Clustering unstructured life sciences experiments with unsupervised machine learning

Natural language processing for unstructured life sciences texts

MATHIAS DAIL
Clustering unstructured life sciences experiments with unsupervised machine learning

Natural language processing for unstructured texts

Mathias Dail

Master in Computer Science

School of Electrical Engineering and Computer Science

**Supervisor:** Elena Troubitsyna

**Examiner:** Olov Engwall

**Tutor:** Mats Kihln, Dassault Systèmes

**Swedish title:** Klustrering av naturvetenskapliga experiment med oövervakad maskininlärning.

**Date:** June 16, 2019
Abstract

The purpose of this master’s thesis is to analyse different types of document representations in the context of improving, in an unsupervised manner, the searchability of unstructured textual life sciences experiments by clustering similar experiments together. The challenge is to produce, analyse and compare different representations of the life sciences data by using traditional and advanced unsupervised Machine learning models. The text data analysed in this work is noisy and very heterogeneous, as it comes from a real-world Electronic Lab Notebook.

Clustering unstructured and unlabeled text experiments is challenging. It requires the creation of representations based only on the relevant information existing in an experiment. This work studies statistical and generative techniques, word embeddings and some of the most recent deep learning models in Natural Language Processing to create the various representation of the studied data. It explores the possibility of combining multiple techniques and using external life-sciences knowledge-bases to create richer representations before applying clustering algorithms. Different types of analysis are performed, including an assessment done by experts, to evaluate and compare the scientific relevance of the cluster of experiments created by the different data representations. The results show that traditional statistical techniques can still produce good baselines. Modern deep learning techniques have been shown to model the studied data well and create rich representations. Combining multiple techniques with external knowledge (biomedical and life-science-related ontologies) have been shown to produce the best results in grouping similar relevant experiments together.

The different studied techniques enable to model different, and complementary aspects of a text, therefore combining them is a key to significantly improve the clustering of unstructured data.
Sammanfattning

# Contents

1 Introduction

1.1 Context ................................................................. 1
1.2 Problem specification .................................................. 2
   1.2.1 Electronic lab notebook data ................................... 2
   1.2.2 Purpose of doing clustering ................................... 2
   1.2.3 Challenges ......................................................... 2
1.3 Research questions ................................................... 4
   1.3.1 Ethics .............................................................. 4
   1.3.2 Sustainability and societal aspects .............................. 4
1.4 Overview ............................................................... 5

2 Background

2.1 Artificial Neural Networks ............................................ 6
2.2 Text representations .................................................. 7
2.3 Models used to create different representations of the data ..... 9
   2.3.1 Word-based techniques .......................................... 9
   2.3.2 Word embeddings based techniques ............................. 12
   2.3.3 Contextualized string embeddings and Language Models .... 15
   2.3.4 Knowledge-based models ........................................ 20
2.4 General data analysis techniques ................................... 21
   2.4.1 Dimensionality reduction techniques ............................. 21
   2.4.2 Visualization techniques ......................................... 22
2.5 Analysis of the clusters of the data representations ............ 22
   2.5.1 Internal evaluation metrics ..................................... 23
   2.5.2 Clustering algorithms ........................................... 25

3 Method

3.1 Data specification and preprocessing ............................... 26
   3.1.1 Dataset specification ............................................ 26
   3.1.2 Text type and noisy data ........................................ 26
   3.1.3 Preprocessing and cleaning ..................................... 27
3.2 Notation ........................................................................ 28
3.3 Creating vector representations of the data ....................... 28
   3.3.1 Word-based representations ...................................... 28
   3.3.2 Word embeddings based techniques .............................. 31
   3.3.3 Document representations with Contextualized string em-
        beddings .............................................................. 32
   3.3.4 Knowledge-based models ......................................... 36
   3.3.5 Concatenations of multiple representations ................. 38
3.4 Clusterability analysis of the data representations ............. 39
   3.4.1 Visualization as a qualitative comparison method ............ 40
   3.4.2 Evaluation of the number of potential clusters in the data 40
   3.4.3 Distance metrics .................................................. 41
   3.4.4 Clustering algorithms ............................................. 41
3.5 Analysis of the nearest-neighbours clustering of the represen-
   tations ........................................................................ 42
   3.5.1 Nearest neighbor and local analysis ............................ 43
   3.5.2 Human assessment of the nearest-neighbours clustering ... 43
3.6 Implementation .......................................................... 45
4 Results

4.1 Representations
4.1.1 Vector-spaces dimensions
4.1.2 Low-dimensional representations visualizations
4.2 Clusterability analysis of the representations
4.2.1 Evaluation of the number of potential clusters in the data
4.2.2 Elbow method with K-Means
4.2.3 Density-based clustering
4.2.4 Internal metrics with studied groups as labels
4.3 Evaluation of the nearest-neighbours clustering of the representation
4.3.1 Similarity comparison of the nearest-neighbourhoods
4.3.2 Human assessment of the nearest-neighbours clusterings
4.3.3 Correlation of the internal metrics with human rating

5 Conclusions and discussion

5.1 Discussion of the results
5.1.1 Qualitative results
5.1.2 Quantitative results
5.2 Main results
5.3 Conclusions
5.3.1 Comparison with related work
5.4 Future work
1 Introduction

1.1 Context

Data mining techniques have been an important branch of artificial intelligence for many decades, allowing the search for valuable information in very large volumes of data. The increasing size of databases has created the need for new information management technologies for handling knowledge efficiently. The amount of life sciences and biomedical literature is exploding, and its volume is getting overwhelming. As an example, the biomedical resource PubMed, developed and maintained by the National Center for Biotechnology Information (NCBI), comprises over 29 million citations for biomedical as of March 2019 [1]. The documents containing valuable information for researchers are constantly added to the literature. As a result, there is a increasing demand for text mining tools for extracting information in life sciences.

Very recent progress in Natural Language Processing (NLP) and Machine Learning made biomedical and life sciences text mining scalable and thus possible to use in large-scale real-world applications.

Word2Vec [51] has been one breakthrough in NLP in the recent years. However, when dealing with specific text corpora (life sciences, biomedical texts, etc.), Word2Vec needs to be adapted because of the large differences in vocabulary and wording between a life sciences corpus and a general domain corpus [59].

Deep learning, a specific branch of Machine Learning, has proven useful in NLP as well. Deep learning architectures require less feature engineering and can handle a large quantity of data.

Some type of Artificial Neural Networks [9] like the sequential Long Short-Term Memory (LSTM) and Conditional Random Field (CRF) have recently improved performance in biomedical and life sciences named entity recognition (NER) [27]. Other deep learning based models have made improvements in life sciences text mining tasks such as relation extraction (Bhasaran and Natarajan, 2018) and question answering (Wiese et al., 2017).

However, Deep learning needs a lot of labeled training data (LeCun et al., 2015). It is, in scientific text mining tasks, a very costly and almost impossible task. The construction of a large training set requires experts in those fields. Therefore, most text mining models cannot exploit the full power of deep learning because of the lack of training data. To address that problem, recent work has focused on training multi-task models (Wang et al., 2018).

Recent models like ELMo [58] and BERT [17] have proved the value of contextualized representations produced by deeper structures like bidirectional language models for transfer learning. These contextualized representations, extracted from a general domain corpus such as Wikipedia, have been efficiently used in biomedical text mining. Current research is focusing on leveraging the large amounts of unstructured life sciences and biomedical data in an unsupervised way to improve the performance of deep learning NLP models [41].
1.2 Problem specification

This thesis was written at a company developing and maintaining a real-world Electronic Lab Notebook (ELN) platform.

1.2.1 Electronic lab notebook data

In the past years, the use of Electronic Lab Notebooks (ELNs) as laboratory document platform, has become mainstream. These new platforms change the way research is done. Therefore, an ELN must deal with the needs of the various scientists who create, build and document their experiments on the platform. The number of experiments on the ELN is growing every day and the motivation is to allow scientists to do their research while leveraging what already exist on the platform. Experiments containing valuable information could help researchers working in different but related fields. As a result, there is a demand for data mining tools and other unsupervised techniques that could help ELN users to explore the data in an efficient and useful way. This requires the supervised creation of representations of the data, to enable the use of clustering algorithms to group related and relevant experiments together (according to different criteria).

The aim of this thesis is, therefore, to explore what kind of machine learning and deep learning techniques can increase the quality of the clusterings (i.e. groupings of experiments) in the data available to the researchers on the ELN platform. [34]

1.2.2 Purpose of doing clustering

The purpose of clustering is to make sense of and extract knowledge from large datasets (structured or not). When working with huge volumes of unstructured data, it only makes sense to try to partition the data into some logical groupings before attempting to analyze it at a finer level. These logical structures called clusters are groupings of data points that share similar attributes.

In this thesis, the Electronic Lab Notebook dataset is large and completely unstructured. The main objective is to group similar experiment in the dataset based on their scientific or experimental content. From a further perspective, clustering enable us to link the knowledge existing in the data, to improve its overall value. Today, the data in the ELN is completely ungrouped and scattered, therefore clustering is a good starting point to extract knowledge from the data. It will provide a way to browse the ELN scientific corpus, in a relevant manner, for both researchers using the platform, and business analysts in need to have a global and structured view of the data.

1.2.3 Challenges

There is a number of challenges when handling the data that should be addressed. Namely:

- The corpus is highly heterogeneous and composed of noisy and unstructured scientific experiments in life sciences.
• The corpus ranges across different scientific domains in life sciences: biology, chemistry, cytology, immunology and genetics.

• The data is fully unlabeled: for a given experiment in the ELN corpus, almost nothing is known beforehand.

• The data is multilingual (e.g., English, French, Swedish) because researchers from different countries work on the same platform.

• Some experiment can range from a few lines to a few pages. They can be very well written (articles, full sentences, correct grammar) or very badly written (word shortcuts, parts of sentences, no punctuation).

• Many abbreviations are used and any kind of specific data 'object' can be present in an experiment: numerical arrays, tables (e.g., excel sheets), etc.

• Potentially all types of meta-data can be present in an experiment, including personal information (email, links, names).

Then there are inherent challenges, in this work, related to the field of Natural Language Processing. Namely:

• Unstructured texts documents have to be represented numerically to make them mathematically computable, to enable the use of algorithms on them (e.g., clustering algorithms)

• There are many types of representations, working on different atomic levels: words, sentences, characters, etc.

• There is no prior knowledge on which representation to use to cluster life science experiment in the most accurate and meaningful way.

The main objective is to cluster experiments based on the most relevant information (for the scientific community working on the ELN) existing in this noisy corpus. The purpose of doing a clustering is to group similar experiments (based on some criteria e.g. the scientific domain) to allow a better use of the work done by researchers of the ELN platform. As the experiments are fully unstructured, there is a real need to provide a better search and browsing in this massive scientific research platform, therefore clustering can provide a way to browse this scientific corpus.

The unsupervised aspect of this work forces the experimentation to be structured as follows: create a useful and meaningful representation of the corpus of experiments with a given Machine learning model, cluster this representation to group similar experiment together and evaluate the quality of this clustering.

There is a variety of Machine learning models in the field of Natural Language Processing. They work on different aspects of a text document. Therefore, this work explores different types of models, and their different types of representation, to evaluate which ones are the best suited to cluster life sciences text experiment in a meaningful way.
1.3 Research questions

This thesis focuses on two aspects: determining the different representations of a textual experiment and evaluating which ones are more promising for improving clustering. The main research question is:

What unsupervised machine learning techniques allow for an improved clustering of unstructured life sciences text experiments?

This question comes with additional research questions:

What are the different types of Machine learning techniques allowing the creation of useful representations of the studied scientific text data?

Can the combinations of representations give more accurate results?

Which technique and its representation produce the most relevant clustering of the data?

1.3.1 Ethics

The ELN data studied in this thesis is confidential and thus no sensitive information will be published. Any text analytic tool, working with confidential data, for example in healthcare or in confidential research, should be built with confidentiality in mind. As AI spreads through healthcare, Organizations using AI-based search tools should establish clear ethical standards.

In the context of this work, the techniques studied can be used in any context, as they are unsupervised. The Machine learning models, trained during this work, do not keep any sensitive information in memory.

1.3.2 Sustainability and societal aspects

This thesis attempts to improve the extraction of valuable information in large quantities of unstructured text, by clustering them, which would allow scientists to do their research more efficiently.

These automated systems are getting more and more valuable as the demand is constantly increasing. This is the case in scientific research but also for non-scientific domains where a lot of unstructured data is available. The systems studied in this work can leverage cross-disciplined data and thus could help many fields to progress faster.
1.4 Overview

This thesis will be structured as follows. The Background of the work will be described in details, introducing the Machine learning and Natural language processing (NLP) techniques that will be used in this work. The main challenge of NLP, and of this work, is to create rich representations of the textual experiments enabling relevant and useful clusterings of the data. Therefore, different types of techniques will be explored to create representations of the data, in order to evaluate which ones are the most suited for the clustering task. The Method used will then be detailed, including a specification of the data studied in this work. It will details the different Machine learning models, described in the Background, used to create representations of the data and the different data pipelines. It will describe in details how the techniques are combined. Two main analyses will be performed on the representations, first to evaluate their clusterability and then to evaluate the quality of the clusterings. Several dimensionality reduction and visualizations techniques are used and will be detailed. The Results will show the visualizations of the representations of the data, the analysis of the clusterings of the different representations of the data and the main results. Some correlations between several metrics will also be analysed. Finally, a conclusion section will detail the results of this work including the best representations and detailed answers to the research sub-questions. Quantitative and qualitative observations will also be detailed. A discussion will be made on the relation between the results and the related work in the field. This thesis will end by discussing the possible orientations for future work.
2 Background

This section will first describe Artificial Neural Networks, the foundation of some models used in this work, and Text representations in the general context of document clustering.

Then, each model used to create a unique representation of the data will be detailed. The models are categorized, in this work, in 4 groups: Word-based, Word embeddings-based, Language Models and Knowledge-based Models. For some of these categories, some theoretical aspects will be described before detailing the models themselves.

This section will then introduce the dimensionality reduction techniques, the modern visualizations techniques and the clustering algorithms used in this work. Every part of this section will be useful to create the Method and to produce the Results.

2.1 Artificial Neural Networks

Artificial Neural Networks (ANN) are non-linear functions with many parameters [73]. They are made of multiple non-linear and simple operations. The theory behind the ANN is inspired by how the brain uses neurons in complex networks for learning. It consists of a network of nodes, built in layers (at least one layer). Each layer consists of a number of nodes. Each node represents a neuron and has several weights attached to it. Each weight, $w_{ab}$, is attached to the connection between the current node, $a$, and a node, $b$, from a previous layer. The features of some data is passed into the network at the input layer, where each feature is represented as one input node. The network tries to learn to reproduce targets associated with every data sample (i.e., labels). They are represented as the output layer with one node for each target feature. One famous example of ANN is the Multi Layer Perceptron [25]. It has an input and output layer as well as several hidden layers in between.

Weights of the network, represented by the arrows in the illustration above, Figure 1, are initialized (usually randomly) and are then progressively changed during the training process via back-propagation [39].

Figure 1: Illustration of an artificial neural network
2.2 Text representations

When dealing with Machine Learning methods, a key step is the definition of the representation which describes the structure of the data. Different representations emphasize different aspects of the main problem and could produce different results. This choice is even more crucial in text analytics because there is neither obvious nor unique way to represent a text document. Therefore, as the choice of the data representation can have a significant impact on the following clustering of the text experiments, this thesis will use, compare and combine different type of representations.

One of the most widely-used used representation system, and the simplest possible, is the Vector Space Model (VSM) [64]. In the VSM, a document $d$ is represented as a vector in the word space, $d = (w_1, w_2, ..., w_{|V|})$ where $|V|$ is the size of the vocabulary. In the simplest version each $w_i \in [0, 1]$ is a binary variable which indicates whether word $w_i$ is present in the document (value 1) or not (value 0).

A document can also be represented by counting how many times each word occur, the traditional Bag-Of-Words (BoW) representation. These representations are focusing on the content of the text, by analyzing the presence (or absence) of words in the document. A document can also be expressed as a set of n-grams, i.e, a contiguous sequence of n words (items) from a given text string, aiming at catching the dependencies between grouping of entities. However, Bag-Of-Words representations fail to capture similarities between words and phrases and suffer from sparsity and dimensionality explosion. Moreover, by treating words as independent tokens, the temporal information is lost (grammar, flow of sentences, etc.) making it impossible to model long semantic dependencies.

New techniques for representing words, sentences or even documents emerged in the last decades. Word embeddings were first discussed in the research area of distributional semantics [28]. The fundamental principle is to 'quantify and categorize' similarities between linguistic items in a text based on their distributional characteristics in samples of some text. Firth is often cited for saying that "a word is characterized by the company it keeps" [20].

The technique of representing words as vectors has been invented in the 1960s with the development of the vector space model for information retrieval. In 2000 Bengio et al. provided the "Neural probabilistic language models" [7], to reduce the high dimensionality words representations in contexts by learning a distributed representation for words. This is the foundation of modern NLP and text analytics. There are 2 types of Word embeddings: the first one in which words are expressed as vectors based on the co-occurrence of words in a text, and the second in which words are expressed as vectors based on the contexts in which they occur.

A breakthrough in distributional semantics (and NLP) was achieved in 2013. Mikolov et al proposed the Word2Vec model [51], a word embedding toolkit, which can train vector space models more efficiently than the techniques existing at that time. It popularized the use of words embeddings and more generally
vector representations. Nowadays, new embedding techniques rely mostly on neural network-based architectures [53]. The most important aspect of using word embeddings, i.e., vectors to represent textual entities (words, sentences, paragraphs or even entire documents) is that these vectors store meaning. Word embeddings have demonstrated their effectiveness in storing valuable syntactic and semantic information [51]. If two vectors, representing respectively the semantics of two documents, are close (according to a given distance, for example the Euclidean distance), these two documents have related contents.

The main advantage of these distributional-based approaches is that they exploit semantic similarities between words, and produce highly compact embeddings. Recent work has shown that embeddings with several hundred dimensions achieve best accuracies in classification and information retrieval [35]. Some of these approaches include weighted word combination models [5], Doc2Vec [40]. The word combination models directly aggregate word representations in a given document through averaging or another function. These approaches are easy to implement and achieve highly competitive performance. Unlike Bag-Of-Words representations, the resulting embeddings are an order of magnitude smaller in size and don’t suffer from sparsity or dimensionality explosion problems. However, by averaging together word representations to create a document representation, temporal information is lost.

It is easy to imagine examples of documents that contain almost the same words, but have very different meanings due the word order. Averaging and other aggregation models that ignore word order are unlikely to perform well on more complex NLP tasks.

