POLITECNICO DI MILANO
Corso di Laurea Magistrale in Computer Science and Engineering
Dipartimento di Elettronica, Informazione e Bioingegneria

Automatic data integration for Genomic Metadata through Sequence-to-Sequence models

GeCo Lab

Relatore: Mark Carman
Correlatori: Anna Bernasconi, Arif Canakoglu, Michele Leone

Tesi di Laurea di:
Giuseppe Cannizzaro, matricola 883290

Anno Accademico 2018-2019

Abstract

While exponential growth in public genomics data can afford great insights into biological processes underlying diseases, a lack of structured metadata often impedes its timely discovery for analysis. In the Gene Expression Omnibus, for example, descriptions of genomic samples lack structure, with different terminology (such as "breast cancer","breast tumor", and "malignant neoplasm of breast") used to express the same concept. To remedy this, two models were learnt to extract salient information from this textual metadata. Rather than treating the problem as classification or named entity recognition, it has been modeled as machine translation, leveraging state-of-the-art sequence-to-sequence (seq2seq) models to directly map unstructured input into a structured text format. The application of such models greatly simplifies training and allows for imputation of output fields that are implied but never explicitly mentioned in the input text. Two types of seq2seq models have been experimented: an LSTM with attention and a transformer (in particular GPT-2), noting that the latter out-performs a multi-label classification approach, also using a transformer architecture (RoBERTa). The GPT-2 model showed a surprising ability to predict attributes with a large set of possible values, often inferring the correct value for unmentioned attributes. The models were evaluated in both homogeneous and heterogeneous training/testing environments,indicating the efficacy of the transformer-based seq2seq approach for real data integration applications.

Sommario

Nonostante la crescita esponenziale di archivi pubblici di dati genomici possa facilitare il processo di scoperta di fattori genomici che determinano malattie, la mancanza di una struttura nei metadati spesso impedisce agisce da freno. Gene Expression Omnibus, per esempio, raccoglie descrizioni di campioni genomici che mancano di struttura, presentando spesso diverse terminologie per indicare lo stesso concetto ("Breast cancer", "Breast tumor", "malignant neoplasm of breast" ecc.). Per far fronte a questo problema, questa tesi presenta la sperimentazione di modelli basati su reti neurali che, attraverso il Machine Learning, estraggono l'informazione rilevante dalla descrizione testuale di campioni. Invece di trattare il problema come classificazione o Named Entity Recognition, questo é stato modellato come Machine Translation, utilizzando lo stato dell'arte dei modelli Sequence-to-Sequence (seq2seq) per mappare direttamente il testo d'ingresso - privo di struttura - ad un formato di testo strutturato. L'uso dei suddetti modelli semplifica enormemente la fase di training e permette l'identificazione di campi d'uscita che erano deducibili, ma mai esplicitati nel testo d'ingresso. Due tipi di modelli di traduzione sono stati sperimentati: una rete neurale basata sulla struttura Encoder-Decoder che sfrutta LSTM e il meccanismo di attenzione; e un modello basato sulle celle Transformer (nello specifico il GPT-2); notando come quest'ultimo sia in grado di superare le performances di un classificatore multi-label, anch'esso basato sui Transformers (RoBERTa) . Il GPT-2 ha mostrato capacitá soprendenti nel predire attributi con una vasta gamma di possibili valori, spesso inferendo il valore corretto da altri attributi non specificati nel testo d'ingresso. I modelli sono stati valutati in ambienti di allenamento/test sia omogenei che eterogenei, denotando l'efficacia del modello seq2seq basato sui transformers in reali applicazioni di integrazione di dati.

Thanks

I thank my parents, who raised me by giving me all the tools necessary to face this journey and have always supported my choices.
I thank my brother, whose experience has illuminated my path.
I thank Alessia, for being a supportive companion and for being able to show me that I can always improve.
I thank all my friends, for being source of relax during hard time.
I thank the MAMA group, which allowed me to undertake this work. A special thank to Michele whose Biology background and patience have helped the development of the thesis.
I thank Marettimo family, for being the Marettimo family.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Next generation sequencing (NGS) technologies are producing data with significantly higher throughput and lower cost. As a result, recent years have seen an exponential growth in publicly available gene expression datasets such as NCBI's GEO [1] or SRA [18]. These repositories hold great value in terms of research possibilities, particularly when integrated with one another. Data integration has become one of the biggest challenges for genomic repositories, mostly due to the heterogeneity of databases structures. The lack of standardization for metadata has brought each consortium to enforce some rules autonomously, often proposing a poor conceptual model which makes it impossible for researchers to perform adequate queries on those repositories. The GeCo project has proposed a standard for genomic metadata, the Genomic Conceptual Model (GCM)[4] to homogeneously describe semantically heterogeneous data and lay the groundwork for providing data interoperability, which, coupled with the GenoMetric Query Language (GMQL) [2], a high-level, declarative query language, used to query thousands of samples of processed data, provides a useful tool for researchers in the biological and bioengineering field. One of the main focuses of the project consists in the integration of data from other sources, such as the above-mentioned GEO, which collects millions of genomic samples and the associated metadata, collected under few, generic fields which make the integration process very hard to execute without human intervention. While the GCM contains very specific attributes for metadata (such as "Age", "Tissue", "Cell Line"), allowing a large number of different types of queries, GEO contains only some generic fields such as "Characteristics" or "Description" filled with a long, plain text description of the genomic sample of reference. The large number of samples collected into GEO database has increased the weight of the problem, making it the curse of the repository

and pushing a lot of effort on finding a solution for this problem.

