; then by using the right text segmentation method we do not lose too much semantic information; finally the chosen text classification model is effective. Thanks to the multinomial classifier, we managed to classify a total 11,153 abstracts about 3D-printing into 3 categories: clinical studies, in vitro/in vivo studies and non biomaterials studies. The use of text analysis techniques on the 3D-printing literature permitted several things: (1) to discover 5 diseases that are treated/addressed using the technology (mitral valve disease, diplopia, aortic aneurysm, infection and cancer, (2) to identify 14 parts of the body that can be replaced by 3D-printed implant, (3) to see which are the most commonly mentioned biomaterials in this field (alginate, GelMA, PCL, PEG and calcium phosphate) and (4) to learn new applications of the technology which do not deal with prosthetics or implants such as biosensors fabrication, drug delivery systems manufacturing or surgical guides creation. Future work could build on the collections and code from this project to develop a 2D-steps classifier which will first distinguish biomaterials articles from non biomaterials articles (i.e. binary classification) and then classify relevant articles into the following categories: in vitro studies, in vivo studies and clinical studies. Another approach would be to implement a multi-label classification model that can assign several tags to an abstract based on its content.

Economic Analysis

The main advantage of this project is the low costs it generated. Indeed, this project did not require the purchase of special equipment. All we needed was a computer and internet access. Likewise, all of the resources used during the project are available online and are completely free. For example, the programming part of the project was done using free software, open source Python libraries and openly available code. The datasets were created using tools from the United States National Library of Medicine. So if we want to assess the overall cost of the project, we simply have to estimate the hours worked on the computer and deduct the electricity consumption that has been generated. Moreover, we will also take into account the salary of the different actors of the project which are the student and his two supervisors. For this analysis, the student salary will be estimated to be the salary of a junior engineer (with 0 to 2 years of experience) that is to say 2500€/month approximately.

Table : Data regarding the numbers of hours worked on the computer, the power usage of the computer and the electricity tarification (in France)

Number of hours worked per day	Number of days worked per week	Duration of the project	Power used by the laptop	Electricity price of 1kWh
7h	**5 days**	**4 months**	**60W**	**7.80€**

First of all, we need to calculate the global electricity consumption generated by the project. To do so, we assumed that we dedicated approximately 35h per week for the project in a 4 months period. Indeed as a double diploma student, I still had some courses to follow with the UPC in parallel with this project so I had less time to work on it compared to regular master students. That is why we have 35h of work per week instead of 40h. As well, the 4 months period of time was chosen because at the beginning of the project I was mostly learning about Python so I have not started programming until the month of October. In the same vein, I spent the last month of the project working on the report and analysing my results which explains why I cut off 2 months from the duration of the project. Finally the power used by the laptop has been estimated to 60W. Taking these considerations into account, we obtained the following electricity consumption:

$$global\ electricity\ consumption\ =\ (60W * 7h * 5\ days * 4\ weeks * 4\ months)/1000\ =\ 33.6kWh$$

The cost associated with this electricity consumption can be easily calculated using the price of 1kWh. To find the price of the electricity, we went on the website “total.direct-energie.com” which is one of the main electricity suppliers in France and looked at the tarification grid for computers. [60]

$$\textit{cost generated by the electricity consumption} = 33.6 * 7.80€ = 262.08€$$

Table : Data regarding the numbers of hours worked by the actors of the project and their hourly wage

Number of hours worked by the student	Estimated hourly Pay for a junior engineer	Number of hours worked by each supervisor	Estimated hourly Pay for the supervisors
560h	**18.00€**	**30h**	**50.00€**

Then we estimated the labor costs of the project. To do so, we multiplied the average hourly pay of each actors by the number of hours worked and we obtained the following result:

$$\textit{salary of the student} = 560h * 18.00€ = 10080€$$

$$\textit{salary of the supervisors} = 30h * 50.00€ * 2 = 3000€$$

Afterwards, the labor costs and the electricity cost were additionated to get the global cost of the project.

$$\textit{global cost generated by the project} = 262.08€ + 10080€ + 3000€ \simeq 13400€$$

So we found out that the overall cost of this master thesis project was 13400€ which corresponds to approximately 3350€ per month if we do not count the purchase cost of a laptop which varies between 450€ to 2000€ depending on the model.