Finally, Recurrent neural networks (RNN) models [29] as embedding models, ingest the document one word at a time and their hidden neural activations are taken as the final document embedding, after the entire document has been processed. This approach provides a way to model temporal aspects of the word sequence. However, the sequential nature of the RNN creates disadvantages. Indeed, many of the commonly used RNN architectures, such as LSTM [30], gate information from already seen input at each recurrence step. This has an undesirable effect where more weight is put on the most recent word ‘read’ by the network. It can then “forget” earlier parts of the document. That’s one reason why new techniques, not based on RNNs, are emerging to be able to capture important information anywhere within a document. One example is the very recent concept called Attention [74]. This mechanism, tries to mimic to human brain: each time the Attention-based model predicts an output word, it only uses parts of the input where the most important information is concentrated. It only pays attention to some of the input words in a sequence.

In this work, a representation of a given experiment will refer to a real-valued or binary vector. This vector dimensionality depend on the model used to create it.
2.3 Models used to create different representations of the data

This section will describe the different models used in this work, handling text in different ways. These models are used to create different representations of a given text document. The following table is the list of different techniques that will be described in this section. The tables shows the atomic unit used by each model and the level at which they work.

<table>
<thead>
<tr>
<th>Technique name</th>
<th>Atomic unit</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF-IDF</td>
<td>Words</td>
<td>Corpus</td>
</tr>
<tr>
<td>LDA</td>
<td>Words</td>
<td>Corpus</td>
</tr>
<tr>
<td>Word2Vec/Doc2Vec</td>
<td>Words</td>
<td>Local context</td>
</tr>
<tr>
<td>Glove</td>
<td>Words</td>
<td>Corpus</td>
</tr>
<tr>
<td>Flair</td>
<td>Contextual string embeddings</td>
<td>Sentences</td>
</tr>
<tr>
<td>BERT</td>
<td>Contextual string embeddings</td>
<td>Sentences</td>
</tr>
</tbody>
</table>

2.3.1 Word-based techniques

The following techniques, TF-IDF and LDA, use words as atomic unit, like in the bag-of-words representation, and to do so, they are using a Dictionary that maps all the unique words in a corpus to an ID (usually a simple integer).

2.3.1.1 TF-IDF: term frequency-inverse document frequency

TF-IDF or Term Frequency Inverse Document Frequency [61] is a measure that is based on the idea of associating a score to the importance of a word in a document based on how often it appears in that document and a given collection of documents.

The intuition for this measure is as follows. If a word appears frequently in a document, then it should be important and it should have a high score. But if a word appears in too many other documents, it’s probably not a unique identifier. Therefore, the method should assign a lower score to that word. The formula is, for a term \( t \) in a document \( d \), in a corpus \( D \):

\[
\text{TF-IDF}(t, d, D) = tf(t, d) \times idf(t, D)
\]  

(1)

\( tf(t, d) \) is the number of times each word appeared in each document, i.e \((\text{Number of times term } t \text{ appears in a document } d) / (\text{Total number of terms in the document})\). Then,

\[
idf(t, D) = \log \left( \frac{|D|}{1 + |\{d \in D : t \in d\}|} \right)
\]  

(2)

The \( idf \) function represents the idea that the specificity of a term can be calculated as an inverse function of the number of documents in which it occurs [32]. These calculations can be performed for every term in a document, for every document, to create a document-term matrix with the associated TF-IDF
weight of each term.

To illustrate the TF-IDF calculation, let’s use a simple example made of 3 documents: d0:”Simple example with Cats and Mouse”, d1:”another simple example with dogs and cats” and d2:”another simple example with mouse and cheese”. The Bag-Of-Words representation of that document is as follow.

<table>
<thead>
<tr>
<th></th>
<th>and</th>
<th>another</th>
<th>cats</th>
<th>cheese</th>
<th>dogs</th>
<th>example</th>
<th>mouse</th>
<th>simple</th>
<th>with</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 2: Example of a bag-of-words representation

The associated TF-IDF document-term matrix is as follows.

<table>
<thead>
<tr>
<th></th>
<th>and</th>
<th>another</th>
<th>cats</th>
<th>cheese</th>
<th>dogs</th>
<th>example</th>
<th>mouse</th>
<th>simple</th>
<th>with</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.000000</td>
<td>0.067578</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.0</td>
<td>0.067578</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>0.0</td>
<td>0.057924</td>
<td>0.057924</td>
<td>0.000000</td>
<td>0.156945</td>
<td>0.0</td>
<td>0.000000</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>0.0</td>
<td>0.057924</td>
<td>0.000000</td>
<td>0.156945</td>
<td>0.000000</td>
<td>0.0</td>
<td>0.057924</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Figure 3: Example of a TF-IDF document-term matrix

As illustrated above in Figure 3, the document-term matrix is usually sparse and very large. Documents are usually made of a few terms composing the total vocabulary. This is one reason to apply dimensionality reduction on the resulting document-term matrix to obtain a denser representation of smaller dimension.

2.3.1.2 LDA topic modeling : Latent Dirichlet Allocation

The Latent Dirichlet allocation [10] (LDA) technique is a generative statistical model that allows sets of observations to be explained by unobserved groups (latent variables) that ‘explain’ why some parts of the data are similar and regroup them. This is very attractive for this work, as the studied data is composed of fully unstructured and unlabeled documents.

In LDA, observations are words in a document (i.e., experiment) and the principle of LDA is that each document is a mixture of a number of topics and each word is associated to one of the document’s topics. This allows a document to be composed of many topics. The number of topics has to be chosen before the procedure and must be known before-hand or can be evaluated using methods like Topic Coherence [62], calculating a score for topics where words appear together in documents relative to how often alone they appear in documents. This is done to “imitate” human judgement.

The topic distribution is based on the assumption that it has a sparse Dirichlet prior [37]: the assumption here is that documents cover only a small set of topics and that topics use only a small set of words frequently.
LDA has been very successful in practice, resulting in a relatively good disambiguation of words and precise assignment of documents to topics. As several variants of LDA exist, the most commonly used is the LDA with Dirichlet distributed topic-word distributions. The parameters of the LDA model are defined as follows and won’t be detailed too much:

\(\alpha\) is the parameter of the Dirichlet prior on the per-document topic distributions. 
\(\beta\) is the parameter of the Dirichlet prior on the per-topic word distribution. 
\(\theta_m\) is the topic distribution for a document \(m\). 
\(\varphi_k\) is the word distribution for topic \(k\). 
\(z_{mn}\) is the topic for the \(n\)-th word in document \(m\). 
\(w_{mn}\) is the specific word.

Figure 4: Plate notation of LDA with Dirichlet-distributed topic-word distributions

As illustrated in Figure 4, words are the only observable variables, and the other variables are latent. Additionally, and as advised in [48], that the use of ngrams, especially bigram and trigrams, can improve the results. A bigram or trigram is respectively a sequence of two or three adjacent words (entities in the text). They are respectively n-grams for \(n=2\) and \(n=3\).

The process of learning the distributions and parameters listed above is solved with Bayesian inference [71]. There are several inference techniques, the one used in this work is Gibbs sampling [23].

A simple illustration of a topic distribution learned from LDA when \(K=3\) topics is shown below, on a supposed general-domain corpus of documents made of articles.

| Topic 1 | | Topic 2 | | Topic 3 |
|---------|-----------------|-----------------|-----------------|
| term    | weight          | term            | weight          | term            | weight          |
| biology | 0.011           | space           | 0.021           | politics        | 0.38            |
| university | 0.009         | nasa            | 0.006           | washington      | 0.21            |
| moth    | 0.008           | earth           | 0.006           | congress        | 0.11            |
| caterpillar | 0.007      | moon            | 0.004           | president       | 0.7             |

Figure 5: Simple example of a LDA topic-word distributions with 3 topics, showing the top 4 words of each topic

In the Figure 5, which is just one example on how to illustrate LDA, the topics are represented by its top 4 word. For example, the Topic 2 appears to match
the news discussing science while Topic 3 corresponds to politics. Each word associated with a topic has a weight, indicating its importance to the topic, this is why LDA topics are usually analysed by looking at its most important (highest weights) words.

2.3.2 Word embeddings based techniques

In this section, models using the concept of word embeddings are detailed. These models learn a geometrical encoding (vectors) of words from their co-occurrence information (how frequently they appear together in a large text corpora). The following techniques use words as an atomic unit.

2.3.2.1 Word2Vec

The Word2Vec [51] (for word to vector) tool used in this thesis implements both the skipgram and CBOW approaches of Mikolov et al. The thesis will only focus on the Continuous Bag of Words (CBOW) approach because it generally produces better results and the more computationally efficient model of the two. (Mikolov et al. (2013b) recommends CBOW as more suitable for larger datasets).

CBOW, one of the Word2Vec based algorithms, is based on the idea of learning word representations that can predict a word given its surrounding words, the context. The input layer corresponds to the context (surrounding words) and the output layer corresponds to signals for the prediction of the target word. The context is only made of elements that are at the left of the target word in the sequence. The neural network tries to learn features that look at the context words in a window and tries to predict the next word.

The Word2Vec model learns to predict the next element during the training, by adjusting the weights of the network, so that the probability of the next word is maximized, as compared to other words in the vocabulary. After many training iterations, the weights become satisfying enough and can then be used as the vectorised representations of words. Word2Vec introduces a technique called negative sampling that estimates the probability of an output word by learning to distinguish it from a noise distribution. Very frequent words ("the", "and", "to", etc.) are not informative as context features. To deal with that problem, Word2Vec implements a method to reduce their effect [26].

This is controlled by a parameter $t$ and words that occur with higher frequency than $t$ are sub-sampled (only a portion is kept). The Word2Vec paper suggests $t = 1 e - 3$ based on empirical observations. To summarize, the main parameters of this model are:

1. The size of the windows (for example 5 words to either side of the middle word).
2. The dimension of the embedding (usually between 200 and 400. By default 300)
3. The negative sub-sampling (usually $t = 1e^{-3}$)

A very simple example is shown below to illustrate the sequential training process of Word2Vec.

![Source Text]

*Figure 6: Illustration of the Word2Vec window sliding through a text*

As the context window is sliding, like in Figure 6, the word in the middle of the window, the target word (in blue), is updated. The training process adjusts the weights of networks to maximize the probability of the target word considering a given context: the words close to the target word from the left and right. In Figure 6, the context window has a size of 2 (2 words before and after the target word). This process creates embedding for the words that are conditioned on their surrounding words.
2.3.2.2 Doc2Vec

As seen in the section 2.3.2.1, the Word2Vec training process can be seen as a self-supervised learning task of predicting target word class given the input context. However, the context can be more generic than just words. Doc2Vec explores this by adding additional input nodes representing documents as an additional context. Each additional node can be thought of just as an ID for each input document.

The following figure is adapted from [8].

![Doc2Vec Diagram](image)

Figure 7: The modified Word2Vec neural network for the Doc2Vec model

The objective of Doc2Vec learning is, for all target and context words and documents:

\[
\max \sum \log(P(\text{target word}|\text{context words, document context}))
\]

In the original Paragraph Vectors paper [40] on which Doc2Vec is based, every document gets its own unique ID tag. The Doc2Vec training process will produce a unique doc-vector per document as output.

Compared to the Word2Vec process, both word embeddings \( W \) and document embeddings \( D \) for documents in the corpus are trained, as illustrated in Figure 7. The document embeddings will be the final representation used in this work, for this Doc2Vec model.

2.3.2.3 Glove: Global Vectors for Word Representation

This section details a count-based model called Glove, or GloVe [56] for Global Vectors for Word Representation.

First, a concept named global matrix factorization needs to be briefly introduced. It is the process of using matrix factorization methods from linear algebra to perform rank reduction on a large term-frequency matrix. These matrices usually represent term-document frequencies, in which the rows are words and the columns are documents.

Low-rank approximations to the term frequency matrices give reasonably sized vector space embeddings of a global corpus statistics.
Shallow-neural-network methods, like the Word2Vec method described in a previous section, learn word representations from local context windows. Glove works similarly as Word2Vec. While the latter is a “predictive” model that predicts context given word, Glove learns by constructing a co-occurrence matrix (words X context) that counts how frequently a word appears in a context. Glove derive ‘meaning’ directly from the statistics of the studied corpus. Glove is global in the sense that it uses co-occurrence frequencies, so a measure that is global with respect to the data.

The Glove paper tells, however, that context window-based methods suffer from the disadvantage of not learning from the global corpus statistics [56]. As a result, repetition and large-scale patterns may not be learned as well with these models as they are with global matrix factorization. It has to be seen as a trade-off and that’s why it can be interesting to combine different techniques together to try to leverage their respective advantages.

2.3.3 Contextualized string embeddings and Language Models

Modern approaches to modeling text in NLP use recurrent neural networks (RNN). Current state-of-the-art approaches use the LSTM architecture [30] [68], a variant of RNN, or the bidirectional recurrent neural networks (BiLSTMs) as a language modeling architecture [76]. The following approaches don’t use words directly as input but instead treat texts as a string made of characters.

2.3.3.1 Sequence to sequence (Seq2Seq) models

A sequence-to-sequence architecture is a neural network that transforms a given sequence of elements, such as the sequence of words in a sentence, into another sequence. A common choice for this type of model is Long-Short-Term-Memory (LSTM)-based models. With sequence-dependent data, the LSTM [30] modules can give meaning to the sequence it reads while remembering, to simplify, the past sentences it has read.

Sentences, for example, are sequence-dependent since the order of the words is very important for ‘understanding’ the sentence. That’s why LSTM are a natural choice for this type of data. Seq2Seq models consist of an Encoder and a Decoder. The Encoder takes the input sequence and maps it into a higher dimensional space. This is then fed into a Decoder which turns it into an output sequence. These models can also be stacked to become “deep” and construct deepened representations.

2.3.3.2 Flair

These recent techniques have been shown to outperform earlier n-gram or purely statistical based models due to the ability of LSTMs to capture long-term dependencies with their hidden state [33].

Characters are used as the atomic units of language modeling [24], allowing the text of each document to be treated as a sequence of characters. That sequence is then fed to an LSTM which at each point in the sequence is trained to predict the next character.

Training a language model can be viewed simply as a process that tries to learn $P(x_t | x_0, ..., x_{t-1})$, an estimate of the predictive distribution over the next
character given past characters. The joint distribution over sentences is then decomposed as a product of the predictive distribution over characters conditioned on the preceding characters.

The following figure is adapted from [4].

![Figure 8: Extraction of a contextual string embedding for a word (“Washington”) in a sentential context. Two language models shown in red and blue](image)

In the illustration above, Figure 8, two different output hidden states are extracted: From the forward language model (shown in red), the output hidden state is extracted after the last character in the word. This hidden state contains information propagated from the beginning of the sentence up to this point. Then from the backward language model (shown in blue), the output hidden state is extracted before the first character in the word. This contains information taken from the end of the sentence to this point. Both output hidden states are concatenated to form the final embedding.

A word embedding from this kind of model is said to be contextual because it is taken from a larger hidden state representing a larger sequence, like a sentence. This approach has 3 main advantages:

- It can be pre-trained on large unlabeled corpora, it captures word meaning in context and consequently produce different embeddings for polysemous words depending on their usage. For example the word ‘vacuum’ used in a sentence part of a biological related experiment will have a different embedding than the word ‘vacuum’ used in a sentence part of a mechanical related experiment.
- Then another advantage of this approach is the ability to model words and their context as sequences of characters. It both handle rare and misspelled words as well as can model sub-word structures such as prefixes and endings. It is an interesting characteristic, as the studied data is made of life sciences terms (biology, molecular biology, chemistry, physics) that can be written in many forms and are often, especially with chemicals, written with prefixes and suffixes.
- Some modern sequence labeling models often combine different types of embeddings by concatenating each embedding vector to form the final word vectors. In other applications it has been proven to be beneficial to add classic word embeddings [33] for a potentially different and richer latent representation to another embeddings.

The final output embedding of a word in this context is the concatenation of the Contextual String embeddings of that word with the Glove embedding of that word.
2.3.3.3 Attention, Transfer learning and Transformer architecture

Transfer learning, Attention, and the Transformer architecture [74] have to be explained briefly before introducing BERT-based models. Firstly, Transfer learning is a machine learning method where a model that is developed for a specific task is reused as the starting point for a model on a second task. Transfer Learning is disruptive in Machine learning in that it uses pre-trained models that have been used for another task to start the development process on a new task or even a new data. Moreover, learning word representations from a large amount of unannotated text, in an unsupervised way, is a long-established method. Training on specific NLP tasks (e.g. language modeling) where word representations were some output of the NLP tasks, or direct optimization of word representations was done to obtain word representations. While earlier work on word representations was focused on learning context independent representations, like with TF-IDF or LDA, recent work have focused on learning context dependent representations using specific NLP tasks. For instance, ELMo (Peters et al.,2018) [58] uses a bidirectional language model while CoVe [49] uses machine translation to embed context information into word representations.

Secondly, attention-mechanism [74] needs to be introduced, as it is a core mechanism used by Transformers. A Transformer is a sequence-to-sequence architecture (or Seq2Seq), as described briefly in section 2.3.3.1. Attention looks at an input sequence and decides, for each word in that sequence, and at each step, which other parts of the sequence are important. It is based on the idea of copying humans in the way they read a text. The intuition is simply that in general, people tend to keep in mind (or memory) the important keys or elements in a text. This allows the context to be kept into ‘memory’ during the read. This mechanism improved greatly the results on many tasks in Natural Language Processing, especially Machine Translation, as the language models are able to find interesting mappings between different parts of the input sequence and corresponding parts of the output sequence.

An attention-mechanism works in a similar fashion for a given sequence. For every input, the current text sequence, that the Encoder reads, the attention-mechanism takes into account several other inputs at the same time and decides which ones are important by attributing them some weights. The Decoder will then take as input the encoded sentence, and the weights provided by the attention-mechanism.

Finally, like LSTM, the Transformer is an architecture based on an Encoder and a Decoder, but it does not make use of any Recurrent Networks (no LSTM), it is a fundamentally different architecture. To describe the difference between the Transformer and the LSTM without delving too deep into the details, it is important to understand the motivation behind the Transformer architecture. In the recent years, RNN (like LSTMs) based on encoder-decoder scheme were and still are the most popular choice of language models. However, recurrent models, because of their sequential nature are not allowing for parallelization during training, and so have a problem with the learning long-term dependencies [30] from memory. The Transformer architecture reduces the number of sequential operations to relate two symbols from input/output sequences to a
constant $O(1)$ with respect to the size of the sequence, and it achieves this with a multi-head attention mechanism [74].