Different strategies for annotating and curating GEO database metadata have been developed in the last years. They can be essentially divided into three main categories: manual curation, extraction of metadata from the gene expression profiles information and automated natural language processing (NLP) techniques which extract information from the above mentioned generic fields of interest[35]. Manual metadata curation, despite being the most accurate method to infer knowledge, is time-consuming and practically unfeasible, as the volume of biological data grows rapidly. The extraction from gene expressions reduces the accuracy so much that some information can't be mined. Natural Language Processing techniques seem to be the more promising way to solve the problem. The related work section will show that the major contributions approached only small sub-tasks, such as the identification of a single label. In addition, the chapter will highlight the strengths and weaknesses of each methodology, pointing out the need for a system which uses new techniques able to overcome the problems of the state of the art and capable of handling the more tasks possible. This thesis presents a novel approach to the metadata integration task through NLP and it proves that the embraced methodology is superior to previous literature from an accuracy and generalization point of view. The type of models used in this thesis is Sequence-to-Sequence models; in particular, two of them were trained and tested: an LSTM-based encoder/decoder [21] and OpenAI GPT-2 [25].

The work is performed through 2 different experiments, both being a comparison between the proposed models and a multi-label classification approach; the first experiment was performed on Cistrome data [22], a collection of more than 44.000 samples labeled with four attributes; the second experiment was performed on ENCODE data [16], one of the major genomic archive which allowed downloading more than 16.000 samples with the associated sixteen different labels. The two experiments aimed to prove the effectiveness of seq-to-seq models highlighting the strengths of the proposed approach with respect to the most standard classification. The results showed that seq-to-seq models can reach higher performances with respect to the state of the art and the baseline classifier, being able to extract correct information even with a messy input text that could trick a human reader.

1.2 Summary

The thesis is structured as follows:

- *Chapter 2* lays the ground for the methods and the technologies used in following chapters. In this chapter the task and the main research questions are described.
- *Chapter 3* exposes the related works for the given task.
- *Chapter 4* describes the approach adopted to face the task, focusing on data format and proposed models.
- *Chapter 5* presents the details of the datasets used for the experiments
- *Chapter 6* describes the experiments performed and an answer to the research questions will be given.
- *Chapter 7* resumes the work done and draws the conclusions

Chapter 2

Background

In this chapter we present the background knowledge necessary for a full understanding of the experiments and results exposed in this thesis. The first Section 2.1 describes the GEO repository structure; the second Section 2.2 illustrates the target Genomic Conceptual Model. Then, Section 2.3 exposes the task and the main research questions for this work.

2.1 NCBI Gene Expression Omnibus

The NCBI Gene Expression Omnibus repository is an international public archive which collects and distributes genomics data of high-throughput microarray (A.1.1) and Next-Generation Sequence (A.1.3) techniques. Different research institutes and universities from all over the world upload their experimental data on such platform and a huge amount of approaches have been developed to reanalyze its dataset collections.

The core items of the archive are GSM (or GEO Sample), GSE (or GEO Series) and GPL (or GEO Platform).

Each record corresponding to a single genomic sample is called GSM. Each GSM is composed of a region file, containing the genomic information, and a metadata file. Each GSM is associated to a unique identifier which appears in the form of "GSM" + integer_number (e.g. "GSM123151").

Given that it's likely that an experiment would produce more than one genomic sample, the GSE groups GSM that belong to the same experiment. So Series act as super-category for GSM.

Another type of record present in the GEO repository, is the GPL, which collects information about the type of experiment, such as the technique adopted, the instrumentation used, etc.

The three above-mentioned elements represent the core of the database; on top of the GSM, GSE, GPL structure, GEO collects two other types of data: Datasets and Profile, but they are not interest topic for this thesis.