References

[1] Udayabhanu Jammalamadaka and Karthik Tappa. (PDF) Recent Advances in Biomaterials for 3D Printing and Tissue Engineering. (2018) Available from: https://pubmed.ncbi.nlm.nih.gov/29494503/ [accessed Jan 18 2021].

[2] PubMed. https://pubmed.ncbi.nlm.nih.gov/.

[3] Osnat Hakimi, Josep Luis Gelpi, Martin Krallinger, Fabio Curi, Dmitry Repchevsky, and Maria-Pau Ginebra. (PDF) *The Devices, Experimental Scaffolds, and Biomaterials Ontology (DEB): A Tool for Mapping, Annotation, and Analysis of Biomaterials' Data.* (February 2020)

[4] Elizabeth D. Liddy. (PDF) *Natural Language Processing*. (2001) Available from: https://surface.syr.edu/cgi/viewcontent.cgi?referer=http://scholar.google.fr/&httpsredir=1&article=1019&context=cnlp [accessed Dec 04 2020].

[5] Quoc Dinh Truong, Hiep Xuan Huynh, and Cuong Ngoc Nguyen. (PDF) *An abstract-based approach for text classification*. (March 2016) Available from: https://www.researchgate.net/publication/301328065_An_Abstract-Based_Approach_for_Text_Classification#fullTextFileContent [accessed Dec 04 2020].

[6] DEBBIE pipeline. https://github.com/ProjectDebbie/DEBBIE_pipeline

[7] Medline Ranker. http://cbdm-01.zdv.uni-mainz.de/~jfontain/cms/?page_id=4

[8] Gold standard set. https://github.com/ProjectDebbie/gold_standard_set

[9] Saeideh Kholgh Eshkalak, Erfan Rezvani Ghomi, Yunquian Dai, Deepak Choudhury, and Seeram Ramakrishna. (PDF) *The role of three-dimensional printing in healthcare and medicine.* (September 2020) Available from: https://www.sciencedirect.com/science/article/pii/S0264127520304743# [accessed Jan 12 2021].

[10] Dina Radenkovic, Atefeh Solou, and Alexander Seifalian. (PDF) *Personalized development of human organs using 3D-printing technology.* (December 2020) Available from: https://www.sciencedirect.com/science/article/pii/S0306987715004715 [accessed Jan 12 2021].

[11] Muhammad Pervez Akhter, Zheng Jiangbin, Irfan Raza Naqvi, Mohammed Abdelmajeed, Atif Mehmood, and Muhammad Tariq Sadiq. (PDF] *Document-Level Text Classification Using Single-Layer Multisize Filters Convolutional Neural Network.* (February 2020) Available from: https://www.researchgate.net/publication/339547689_Document-Level_Text_Classification_Using_Single-Layer_Multisize_Filters_Convolutional_Neural_Network [accessed Jan 18 2021]

[12] Kwan yi. (PDF) *Survey of Automated Classification based on Classification Schemes: A Study of Mythology*. (November 2014) Available from: https://www.researchgate.net/publication/271505822_Survey_of_Automated_Classification_based_on_Classification_Schemes_A_Study_of_Methodology [accessed Jan 18 2021]

[13] Sebastien Raschka. (PDF) *Naive Bayes and Text Classification I: Introduction and Theory* (October 2014) Available from: https://arxiv.org/pdf/1410.5329.pdf [accessed Dec 07 2020]