This mechanism allows the network to model dependencies in the studied sequence regardless of their distance in input or output sentence. The novel approach of Transformer is however, to eliminate recurrence completely and replace it with Attention to handle the dependencies between input and output. The Transformer takes its advantage by using Attention entirely and by applying a resulting mechanism called Self-Attention or Intra-Attention. This new building block is spreading quickly in NLP as it is a very efficient architecture, which is able to leverage a lot of data more efficiently than previous models. [69]

2.3.3.4 BERT

BERT (Devlin et al., 2018) is a recent language representation model which stands for Bidirectional Encoder Representations from Transformer [17] that uses the Attention mechanism in a Transformer based architecture. It is a contextualized word representation model. It has a very specific pre-training process, compared to the other existing language models, as it is based on a masked language model using bidirectional Transformers, described in section 2.3.3.3. Due to the nature of language modeling where future words cannot be seen, previous bidirectional language models (biLM) were limited to a combination of two unidirectional language models (left-to-right and right-to-left). However, BERT uses a masked language model that randomly mask words in a sentence, during the training process, to then predicts these masked words. This has been shown to produce better bidirectional representations [17].

The BERT model used in this thesis is a multi-layer bidirectional Transformer with the following characteristics: 12 layers, hidden layer size of 768, 12 attention heads and 110 millions trainable parameters in the network.

The pre-trained model is made available by Devlin et Al. and the unsupervised training was performed using a multi-lingual corpus (BooksCorpus (800M words) (Zhu et al., 2015) and the entire Wikipedia dump for 104 languages, as illustrated in Figure 9.

The following illustration is adapted from the BioBERT paper [41].

![Figure 9: Illustration of BERT pretraining](image)

This model can be combined with another task (sentence prediction, sentiment...
analysis, Semantic Role Labeling, etc.) but it can also be used directly, (as it is available pre-trained) to do sentence embedding, using the BERT model as a feature-extractor. It is illustrated in Figure 10. As BERT is a stacked model of 12 layers, a different representation is created at each layer. The question is then to know which one to use and this is explored in the BERT paper [17], where it shows that the embeddings of each layers are focused on different aspects of a text. Notably, using the second-to-last layer (the 11th) or the sum of the last four layers have been shown to produce the best sentence embeddings, when compared on different downstream tasks (Next sequence prediction, translation and classification).

The following illustration is adapted from [41].

![Figure 10: Illustration of extracting BERT hidden layer embeddings](image)

The resulting embedding for every BERT-based architecture, for a given string passed as input to the model is a 768-dimensional vector (the size of the hidden layer in the BERT architecture).

### 2.3.3.5 BioBert and fine-tuning

BioBERT [41] is a pre-trained language representation model for the biomedical domain, based completely on BERT. Its training process is done by initializing BioBERT with BERT which was pre-trained on general domain corpora (as explained in the previous section 2.3.3.4).

Then, BioBERT is fine-tuned on biomedical domain corpora (PubMed abstracts and PMC full-text articles).

The fine-tuning process is exactly the same as a pre-training process, but it is done on already trained weights (the BERT initialization). The original BERT was pre-trained on all of Wikipedia, in 104 languages, and BooksCorpus.

The following illustration is adapted from [41].

---

19
Lee et al. performed the pre-training of the BioBERT with 200,000 and 270,000 steps for PubMed and PMC respectively.

The authors of BioBERT showed that this fine-tuned BERT architecture transfers the knowledge from a large amount of biomedical texts to biomedical text and that BioBERT is competitive with state-of-the-art techniques and outperforms them on three representative biomedical text mining tasks: biomedical named entity recognition, biomedical relation, and biomedical question answering [41]. Architecture wise, BioBERT and BERT are the same model but they leverage different training data.

2.3.4 Knowledge-based models

2.3.4.1 Ontology-Entity based document representation

In the context of knowledge sharing, the term ontology refers to the clear specification of a conceptualization [60]. It is a description of the concepts and relationships that can exist for an agent or a community of agents. It can be viewed as set-of-concept-definitions. There exist general domain ontologies (such as WordNet [52]) and also specific domain ontologies (e.g. the biomedical MeSH [66]).

Within each of the studied life sciences experiments in the data, one concept might be represented in different forms or in abbreviations. For example, ‘T helper cells’ could be represented as ‘CD4 cells’ or ‘CD4’, or even as ‘T lymphocyte’. Therefore, document representations can make use of external knowledge bases, such as ontologies to add richer and more fine-grained information to the representations. This is why this possibility is explored in this thesis. When using a medical ontology, it is possible to add labels (scientific categories, identifiers, links to a particular family of concepts, etc.) to the words (or grouping of words) in a document. These ‘highlighted’ words in a text are referred to as ‘concept’ [54]. They supposedly carry more information regarding the scientific content of the document.

Knowledge-based models can focus on different aspects: using only the concepts extracted from each document to build their representation, creating a weight-based system [45] that add more weights to the concepts-words than the rest of words in a document, using the concept hierarchy knowledge existing in some ontologies [44], to then build the document representation. Some make use of
the associations between concepts to build document vectors representation [77]. Overall, ontologies as background knowledge can improve exploring documents, by associating important concepts identified in a document to a meaningful and clear structure (e.g. hierarchy of concepts in life sciences) [6].

2.4 General data analysis techniques

This section introduces general data analysis techniques that are used on the representations (vectors) of the data.

2.4.1 Dimensionality reduction techniques

In Machine learning, an important problem when dealing with high dimensional data is known as The Curse of Dimensionality. [75]

Bellman used this expression in 1961 to refer to the phenomenon where many algorithms that work fine in low dimensions become intractable, and thus less effective, when the input is high-dimensional [18].

Therefore, reducing the dimension of a dataset before doing any analysis is often necessary. Dimension reduction can also be used to do feature extraction to make the representation of some data more 'compact'. It is often necessary when clustering the data. [67].

It can allow a better analysis of non-linear, non-local structures in the data that are not 'easy' to comprehend in a high dimensional representation. Additionally it can remove noise in the data and reduce Overfitting [70]. And finally, as it will be used several times in this thesis, dimensionality reduction is helpful to do visualization of the data. Indeed, a high-dimensional data representation can be reduced to 2 (or 3) dimensions to then be plotted and analyzed.

2.4.1.1 PCA: Principal component analysis

The Principal component analysis or PCA [31] is a data analysis technique that can be used to analyze the structure of a data set or transform the representation of that data into a vector space with a lower number of dimensions. The context for PCA as an exploratory data analysis tool [65] involves a dataset with observations on p numerical variables and n data values. These data values define p n-dimensional vectors \(x_1, \ldots, x_p\) or, equivalently, an \(n \times p\) data matrix \(X\), whose jth column is the vector \(x_j\) of observations on the jth variable. This technique seeks a linear combination of the columns of matrix \(X\) with maximum variance.

PCA reduces data by geometrically projecting it onto lower dimensions called principal components (PCs). The goal is finding the best summary of the data using a limited number of PCs (usually much lower than the total number of PCs). The first principal component is generated in order to minimize the distance between the data and their projection on that PC. By minimizing this distance, it also maximizes the variance of the projected points. The additional requirement is that they are uncorrelated with all previous PCs. For example,
projection onto PC number 1 is uncorrelated with projection onto PC number 2. This implies that the PCs are geometrically orthogonal to each other. This requirement of no correlation means that the maximum number of PCs possible is either the number of samples or the number of features, whichever is smaller.

The use of PCA in this thesis is to take the first \( p \) Principal Components (PCs) such that the cumulated explained variance by the \( K \) PCs is at least 99%. It allows to create a new representation of the data that is of lower dimensionality, while still keeping most of the information of the original data.

2.4.2 Visualization techniques

2.4.2.1 T-SNE

T-SNE or t-Distributed Stochastic Neighbor Embedding [72], is a data dimension reduction capable of retaining the local structure of the data while also revealing more global structure like clusters. It is a non-linear (as opposed to PCA), non-parametric technique, also called a manifold learning technique. Manifold learning [14] is a modern approach to non-linear dimensionality reduction. Algorithms for this task are based on the idea that the dimensionality of many data sets is only artificially high and that the ‘real’ structure can be ‘explained’ on a much lower dimension.

T-SNE has been shown better than previous methods at creating a mapping revealing structure at different scales [72]. It reduces the dimensionality of the data mainly based on its local structures, it doesn’t conserve intrinsic distances and density of the original data, therefore it should be used mostly for visualizations.

2.4.2.2 UMAP

UMAP or Uniform Manifold Approximation and Projection for Dimension Reduction [50], is a recent manifold learning technique, similar to T-SNE for dimension reduction. It is built from a theoretical framework based in Riemannian geometry and algebraic topology. The UMAP algorithm has been shown as effective as t-SNE, and has the advantages to being usable as a general purpose dimension reduction technique for machine learning, as opposed to T-SNE, meaning that UMAP can be used before other tasks than visualization only, as opposed to the previously described, T-SNE.

2.5 Analysis of the clusters of the data representations

This section focuses on the clusterability analysis of the different representations created from the data. This section will detail the techniques used to cluster the data in the different representations. In this thesis, clustering the data is a sub-objective that will be evaluated qualitatively and quantitatively.

In [2], it is observed that when a data set is well clusterable, it is computationally “easy” to find a near optimal clustering of it. Moreover, it is admitted that certain clustering algorithms (the one used in the
following) perform well, and [2] shows that when they do not produce satisfactory clusterings, it is because of insufficient clustering structure in the data.

2.5.1 Internal evaluation metrics

This section will describe several internal metrics functions [63] used in this thesis. These functions help to evaluate, in an unsupervised context, if a clustering of points is valid [16]. It is also beneficial to compare different clusterings of the same representation of the data when there are no labels available.

2.5.1.1 Notation

There are N observations (or data points). The data is assumed to be partitioned in K clusters. Let’s denote by \( M_1, \ldots, M_N \) the points representing all the observations, or data point. The coordinates of \( M_i \) are the coefficients of the i-th row of the data matrix A.

Let’s denote \( I_k \) the indices of all observations (or data points) in a cluster \( k \).

2.5.1.2 Davies-Bouldin index

This Davies-Bouldin index [16] is a function aimed at scoring the compactness and separation of clusters of data points. It is made of two main sub-measures, the distance between the means of two clusters and the scatter within a given cluster.

Let’s denote by \( \delta_k \) the mean distance of the data points in a cluster \( C_k \) to their barycenter \( G_k \). 

\[
\delta_k = \frac{1}{n_k} \sum_{i \in I_k} \| M_i^k - G_k \| 
\]

And let’s denote by 

\[
\Delta_{kk'} = \| G_k - G_{k'} \| 
\]

the distance between the barycenters of the clusters \( C_k \) and \( C_{k'} \).

The Davies-Bouldin index is then defined by:

\[
DB = \frac{1}{K} \sum_{k=1}^{K} \max_{k' \neq k} \left( \frac{\delta_k + \delta_{k'}}{\Delta_{kk'}} \right) 
\]

A lower DB index indicate a better clustering.

2.5.1.3 Silhouette

The Silhouette value [57] is a measure of the similarity of a data point compared its own cluster, the cohesion, compared to other clusters, the separation. The Silhouette value ranges from \(-1\) to \(+1\). With a high value indicating a ‘good’ clustering. Let’s consider for each point in the data, its mean distance to every clusters. First the intra-cluster mean distance is then defined, for each point \( M_i \),
to the other points of its own cluster $C_k$:

$$a(i) = \frac{1}{n_k - 1} \sum_{i' \neq i, M_{i'} \in C_k} d(M_i, M_{i'}) \quad (6)$$

Then let’s evaluate the mean distance $\delta(M_i, C_{k'})$ of $M_i$ to the points of all the other clusters $C_{k'}$:

$$\delta(M_i, C_{k'}) = \frac{1}{n_{k'}} \sum_{M_{i'} \in C_{k'}} d(M_i, M_{i'}) \quad (7)$$

Let’s denote $b(i)$ the smallest of these mean distances:

$$b(i) = \min_{k \neq k'} \delta(M_i, C_{k'}) \quad (8)$$

Then a quotient $s(i)$ is calculated, called the silhouette width of the point, and is a quantity between -1 and 1. A value near 1 indicates that the point $M_i$ is affected to the right cluster, and a value close to -1 indicate that this point should become to another cluster.

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (9)$$

Finally, the global silhouette index is the mean of the mean silhouettes width of a cluster $C_k$, through all the clusters:

$$SIL = \frac{1}{K} \sum_{k=1}^{K} \left( \frac{1}{n_k} \sum_{i \in I_k} s(i) \right) \quad (10)$$

### 2.5.1.4 Calinski-Harabasz index

First let’s define the within-cluster dispersion $WGSS_k$ i.e. the sum of the squared distances between the data points $M_i^k$ and the barycenter $G^k$.

$$WGSS_k = \sum_{i \in I_k} \| M_i^k - G^k \|^2 \quad (11)$$

Then the pooled within-cluster sum of squares $WGSS$:

$$WGSS = \sum_{k=0}^{K} WGSS_k \quad (12)$$

Now let’s define the between-group dispersion measure. It is defined as the dispersion of the barycenters $G^k$ of each cluster compared to the barycenter $G$ of the whole dataset.

$$BGSS = \sum_{k=1}^{K} n_k \| G^k - G \| \quad (13)$$

The Calinski-Harabasz index is defined by the following formula:
\[ CH = \frac{N - K \cdot BGSS}{K - 1 \cdot WGSS} \]  

(14)

A high Calinski-Harabasz index indicate a good clustering.

### 2.5.2 Clustering algorithms

Clustering algorithms [21] can be categorized into four groups: partitional, hierarchical, density-based and grid clustering. In this thesis, the focus is made on a partitional clustering algorithm, K-Means, and a density-based algorithm named HDBSCAN.

#### 2.5.2.1 Partitional clustering algorithms: K-Means

K-means is one of the most known clustering paritional algorithms. It will be used later as a first approach of clustering.

This algorithm takes as input the number K of clusters. First, the K cluster centers are initialized randomly and are then iteratively updated to better fit the given dataset. This is done until a given convergence criteria is met (e.g. minimizing the least square error).

#### 2.5.2.2 Density-based clustering algorithms

Density-based clustering is based on the idea that a cluster is defined by a higher density of data points than their surroundings. The more often used density-based algorithm is DBSCAN [11] which is based on the concept of density-reachability [19]. There exist a hierarchical version of the DBSCAN algorithm, called HDBSCAN [12].

A point A is density-reachable from another point B if they are within a fixed distance from each other, and if there are sufficiently many points (the neighbor threshold), surrounding the point A, so that a cluster can be formed including them both. One characteristic of density-based clustering algorithm, as opposed to a traditional K-Means algorithm, is that they don’t have to cluster all the points in a dataset, and some can be discarded as “noise”. Density-based clustering algorithms are better suited to cluster the data with non-convex clusters (or of any arbitrary shape) that could be present in the different representations studied in this work.
3 Method

This section will describe the methodology used in this work. First, the data specification will be detailed with a preprocessing step. The creation of different representations of the data will be described. Two main analyses will be presented. A clusterability analysis of the different representations will be done, by performing several clusterings. Visualization will be performed on the clusterings and on the representations, as well with the use of internal metrics, to evaluate quantitatively both the clusterings and the representations. Finally, an analysis will be performed to evaluate the nearest-neighbours clustering of the different representations of the data, by including human evaluation to score the different representations. This last analysis will help to draw the final results and conclusions on the different representations and their relevance for clustering the ELN data.

3.1 Data specification and preprocessing

3.1.1 Dataset specification

The corpus is extracted from a real-world Electronic Lab Notebook (ELN). It is made of a set of 75,000 life sciences experiments, totaling more than 84 million raw words (683,747,281 characters).

The data is almost completely unstructured:

1. Each experiment has a Group-ID (integer) identifying the group of researchers (organization, lab, client, etc.) originated. There are 730 different Groups in the data.

2. Each experiment is made of plain text (UTF-8 encoding), unlabeled and unstructured.

3.1.2 Text type and noisy data

The data is constituted of real-world experiments, written by researchers in life sciences. The corpus of experiment ranges across different scientific domains: molecular biology, biochemistry, cytology and immunology. It is composed of experiments in different languages (e.g., English, French, Swedish), without any metadata on which language is used in a given experiment. Experiments can range from a few lines (50 words) to a few pages (1000-2000 words).
Figure 12: Length distribution of the data (in number of characters)

Experiments can range from very well written articles to very poorly written experiments, i.e. from long and valid sentences to a few sentences (as illustrated in Figure 12) (sometimes improper sentences).

Researchers use a great number of abbreviations. Experiments are also constituted of numerical arrays and tables (excel sheets, other formats), present in the text content of the experiment itself. There is also a lot of meta-data, with some specific structures: chemical structures and software-specific arrays, chemical compounds and traces left by several programs (software) used by researchers. Additionally, there is a lot of personal data: email, URLs, keywords, paths (Windows or Unix file-paths). Overall this is a very noisy and heterogeneous data set.

3.1.3 Preprocessing and cleaning

Some aspects of the data that are obviously irrelevant to the study, can be cleaned before hand: emails, URLs, file paths and some obvious redundant metadata keywords.

Some aspect of the data, on the other hand, has to be cleaned: the chemical tables, and other numerical tables. 'Single numbers' (that are not part of any word or entity) are also removed during this preprocessing phase (some numbers have to be kept, like for example in chemicals compounds e.g. '3,5-di-tert-butyl'). This is done because statistical approaches are based on the statistical similarities of textual entities in a document as opposed to the rest of the corpus. Therefore, keeping 'single numbers' before creating a document representing often results in documents being associated together because of the numerical characters they have in common. They are, therefore, considered, in this thesis, as noise (like personal data and meta data). This cleaning step is done by using regular expressions [3] to perform automatic searches of these specific patterns in the text to remove them.
3.2 Notation

In this thesis, the following notations will be used:

1. Experiments will also be referred as "documents".

2. A word or token is a string of contiguous characters between two spaces, or between a space and punctuation marks.

3. A sentence refers to any string of characters.

4. The ELN data (Electronic lab notebook experiments) will be referred as the 'corpus' or the 'data'.

3.3 Creating vector representations of the data

One major aspect of this work is to explore different types of vector representations (embeddings) of the corpus of 75,000 experiments. The concepts of representing a given word meaning by a vector is expanded in this thesis to represent an experiment meaning. Vector representations enable direct comparison between the vectors (i.e., the experiments). It enables a direct clustering technique where, for a given experiment represented by a vector, it is possible to query its nearest neighbors (based on the distance between vectors) to extract the most scientifically related experiments in the corpus.