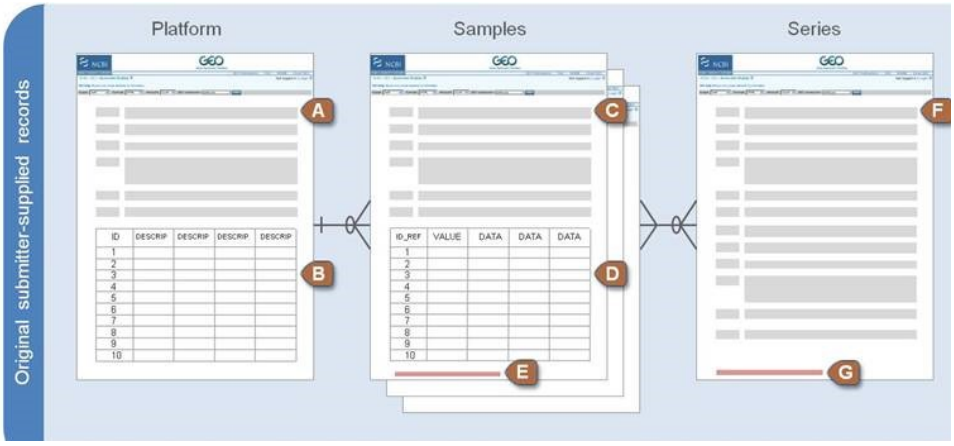


Figure 2.1: GEO records structure

In recent times, new generation gene sequencing platforms have been developed, they allow to execute experiments at a significantly lower cost with respect to the previous years, this has allowed a growing number of institutes and organizations to perform such experiments enlarging the number of available data on public repositories. At the time of writing GEO archive alone can boast more than 3.000.000 samples and this makes the archive one of the most targeted for research on gene expressions.

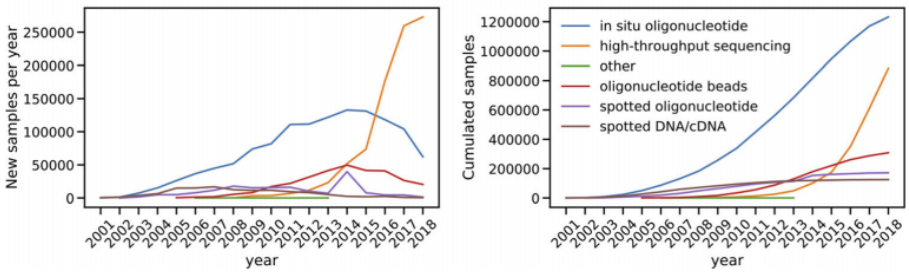


Figure 2.2: Sample upload growth through time [34]

Of the above-mentioned records, particular focus must be given to the GSM in that they represent an actual genomic sample. Each GSM include a ".bed" file which represents the processed genomic data (for example a DNA or RNA), in addition a file containing the related metadata can be visualized. Metadata file contains information about the cell on which the experiments were performed on (such as the organism, the Sex of the organism etc.).

GEO provides a web interface to facilitate the visualization of metadata [1], an example of metadata file is shown in Figure 2.3.

[1] `https://www.ncbi.nlm.nih.gov/geo/`

Sample GSM102371	Query DataSets for GSM102371
Status	Public on May 30, 2006
Title	Anidulans_glucose_1
Sample type	RNA
Source name	Aspergillus nidulans, growth in glucose
Organism	Aspergillus nidulans
Characteristics	Strain: Aspergillus nidulans A187 (pabaA1 ya1) Maximum specific growth rate: 0.22 Temperature: 30 degrees C pH: 6.0
Biomaterial provider	Fungal Genetics Stock Center, Kansas City, Kansas 66160-7420 USA
Growth protocol	Batch cultivation was in a chemically defined medium as described by Agger et al (2002), with the following modifications: NH4Cl was used as the nitrogen source, in a concentration of 12.2 g/L, and three different carbon sources were tested, namely glucose, glycerol, and ethanol (10 g/L). Yeast extract was added to the fermenter in a concentration of 3 mg/L in order to encourage the germination of spores. Furthermore, the nutritional supplement p-aminobenzoic acid (PABA) was added to the medium in a concentration of 1 mg/L, as well as the antifoam agent 204 (Sigma), in a concentration of 0.05 mL/L.
Extracted molecule	total RNA
Extraction protocol	Total RNA was isolated using the Qiagen RNeasy Mini Kit, according to the protocol for isolation of total RNA from animal tissues. For the purpose, approximately 20-30 mg of frozen mycelium were placed in a 2 mL Eppendorf tube, pre-cooled in liquid nitrogen, containing three RNase-treated steel-balls (two balls with a diameter of 2 mm and one ball with a diameter of 5 mm). The tubes were then shaken in a Retsch Mixer Mill, at 3 degrees C, during 6-8 minutes, until the mycelia were ground to powder, and thus ready for extraction of total RNA. The quality and quantity of the total RNA extracted were determined by spectrophotometric analysis and by gel electrophoresis. The total RNA was stored at -80 degree C until further processing.
Label	biotin
Label protocol	Biotin-labeled cRNA was prepared from approximately 10 micro-g of total RNA, according to the protocol described in the Affymetrix GeneChip® Expression Analysis Technical Manual (2004). An additional cleanup step was performed before fragmentation, using the Qiagen RNeasy Mini Kit (protocol for RNA Cleanup), in order to guarantee good-quality cRNA

Figure 2.3: Excerpt of a GSM

Besides the web portal, GEO shares an SQLite B.5 file containing only metadata present in the archive called GEOMetaDB [41]. There are different tables in the DataBase, but, for the aim of this work, the GSM table will be described.