[14] Scikit-learn library. https://scikit-learn.org/stable/

[15] Agung B. Prasetijo, R. Rizal Isnanto, Dania Eridani, Yosua Alvin Adi Soetrisno, M. Arfan, and Aghus Sofwan. (PDF) *Hoax Detection System on Indonesian News Sites Based on Text Classification using SVM and SGD.* (October 2017) Available from: http://eprints.undip.ac.id/69088/1/C06_CSI-09_Prasetijo.pdf [accessed Dec 07 2020]

[16] Ameni Bouaziz , Christel Dartigues-Pallez , Celia da Costa Pereira , Frederic Precioso, and Patrick Lloret. PDF) *Short Text Classification Using Semantic Random Forest*. (September 2020) Available from: https://www.researchgate.net/profile/Celia_Da_Costa_Pereira/publication/300335247_Short_Text_Classification_Using_Semantic_Random_Forest/links/5cff88474585157d15a225d0/Short-Text-Classification-Using-Semantic-Random-Forest.pdf[accessed Jan 12 2021]

[17] Marina Skurichina and Robert P. W. Duin. (PDF) Bagging, Boosting and the Random Subspace Method for Linear Classifiers. (2002) Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.19.5455&rep=rep1&type=pdf [accessed 12 Jan 2021]

[18] Bruno Trstenjak, Sasa Mikac, and Dzenana Donko. (PDF) *KNN with TF-IDF Based Framework for Text Categorization.* (March 2014) Available from: https://www.sciencedirect.com/science/article/pii/S1877705814003750 [accessed Dec 07 2020]

[19] Ah-Hwee Tan, Kent Ridge,and Heng Mui Keng Terrace. (PDF) *Text Mining: The state of the art and the challenges*. (November 2000) Available from: https://www.researchgate.net/publication/2471634_Text_Mining_The_state_of_the_art_and_the_challenges [accessed Jan 18 2021]

[20] Swapna Gottipati,Venky Shankararaman, and Jeff Rongsheng Lin. (PDF) *Text analytics approach to extract course improvement suggestions from students' feedback.* (2018) Available from: https://ink.library.smu.edu.sg/cgi/viewcontent.cgi?article=5079&context=sis_research [accessed Jan 18 2021]

[21] PubTator Central. https://www.ncbi.nlm.nih.gov/research/pubtator/

[22] Gene database. https://www.ncbi.nlm.nih.gov/gene?term=.

[23] CTD database. http://ctdbase.org/

[24] Taxonomy database. https://www.ncbi.nlm.nih.gov/taxonomy

[25] MeSH database. https://www.ncbi.nlm.nih.gov/mesh/?term=.

[26] SNP database. https://www.ncbi.nlm.nih.gov/snp/.

[27] Rubayyi Alghamdi and Khalid Alfalqi. (PDF) *A Survey of Topic Modeling in Text Mining*. (2015) Available from: https://thesai.org/Downloads/Volume6No1/Paper_21-A_Survey_of_Topic_Modeling_in_Text_Mining.pdf [accessed Jan 18 2021]

[28] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. (PDF) *Latent Dirichlet Allocation.* (2003) Available from: https://jmlr.org/papers/volume3/blei03a/blei03a.pdf [accessed Jan 12 2021]

[29] Shashank Kapadia GitHub account. https://github.com/kapadias/mediumposts/blob/master/natural_language_processing/topic_modeling/notebooks/Introduction%20to%20Topic%20Modeling.ipynb

[30] pyLDAvis package. https://pypi.org/project/pyLDAvis/

[31] David M. Blei, Thomas L. Griffiths, Michael I. Jordan, and Joshua B. Tenenbaum. (PDF) *Hierarchical Topic Models and the Nested Chinese Restaurant Process.* (2004) Available from: http://www.cs.columbia.edu/~blei/papers/BleiGriffithsJordanTenenbaum2003.pdf [accessed Jan 03 2021]