However, the techniques, described in section 2.2, create different vector representations, embedding meaning differently, and handle text at different atomic levels (words, characters). To explore which representation is the best suited for the clustering of life sciences experiments, it is necessary to explore a variety of models. Therefore, the goal of this section is to create different representations of the data which can be compared and evaluated to assess quantitatively and qualitatively how they improve clustering of the corpus.

In the following, the methodology used to create the representations of the data will be detailed.

The methodology used in this work is to explore many different types of representations, and different machine learning or statistical techniques. The following subsections will describe, in an exhaustive manner, the techniques and the associated data pipelines that will be used to produce the different vector representations of the ELN data.

3.3.1 Word-based representations

The following techniques use words as atomic units, like in the bag-of-words representation, and to do so, they are using a Dictionary that maps all the unique words in a corpus to an ID (usually a simple integer).

3.3.1.1 TF-IDF: term frequency-inverse document frequency

A representation of the corpus is created with the TF-IDF technique. The conversion of an experiment in the corpus to a vector of TF-IDF weights is done
by using the equations 1 and 2 in section 2.3.1.1.

When dealing with real-world documents, the number of dimensions increase dramatically because the number of dimensions equals to the number of unique terms in the entire set of documents. In this work, it can reach thousands of words. Therefore, in practice, a TF-Idf "Vectorizer" is to used to produce the matrix of TF-IDF weights. Some parameters, here $min_{df} = 5$ and $max_{df} = 0.95$, serve to reduce the number of features, to reduce the complexity of the TF-IDF model. The arguments $min_{df}$ and $max_{df}$ are used for removing terms that appear, respectively, too infrequently or too frequently.

Following the TF-Idf Vectorization, a Truncated SVD is performed with 100 components (suitable number of component found by manual inspection). Truncated SVD or singular-value decomposition is a factorization-reduction method of a matrix. It is used here as a dimensionality reduction technique similarly to the Principal Component Analysis (PCA). PCA is not used here because it can be viewed as an equivalent to Truncated SVD on centered data (per-feature mean substraction), but it cannot be achieved efficiently memory-wise for such a large data.

![Figure 13: Illustration of the creation of the TF-IDF representation](image)

The resulting representation for a single experiment in the data is then a 100 dimensional real vector.

### 3.3.1.2 LDA topic modeling : Latent Dirichlet Allocation

A representation of the corpus is created with the LDA topic modeling technique described in section 2.3.1.2.

As described in 2.3.1.2, bigrams and trigrams (respectively a sequence of two or three adjacent words in the text) are added to the bag-of-words representation. Bigrams and Trigrams are joined by underscores are thus treated as single tokens. During experimentation, it is observed that this process helps LDA produce more relevant and self-explanatory topics.

One issue with LDA is the number of topics that have to be chosen beforehand. This parameter has a big impact on the topics generated. Indeed, topics are not guaranteed to be well interpreted, and it can be hard to decide on how many topics are needed while doing manual inspections of the results. To be able to evaluate (if possible) the needed number of topics to better represent the data, an internal metric has been developed, named Topic Coherence. As detailed in [62], each generated topic consists of words, and the topic coherence measure is applied to the top N words (usually around 10) from the topic. It is defined as the average of the pair-wise word-similarity scores of the words in the topic.
To simplify, it is an automated process, testing the "quality" of generated topics. The goal is to generate 'coherent' topics i.e. topics with high topic coherence scores. Good topics can generally be described by a short label or a few terms; Therefore, this is what the topic coherence attempt to measure.

The coherence value is produced from a pipeline that is not relevant to this thesis and will not be detailed. The number of topics evaluated to fit the data better is assumed to yield the highest coherence score. However, too many topics produce repeated keywords in topics. The goal is then to find the lowest number of topics that represents actual groupings in the data.

Figure 14: Coherence values in the y-axis for the associated number of topics in the x-axis

Finally, one LDA model with 30 topics is selected, based on its highest coherence score of 0.610, as seen in Figure 14, and another one with 10 topics, although not associated with the highest coherence score. The latter was selected because the topics were judged, during inspection, to be coherent. It has to be noted that the difference in coherence values was not significant (bounded between 0.52 and 0.62).

The inputs to the topic model LDA are the bag-of-words representation for all the documents in the corpus, and the number of topics. Stop words are filtered similarly to the filtering in section 2.3.1.1. Only words that are present in at least 10 documents, but not more than 20% of all documents in the corpus are kept in the representation.

The output of LDA algorithm are 2 matrices: a document to topic matrix and a topic to word matrix. The Document-Topic matrix is of (N,K) dimensions where N is number of experiments and K is number of topics in the vocabulary. The Topic-Word matrix is of (K,M) dimensions where M is the number of words in the vocabulary. The vector representation for a given experiment is the corresponding row in the document-topic matrix.
The resulting representations for a single experiment in the data is then respectively, for the two LDA models selected, a 10 and 30 dimensional real vector.

### 3.3.2 Word embeddings based techniques

#### 3.3.2.1 Doc2Vec

A representation of the corpus is created with the Doc2Vec technique described in section 2.3.2.2. As the goal of this thesis is to explore the representation of every experiment in the data, each experiment is assigned a unique ID and then the Doc2Vec training process will produce a unique doc-vector per document as output. Compared to the Word2Vec process, at the end of the Doc2Vec training process, different elements are available: word embeddings W and document embedding D for documents in the training corpus. In this section, the word embeddings are discarded as the focus is on the document representations.

The resulting representation for a single experiment in the data is a 100 dimensional real vector. Additionally, another representation is created by applying UMAP to the 100-dimensional Doc2Vec UMAP representation. The parameters of the UMAP transformation: number of neighbors = 10, min dist = 0, n components = 15. This representation is created to explore the impact of a much lower-dimension embedding on the results.
The resulting representation for a single experiment in the data is a 15 dimensional real vector.

3.3.3 Document representations with Contextualized string embeddings

3.3.3.1 Pooling

In this thesis, the aim is to get document embeddings. Document embeddings are created from the embeddings of all words in a document. The technique used here is called pooling, which applies an operation over all word embeddings in a document to produce the final document embedding. In this case the pooling operation is the ‘mean’, which gives the mean vector of all words embeddings present in a document. Other operations could be explored like min, max or even custom weighted average. The pooling has the advantage of producing a fixed vector length representation for each document, regardless of the size of that document. This could become a problem when dealing with very large documents, as the meaning of different section would be ‘averaged’ in the final vector, but this is not explored in this thesis.

3.3.3.2 Concatenation of vectors

The concept of concatenation has to be briefly introduced. Concatenating vectors together is simply taking them and putting them together, to form a new larger vector. The output vector dimensionality depends on the dimensions of the vectors taken as input to the concatenation.
3.3.3.3 Flair

A representation of the corpus is created with the Flair technique described in section 2.3.3.2. For the Flair-based techniques, the maximum string length is 64 characters. A mean-pooling operation is applied on the Flair words embeddings of each word in a document to produce the final document representation. The following flowcharts explain the process for a given experiment:

Figure 19: Illustration of the creation of the Flair representation

In the rest of this thesis, the term ‘Flair’ will refer to the Contextual String Embeddings. The resulting representation for a single experiment in the data is a 1024 dimensional real vector, because the two language models from the bi-directional LSTM network have 512 units.

3.3.3.4 Flair-Glove

A representation of the corpus is created with the Flair-Glove stacked embeddings technique described in section 2.3.3.2. The maximum string length is 64 characters. The final output embedding of a word in this context is the concatenation of the Contextual String embeddings of that word with the Glove embedding of that word.

A mean-pooling operation is applied on the stacked Flair and Glove word embeddings of each word in a document to produce the final document representation.

The following flowcharts explain the process for a given experiment:
The term ‘Flair-Glove’ will refer, in this work, to the embedding (i.e., the vector representation) made of the concatenation of the Flair and Glove embedding. The resulting representations for a given string in the data, is a 1124 dimensional real vector for the Flair-Glove stacked embedding.

### 3.3.3.5 BERT-based document representation

This section details how documents representations are created from the string representations created by BERT-based models (BERT, BioBERT and ELN-BioBERT), described in section 2.3.3.4. As said in section 2.3.3.4, the BERT model can be combined with another task (sentence prediction, sentiment analysis, Semantic Role Labeling, etc.) but it can also be used directly, to do sentence embedding, using the BERT model as a feature-extractor. This is how it is used in this thesis. In this case, the contextualized sentence embedding is extracted from the last two layers of the BERT architecture [17], but other combinations are possible and are not explored in this thesis. The resulting embedding for every BERT-based architecture, for a given string (maximum 64 characters long) passed as input to the model is a 768-dimensional vector (the size of the hidden layer in the BERT architecture).

In the following, the procedure described will only refer to BERT, as the three BERT-based models only differ in their pre-training (in essence this is the same model trained differently).

As described in section 2.3.3.3, BERT can be used as a feature-extractor (a sentence encoder) that is, an embedding is created for any given string passed as input to the model. The maximum string length is 64 characters, for comparison purposes with Flair.

To use BERT as a document encoder in this work, a procedure was designed to create a single representation of a full document. For every document in the data, the following process is applied:

1. Divide each document into smaller string chunks of a maximum of 64
characters. The chunking procedure does not cut words. If during the chunking, adding a word to the current chunk would create a string longer than 64 characters, that word is ignored (and will be included in the next chunk).

2. Each list of n string (for each document) is then passed to the BERT model, which outputs a list of n 768-dimensional embedding (real vectors).

3. Finally, a mean-pooling operation is applied on every list of n vectors to create an averaged 768-dimensional vector, representing each document.

The following flowcharts explain the process for a one given experiment:

Figure 21: Illustration of the creation of the BERT-based representations

The final resulting embedding, or representation, for a document is then a 768-dimensional vector. This is the exact same process that is performed for BERT, BioBERT and ELN-BioBERT.

3.3.3.6 Fine-tuned ELN BioBERT

In this section, the pre-trained BioBERT model, described in section 2.3.3.5 and available publicly [42], is fine-tuned on the ELN corpus data. This creates, in addition to BERT and BioBERT, a new model very close to BioBERT that is fine-tuned on the studied corpus.

The following illustration is inspired from BioBERT [41].
After a few inspections, the fine-tuning was performed for 20,000 steps on a small subset of the data, with the following hyper-parameters. The subset of the data used was composed randomly and made of 25,000 experiments (28,441,120 words). This was decided because of the strong computational requirement (in both time and memory) that such a process imposes.

Table 2: Hyper-parameters used during the fine-tuning of the BioBERT architecture to produce ELN-BioBERT

<table>
<thead>
<tr>
<th>Hyper-parameter</th>
<th>Values used</th>
</tr>
</thead>
<tbody>
<tr>
<td>training batch size</td>
<td>64</td>
</tr>
<tr>
<td>max sequence length</td>
<td>64</td>
</tr>
<tr>
<td>max predictions per sequence</td>
<td>10</td>
</tr>
<tr>
<td>number of training steps</td>
<td>20,000</td>
</tr>
<tr>
<td>number of warm up steps</td>
<td>100</td>
</tr>
<tr>
<td>learning rate</td>
<td>2e-5</td>
</tr>
</tbody>
</table>

3.3.4 Knowledge-based models

3.3.4.1 Ontology-Entity based document representation

A representation of the corpus is created with the Ontology-Entity based technique described in section 2.3.4.1. Different techniques are possible: using only the concepts extracted from each document to build their representation, creating a weight-based system [45] that add more weights to the concepts-words than the rest of words in a document to then build the document representation or some kind of or a hybrid system.

To extract the various life-sciences concepts that could exist in every experiment composing the data, the BeCAS tool is used, the Biomedical Concept Annotation System [13].

The extraction technique makes use of multiple different knowledge bases such as UMLS (Bodenreider, 2004), LexEBI (Sasaki et al., 2008), Jochem (Hettne et
al., 2009), the Gene Ontology (GO) and NCBI (Geer et al., 2010). An example of the annotation system is shown below.

Figure 23: Example of the concept extraction on a public sample, each color represents a different concept

![Figure 23](image)

Figure 24: Categories associated with their color with the associated semantic concept

Here, the choice is made to create a document representation based solely on the concepts extracted from the documents. This is illustrated by Figure 23, where the extracted concepts are colored according to their scientific context, as shown in Figure 24. There could be a possibility to create multiple representations for each semantic type extracted in a document, but here the representation is created only from the annotated words (i.e. labeled by the annotation tool), the other words of the document are discarded.

The arbitrary choice made here is to create a Flair-Glove representation, as described earlier in section 2.3.3.4, of each extracted concept-words (so a 1124-dimensional real vector), and then to do a mean-pooling to create a mean vector (also 1124-dimensional).

In the following illustration, the Flair-Glove process is simplified into one block, as it was detailed in section 2.3.3.2.
This final document representation based on its extracted concepts will be referred, for the rest of this thesis, as LSCO-Flair-Glove, for Life-sciences-Concepts-Only-Flair-Glove.

3.3.5 Concatenations of multiple representations

In this work, the choice is to study the effects of combining (by concatenating) several representations into new ones. For example, given two vectors for the same document in two different representations, a third vector is produced by concatenating the two vectors. Then a linear dimensionality reduction with PCA, described in section 2.4.1.1, is performed. This dimensionality reduction is made to remove any potentially correlated features between the two embeddings. Here, PCA is done to keep at least 0.99 of the variance (99% of the total information). Depending on the embeddings concerned by the concatenation, the number of Principal Components kept to produce the resulting embedding is variable, resulting in a final vector having a variable dimension (but still lower than the sum of the original embeddings used). The following is an illustration of the process:

![Figure 26: Illustration of a representation created after concatenation and PCA dimensionality reduction](image)

The notation used to describe a representation created by the concatenation and PCA process, described above, for a given representation A and B, will be
referred as ‘A + B + PCA’. A list of seven representations are created from a combinations of the existing representations. They are created from the process illustrated in Figure 26. The following list enumerates the combinations:

1. LSCO-Flair-Glove and LDA (30 topics) are combined. This representation named ‘LSCO-Flair-Glove + LDA (30 topics) + PCA’ is made of 178-dimensional documents vectors.

2. LSCO-Flair-Glove and Flair-Glove are combined. This ‘LSCO-Flair-Glove + Flair-Glove + PCA’ representation is made of 275-dimensional vector documents vectors.

3. ELN-BioBERT and LSCO-Flair-Glove are combined. This representation named ‘ELN-BioBERT + LSCO-Flair-Glove + PCA’ is made of 596-dimensional vector documents vectors.

4. ELN-BioBERT and Flair-Glove are combined. This ‘ELN-BioBERT + Flair-Glove + PCA’ representation is made of 589-dimensional documents vectors.

5. ELN-BioBERT, Flair-Glove and LDA (30 topics) are combined. This ‘ELN-BioBERT + Flair-Glove + LDA (30 topics) + PCA’ representation is made of 466-dimensional documents vectors.

6. ELN-BioBERT, Flair-Glove and LSCO-Flair-Glove are combined. This ‘ELN-BioBERT + Flair-Glove + LSCO-Flair-Glove + PCA’ representation is made of 537-dimensional documents vectors.

7. Flair-Glove, LSCO-Flair-Glove and LDA are combined. This ‘Flair-Glove + LSCO-Flair-Glove + LDA (30 topics) + PCA’ representation is made of 241-dimensional documents vectors.

3.3.5.1 Random projections representation

For comparison purposes, a representation is created randomly. For each document in the corpus, a 10-dimensional vector is generated with random real numbers between -1 and 1. This representation will be used to produce the lowest baseline possible, as there is no meaning in random vector projections.

3.4 Clusterability analysis of the data representations

In this thesis, clustering the data is a sub-objective that can be evaluated qualitatively and quantitatively. This section will detail the techniques applied to cluster the different representations and how visualization and internals metrics are used to evaluate the clusterings.

In [2], it is observed that when a data set is well clusterable, it is computationally “easy” to find a near optimal clustering of it. Moreover, it is admitted that certain clustering algorithms (the one used in the following) perform well, and [2] shows that when they do not produce satisfactory clusterings, it is because of insufficient clustering structure in the data.
From this thesis perspective, the existence of a clusterable structure in the data is a priori unknown. Each technique produces a representation expected to be different that the others representations considered. As a result, it is not acceptable to draw any conclusion from the analysis of only a single representation of the data. Therefore, the goal of this section is to evaluate the clusterability of the different representations of the data.

This section will consist first in a visualization of the representations in 2 dimensions. An attempt to cluster the representations via two different clustering algorithms will be performed. The number of clusters will be evaluated if possible via an ‘elbow’ method. This section is explorative, as there is no assumption on the shape or topology of the data.

3.4.1 Visualization as a qualitative comparison method

The visualizations using the T-SNE algorithm can be useful in providing insight as a qualitative assessment. It has been shown that T-SNE is able, in a lot of applications, to recover well-separated clusters, and to a less extent approximates a form of spectral clustering [43]. The final T-SNE transformation gives data points that reflect the structure of the high dimensional data and are able to be plotted on a 2D or 3D graph. In the results, the 2D T-SNE visualizations of the different representations of the data will be shown.

This quantitative visualized analysis will consist of using a fixed set of 35 groups of experiments, in the data, and to color them in the same way, in all visualizations. These 35 groups will be referred to as the ‘studied groups’. This analysis allows the visualizations to be compared by observing where these specific groups are, regarding the other points, and the other groups. It can show how some groups of experiments (data points in the visualizations) are ‘clustered’, if they are isolated or if then are ‘spread’ out, in a representation of the data. Furthermore, visualizing allows to understand, on a global scale, how different or similar is the shape of the data between the different representations. This qualitative assessment is only comparative.

3.4.2 Evaluation of the number of potential clusters in the data

The “elbow” method is a technique that helps decide on the number of clusters existing in a data set, by using an internal metric. One should decide on the number of clusters in a progressive manner, so that adding another cluster doesn’t give a significantly better modelling of the data. The first clusters will add so much information, but at some point the gain will drop and gives an angle in the graph if the data is well clusterable. The function used could be any relevant metric, depending on the domain. One could use external metrics like the sum of squared errors if some labels associated with the document were available (i.e., a supervised case). However, in this thesis, the problem is fully unsupervised. This constrains the study to only use internal metrics [63] as a scoring function, like the Davies Bouldin index, the Silhouette technique, or the Calinski-Harabasz index, detailed in section 2.5.1.

The Silhouette score and the Calinski-Harabasz index have been proven valuable
on a wide range of applications and types of data, and thus will be used to produce the elbow plots.

### 3.4.3 Distance metrics

Distance measure are used almost in all clustering algorithms and play a very crucial role [55]. Distance measures are also used in other applications that clustering, notably later in this thesis when getting the nearest neighbors of a point.