The list of the GSM attributes is shown in Table 2.1:

Key	Description
GSM	GEO accession ID for the sample
Title	Title that describes the sample
Series ID	ID corresponding to the GSE which the sample belongs to
GPL	GEO accession number for the platform
Submission date	Date of submission of the experimental data
Last update date	Date of the last update
Type	Type of the sample e.g., RNA or DNA
Source Name	Biological material and the experimental variable
Organism	Organism from which the biological material was derived
Characteristics	List all available characteristics of the biological source
Molecule	Type of molecule extracted from the biological material
Label	Compound used to label the extract
Treatment Protocol	Treatments applied to the biological material prior to extract preparation
Extraction Protocol	Protocol used to isolate the extract material
Label Protocol	Protocols used to label the extract.
Hybridization Protocol	Protocols used for hybridization, blocking and washing, and any post-processing steps
Description	Any additional information not provided in the other field
Data Processing	Details of how data were generated and calculate.
Contact	Details about the person to whom the data are attributable
Supplementary File	Additional material referred to the related experiment

Table 2.1: List of main attributes for each GSM sample

Of the above fields, just 5 of them contain the only target information (GPL, Type, Organism, Molecule, Label), the remaining ones, which represent the large majority, contains plain text without any constraint of structure or terms, leaving to submitters the freedom to fill them with any type of lyric, as shown in Figure 2.3.

The result of this choice is that each Metadata file associated to GSM is easily human readable, but very hard to be processed by algorithms.

This leads to the **problem** of the GEO repository: *The conceptual model allows to perform queries only on those structured fields, restricting the types of possible queries.*

To understand the issue let's suppose that a certain institute is performing studies on human DNA and, to accomplish that, it needs to collect the largest number of samples of DNA belonging to **male** humans **50 years older**. Those two attributes are not explicit in any of the fields described above, they are often wrapped under "Characteristics" or "Description" together with a lot of other noisy information. Usually submitters try to provide some structure in the "Characteristics" field, however there's no agreement about neither the type of structure, nor other important properties such as the units of measure, making it impossible to perform this kind of queries, which are often the most useful.

2.2 Genomic Conceptual Model

GEO is not an isolated case, genomic repositories lack of a unified standard for metadata, thus each consortium enforces some rules autonomously. The lack of a unified structured conceptual model has pushed the birth of the GCM [2][4] proposed by GeCo laboratory at Politecnico of Milan. As shown in Figure 2.4, the schema is way more complex and detailed with respect to the one proposed by GEO.

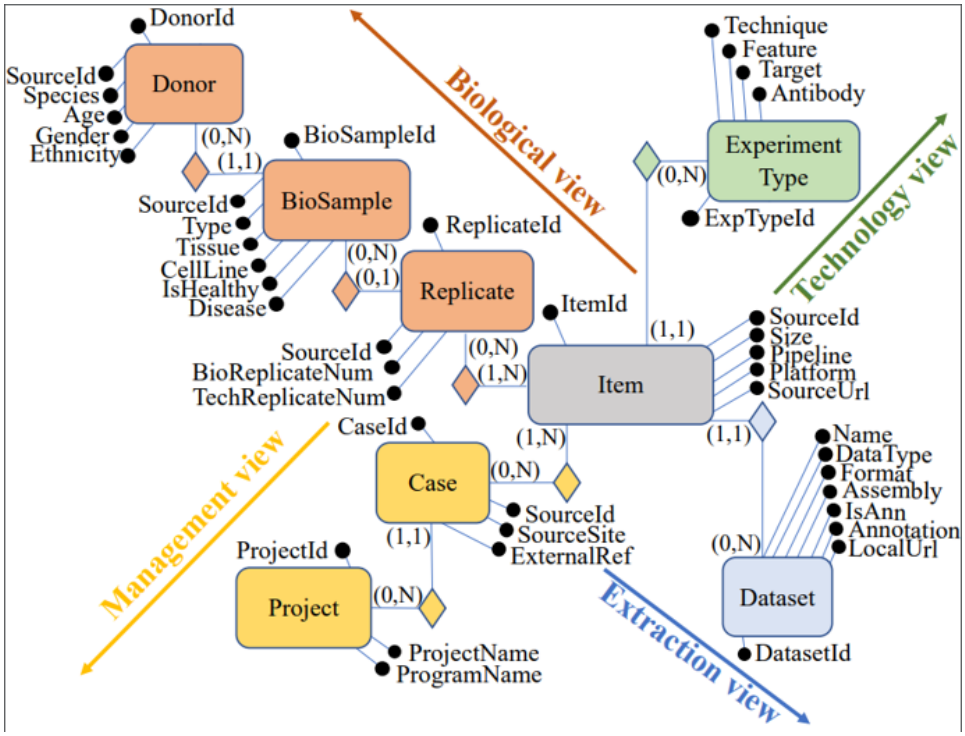


Figure 2.4: GCM schema [3]

It's centered around the resource "Item", typically a file containing ge-

nomic regions. There are four main views through which the Item is described:

- Management: Information about the organizational process for the production of the item
- Extraction: Information about the extraction process
- Technology: Information about the experiment
- Biological: Information about the biosample

Aside from the mere proposition of a structured Conceptual Model, GeCo Lab aims to collect data and metadata from different sources.