[32] Pingan Liu, Lei Li, Wei Heng, and Boyuan Wang. (PDF) *HDLA based text clustering* (October 2012) Available from: https://www.researchgate.net/publication/261201482_HLDA_based_text_clustering [accessed Jan 03 2021]

[33] Blei, D. M., Griffiths, T. L., and Jordan, and M. I. (PDF). *The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies.* (January 2010) Available from: http://cocosci.princeton.edu/tom/papers/ncrp.pdf [accessed Jan 03 2021]

[34] Joe Wandy GitHub account. https://github.com/joewandy/hlda/blob/master/notebooks/bbc_test.ipynb

[35] Cristian Padurariu, and Mihaela Breaban. (PDF) *Dealing with Data Imbalance in Text Classification.* (January 2019) Available from: https://www.researchgate.net/publication/336538175_Dealing_with_Data_Imbalance_in_Text_Classification [accessed Dec 07 2020]

[36] Ebot. https://www.ncbi.nlm.nih.gov/Class/PowerTools/eutils/ebot/ebot.cgi

[37] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. (PDF) Text Classification Algorithms: a Survey. (April 2019) Available from: https://www.researchgate.net/publication/332463886_Text_Classification_Algorithms_A_Survey [accessed Jan 04 2021].

[38] Vandana Korde. (PDF) *Text Classification and Classifiers:A Survey*. (March 2012) Available from:https://www.researchgate.net/publication/276196340_Text_Classification_and_ClassifiersA_Survey [accessed Dec 07 2020]

[39] Train.test.split function in Scikit-learn
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

[40] Cross-validation. https://scikit-learn.org/stable/modules/cross_validation.html

[41] Scikit-learn library. https://scikit-learn.org

[42] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html

[43] Maqbool Ali, Syed Imran Ali, Dohyeong Kim, Taeho Hur, Jaehun Bang, Sungyoung Lee, Byeong Ho Kang, Maqbool Hussain, and Fengfeng Zhou.(PDF) *uEFS: An efficient and comprehensive ensemble-based feature selection methodology to select informative features*. (August 2018) Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6112679/ [accessed Dec 07 2020]

[44] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html

[45] https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html

[46] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

[47] NLTK library. https://www.nltk.org/

[48] Benjamin Ho, Sum Huan Ng, Holden King Ho Li, and Yong-jin Yoon. (PDF) *3D Printed Microfluidics for Biological Applications.* (July 2015) Available from:
https://www.researchgate.net/publication/280404930_3D_Printed_Microfluidics_for_Biological_Applications [accessed Jan 12 2021]

[49] Boris Kosic, Aleksandra Dragicevic, Zorana Jeli, and Gabriel-Catalin Marinescu. (PDF) *Application of 3D Printing in the Metamaterials Designing*. (September 2019) Available from : https://link.springer.com/chapter/10.1007/978-3-030-30853-7_1 [accessed Jan 12 2021]

[50] Sarwar Beg, Waleed H. Almalki, Arshi Malik, Mohd Farhan, Mohammad Aatif, Ziyaur Rahman, Nabil K. Alruwaili, Majed Alrobaian, Mohammed Tarique, and Mahfoozur Rahman. (PDF). *3D-Printing for drug delivery and biomedical applications.* (July 2020) Available from:
https://www.sciencedirect.com/science/article/pii/S1359644620302841#:~:text=3D-printing%20technology%20has%20been,shift%20in%20the%20healthcare%20industry.[accessed Jan 12 2021]

[51] Manisha Pandey, Hira Choudhury, Joyce Lau Chui Fern, Alice Teo Kee Kee, Janice Kou, Jane Lee Jia Jing, How Chiu Her, Hong Sin Yong, Hon Chian Ming, Subrat Kumar Bhattamisra, and Bapi

Gorain. (PDF). *3D-Printing for oral drug delivery: a new tool to customize drug delivery.* (March 2020) Available from:
https://link.springer.com/article/10.1007%2Fs13346-020-00737-0 [accessed Jan 12 2021]