In the context of this thesis, data is unlabeled, and it is not possible to test easily whether a distance measure gives better results than another. Various metrics exist (L1-norm, cosine distance, etc.). As we can’t explore many metrics in the unsupervised context of this thesis, the choice, after some experimentation, is to use the classic Euclidean distance:

The following defines the Euclidean distance for two points a and b, with d dimensions.

\[
d_{\text{euclidean}} = \sqrt{\sum_{i=1}^{d} (a_i - b_i)^2}
\] (15)

### 3.4.4 Clustering algorithms

Clustering algorithms [21] can be categorized into four groups: partitional, hierarchical, density-based and grid clustering. In this thesis, the focus is made on a partitional clustering algorithm, K-Means, and a density-based algorithm named HDBSCAN.

#### 3.4.4.1 Traditional clustering algorithms: K-Means

K-means is one of the most known clustering algorithms. It will be used later as a first approach of clustering.

This algorithm takes as input the number of clusters K. First, the K cluster centers are initialized randomly and are then iteratively updated to better fit the given dataset. This is done until a given convergence criteria is met (e.g. the least square error being below a threshold).

#### 3.4.4.2 Density-based clustering algorithms: HDBSCAN

In this thesis, the Hierarchical version of the DBSCAN algorithm will be used, also called HDBSCAN [12]. This is a density-based clustering algorithm, like described in section 2.5.2.2.

DBSCAN and HDBSCAN share a common parameter: the minimum number of data points in a sample, i.e., the number of samples in a neighbourhood for a point to be considered a core point (i.e. as opposed to a noisy point). DBSCAN has a parameter \( \epsilon \), which is the radius those neighbors have to be in for the core to form. As opposed to that, HDBSCAN has a parameter minimum cluster size, which is how big (regarding the number of points) a cluster needs to be to actually form. This is usually more intuitive to set that parameter than epsilon. An additional advantage is that the HDBSCAN is generally faster than DBSCAN. HDBSCAN is built for real-world data with varying density, which
is the case of this thesis. Finally, by tweaking another parameter, the minimum cluster size parameter, it is possible to extract bigger or smaller clusters in the data.

Additionally, UMAP has been shown in other applications to improve density-based clustering methods like DBSCAN and HDBSCAN [50]. It is beneficial to reduce the dimensionality of an embedding before attempting a clustering [75]. Therefore UMAP will be used a non-linear dimensionality reduction technique. The UMAP embedding of each representation will then be passed as input to the HDBSCAN to be clustered. UMAP has several hyper-parameters that can transform the resulting embedding.

![Figure 27: Illustration of UMAP and HDBSCAN used together to do clusterings](image)

The hyper-parameters will be set by doing manual inspections (visualizations) of the resulting clusterings.

### 3.5 Analysis of the nearest-neighbours clustering of the representations

This section aims to assess the quality of the representations of the data, locally, by exploring the nearest-neighbours clustering of each point (experiments) in the various representations. The fundamental assumption of using vector representation is that, if two given experiments are close semantically (related content), their vector representations will be close in the embedding space as well.

The nearest neighborhood of a point is defined here by its N-nearest neighbors. They are extracted by calculating the pair-wise Euclidean distance to all the other vectors (representing experiments) in the vector space, ordering them by lowest distance and then taking the first N vectors.

The distance used in the nearest-neighbor extraction process is the Euclidean distance; the vectors composing the representations are normalized (change their length to unit length, turning them unit vectors) beforehand.

For every representation, it is possible to explore the local neighborhood of every point up to its N-nearest neighbors. N is a parameter that allows us to look more or less locally around a considered data point. However, this analysis is only interesting “locally” and here, the choice is made to look closely around a point, with \( N = 4 \). Two analysis will be performed in this section: a nearest neighbor similarity comparison between the representation and a human evaluation of the clusterings of the representations.
3.5.1 Nearest neighbor and local analysis

In this section, an experimental analysis is used to compare the different representations on a local scale. This is performed to compare the local neighborhoods of every representation. This comparison can be useful to evaluate how similar two different representations are locally. This nearest neighborhood exploration is helpful to understand, when 2 representations are considered, and for the same fixed experiment (i.e. a point of interest in the data), how much the two representations share similar experiments in the set of the N-nearest neighbors of that experiment (the point of interest). With that approach, it is only possible to compare in pair-wise manner all the different representations produced by all the techniques considered.

In Figure 28, the point in red is the point of interest, the points in blue are its N-nearest neighbors with N=4 and the points in black are the surrounding points too far to included in its neighbors. The experimentation is to inspect, for the same fixed point of interest, whether some of the nearest neighbors (the blue points) are the same (i.e. referring to the same experiment in the original data) in other representations produced by other techniques and to what extent. That experiment will be performed for every point and for every representation and the percentage of similarity of the nearest neighbors shared between the representation is calculated. This quantitative assessment allows to tell whether, for two given representations, they are ‘similar’ or ‘dissimilar’ when explored locally.

3.5.2 Human assessment of the nearest-neighbours clustering

This section will describe the method for the human assessment of the nearest-neighbours clustering.

3.5.2.1 Assessment

The various metrics and quantitative results extracted from the representations isn’t enough to rank and compare them, when considering the task of clustering.
the experiments in a scientifically relevant way. The only way to analyse which representation is best for clustering, when compared to others and when dealing with fully unsupervised techniques and unlabeled data, is to involve some human assessment. This is what will be described in this section. The decision is to get human annotators, which are experts in life sciences, to rank an experiment in a representation with its N-nearest neighboring experiments. Here N is 4.

The goal is to determine which representation yield, for a fixed target experiment, the most scientifically related N-nearest neighbors. The definition of relatedness is partly subjective even for scientific experiments because of different criteria that could be taken into account. In this thesis, the relatedness is evaluated with the help of human annotators on two criteria. The first criteria is the scientific content (type of chemistry, biology, disease being studied, etc.) and the second is the writing format (type of experiment, phrasing, etc.). In other words these two criteria are focused on the content and the format of a document.

To perform an evaluation and compare the results for the different representation, a fixed set of target experiments (built from a random subset of the data) is created and is used to extract their N-nearest neighbors in the $N_{rep}$ representations included in the assessment. To clarify, the annotators will be shown, for each representation, the same target experiment with its 4 nearest neighbors. The target experiment is always the same, but the neighbors will differ according to the representations. At the end of assessment, each annotator will have rated (on both criteria) $N_{rep} \times N_{tar}$ evaluation sets (1 target + 4 neighbors = 5 experiments), where $N_{tar}$ is the number of target experiment per representation.

This evaluation is done on an assessment platform made of a login page (to identify the annotator) and the assessment page. Concretely, the platform will present a target experiment with its 4 nearest neighboring experiments, and ask a given annotator to rate it, on a scale of 1 to 5 (possible ratings are 1, 2, 3, 4 and 5) for each criteria (scientific content and writing format). The sets shown are randomized to ensure a proper evaluation with no bias.

As the human assessment takes a relatively long time per evaluation set, the evaluation has to be restrained to a lower number of $N_{rep}$ representations than the 18 representations of the data studied in this thesis. To assess all the techniques, it could be possible to do the evaluation with a lower number of documents per technique, but lowering implicitly the conclusions drawn from the results as the subset of experiments evaluated has to be statistically significant. The choice in this evaluation is then to focus on a lower number of representation, allowing the annotators to evaluate more experiments per representation. To make that choice, the section 3.5.1 can help to not pick representations that already too similar locally, because then the extracted ratings of such representations would be, as expected, similar. The choice here is to pick a relatively diverse set of representations. The results and parameters of this assessment will be detailed later.
3.5.2.2 Inter-annotator agreement

The inter-annotator agreement is the degree of agreement among raters. It is a score of how much consensus there is in the ratings given by the annotators on the platform. This can be calculated via different methods depending on the type of annotations.

The human assessment, in this case, produces ordinal ratings (a ranking between 1 and 5). A fitting statistical measure for that kind of rating is the Krippendorff’s alpha coefficient [38].

The Krippendorff’s alpha coefficient is a value between 0 and 1. Closer to 1 means a perfect agreement, and closer to 0 means a very bad or nonexistent agreement.

3.6 Implementation

3.6.1 Programming language and libraries

The programming language used is Python 3.

General machine learning and NLP tools were used from Gensim, NTLK, Tensorflow, Pytorch and Sklearn.

Internal metrics are used from the Sklearn package as well.

Plots are produced with the Bokeh, Plotly and Matplotlib library.

Data management was done with the Pandas and Numpy package. BERT and BioBERT models and tools are openly available, as well with the Flair model.

The sentence embedding extraction process was performed with the Bert-As-Service package.

The annotation platform was built with the Flask Framework (Python).

3.6.2 Hardware

An AWS EC2 instance with a Tesla K80 GPU was used to perform experimentations.
4 Results

This section will describe the results of this work and the different representations of the data. Then visualizations of the representations will be shown. The two main analyses will be detailed: the clusterability analysis and the nearest-neighbours clustering analysis. The clusterability analysis section will detail the elbow method used to evaluate the number of clusters in the data. The different clusterings algorithms are used on the representations and their results will be visualized and detailed quantitatively. The human evaluation of the nearest-neighbours clustering will be detailed, and final ratings will be attributed to the different representations. Finally, correlations between internal metrics and humans ratings will be studied.

4.1 Representations

4.1.1 Vector-spaces dimensions

Table 3: List of the 18 representations with their corresponding vector dimensions

<table>
<thead>
<tr>
<th>Technique name</th>
<th>dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF-IDF with Truncated SVD</td>
<td>100</td>
</tr>
<tr>
<td>LDA with 10 topics</td>
<td>10</td>
</tr>
<tr>
<td>LDA with 30 topics</td>
<td>30</td>
</tr>
<tr>
<td>Doc2Vec-UMAP-15</td>
<td>15</td>
</tr>
<tr>
<td>Doc2Vec</td>
<td>100</td>
</tr>
<tr>
<td>Random-projections</td>
<td>10</td>
</tr>
<tr>
<td>Flair</td>
<td>1024</td>
</tr>
<tr>
<td>Flair-Glove</td>
<td>1124</td>
</tr>
<tr>
<td>BERT</td>
<td>768</td>
</tr>
<tr>
<td>ELN-BioBERT</td>
<td>768</td>
</tr>
<tr>
<td>LSCO-Flair-Glove</td>
<td>1124</td>
</tr>
<tr>
<td>LSCO-Flair-Glove + LDA (30 topics) + PCA*</td>
<td>178</td>
</tr>
<tr>
<td>LSCO-Flair-Glove + Flair-Glove + PCA*</td>
<td>275</td>
</tr>
<tr>
<td>ELN-BioBERT + LSCO-Flair-Glove + PCA*</td>
<td>596</td>
</tr>
<tr>
<td>ELN-BioBERT + Flair-Glove + PCA*</td>
<td>589</td>
</tr>
<tr>
<td>ELN-BioBERT + Flair-Glove + LDA (30 topics) + PCA*</td>
<td>466</td>
</tr>
<tr>
<td>ELN-BioBERT + Flair-Glove + LSCO-Flair-Glove + PCA*</td>
<td>537</td>
</tr>
<tr>
<td>Flair-Glove + LSCO-Flair-Glove + LDA (30 topics) + PCA*</td>
<td>241</td>
</tr>
</tbody>
</table>

*All representations produced after a PCA dimensionality reduction are done systematically to keep 0.99 of the variance (99% of the initial information). Therefore, the dimension of the output can vary between the different inputs used for PCA.

4.1.2 Low-dimensional representations visualizations

This section will detail and show the 2D T-SNE visualizations of the different representations produced by each technique considered in this thesis. Each point represents an experiment that is a document in the representation
As T-SNE is used, the distances and data densities are mostly not representative of the original distances and densities present in each high dimensional vector-space considered. As in T-SNE and other Nonlinear dimensionality reduction techniques, the clusters visualized can be influenced more or less by the parameters. Some “clusters” (areas with a high density of points) can be shown to appear even in non-clustered data, and thus false any direct conclusions drawn from the visualizations. It explains partly why the visualizations will be used as a first approach only, as described in section 3.4.1.

Here, the parameters have been chosen after manual inspection of different plots with a range of different parameters. It appears that the visualizations are consistent across a wide range of parameters. The T-SNE parameters used will be the same across this thesis and they have been shown during several trials that they produce stable results (and provoke no unwanted clusters, as much as it is possible to tell visually).

For clarity and to reduce the computational cost, the T-SNE plots only display a random subset of 10 000 points. This sub-sampling is done after the T-SNE transformation and so does not affect the resulting 2D representation. In the following plots, the 35 ‘studied groups’ of experiments are colored, the others will be displayed as gray. The points with the same color across all the following plots correspond to the same experiment in the original data. Obviously, to be able to compare the results, the exact same set of data points is used for each visualization.

The following T-SNE visualizations can allow us to analyse and compare the different representations. Indeed, as T-SNE preserves the topology of the high-dimensional representations, we can visualize and make observations on the different representations:

In Figure 29, many colored groups are well clustered together (the yellow and dark red groups) and many small clusters appears. In Figure , some clusters are visible, representing the topics generated by LDA. The yellow, black and blue groups of points seems to be clustered in their respective topic. In Figure 31, the data looks more clustered than in Figure 30, the colored groups appear to be well separated, in different topic-clusters. In Figure 32, no clear cluster is standing out and points from the studied colored groups are not close to each others. In Figure 33, The UMAP transformation appears to create small clusters but the studied groups are not clustered together. This would indicate that the Doc2Vec representation will fail to cluster related experiments.

The projections are random in the ”Random-projections” representation. So, in Figure 34, no structure is visible, as expected. In Figure 35, two main clusters tend to stand out. Small and isolated clusters appeared, and studied groups appear to be clustered together. The dark red group is compact.

In Figure 36, and compared to Figure 35, the two main clusters are visible too, the colored groups (notably the Yellow and blue one) appears to be slightly
more clustered together although the dark red group appears to be spread.
In Figure 37, and compared to Figure 36, the visualization seems similar, the 
clusters are more compact and the studied groups are clustered together.
In Figure 38, the visualization looks very similar to Figure 37, no clear difference 
is observed.
In Figure 39, surprisingly, the two main clusters of Figure 36 and 37 are also 
visible. The studied groups are clustered together but not in a very compact 
manner. An additional small cluster on the right is visible.
In Figure 40, the visualization, when compared to the Figure 39, shows the 
studied groups to be significantly clustered together. The black and blue groups 
of points are compact. It appears that the right cluster in Figure 39 has been 
divided into more separated clusters.
In Figure 41, studied groups (yellow, blue, black, purple and dark red) are 
clustered quite closely.
Finally, in Figure 42, the clusters are separated and compact, we can observe 
three main clusters with several medium clusters, composed of some groups 
(yellow, blue and black).

Figure 29: T-SNE visualization (2d) - TF-IDF with Truncated SVD
Figure 30: T-SNE visualization (2d) - LDA with 10 topics

Figure 31: T-SNE visualization (2d) - LDA with 30 topics
Figure 32: T-SNE visualization (2d) - Doc2Vec

Figure 33: T-SNE visualization (2d) - Doc2Vec-UMAP-15-dimensional
Figure 34: T-SNE visualization (2d) - Random-projections

Figure 35: T-SNE visualization (2d) - Flair
Figure 36: T-SNE visualization (2d) - Flair-Glove

Figure 37: T-SNE visualization (2d) - BERT
Figure 38: T-SNE visualization (2d) - ELN-BioBERT

Figure 39: T-SNE visualization (2d) - LSCO-Flair-Glove
Figure 40: T-SNE visualization (2d) - LSCO-Flair-Glove + LDA (30 topics) + PCA

Figure 41: T-SNE visualization (2d) - LSCO-Flair-Glove + Flair-Glove + PCA
Figure 42: T-SNE visualization (2d) - ELN-BioBERT + LSCO-Flair-Glove + PCA

Figure 43: T-SNE visualization (2d) - ELN-BioBERT + Flair-Glove + PCA
Figure 44: T-SNE visualization (2d) - ELN-BioBERT + Flair-Glove + LDA (30 topics) + PCA

Figure 45: T-SNE visualization (2d) - ELN-BioBERT + Flair-Glove + LSCO-Flair-Glove + PCA
An important note is that T-SNE can produce different representations for the same data: some plots can be rotated or flipped according to an arbitrary axis. Here, what is considered relevant is the topology of the representations. In other words, the global structure: the closeness of different high-density regions, the global clusters and the proximity of the different subclusters (small or medium groupings of points) with other groupings.

It is observed that in almost any representations, produced by very different machine learning techniques, some clusters (for example yellow, orange, light blue and dark red colored group-related points) are present. Even though the global structure, or topology, is not the same across various representations, it shows that from a global perspective, some restricted parts of the data are very “similar” and this “similarity” is picked up by almost all techniques, even the traditional statistical ones. It is also observed that some representations are very similar, topologically wise, but this is mostly observed this in the last 6 techniques made of a combination of the first techniques. This is interesting, as it helps to visually understand what combinations of techniques produce a very different representation than the others. It can show how much a representation impacted a combination e.g. in some plots adding LDA did not change the visualization. Additional interpretations of these visualizations are made in sections 4.3 and 4.2.

4.2 Clusterability analysis of the representations

The goal of this section is then to evaluate, with methods described in section 3.4, the clusterability of the representations produced by the different techniques studied in this thesis.
4.2.1 Evaluation of the number of potential clusters in the data

In the following section, the focus is made only on the Silhouette and the Calinski-Harabasz index. Like the traditional elbow method, a K-means clustering is performed on the data followed by an internal evaluation of the clustering with the Silhouette and Calinski-Harabasz index. The elbow method is done by varying the number of clusters passed as input to K-Means between K=2 and K=100. (Going well above K=100 did not produce any interesting result)

4.2.2 Elbow method with K-Means

First, let’s do an elbow method analysis with the Calinski-Harabasz index. With this index, the higher the score the better, so here the search for an elbow would be an “upward peak”.

![Elbow method plots](image)

Figure 47: Analysing the potential number of clusters in the data using elbow plots. Performed with the Calinski-Harabasz index, for K-means clustering. Part 1/3
Figure 50: Analysing the potential number of clusters in the data using elbow plots. Performed with the Calinski-Harabasz index, for K-means clustering. Part 2/3
Random-projections

ELN-BioBERT + Flair-Glove + LSCO-Flair-Glove + PCA

LSCO-Flair-Glove + LDA (30 topics) + PCA

Flair-Glove + LSCO-Flair-Glove + PCA

ELN-BioBERT + Flair-Glove + LDA (30 ELN-BioBERT + LSCO-Flair-Glove + topics) + PCA

PCA
Figure 53: Analysing the potential number of clusters in the data using elbow plots. Performed with the Calinski-Harabasz index, for K-means clustering.