The curators then provide GenoSurf [2] [5] a web portal for attribute based and keyword-based searches through well-defined interfaces.

At the time of writing, GenoSurf collected more than 40 million metadata files, but only 7 million of them found a correspondence of attributes that allowed the integration of metadata from the original source, into the integrated database.

The integration process is thwarted by the lack of structure in the sourced metadata files, such as the ones from GEO.

Two of the major sources for data and metadata for GenoSurf are TCGA [37] and ENCODE [28]. The conceptual model of the two main sources allows to algorithmically realize the integration process through Schema Matching (A.2.1) [26]. Another target source of processed genomic data is NCBI's GEO archive, but given the structure of the repository, metadata are the main source of challenges for the laboratory. While some of the attributes of the GCM find a correspondence in the GEO schema (such as GEO:Organism with GCM:Species) and a simple schema matching rule can overcome the problem, others - often of major importance - don't.

A manual curation of the process is practically unfeasible given the large amount of samples to be analyzed, hence the need for an automatic integration process is glaring.

The creation of a standard conceptual model for genomic metadata, is thus held back by the **Problem** of the GEO repository. The following Section will describe the **task** and the challenges that the proposed system will need to face; then, the main **research questions** of the thesis will be exposed.

[2] `http://geco.deib.polimi.it/genosurf/`

2.3 The Task

The previous Section highlighted the reasons for the need of the creation of a system which automatically extracts relevant information from unstructured metadata files in GEO repository. For the purpose of this thesis, the target files will only be GSMs in that they represents the core of the GEO archive and contain information that could be integrated in GenoSurf repository.

The proposed task is hence the following: *Extract all the information available in GSM metadata file in a structured form.*

To accomplish the task, it is necessary to develop a system able to process Natural Language and provide an output structure that allows to integrate the extracted information to the attributes of the GCM in a simple way.

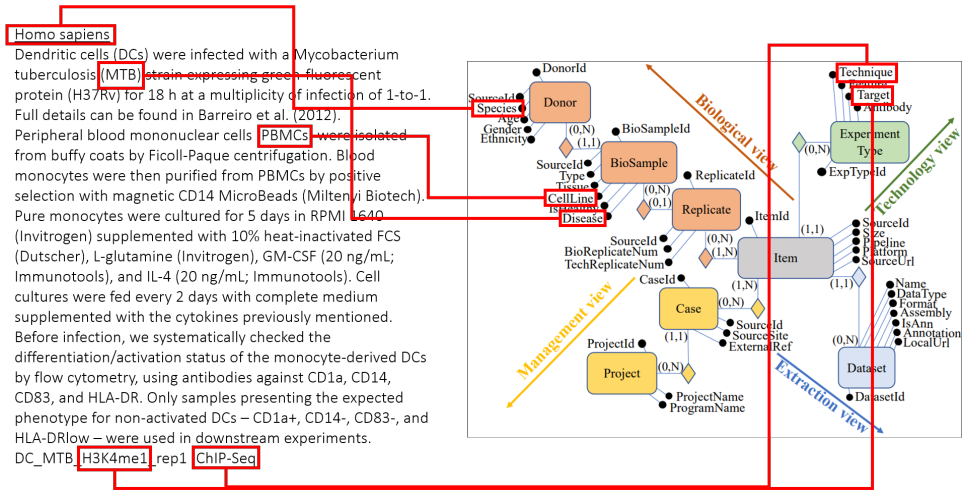


Figure 2.5: Example of an integration process between GEO and GCM

The Figure 2.5 shows an example of the target process. On the left - the plain text description of a GSM[3] is provided, and, on the right, a correspondence for some terms which can be found in the GSM description. As can be noticed, the integration process of this sample requires the extraction of sub-strings from the larger textual description, there are many cases for which this simple matching can't be done and information must be inferred from text.

[3] `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM1565792`

The lack of structure in metadata leaves to the submitters a lot of freedom in the process of description of Samples. This, generates several problems that the proposed system has to face dealing with Natural Language, as shown in Table 2.2:

Adoption of synonyms: In the field of biology, many words (or set of words) can refer to the same entity, think of "Breast Cancer", "Breast adenocarcinoma", "Breast Carcinoma", "Mammary tumor" etc. The system must be able to recognize when synonyms refer to the same entity.
Abbreviations: For example "Stem Cells" which can be subject to numerous different acronyms (hES, HES, ES cells, ESC, ESCs, HESC etc.), or - referring to breast cancer - "Br. Can.", "Br. Cancer", "Breast c." etc. The system must be able to recognize the entities that some abbreviations may refer to.
Words scattered in paragraphs: For example a simple match for "Breast adenocarcinoma" would fail with the following cases "Breast cells subject of adenocarcinoma", or "Adenocarcinoma cells, belonging to a patient of [...] extracted from breast". The system must be able to locate the target information across the entire sentence.
Hidden information: Often, a lot of information is not explicit, but it is inferable from text - at least for a human reader with an high level biology background - for example, the Cell Line "K562" implies that the cell was immortalized and was labelled with that acronym, but that simple sequence of characters implies that the cell belongs to a woman, 53 years old with Cronic Myelogenous Leukemia. The system must be able to deduce information from the description provided by the GSM.
New knowledge: The field of genomic research is seeing a rapid evolution, new discoveries are made and new methodologies are constantly being developed. This also means that a list of target values for each attribute can't always be available. Thus the proposed system should be capable of handling cases of samples with new, unseen values and be able to extract the correct values, generalizing from previous knowledge.