[52] Dolores R. Serrano, Mari C. Terres, and Aikaterini Lalatsa. (PDF). Applications of *3D-printing in cancer.* (July 2018) Available from:
https://www.researchgate.net/publication/327028869_Applications_of_3D_printing_in_cancer [accessed Jan 12 2021]

[53] Yuanyuan Xu, Xiaoyue Wu, Xiao Guo, Bin Kong, Min Zhang, Xiang Qian, Shengli Mi, and Wei Sun. (PDF). *The Boom in 3D-Printed Sensor Technology.* (May 2017) Available from:
https://www.researchgate.net/publication/317108041_The_Boom_in_3D-Printed_Sensor_Technology [accessed Jan 12 2021]

[54] Tao Han, Sudip Kundu, Anindya Nag, and Yongzhao Xu. (PDF). *3D-Printed Sensors for Biomedical Applications: A Review.* (April 2019) Available from:
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6480222/#__ffn_sectitle [accessed Jan 12 2021]

[55] Philip Tack, Jan Victor, Paul Gemmel, and Lieven Annemans. (PDF). *3D-printing techniques in a medical setting: a systematic review.* (October 2016) Available from:
https://link.springer.com/article/10.1186/s12938-016-0236-4 [accessed Jan 12 2021]

[56] M.Figliuzzi, F. Mangano, and C. Mangano. (PDF) *A novel root analogue dental implant using CT scan and CAD/CAM: selective laser melting technology.* (February 2012) Available from: https://pubmed.ncbi.nlm.nih.gov/22377004/ [accessed Jan 03 2021]

[57] Augusto Palermo, and Bernardo Innocenti. (PDF). *The role of 3D-Printing in Medical Applications: A State of the Art.* (March 2019) Available from:
https://www.hindawi.com/journals/jhe/2019/5340616/ [accessed Jan 12 2021]

[58] Qian Yan, Hanhua Dong, Jin Su, Jianhua Han, Bo Song, Qingsong Wei, and Yusheng Shi. (PDF). *A Review of 3D-Printing Technology for Medical Applications.* (October 2018) Available from:
https://www.sciencedirect.com/science/article/pii/S2095809917306756 [accessed Jan 12 2021]

[59] Sven Duda, Sascha Hartig, Karola Hagner, Lisa Meyer, Paula Wessling Intriago, Tobias Meyer, and Heinrich Wessling. (PDF) *Potential risks of a widespread use of 3D printing for the manufacturing of face masks during the severe acute respiratory syndrome coronavirus 2 pandemic.* (December 2020)
Available from: https://www.futuremedicine.com/doi/full/10.2217/3dp-2020-0014 [accessed Jan 03 2021]

[60]
https://total.direct-energie.com/particuliers/electricite/tout-savoir-electricite/consommation-electricite

FIGURES

Figure 1: screenshot from pubmed.ncbi.nlm.nih.gov [Oct 2020]

Figure 2:
https://raw.githubusercontent.com/ritchieng/machine-learning-stanford/master/w3_logistic_regression_regularization/multiclass_classification.png

Figure 3:
https://cdn.analyticsvidhya.com/wp-content/uploads/2019/05/Capture.jpg1-300x136.jpg

Figure 4:
https://rasbt.github.io/mlxtend/user_guide/general_concepts/gradient-optimization_files/ball.png

Figure 5:
https://res.cloudinary.com/dyd911kmh/image/upload/f_auto,q_auto:best/v1526467744/voting_dnjweq.jpg

Figure 8: screenshot from Pubtator Central
(link: https://www.ncbi.nlm.nih.gov/research/pubtator/)

Figure 9:

https://community.alteryx.com/t5/image/serverpage/image-id/124666i826A507A4B5E3107?v=1.0