Part 3/3

The elbows in the plots of LDA with 10 and 30 topics, in Figure 47 appears also at respectively 10 and 30 topics, which is obviously expected, because LDA is a generative statistical model that allows sets of data-points to be explained by unobserved groups it creates clusters for the “topics” it generated.

Apart from the clears elbows (or peaks) found in the elbow plots of LDA based representations, the observation is that no elbow stand out in the rest of the elbow plots. It suggests that there are no convex clusters in the data, or at least not more than one or two large cluster, from the point of view of partitional clustering.

It can be seen that, in Figure 50, the elbow plot of Random-projections behaves strangely. It is obviously an anomaly as this representation is random and thus not representative of the actual structure of the data.

The same elbow method is performed with another metric as the function: the Silhouette method. With this index the higher the score, the better in terms of clustering, so here the search for an elbow would also be an “upward peak”.
As seen in Figure 54, the anomaly of Random-projections disappeared. Only four elbow plots are shown here, to illustrate the Silhouette method, as the conclusions drawn are the same than with the Calinski-Harabasz metric: There is only one clear ‘elbow’, in Figure 47, in the elbow plot of LDA with 10 and 30 topics plots, as expected. The other elbow plots show no ‘elbow’, and thus no sign, from this evaluation, of any significantly clustered data. This is not surprising as the data is noisy and very heterogeneous.

4.2.3 Density-based clustering

This section attempts to cluster the data with a density-based clustering algorithm: HDBSCAN. The HDBSCAN is applied on an UMAP embedding, as described in 3.4.4.2

The UMAP hyper-parameters used before clustering are in the following table.
Table 4: UMAP hyperparameters

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>nb neighbors</td>
<td>20</td>
</tr>
<tr>
<td>nb components</td>
<td>2</td>
</tr>
<tr>
<td>min dist</td>
<td>0</td>
</tr>
<tr>
<td>metric</td>
<td>euclidean</td>
</tr>
<tr>
<td>nb epochs</td>
<td>none (default)</td>
</tr>
<tr>
<td>learning rate</td>
<td>1.0 (default)</td>
</tr>
<tr>
<td>init</td>
<td>spectral (default)</td>
</tr>
<tr>
<td>spread</td>
<td>1.0 (default)</td>
</tr>
<tr>
<td>local connectivity</td>
<td>1.0 (default)</td>
</tr>
<tr>
<td>negative sample rate</td>
<td>5 (default)</td>
</tr>
<tr>
<td>transform queue size</td>
<td>4.0 (default)</td>
</tr>
</tbody>
</table>

Table 5: Different parameters used for HDBSCAN

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Values used</th>
</tr>
</thead>
<tbody>
<tr>
<td>min samples</td>
<td>20</td>
</tr>
<tr>
<td>min cluster size</td>
<td>75, 100, 300, 500</td>
</tr>
</tbody>
</table>

In the following table, the resulting internal metrics on the resulting clustering are detailed, with the number of clusters extracted from each clustering.

Table 6: Internal metrics on the HDBSCAN clusterings after UMAP transformation for min cluster size = 75

<table>
<thead>
<tr>
<th>Representation name</th>
<th>Calinski-Harabasz</th>
<th>Davies-Bouldin</th>
<th>Silhouette</th>
<th>Cluster count</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF-IDF with Truncated SVD</td>
<td>192.932</td>
<td>2.769</td>
<td>-0.031</td>
<td>59</td>
</tr>
<tr>
<td>LDA with 10 topics</td>
<td>455.639</td>
<td>1.263</td>
<td>-0.070</td>
<td>75</td>
</tr>
<tr>
<td>LDA with 10 topics</td>
<td>343.552</td>
<td>1.773</td>
<td>-0.038</td>
<td>76</td>
</tr>
<tr>
<td>Doc2Vec-UMAP-15</td>
<td>347.203</td>
<td>2.082</td>
<td>-0.124</td>
<td>79</td>
</tr>
<tr>
<td>Doc2Vec</td>
<td>68.921</td>
<td>5.800</td>
<td>-0.974</td>
<td>25</td>
</tr>
<tr>
<td>Random-projections</td>
<td>187.875</td>
<td>241.976</td>
<td>-3.040</td>
<td>44</td>
</tr>
<tr>
<td>Flair</td>
<td>82.990</td>
<td>3.239</td>
<td>-0.108</td>
<td>38</td>
</tr>
<tr>
<td>Flair-Glove</td>
<td>143.80</td>
<td>3.11</td>
<td>-0.168</td>
<td>51</td>
</tr>
<tr>
<td>BERT</td>
<td>144.465</td>
<td>3.344</td>
<td>-0.125</td>
<td>28</td>
</tr>
<tr>
<td>ELN-BohBERT</td>
<td>148.759</td>
<td>3.559</td>
<td>-0.119</td>
<td>54</td>
</tr>
<tr>
<td>LSCO-Flair-Glove</td>
<td>107.661</td>
<td>3.312</td>
<td>-0.206</td>
<td>47</td>
</tr>
<tr>
<td>LSCO-Flair-Glove + LDA (30 topics) + PCA</td>
<td>114.623</td>
<td>2.947</td>
<td>-0.112</td>
<td>52</td>
</tr>
<tr>
<td>LSCO-Flair-Glove + Flair-Glove + PCA</td>
<td>115.782</td>
<td>3.384</td>
<td>-0.096</td>
<td>54</td>
</tr>
<tr>
<td>ELN-BohBERT + LSCO-Flair-Glove + PCA</td>
<td>117.146</td>
<td>3.479</td>
<td>-0.088</td>
<td>57</td>
</tr>
<tr>
<td>ELN-BohBERT + Flair-Glove + PCA</td>
<td>121.733</td>
<td>3.494</td>
<td>-0.104</td>
<td>59</td>
</tr>
<tr>
<td>ELN-BohBERT + Flair-Glove + LDA (30 topics) + PCA</td>
<td>119.853</td>
<td>3.284</td>
<td>0.064</td>
<td>58</td>
</tr>
<tr>
<td>ELN-BohBERT + Flair-Glove + LSCO-Flair-Glove + PCA</td>
<td>129.592</td>
<td>3.297</td>
<td>-0.080</td>
<td>55</td>
</tr>
<tr>
<td>Flair-Glove + LSCO-Flair-Glove + LDA (30 topics) + PCA</td>
<td>128.584</td>
<td>3.196</td>
<td>-0.080</td>
<td>49</td>
</tr>
</tbody>
</table>
Table 7: Internal metrics on the HDBSCAN clusterings after UMAP transformation for min cluster size = 100

<table>
<thead>
<tr>
<th>Representation name</th>
<th>Calinski-Harabasz</th>
<th>Davies-Bouldin</th>
<th>Silhouette</th>
<th>Cluster count</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF-IDF with Truncated SVD</td>
<td>541.394</td>
<td>1.641</td>
<td>-0.082</td>
<td>54</td>
</tr>
<tr>
<td>LDA with 10 topics</td>
<td>413.193</td>
<td>1.731</td>
<td>-0.022</td>
<td>48</td>
</tr>
<tr>
<td>Doc2Vec-UMAP-15</td>
<td>452.303</td>
<td>2.828</td>
<td>-0.136</td>
<td>67</td>
</tr>
<tr>
<td>Doc2Vec</td>
<td>97.906</td>
<td>4.875</td>
<td>-0.058</td>
<td>18</td>
</tr>
<tr>
<td>Random-projections</td>
<td>224.053</td>
<td>201.073</td>
<td>-0.273</td>
<td>34</td>
</tr>
<tr>
<td>Flair</td>
<td>304.867</td>
<td>3.445</td>
<td>-0.118</td>
<td>24</td>
</tr>
<tr>
<td>Flair-Glove</td>
<td>189.1746</td>
<td>2.961</td>
<td>-0.163</td>
<td>37</td>
</tr>
<tr>
<td>BERT</td>
<td>166.805</td>
<td>3.427</td>
<td>-0.120</td>
<td>24</td>
</tr>
<tr>
<td>LSCO-Flair-Glove + LDA (30 topics) + PCA</td>
<td>176.494</td>
<td>3.316</td>
<td>-0.083</td>
<td>37</td>
</tr>
<tr>
<td>LSCO-Flair-Glove + Flair-Glove + PCA</td>
<td>174.172</td>
<td>3.139</td>
<td>-0.066</td>
<td>38</td>
</tr>
<tr>
<td>ELN-BioBERT + Flair-Glove + LDA (30 topics) + PCA</td>
<td>308.627</td>
<td>3.094</td>
<td>-0.095</td>
<td>36</td>
</tr>
</tbody>
</table>

Table 8: Internal metrics on the HDBSCAN clusterings after UMAP transformation for min cluster size = 300

<table>
<thead>
<tr>
<th>Representation name</th>
<th>Calinski-Harabasz</th>
<th>Davies-Bouldin</th>
<th>Silhouette</th>
<th>Cluster count</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF-IDF with Truncated SVD</td>
<td>306.526</td>
<td>3.724</td>
<td>-0.005</td>
<td>12</td>
</tr>
<tr>
<td>LDA with 10 topics</td>
<td>1821.97</td>
<td>1.484</td>
<td>0.092</td>
<td>15</td>
</tr>
<tr>
<td>LDA with 30 topics</td>
<td>838.439</td>
<td>1.798</td>
<td>0.109</td>
<td>13</td>
</tr>
<tr>
<td>Doc2Vec-UMAP-15</td>
<td>715.477</td>
<td>4.711</td>
<td>-0.180</td>
<td>19</td>
</tr>
<tr>
<td>Doc2Vec</td>
<td>178.898</td>
<td>4.849</td>
<td>-0.123</td>
<td>40</td>
</tr>
<tr>
<td>Random-projections</td>
<td>36931.613</td>
<td>128.470</td>
<td>-0.152</td>
<td>14</td>
</tr>
<tr>
<td>Flair</td>
<td>188.618</td>
<td>3.847</td>
<td>-0.074</td>
<td>8</td>
</tr>
<tr>
<td>Flair-Glove</td>
<td>607.891</td>
<td>2.571</td>
<td>-0.110</td>
<td>9</td>
</tr>
<tr>
<td>BERT</td>
<td>384.666</td>
<td>2.382</td>
<td>-0.160</td>
<td>7</td>
</tr>
<tr>
<td>ELN-BioBERT</td>
<td>343.241</td>
<td>3.181</td>
<td>-0.119</td>
<td>15</td>
</tr>
<tr>
<td>LSCO-Flair-Glove + LDA (30 topics) + PCA</td>
<td>696.607</td>
<td>2.583</td>
<td>0.043</td>
<td>7</td>
</tr>
<tr>
<td>LSCO-Flair-Glove + Flair-Glove + PCA</td>
<td>562.665</td>
<td>2.828</td>
<td>0.030</td>
<td>18</td>
</tr>
<tr>
<td>LSCO-Flair-Glove + Flair-Glove + PCA</td>
<td>411.396</td>
<td>2.722</td>
<td>-0.043</td>
<td>11</td>
</tr>
<tr>
<td>ELN-BioBERT + Flair-Glove + LDA (30 topics) + PCA</td>
<td>415.836</td>
<td>2.976</td>
<td>-0.039</td>
<td>12</td>
</tr>
<tr>
<td>ELN-BioBERT + Flair-Glove + LSCO-Flair-Glove + PCA</td>
<td>376.081</td>
<td>2.833</td>
<td>-0.058</td>
<td>13</td>
</tr>
<tr>
<td>ELN-BioBERT + Flair-Glove + LDA (30 topics) + PCA</td>
<td>335.922</td>
<td>2.994</td>
<td>-0.020</td>
<td>14</td>
</tr>
</tbody>
</table>

Table 9: Internal metrics on the HDBSCAN clusterings after UMAP transformation for min cluster size = 500

<table>
<thead>
<tr>
<th>Representation name</th>
<th>Calinski-Harabasz</th>
<th>Davies-Bouldin</th>
<th>Silhouette</th>
<th>Cluster count</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF-IDF with Truncated SVD</td>
<td>391.20</td>
<td>4.664</td>
<td>0.024</td>
<td>7</td>
</tr>
<tr>
<td>LDA with 10 topics</td>
<td>2870.474</td>
<td>1.289</td>
<td>0.018</td>
<td>8</td>
</tr>
<tr>
<td>LDA with 30 topics</td>
<td>804.039</td>
<td>1.859</td>
<td>0.092</td>
<td>12</td>
</tr>
<tr>
<td>Doc2Vec-UMAP-15</td>
<td>1112.633</td>
<td>2.916</td>
<td>-0.130</td>
<td>9</td>
</tr>
<tr>
<td>Doc2Vec</td>
<td>237.372</td>
<td>4.014</td>
<td>-0.041</td>
<td>6</td>
</tr>
<tr>
<td>Random-projections</td>
<td>2613.330</td>
<td>127.342</td>
<td>-0.032</td>
<td>4</td>
</tr>
<tr>
<td>Flair</td>
<td>270.062</td>
<td>4.986</td>
<td>-0.012</td>
<td>7</td>
</tr>
<tr>
<td>Flair-Glove</td>
<td>990.88</td>
<td>2.590</td>
<td>-0.024</td>
<td>6</td>
</tr>
<tr>
<td>BERT</td>
<td>737.022</td>
<td>2.173</td>
<td>0.008</td>
<td>3</td>
</tr>
<tr>
<td>ELN-BioBERT</td>
<td>762.459</td>
<td>2.987</td>
<td>-0.059</td>
<td>7</td>
</tr>
<tr>
<td>LSCO-Flair-Glove + LDA (30 topics) + PCA</td>
<td>1212.644</td>
<td>2.705</td>
<td>0.067</td>
<td>5</td>
</tr>
<tr>
<td>ELN-BioBERT + Flair-Glove + LDA (30 topics) + PCA</td>
<td>910.353</td>
<td>2.569</td>
<td>0.028</td>
<td>6</td>
</tr>
<tr>
<td>ELN-BioBERT + Flair-Glove + LDA (30 topics) + PCA</td>
<td>720.628</td>
<td>2.031</td>
<td>0.027</td>
<td>7</td>
</tr>
<tr>
<td>ELN-BioBERT + Flair-Glove + Flair-Glove + PCA</td>
<td>692.60</td>
<td>2.819</td>
<td>0.025</td>
<td>9</td>
</tr>
<tr>
<td>ELN-BioBERT + Flair-Glove + LDA (30 topics) + PCA</td>
<td>731.702</td>
<td>2.589</td>
<td>-0.028</td>
<td>8</td>
</tr>
<tr>
<td>ELN-BioBERT + Flair-Glove + LSCO-Flair-Glove + PCA</td>
<td>586.060</td>
<td>2.866</td>
<td>-0.021</td>
<td>11</td>
</tr>
<tr>
<td>ELN-BioBERT + Flair-Glove + LSCO-Flair-Glove + PCA</td>
<td>746.631</td>
<td>2.932</td>
<td>0.007</td>
<td>7</td>
</tr>
<tr>
<td>ELN-BioBERT + Flair-Glove + LSCO-Flair-Glove + LDA (30 topics) + PCA</td>
<td>439.965</td>
<td>2.842</td>
<td>-0.037</td>
<td>8</td>
</tr>
</tbody>
</table>

To illustrate visually the HDBSCAN clustering, only three sets of 2D T-SNE plots of HDBSCAN clusterings are shown, while varying the min cluster size.
Each cluster is colored, except for the blue representing the points classified as noise by the HDBSCAN algorithm. On a side note, in each set of plots, the T-SNE visualization can produce flipped visualizations (in any arbitrary axis).

Figure 55: Visualizing the clusterings performed by HDBSCAN on the 'ELN-BioBERT' representation, with different hyper-parameters
(a) ELN-BioBERT + Flair-Glove + LSCO-Flair-Glove + PCA - HDBSCAN
min cluster size = 75

(b) ELN-BioBERT + Flair-Glove + LSCO-Flair-Glove + PCA - HDBSCAN
min cluster size = 100

(c) ELN-BioBERT + Flair-Glove + LSCO-Flair-Glove + PCA - HDBSCAN
min cluster size = 300

(d) ELN-BioBERT + Flair-Glove + LSCO-Flair-Glove + PCA - HDBSCAN
min cluster size = 500

Figure 56: Visualizing the clusterings performed by HDBSCAN on the 'ELN-BioBERT + Flair-Glove + LSCO-Flair-Glove + PCA' representation, with different hyper-parameters
Figure 57: Visualizing the clusterings performed by HDBSCAN on the TF-IDF representation, with different hyper-parameters

It can be observed, in Figures 55, 56 and 57, that with a large min cluster size like 300 and 500, the clustering algorithm picks up more the real general shape of the clusters present in the data. With min cluster size = 75, the algorithm can pick more finely grained cluster and in greater number, while with this parameter set to 500 or higher, it produces only the major clusters.

It is also observed that many points are classified as noise. This is how the HDBSCAN works, while it is also influenced by UMAP hyper-parameters. One solution is to assign noisy points to the nearest cluster (soft clustering).

In this experiment, the goal is not to tell which clustering is more or less “useful” as it can depend what one may look for. This section does the density-based clustering agnostically and its conclusion is that it is possible to cluster the data hierarchically with a density-based clustering algorithm, while varying the min cluster size parameter to extract clusters at a given level.

The data is, as observed in the visualizations and depending of the representation considered, composed of merely 2 or 3 major clusters, not well separated. In some representations, which create artificial clusters like LDA, TF-IDF and Doc2Vec-UMAP-15, any clustering algorithm can find more clusters than with the other representations.
The data is also composed of small and compact clusters, well separated. This is also not surprising given the group-characteristic of the data (some groups are made of a very different and specific vocabulary that the rest of the data). These results show that, even for a variety of different representations produced by very different methods, it is very hard to separate the data into clear-cut clusters, even though the density-based clustering techniques are very suitable to get small clusters made of a single or a few groups working on the same specific subject and to get very large clusters made of a lot of groups working in a general branch of scientific research.

4.2.4 Internal metrics with studied groups as labels

In this section, internal metrics are produced, but instead of using labels produced by a clustering like in section 4.2.3, the labels will be the group IDs associated with each experiment. This experiment serves to quantitatively compare how the representations “cluster” or “group” documents originally from the same Group.