Table 2.2: Problems in free-text metadata description

An analysis of the Related Work and of the State of the art in NLP models, have pushed the proposal of a new approach to the given **task**.

In this work, the information retrieval from unstructured text is faced trough the use of Sequence-to-Sequence models, using a Machine Translation approach.

The job of solving the given **Task**, facing the above-mentioned problems with a Machine Translation approach, brings to light the following **research questions**:

- *Can Sequence-to-Sequence models provide structured information from unstructured plain text?*
- *Can Sequence-to-Sequence models extract correct biological information from plain text overcoming the problems in Table 2.2?*
- *How do Sequence-to-Sequence models perform, in relation with other approaches?*

Given the relevance - in the biology community - of the GEO repository, the task embraced in this thesis is well known. Many have been the attempts to accomplish it, always with partial, bad or simply poor results.

The Related Work Chapter 3 will show what is the state of the art for the task of extraction of metadata from GEO repository, with particular focus to the above-mentioned problems and how the proposed work approaches them.

It will be highlighted the need for a new method and - in the following chapter - will be outlined the structure of the proposed approach, i.e. Sequence-to-Sequence models.

2.4 Summary

This Chapter describes the structure of the NCBI's GEO repository, exposing the structure of the conceptual model proposed by the consortium. The structure of the metadata files associated to GEO samples and the related **Problem** has been showed. Then, the Section 2.2 explains how the GeCo laboratory aims to solve the issue by proposing a structured conceptual model (GCM); the data integration process is held back by the lack of structure in the GEO conceptual model. Being manual curation of the integration practically unfeasible, Section 2.3 describes the need for an automated way to extract relevant information from plain text metadata description in a structured fashion facing the NLP issues described in Table 2.2. Last paragraphs mention the proposed approach for solving the **task**, i.e. Sequence-to-Sequence models, and the main **research questions** are exposed. The following Chapter will describe the Related Works for the problem of extraction of structured information from unstructured metadata files in the GEO archive.

Chapter 3

Related works

To understand in deep the reasons and the need for the work exposed in this thesis, it is necessary to explore techniques and results of previous works that addressed the GEO **task**.

In this chapter we summarise all the works which have faced the problem of extraction of metadata from samples stored in the Gene Expression Omnibus archive. Many different techniques have been adopted, they can be essentially grouped into three categories:

- Manual curation
- Natural Language Processing
- Extraction from gene expression profiles information.

Each section is followed by a summary which recaps what are the pros and cons of the related category. At the end of the chapter, the reader will understand that the approaches adopted so far do not satisfy the needs for a complete fulfilment of the **task**.

3.1 Manual curation

Given the difficulty of the task, many studies have opted for a manual annotation of samples, preferring a higher accuracy, but lower practicality.

3.1.1 STARGEO

STARGEO [13] aims to speed up the process of annotation through crowdsourcing, providing an interface to facilitate the procedure of annotation of samples with disease phenotypes. The annotation is done by selecting a certain GSE, retrieve the textual "Characteristics" field of each GSM and let the annotator build an appropriate Regular Expression (A.2.2) to extract the desired tags of the sample belonging to the desired GSE. This procedure is based on the assumption that the submitter of a certain GSE will adopt the same textual structure for each GSM of the series so a unique RE applied to the description of every sample of a certain series, will match for multiple items, speeding up the process. As a downside, Regular Expressions can't extract most of the useful information hidden in text because suffer of a lot of weaknesses which will be described in details in Section 3.2.2.

The annotators were recruited through social media, with the only requirement of "some graduate level training in the biomedical sciences". The recruitment approach can provide an easy way to build a crowd sourcing team, but is a strategy which presents low reliability.

To overcome the problem, the curators proposed a validation procedure, each sample received multiple annotation by different curators, but in order to make a blind curation and stimulate the individual work, the annotation of other curators were hidden.

STARGEO seems - as the related paper [13] highlights several times - primarily - a study of feasibility for crowd-curation of repositories rather than a method to provide a unified and structured database for biomedical research which is indeed not provided.

3.1.2 SFMetaDB

The first version of the database was published as RNASeqMetaDB [12], a manually curated repository which collected structured labels about "Gene symbol", "Genotype", "Reference" (including title, authors, abstract, Pub Med ID), "Disease", "Tissue Type", "Author" and author's website link extracted from different archives (including GEO), however that version of the database is not available nowadays.

The evolution of RNASeqMetaDB is SFMetaDB [19], a public database containing only 75 manually annotated GSE extracted from GEO. In particular the collected GSE refers only to RNA-sequencing experiments performed on Mouses. The curators provide two labels, an "RNA splicing

factor" and a "Perturbation" information.