Figure 11:
https://github.com/joewandy/hlda/blob/master/notebooks/bbc_test.ipynb

Figure 13:
https://media.mljar.com/blog/validation-learning-not-memorizing/dataset.jpg

Figure 14:
https://monkeylearn.com/static/507a7b5d0557f416857a038f553865d1/5040b/text_process_training.png

Figure 15:
https://scikit-learn.org/stable/_images/grid_search_cross_validation.png

Figure 16:
https://monkeylearn.com/static/afa7e0536886ee7152dfa4c628fe59f0/5040b/text_process_prediction.png

Figure 36:

https://github.com/ProjectDebbie/DEBBIE_pipeline/blob/master/Pipeline_overview.png

Annex A: DEBBIE pipeline

The DEBBIE automated pipeline (mckitric et al, unpublished data) is tool that was created to retrieve biomaterials abstracts from PubMed and annotate them using multiple lexical tools such as DEB (The Device, Experimental scaffolds and medical Device ontology) and the MeSH thesaurus. The annotated abstracts are then deposited in a dedicated database. The database search page is accessible from this link: debbie.bsc.es/search/.

The pipeline process can be divided into the 5 following steps (see Figure 36):

1. Periodic abstract retrieval from PubMed
2. Standardization of the abstract text
3. Binary classification (relevant VS non relevant to the biomaterials field) using an SVM classification model
4. Gate-based annotation of the biomaterials abstracts using terms from DEB and other manually-selected classes from open ontologies such as Nederlandse Publieke Omroep, National Cancer Institute Thesaurus and UBERON
5. Conversion of the resulting gate files to the appropriate format and deposition in the DEBBIE Mongo database

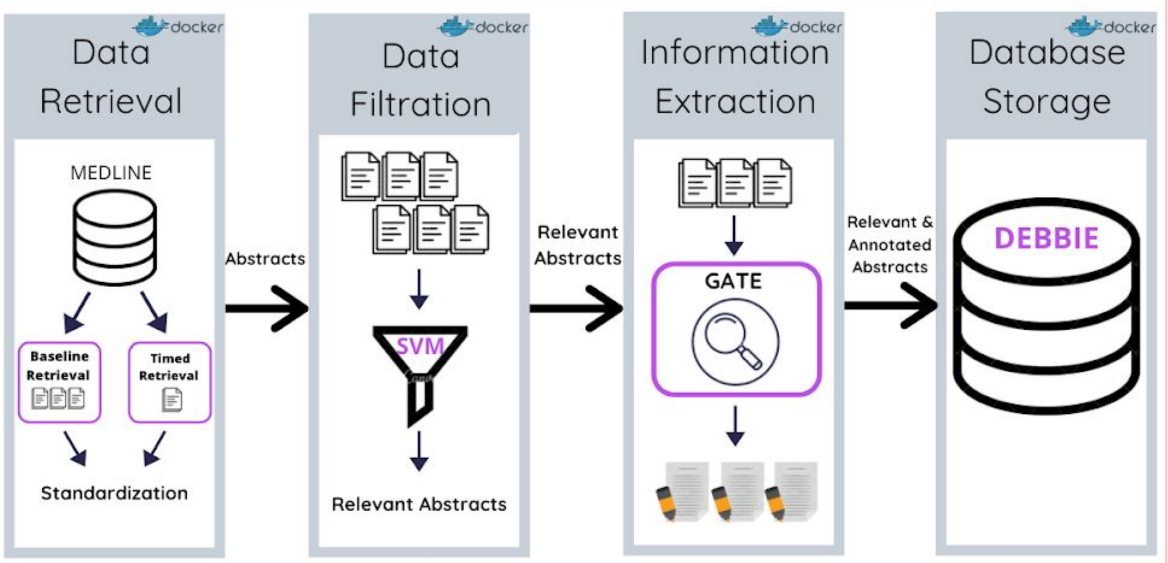


Figure 36 : Presentation of the DEBBIE pipeline