The higher the better for Silhouette and Calinski-Harabasz metrics. And the lower the better for the Davies Bouldin index. The analysis is made on the fixed set of 35 studied groups of experiments, composed randomly from groups of different sizes in the corpus. In this context, the internal metrics Calinski-Harabasz and Silhouette will be higher, and the Davies Bouldin lower, when representations cluster more document from the same groups together.

Table 10: List of representations with the associated group-related Calinski-Harabasz, Davies Bouldin and Silhouette metrics

<table>
<thead>
<tr>
<th>Representation name</th>
<th>Calinski-Harabasz</th>
<th>Davies Bouldin</th>
<th>Silhouette</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF-IDF with Truncated SVD</td>
<td>44.607</td>
<td>4.187</td>
<td>-0.089</td>
</tr>
<tr>
<td>LDA with 10 topics</td>
<td>95.239</td>
<td>5.041</td>
<td>-0.348</td>
</tr>
<tr>
<td>LDA with 30 topics</td>
<td>91.352</td>
<td>3.592</td>
<td>-0.200</td>
</tr>
<tr>
<td>Doc2Vec-UMAP-15</td>
<td>34.692</td>
<td>13.498</td>
<td>-0.375</td>
</tr>
<tr>
<td>Doc2Vec</td>
<td>10.311</td>
<td>7.448</td>
<td>-0.096</td>
</tr>
<tr>
<td>Random-projections</td>
<td>480.72</td>
<td>555.953</td>
<td>-0.561</td>
</tr>
<tr>
<td>Flair</td>
<td>28.846</td>
<td>4.470</td>
<td>-0.142</td>
</tr>
<tr>
<td>Flair-Glove</td>
<td>34.813</td>
<td>4.973</td>
<td>-0.178</td>
</tr>
<tr>
<td>BERT</td>
<td>44.280</td>
<td>4.825</td>
<td>-0.155</td>
</tr>
<tr>
<td>ELN-BoBiBERT</td>
<td>59.420</td>
<td>4.485</td>
<td>-0.156</td>
</tr>
<tr>
<td>LSCO-Flair-Glove</td>
<td>25.555</td>
<td>5.739</td>
<td>-0.152</td>
</tr>
<tr>
<td>LSCO-Flair-Glove + LDA (30 topics) + PCA</td>
<td>29.542</td>
<td>5.218</td>
<td>-0.120</td>
</tr>
<tr>
<td>LSCO-Flair-Glove + Flair-Glove + PCA</td>
<td>30.983</td>
<td>4.926</td>
<td>-0.109</td>
</tr>
<tr>
<td>ELN-BoBiBERT + LSCO-Flair-Glove + PCA</td>
<td>49.281</td>
<td>4.513</td>
<td>-0.120</td>
</tr>
<tr>
<td>ELN-BoBiBERT + Flair-Glove + PCA</td>
<td>58.167</td>
<td>4.304</td>
<td>-0.144</td>
</tr>
<tr>
<td>ELN-BoBiBERT + Flair-Glove + LDA (30 topics) + PCA</td>
<td>50.330</td>
<td>4.433</td>
<td>-0.119</td>
</tr>
<tr>
<td>ELN-BoBiBERT + Flair-Glove + LSCO-Flair-Glove + PCA</td>
<td>48.820</td>
<td>4.421</td>
<td>-0.117</td>
</tr>
<tr>
<td>Flair-Glove + LSCO-Flair-Glove + LDA (30 topics) + PCA</td>
<td>33.253</td>
<td>4.703</td>
<td>-0.104</td>
</tr>
</tbody>
</table>
4.3 Evaluation of the nearest-neighbours clustering of the representation

In this section, and as described in section 3.5, experiments are performed to evaluate, and compare, the nearest-neighbours clustering of the representation, to assess which representation is the most relevant for scientific research.

First an experiment is made to compare the similarity of the nearest-neighbourhoods of the representations, to explore how different are, in the vector space, the representations studied in this work.

Then a human assessment will be performed to evaluate the quality of the representations and establish a ranking according to two criteria (scientific content and writing format), by asking two experts in Chemistry and Life sciences to read and rate some of the experiments that are clustered together.

Finally, correlations between internal metrics and human ratings will be studied, to explore the usability of internal metrics for evaluating in an unsupervised (and automated) way the quality of the nearest-neighbours clustering of a representation.

4.3.1 Similarity comparison of the nearest-neighbourhoods

In this section, an experiment is performed to compare quantitatively the local neighborhoods of the representation, as described in section 3.5. This is performed for every point in a representation for every representation.

The percentage of similarity of nearest neighbors is then detailed, between the 18 representations considered in this thesis, in a pair-wise manner. This percentage is calculated by counting, for every point present in a studied group, the ones that are in both N-nearest neighbors for two considered representations.

To present the following N-nearest neighbors similarity comparison array, shortening of the names of the techniques is needed to improve clarity. A letter is associated with each technique considered, such as follows.
Table 11: List of representations with its associated Letter

<table>
<thead>
<tr>
<th>Representation name</th>
<th>Associated letter</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF-IDF with Truncated SVD</td>
<td>A</td>
</tr>
<tr>
<td>LDA with 10 topics</td>
<td>B</td>
</tr>
<tr>
<td>LDA with 30 topics</td>
<td>C</td>
</tr>
<tr>
<td>Doc2Vec-UMAP-15</td>
<td>D</td>
</tr>
<tr>
<td>Doc2Vec</td>
<td>E</td>
</tr>
<tr>
<td>Random-projections</td>
<td>F</td>
</tr>
<tr>
<td>Flair</td>
<td>G</td>
</tr>
<tr>
<td>Flair-Glove</td>
<td>H</td>
</tr>
<tr>
<td>BERT</td>
<td>I</td>
</tr>
<tr>
<td>ELN-BioBERT</td>
<td>J</td>
</tr>
<tr>
<td>LSCO-Flair-Glove</td>
<td>K</td>
</tr>
<tr>
<td>LSCO-Flair-Glove + LDA (30 topics) + PCA</td>
<td>L</td>
</tr>
<tr>
<td>LSCO-Flair-Glove + Flair-Glove + PCA</td>
<td>M</td>
</tr>
<tr>
<td>ELN-BioBERT + LSCO-Flair-Glove + PCA</td>
<td>N</td>
</tr>
<tr>
<td>ELN-BioBERT + Flair-Glove + PCA</td>
<td>O</td>
</tr>
<tr>
<td>ELN-BioBERT + Flair-Glove + LDA (30 topics) + PCA</td>
<td>P</td>
</tr>
<tr>
<td>ELN-BioBERT + Flair-Glove + LSCO-Flair-Glove + PCA</td>
<td>Q</td>
</tr>
<tr>
<td>Flair-Glove + LSCO-Flair-Glove + LDA (30 topics) + PCA</td>
<td>R</td>
</tr>
</tbody>
</table>

Figure 58: Analysis of the N-nearest-neighboring similarity with N=4

This table provides a quick overview of similarity in representations, in neighborhoods around the data points.
It is seen that the representations made of combinations of others have a similar local neighborhood. Especially N and P (ELN-BioBERT + LSCO-Flair-Glove + PCA and ELN-BioBERT + Flair-Glove + LDA (30 topics) + PCA). This is likely due to the fact that the addition of the small LDA vector doesn’t make a significant difference.

LDA with 10 and 30 topics, and especially Doc2Vec-UMAP-15 are very dissimilar to the various different representations studied. ELN-BioBERT is also surprisingly dissimilar to others, except to BERT, which is expected as ELN-BioBERT is produced after a fine-tuning of BERT, they both share the same training architecture. As expected, it is seen that the Random-projections are completely dissimilar to others. This observation tells us that, while the various techniques studied in this work are very different, their respective representations are comparable.

Doc2Vec is also quite dissimilar to the rest of the representations. Surprisingly, the purely statistical TF-IDF representation is quite similar to the deep learning Flair or Flair-Glove techniques, but not to BERT and ELN-BioBERT.

4.3.2 Human assessment of the nearest-neighbours clusterings

This section will detail the results from the human assessment of the clusterings of the different representations studied in this evaluation. The choice here was to focus on a lower number of representations, allowing the annotators to evaluate more experiment per representation. To make that choice, Figure 58 helped to avoid picking representations that are already too similar locally. The choice here is to pick a relatively diverse set of representation, and the final decision was to do the evaluation of 12 representations, listed in the following table.

<table>
<thead>
<tr>
<th>Representation name</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSCO-Flair-Glove + LDA (30 topics) + PCA</td>
</tr>
<tr>
<td>LSCO-Flair-Glove + Flair-Glove + PCA</td>
</tr>
<tr>
<td>ELN-BioBERT + LSCO-Flair-Glove + PCA</td>
</tr>
<tr>
<td>ELN-BioBERT + Flair-Glove + LSCO-Flair-Glove + PCA</td>
</tr>
<tr>
<td>LSCO-Flair-Glove</td>
</tr>
<tr>
<td>TF-IDF with Truncated SVD</td>
</tr>
<tr>
<td>BERT</td>
</tr>
<tr>
<td>ELN-BioBERT</td>
</tr>
<tr>
<td>Flair-Glove</td>
</tr>
<tr>
<td>Random-projections</td>
</tr>
<tr>
<td>Doc2Vec</td>
</tr>
<tr>
<td>LDA with 30 topics</td>
</tr>
</tbody>
</table>

The parameters of this evaluation, for each annotator:
This evaluation, performed by 2 experts, produced 840 ratings, a sample representative of the data, while remaining low enough to be feasible in a reasonable time. The experiments were picked randomly for the evaluation, to better represent the data, as in our unsupervised context, there is no knowledge on the balance of the dataset (some scientific domain could be more represented than others).

The detailed results of the evaluation for the first human annotator:

<table>
<thead>
<tr>
<th>Table 13: Evaluation parameters, per annotator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of representation evaluated</td>
</tr>
<tr>
<td>Number of different target document per representation</td>
</tr>
<tr>
<td>Number of nearest neighbors shown</td>
</tr>
<tr>
<td>Rating scale</td>
</tr>
<tr>
<td>Total number of target experiment</td>
</tr>
<tr>
<td>Total number of neighbors experiment</td>
</tr>
</tbody>
</table>

Table 14: Results of the assessment for annotator 1 and for scientific content

<table>
<thead>
<tr>
<th>Representation name</th>
<th>No. 1</th>
<th>No. 2</th>
<th>No. 3</th>
<th>No. 4</th>
<th>No. 5</th>
<th>Average rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSCO-Flair-Glove + LDA (30 topics) + PCA</td>
<td>0</td>
<td>4</td>
<td>12</td>
<td>17</td>
<td>2</td>
<td>3.49</td>
</tr>
<tr>
<td>LSCO-Flair-Glove + Flair-Glove + PCA</td>
<td>2</td>
<td>1</td>
<td>14</td>
<td>11</td>
<td>7</td>
<td>3.57</td>
</tr>
<tr>
<td>ELN-BioBERT + LSCO-Flair-Glove + PCA</td>
<td>0</td>
<td>6</td>
<td>14</td>
<td>13</td>
<td>2</td>
<td>3.31</td>
</tr>
<tr>
<td>ELN-BioBERT + Flair-Glove + LSCO-Flair-Glove + PCA</td>
<td>2</td>
<td>7</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>3.03</td>
</tr>
<tr>
<td>LSCO-Flair-Glove</td>
<td>4</td>
<td>11</td>
<td>14</td>
<td>4</td>
<td>2</td>
<td>2.60</td>
</tr>
<tr>
<td>TF-IDF with Truncated SVD</td>
<td>2</td>
<td>10</td>
<td>12</td>
<td>8</td>
<td>3</td>
<td>3.00</td>
</tr>
<tr>
<td>BERT</td>
<td>3</td>
<td>9</td>
<td>12</td>
<td>10</td>
<td>4</td>
<td>3.00</td>
</tr>
<tr>
<td>ELN-BioBERT</td>
<td>1</td>
<td>9</td>
<td>12</td>
<td>12</td>
<td>1</td>
<td>3.09</td>
</tr>
<tr>
<td>Flair-Glove</td>
<td>0</td>
<td>9</td>
<td>12</td>
<td>12</td>
<td>2</td>
<td>2.97</td>
</tr>
<tr>
<td>Random-projections</td>
<td>32</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.00</td>
</tr>
<tr>
<td>Doc2Vec</td>
<td>20</td>
<td>8</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>1.63</td>
</tr>
<tr>
<td>LDA with 30 topics</td>
<td>2</td>
<td>13</td>
<td>13</td>
<td>4</td>
<td>3</td>
<td>2.8</td>
</tr>
</tbody>
</table>

Table 15: Results of the assessment for annotator 1 and for writing format and methodology

<table>
<thead>
<tr>
<th>Representation name</th>
<th>No. 1</th>
<th>No. 2</th>
<th>No. 3</th>
<th>No. 4</th>
<th>No. 5</th>
<th>Average rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSCO-Flair-Glove + LDA (30 topics) + PCA</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>14</td>
<td>4</td>
<td>3.46</td>
</tr>
<tr>
<td>LSCO-Flair-Glove + Flair-Glove + PCA</td>
<td>1</td>
<td>5</td>
<td>13</td>
<td>11</td>
<td>5</td>
<td>3.4</td>
</tr>
<tr>
<td>ELN-BioBERT + LSCO-Flair-Glove + PCA</td>
<td>0</td>
<td>8</td>
<td>12</td>
<td>10</td>
<td>5</td>
<td>3.34</td>
</tr>
<tr>
<td>ELN-BioBERT + Flair-Glove + LSCO-Flair-Glove + PCA</td>
<td>2</td>
<td>10</td>
<td>14</td>
<td>7</td>
<td>2</td>
<td>2.94</td>
</tr>
<tr>
<td>LSCO-Flair-Glove</td>
<td>7</td>
<td>12</td>
<td>8</td>
<td>7</td>
<td>1</td>
<td>2.51</td>
</tr>
<tr>
<td>TF-IDF with Truncated SVD</td>
<td>4</td>
<td>12</td>
<td>10</td>
<td>5</td>
<td>4</td>
<td>2.8</td>
</tr>
<tr>
<td>BERT</td>
<td>3</td>
<td>11</td>
<td>9</td>
<td>8</td>
<td>4</td>
<td>2.97</td>
</tr>
<tr>
<td>ELN-BioBERT</td>
<td>5</td>
<td>9</td>
<td>11</td>
<td>6</td>
<td>4</td>
<td>2.86</td>
</tr>
<tr>
<td>Flair-Glove</td>
<td>3</td>
<td>10</td>
<td>12</td>
<td>9</td>
<td>1</td>
<td>2.86</td>
</tr>
<tr>
<td>Random-projections</td>
<td>35</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.9</td>
</tr>
<tr>
<td>Doc2Vec</td>
<td>22</td>
<td>5</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>1.63</td>
</tr>
<tr>
<td>LDA with 30 topics</td>
<td>4</td>
<td>5</td>
<td>11</td>
<td>12</td>
<td>3</td>
<td>3.14</td>
</tr>
</tbody>
</table>

The detailed results of the evaluation for the second human annotator:
Table 16: Results of the assessment for annotator 2 and for scientific content

<table>
<thead>
<tr>
<th>Representation name</th>
<th>No. 1</th>
<th>No. 2</th>
<th>No. 3</th>
<th>No. 4</th>
<th>No. 5</th>
<th>Average rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSCO-Flair-Glove + LDA (30 topics) + PCA</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>15</td>
<td>13</td>
<td>4.11</td>
</tr>
<tr>
<td>LSCO-Flair-Glove + Flair-Glove + PCA</td>
<td>1</td>
<td>3</td>
<td>10</td>
<td>8</td>
<td>13</td>
<td>3.83</td>
</tr>
<tr>
<td>ELN-BioBERT + LSCO-Flair-Glove + PCA</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>13</td>
<td>14</td>
<td>4.17</td>
</tr>
<tr>
<td>ELN-BioBERT + Flair-Glove + LSCO-Flair-Glove + PCA</td>
<td>0</td>
<td>2</td>
<td>10</td>
<td>6</td>
<td>17</td>
<td>4.09</td>
</tr>
<tr>
<td>LSCO-Flair-Glove</td>
<td>2</td>
<td>2</td>
<td>11</td>
<td>12</td>
<td>8</td>
<td>3.63</td>
</tr>
<tr>
<td>TF-IDF with Truncated SVD</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>14</td>
<td>9</td>
<td>3.83</td>
</tr>
<tr>
<td>BERT</td>
<td>3</td>
<td>7</td>
<td>10</td>
<td>11</td>
<td>4</td>
<td>3.17</td>
</tr>
<tr>
<td>ELN-BioBERT</td>
<td>0</td>
<td>1</td>
<td>10</td>
<td>9</td>
<td>15</td>
<td>4.09</td>
</tr>
<tr>
<td>Flair-Glove</td>
<td>1</td>
<td>3</td>
<td>9</td>
<td>13</td>
<td>9</td>
<td>3.74</td>
</tr>
<tr>
<td>Random-projections</td>
<td>34</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.54</td>
</tr>
<tr>
<td>Doc2Vec</td>
<td>17</td>
<td>17</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1.54</td>
</tr>
<tr>
<td>LDA with 30 topics</td>
<td>1</td>
<td>8</td>
<td>13</td>
<td>6</td>
<td>7</td>
<td>3.29</td>
</tr>
</tbody>
</table>

Table 17: Results of the assessment for annotator 2 and for writing format and methodology

<table>
<thead>
<tr>
<th>Representation name</th>
<th>No. 1</th>
<th>No. 2</th>
<th>No. 3</th>
<th>No. 4</th>
<th>No. 5</th>
<th>Average rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSCO-Flair-Glove + LDA (30 topics) + PCA</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>11</td>
<td>16</td>
<td>4.11</td>
</tr>
<tr>
<td>LSCO-Flair-Glove + Flair-Glove + PCA</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>9</td>
<td>18</td>
<td>4.2</td>
</tr>
<tr>
<td>ELN-BioBERT + LSCO-Flair-Glove + PCA</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>10</td>
<td>21</td>
<td>4.49</td>
</tr>
<tr>
<td>ELN-BioBERT + Flair-Glove + LSCO-Flair-Glove + PCA</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>9</td>
<td>19</td>
<td>4.31</td>
</tr>
<tr>
<td>LSCO-Flair-Glove</td>
<td>0</td>
<td>3</td>
<td>5</td>
<td>15</td>
<td>12</td>
<td>4.03</td>
</tr>
<tr>
<td>TF-IDF with Truncated SVD</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>15</td>
<td>12</td>
<td>4.06</td>
</tr>
<tr>
<td>BERT</td>
<td>3</td>
<td>5</td>
<td>11</td>
<td>12</td>
<td>4</td>
<td>3.26</td>
</tr>
<tr>
<td>ELN-BioBERT</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>8</td>
<td>19</td>
<td>4.26</td>
</tr>
<tr>
<td>Flair-Glove</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>8</td>
<td>16</td>
<td>4.03</td>
</tr>
<tr>
<td>Random-projections</td>
<td>35</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.0</td>
</tr>
<tr>
<td>Doc2Vec</td>
<td>11</td>
<td>14</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>1.97</td>
</tr>
<tr>
<td>LDA with 30 topics</td>
<td>0</td>
<td>2</td>
<td>19</td>
<td>4</td>
<td>10</td>
<td>3.63</td>
</tr>
</tbody>
</table>

The final summary of the evaluation results is then produced from the previous tables.