Despite the type of labels extracted for this repository can certainly provide useful information for specific biological research, the annotation of a limited case of experiments at Series level and the low number of collected entries make this archive hardly helpful.

3.1.3 CREEDS

CREEDS [36] is a crowd sourcing project (simlar to STARGEO 3.1.1 born with the aim of annotating the various information regarding the disease, drug and gene perturbation expression signatures present on the Gene Expression Omnibus Database (GEO).

All these features were obtained thanks to the work of 70 participants from over 25 different countries. Participants were asked to identify samples that concerned the comparison of *normal* versus *diseased* tissues or gene perturbation experiments. Subsequently, metadata were extracted with the GEO2Enrich [11] tool - a Google Chrome extension which explores data to extract gene signatures - and stored in a local database. Finally, following a further manual inspection process to improve accuracy and quality, the human-extracted signatures were used as a gold standard for training machine learning classifiers for automated signature extraction, but this approach will be described in Section 3.2.4

3.1.4 Considerations

Manual curation - despite providing often high-quality data - does not scale up; Crowd validation techniques can overcome the problem for now, but will suffer of the same issue in the future, as the number of samples grows exponentially. In addition, crowd validation presents the problem of reliability of the annotations; some projects - such as STARGEO - increase the reliability through a double-blinded review process, but increasing the burden on the curation task many folds [35].

The need for automated techniques is straightforward if the number of manually annotated samples is compared to the - continuously growing - number of publicly available genomic samples 2.2. The choice for a manual curation seems, however, to be the most accurate. The annotation resulting from a human check often represents ground truth data, very useful for training and testing automatic techniques. In the following chapter we analyse the contributions which opted for an automatic approach, exploiting Natural Language Processing techniques applied to textual description of GSM or GSE (which are described in a fashion similar to the one described for GSM).

3.2 Natural Language Processing approaches

In this section will be showed the major contributions of the work that exploited both "classical" Text Mining techniques (such as Regular Expressions or Hierarchical Clustering) and "modern" NLP approaches (such as Named Entity Recognition) to extract information from GEO archive.

3.2.1 Predicting structured metadata from unstructured text

In this section we describe the approach to extract metadata from GEO samples adopted by Posch et al. [24]. The theoretical purpose of the work is to classify GSM extracted from GEO using both as input and output the fields present in the related metadata file.

To accomplish the task, two Support Vector Machines (SVM A.2.19) were trained and tested on all the GEO samples available at the time (2015).

The two classifiers predict classes using term frequency - inverse document frequency (TF-IDF - A.2.18) features a Latent Dirichlet Allocation model (LDA - A.2.20), to reduce the dimensionality of the unstructured data. In addition, a majority voting of the prediction of the two SVM classifiers was tested.

The authors split metadata fields into *structured* and *unstructured* categories, as shown in Table 3.1, which were used as target labels and input text, respectively.

Structured	**Unstructured**
GPL	Title
Type	Source Name
Organism	Treatment Protocol
Molecule	Extraction Protocol
Label	Label Protocol
	Hybridization Protocol
	Description
	Data Processing

Table 3.1: Classification of fields for Labels and Input Text

This means that the output of the system is be composed of a collection of labels which were already available in a structured form in the GEO archive. Consequently, the aim of the work is to test the effectiveness of the three models, rather than provide a useful tool for the problem of metadata extraction from GEO.

Performances were evaluated using F-1 score, Precision and Recall. Results showed that only the SVM which uses TF-IDF is able to reach high performances and it is able to do that only on the sub-task of classifying

Molecule, **Organism** and **Type**, while for **Label** and - particularly - **GPL** low performances were reached, as shown in Figure 3.1.

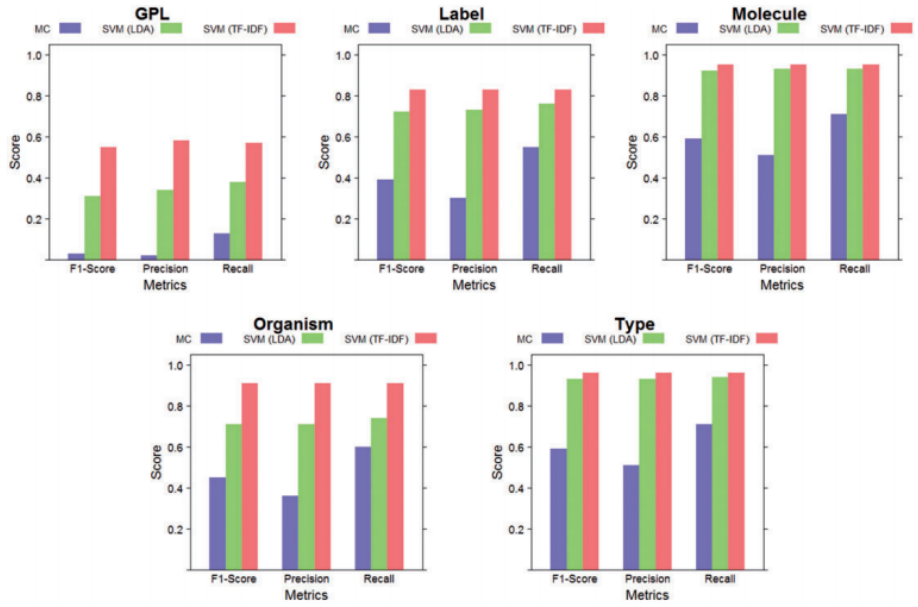


Figure 3.1: Performances of the three classifiers for each class [24]

3.2.2 ALE

Automatic Label Extraction from GEO' [10] presents another approach which makes use of two different techniques to extract information from plain text metadata of GEO samples: Regular Expressions and Machine Learning.