Table 18: Summary of the final results of the human assessment

<table>
<thead>
<tr>
<th>Representation name</th>
<th>Average scientific content rating</th>
<th>Average writing format rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSCO-Flair-Glove + LDA (30 topics) + PCA</td>
<td><strong>3.80</strong></td>
<td><strong>3.62</strong></td>
</tr>
<tr>
<td>LSCO-Flair-Glove + Flair-Glove + PCA</td>
<td><strong>3.70</strong></td>
<td><strong>3.80</strong></td>
</tr>
<tr>
<td>ELN-BioBERT + LSCO-Flair-Glove + PCA</td>
<td><strong>3.74</strong></td>
<td><strong>3.92</strong></td>
</tr>
<tr>
<td>ELN-BioBERT + Flair-Glove + LSCO-Flair-Glove + PCA</td>
<td><strong>3.56</strong></td>
<td><strong>3.61</strong></td>
</tr>
<tr>
<td>LSCO-Flair-Glove</td>
<td><strong>3.16</strong></td>
<td><strong>3.27</strong></td>
</tr>
<tr>
<td>TF-IDF with Truncated SVD</td>
<td><strong>3.42</strong></td>
<td><strong>3.43</strong></td>
</tr>
<tr>
<td>BERT</td>
<td><strong>3.13</strong></td>
<td><strong>3.12</strong></td>
</tr>
<tr>
<td>ELN-BioBERT</td>
<td><strong>3.59</strong></td>
<td><strong>3.56</strong></td>
</tr>
<tr>
<td>Flair-Glove</td>
<td><strong>3.47</strong></td>
<td><strong>3.45</strong></td>
</tr>
<tr>
<td>Random-projections</td>
<td><strong>1.00</strong></td>
<td><strong>1.0</strong></td>
</tr>
<tr>
<td>Doc2Vec</td>
<td><strong>1.59</strong></td>
<td><strong>1.8</strong></td>
</tr>
<tr>
<td>LDA with 30 topics</td>
<td><strong>3.05</strong></td>
<td><strong>3.39</strong></td>
</tr>
</tbody>
</table>

These final results shows the best representations for clustering similar life sciences text experiments, according to the two criterias considered (scientific content and writing format). ELN-BioBERT + LSCO-Flair-Glove + PCA is the best representation for writing format search. LSCO-Flair-Glove + LDA (30 topics) + PCA is the best representation for scientific content search. As commonly done when using results from some annotations or expert assessment, it is possible to produce an inter-annotator correlation score, as described.
in section 3.5.2.2, using the Krippendorff’s alpha with an ordinal metric (the ratings are on a scale of 1 to 5):

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Krippendorff’s alpha on the scientific content ratings</td>
<td>0.49</td>
</tr>
<tr>
<td>Krippendorff’s alpha on the writing format ratings</td>
<td>0.35</td>
</tr>
</tbody>
</table>

This shows that the annotators agreed fairly well during the evaluation. The writing format and methodology criteria appeared to be more subjective to rate, while the scientific content rating was less subjective. These inter-annotator correlation scores give good confidence in the results of the nearest neighborhood evaluation.

4.3.3 Correlation of the internal metrics with human rating

This section will aim at finding correlations between the different internal metrics and the human rating. If an internal metric was to be correlated with human ratings, it would help answer whether any metric is relevant at both global and local scales with the studied data. As the number of human ratings is relatively low, linear regressions can’t produce statistically significant results. Therefore, a possibly valuable analysis is to plot human ratings and internals metrics together to observe trends.

The first analysis will attempt to find any correlation between human ratings and group-related internal metrics (when Group-IDs are used as labels). The following plots will allow an analysis of the correlations between the human ratings and the group-related internal metrics (Table 4).
Both Calinski-Harabasz (left column), Davies-Doubin index (middle column) show no correlations with any human ratings, as shown in Figure 59. However, the Silhouette score shows a significant correlation with both ratings. It would appear that the Silhouette measure (when group IDs are used as labels) is a good indicator of how "good" (i.e. according to a human local evaluation) is a representation of the data. One clear explanation is that experiments from the same group are usually scientifically similar, and thus related.
The next set of analysis will be on HDBSCAN clusters. Correlation with K-means-based metrics will be not be studied in this section.

The following plots will allow an analysis of the correlations between the human ratings and the internal metrics produced from the HDBSCAN clustering with a min cluster size of 75 (Table 7).

Figure 60: Comparing the scientific ratings (left column), writing format ratings (right column) with the 3 internal metrics on HDBSCAN clusters (min cluster size = 75): Calinski-Harabasz (top row), Davies-Doublin index (middle row) and Silhouette score (bottom row)
The correlations, as shown in Figure 60, seems to be weaker, even though a slight correlation with the Silhouette score is observed. The following plots will allow an analysis of the correlations between the human ratings and the internal metrics produced from the HDBSCAN clustering with a min cluster size of 500 (Table 10).

Figure 61: Comparing the scientific ratings (left column), writing format ratings (right column) with the 3 internal metrics on HDBSCAN clusters (min cluster size = 500): Calinski-Harabasz (top row), Davies-Doublin index (middle row) and Silhouette score (bottom row).

Again, as shown in Figure 61, the conclusion is the same as before: the cor-
relations, in the above case, seems to be weaker, even though a slight one is observed with the Silhouette score.

The conclusions of this section, on internal metrics, are:

1. Group-related internal metrics (i.e. when using group-IDs as clusters labels) are significantly correlated to human ratings when using the Silhouette method. Human annotators produced better ratings when experiments from the same groups appeared to be clustered together, even though it was observed that the N-nearest neighbors of an experiment generally don’t belong to the same Group.

2. The Silhouette score appears to be the most relevant internal metrics studied in this thesis, when compared to the associated human rating on the nearest-neighbours evaluation.

3. The Calinski-Harabasz and Davies-Bouldin indexes appear to not be relevant to analyse the quality of a representation.
5 Conclusions and discussion

This section will draw the final conclusions about the different techniques studied in this work. A discussion about quantitative and qualitative results will be made. This section will end with the possible future work and directions.

5.1 Discussion of the results

This section discusses both qualitative and quantitative results obtained in this project.

5.1.1 Qualitative results

The different techniques explored in this work produced high-dimensional representations of the experiments of the data. It was then possible to use the T-SNE non-linear dimensionality reduction technique to visualize them in two dimensions.

Visualizing the representation allowed us to assess them quantitatively:

- Words-based techniques like TF-IDF and LDA produce a very clustered data. Some studied groups appeared to be in the same topic-cluster, as it could be expected (some groups have a specific vocabulary).

- The deep learning techniques like Flair, BERT and ELN-BioBERT produced a significantly clustered representation, as the colored groups appeared to be clustered together, and that some high-density regions appeared in the data, as opposed to the other representations.

- The deep learning techniques like Flair, BERT and ELN-BioBERT produced a global structure made of two, or in some case three, large clusters.

- The LSCO-Flair-Glove technique, based on external knowledge, produced also two or three global clusters, that appeared to be similar, topologically wise, with the deep learning-based representations. The groups appeared to be clustered together but in a less compact manner than with the deep learning techniques.

- The combination of LSCO-Flair-Glove and LDA with 30 topics produced a similar representation than without LDA, but the clusters were observed to be more compact.

- Overall, the representation produced after a combination of techniques were observed to be more clustered, with clusters more separated than the other techniques. It was observed especially with the ELN-BioBERT + LSCO-Flair-Glove + PCA representation.

Secondly, during the human assessment of the local structure (i.e. the nearest neighbors of the experiments) of the different representations, the human experts made some specific observations.

- The sets of experiments shown during the assessment were found to be very relevant in term of scientific content similarity when deep learning techniques (BERT and Flair) were used to produce the representation.
Experiments from different groups were ‘put’ together and were still relevant. This is an interesting result, as the goal of this work is to improve the searchability of the ELN data by clustering similar experiments.

- It appears that it is more difficult in general, for any technique studied, to create a representation where very similar chemical experiments are closer. It would seem that this is due to the “fine-grained” characteristic of chemical compounds and nomenclature.

- In many cases, very short experiments were matched with longer ones. It appears that long experiments can relate to multiple topics. It is to be noted that this is not necessarily negative.

5.1.2 Quantitative results

The results from the human assessment enabled this work to assign a final human rating on the different representations studied in this work. Important observations from this quantitative evaluation were made:

- Doc2Vec failed to produce valuable representations and scored very low (1.59 and 1.8), slightly above the random projections (1.06 and 1.0).

- The difference between BERT (3.13) and ELN-BioBERT (3.59) shows that fine-tuning the language model on biomedical (PubMed+PMC) and life sciences document (the ELN data) beforehand produced a more accurate representation.

- The LSCO-Flair-Glove technique, based on external knowledge, produced very high ratings when combined to LDA (3.8). It is to be noted that LSCO-Flair-Glove alone got a lower rating (3.16). This would be explained by the fact that LSCO-Flair-Glove serves as a de-noising process, by focusing on the important life sciences concepts existing in an experiment, while LDA capture more general latent topics. They would appear to be complementary.

- The technique ELN-BioBERT + LSCO-Flair-Glove + PCA produced high ratings (3.74 for scientific content and 3.92 for writing format) as opposed to ELN-BioBERT when used alone (3.59), and LSCO-Flair-Glove when used alone (3.13). It shows that combining deep learning-based representations with external knowledge helps to improve the clustering of similar experiments.

- When used alone, ELN-BioBERT score slightly higher on both criteria (3.59 and 3.56) than Flair-Glove (3.47 and 3.45).

The section studying the correlations between human ratings and the internal metrics also gave insight into the value of internal metrics in the unsupervised context of this work:

Group-related internal metrics (i.e. when using group-IDs as clusters labels) are significantly correlated to human ratings.

The Silhouette score is a relevant internal metric regarding the associated human judgment, as opposed to Davies-Bouldin and Calinski-Harabasz indexes.
5.2 Main results

This thesis work, and especially the human assessment of the local neighbourhood around points in different representations, produced interesting results. We will focus in this section on the main results and findings of this work.

1. Using the ontologies as a filter of concepts, and therefore removing the "noisy" part of the textual experiment, caused great improvement when clustering. The ontologies-based representation improved much more the clustering when it was combined with another representation modeling the corpus on a global scale (e.g. topic modeling LDA).

2. Combining totally different vector representations from totally different approaches (topic modeling with LDA, knowledge-based extraction with ontologies, and transfer learning with BERT), produced a powerful hybrid experiment encoding technique. These super-vector-representations were able to model our scientific experiments at a finer, and richer level, and therefore produced the most valuable clusterings.

3. The Silhouette score is the most relevant internal metrics, as it was positively correlated to the associated human rating on the nearest-neighbours evaluation.

This correlation is not observed with the other internal metrics studied, Calinski-Harabasz and Davies-Bouldin indexes. These two metrics are based on the concept of 'analysis-of-variance' in euclidean space [22]. They are based on ratios of sums-of-squares of deviations within and between clusters.

(a) Calinski-Harabasz recognize well convex clusters. It 'prefers' when clusters are same-sized, by the number of points inside.

(b) Davies–Bouldin is similar to the former metric but 'prefers' when clusters are equally-distanced from each other.

(c) On the other hand, the Silhouette score is able to measure the quality of 'clusterization' of each separate 'object' (groupings of points), by measuring how similar a point is to its own cluster ('cohesion') compared to other clusters ('separation').

The Silhouette score is able to capture data structure properties on both a global and local level, as the score is correlated with the 'local' human evaluation of the clusters. The correlation is likely explained by the measure of the 'cohesion' performed by the Silhouette score, as opposed to the other studied metrics. This measure of 'cohesion' appears to be linked to the 'quality' (for scientific relevance) of the nearest-neighbourhood of points in the representations.

4. The fine-tuning of the transfer learning ELN-BioBERT model caused significant improvement over just using the pre-trained BERT model, showing that even though the ELN data is very heterogeneous, fine tuning language models on large biomedical and pharmaceutical domain-specific corpora is a significant step to improve document representations and the clusterings.
5.3 Conclusions

This work aimed at answering the general question of whether it is possible to improve the clustering and the local exploration of experiments in the unstructured and noisy life sciences data, when no supervision is possible (unlabeled and unstructured data).

This work was focused on two aspects: creating representations of the experiments in the data and then analyzing them in different ways. A general survey of the different methods to represent text data was done over a set of 18 different representations of the studied data. The representations were generated from a wide range of unsupervised Natural Language Processing techniques: from traditional and purely statistical tools to very recent deep learning based techniques. Additionally, the possibility of using external knowledge like scientific ontologies was explored.

The different representations were then visualized using modern non-linear dimensionality reduction techniques, T-SNE. Each representation was analyzed on a global scale, as an attempt to find global structure like clusters, and internal metrics were used to compare the results quantitatively.

Finally, a more exploratory analysis was done on the representations on a local scale, to assess the nearest-neighbours clustering. A human evaluation involving life sciences experts was performed to evaluate the nearest neighbors of a set of experiments in the data, in term of scientific relatedness and writing methodology relatedness. This assessment allowed direct conclusions to be drawn on the different representations.

The data has been found to be hard to cluster on a global scale, with traditional algorithm. However a density-based algorithm made it possible by creating reasonable clusters. Additionally, visualizations seem to be useful to evaluate the quality of the representations.

Additionally, some internal metrics appears to be of value to measure beforehand if a representation will be scientifically 'relevant', as the Silhouette score was observed to be positively correlated to human ratings. This is explained by the measure of cohesion in clusters done by the Silhouette score, which appears to be relevant in the context of our experimental and scientific data.

The results show that deep learning and hybrid models produce the best representations for clustering. It has also shown that traditional statistical and generative techniques like TF-IDF and LDA can still produce valuable representations. Finally, this work has shown that hybrid techniques, combining the representations from deep learning techniques with external knowledge bases and topic modeling, are highly effective and can enhance significantly the representations and, therefore, the clustering of the data.
5.3.1 Comparison with related work

Modern deep learning languages models, like Flair-Glove and BERT, have been shown to produce valuable representations regarding the clustering on both scientific content but especially writing methodology. This has proven that very recent techniques are able to capture deep levels of the language, even on noisy and very heterogeneous text data in an unsupervised manner.

This is in accordance with the recent advancements in deep learning-based NLP [17], which is now state-of-the-art in many NLP tasks. This work showed that even in a fully unsupervised context, and when handling relatively short and noisy text data, deep-learning based languages models can improve the quality of the representations, and thus, the clustering of related life sciences experiments. The results of this work showed that hybrid techniques, notably including deep learning and external knowledge bases improved the clusterings. This is also in accordance with recent work [47] using ontologies, and hybrid systems using different representations created at different levels [15].

Finally, this work has shown that new types of unsupervised search systems are possible. In this thesis, and as opposed to traditional Google-like term-based search systems [36], search is done by querying directly on an experiment to get the most similar (and related) experiments in a database.
5.4 Future work

The quality of the global structure (clusters) was not assessed by experts and could be a future step towards understanding the quality of the different representations. Using external knowledge bases like ontologies has been proven very useful and should be explored in more detail, with very specific ontologies (e.g. for chemicals, molecules, genes and diseases.)

One aspect of the data is the variable length of the experiments in the data. As it would be expected, very long experiments can be related to multiple scientific topics and could be related, from a clustering perspective, to different experiments. This could be explored in the future by creating sub-representations of long documents to explain their different aspects.

Fine-tuning the deep learning model was also proven very useful to extract valuable representations. Therefore, it could be worth to experiment fine-tuning BERT-like models on a variety of downstream tasks like Named Entity Recognition with ontology terms.

Fine-tuning on different tasks could help create different models, and so, different representations that are respectively more fitted for clustering given a specific criteria.

It could also be worth exploring pre-processing the corpus with deep-learning-based techniques instead of doing a manual cleaning approach. Also, it would be interesting to explore how to leverage the numbers and numerical tables in the text, as it was mostly cleaned beforehand in this thesis.

Additionally, if more structure was added to the data, for example, the time of publication of the experiment, new types of searches could be possible. It could allow the exploration of the scientific trends in the data and also help understand how past and recent documents interact together in the data representations.

Weight-based systems like TF-IDF have been shown to be also valuable even though very simple, so future work could be done on custom weight-based systems.

Finally, this work has shown that combinations of the different representations (deep learning, ontologies, topic modeling) produced even more valuable representations. Therefore, exploring different types of combinations and new techniques should help to create new and valuable representations of life sciences texts experiments.
Bibliography


[61] Juan Ramos. Using tf-idf to determine word relevance in document queries, 
1999.

[62] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the 
space of topic coherence measures. In WSDM, 2015.

[63] Sandro Saitta, Benny Raphael, and Ian F. C. Smith. A comprehensive 

[64] Gerard Salton and C.S. YANG. On the specification of term values in 

[65] Jonathon Shlens. A tutorial on principal component analysis. In Systems 
Neurobiology Laboratory, Salk Institute for Biological Studies, 2005.

[66] Padmini Srinivasan, Sammitra Bhattacharya, and Viet Ha–Thuc. MeSH: 
a window into full text for document summarization. Bioinformatics, 
27(13):i120–i128, 06 2011.

[67] Michael Steinbach, Levent Ertöz, and Vipin Kumar. The challenges of 
clustering high-dimensional data. In In New Vistas in Statistical Physics: 
Applications in Econophysics, Bioinformatics, and Pattern Recognition. 


[69] Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. 

[70] Igor V Tetko, David J Livingstone, and Alexander I Luik. Neural network 
studies. 1. comparison of overfitting and overtraining. Journal of chemical 

[71] Michael E. Tipping. Bayesian Inference: An Introduction to Principles and 
Practice in Machine Learning, pages 41–62. Springer Berlin Heidelberg, 


as models of neural information processing. Frontiers in Computational 
Neuroscience, 11, 12 2017.

[74] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, 
Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you 

[75] Michel Verleysen and Damien François. The curse of dimensionality in data 
mining and time series prediction. volume 3512, pages 758–770, 06 2005.