In this chapter will be described the extraction of 'Age', 'Tissue' and 'Sex' with the use of Regular Expressions, while the Machine Learning approach will be described in Section 3.3.

Regular Expressions (RE) are sequences of symbols identifying a group of characters; in other words, they describe symbols patterns to be found in an input text [17].

An example could be the simple *.txt, which is a RE that looks for any sequence of characters terminating with a ".txt" string. This will obviously extract all txt filenames in a given text.

The approach is quite simple: apply some regular expressions to extract the 'Age' and 'Sex' attribute, while for 'Tissue' the extraction is performed through a matching (which can be considered a Regular Expression) between words of the metadata description and words taken from the BRENDA tissue ontology [29]. However, the regular expressions applied are not provided in the paper, but results of the method are presented. Regular expressions were

able to find a match for 'Age' only in 34.8% of the samples and 51.2% was the percentage of samples with a match for 'Sex', while for 'Tissue' 100% of the samples were identified with a match. The metrics for performance evaluations adopted are Precision and Recall and were calculated thanks to a manual annotation of more than 38.000 GEO samples collected in what is called the 'Gold Standard Dataset'. Results are resumed in Figure 3.2:

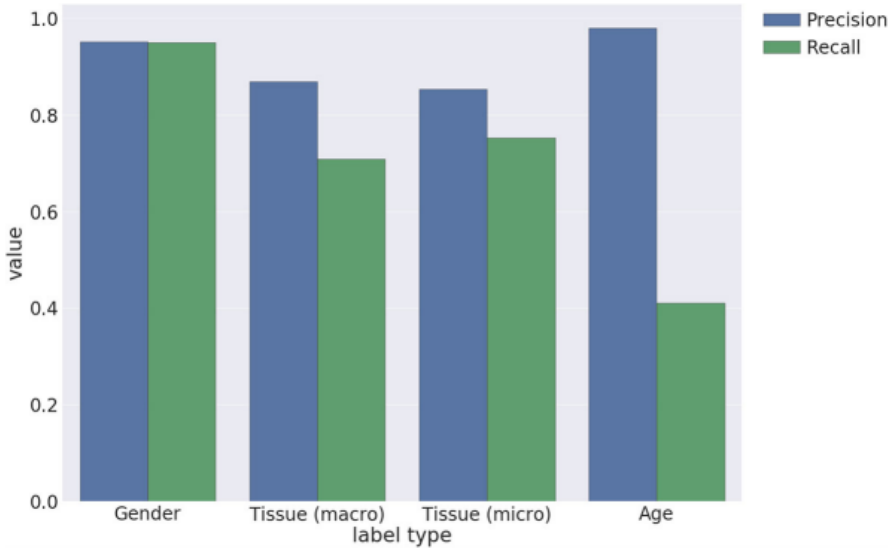


Figure 3.2: Precision and Recall for ALE [10] using RE

The results show that recall is usually lower than precision, this is probably due to the fact that only a small amount of sample could match the provided RE.

Regular Expressions are a really powerful tool for certain tasks, but there are some downsides that make them usable only for specific limited cases, in particular their use is limited to patterns that are:

1. Expressible: There are some values which follow a certain pattern (e.g. sequence of characters starting with 'www' are probably URLs), however the majority of the interesting attributes of biological samples (Figure 2.4) are not expressible through any kind of patterns, thus the extraction through RE is possible only when the submitter of a certain sample adopted fixed rules to describe it (e.g. 'sex: female' which have been proven to work well for this task).

2. Explicit in text: A lot of GEO samples leave the information implicit because - at least for a human reader - they can be inferred from the explicit information. RE cannot overcome this problem.

3. Unique: Whenever REs find multiple matches in text, it's impossible to automatically disambiguate the correct one from the wrong one.

This means that the approach does not successfully interface with the problems described in Table 2.2.

The Automatic Label Extraction from GEO presents interesting results for a limited subset of attributes, but certainly showed that the approach can't cover the majority of the problems related to the GEO archive. However, the 'Gold Standard Dataset' could have been adopted to test other approaches, however, a manual check of the correctness of the annotations have noticed the lack of numerous values which were easily identifiable with a human check making it non suitable for experimentation and revealing an error in the performance evaluation of the ALE.

3.2.3 GEOracle

GEOracle is an R Shiny package which performs Text Mining and Machine Learning techniques classify Series of samples extracted from GEO that contain *perturbation* (A.1.9) data. This type of classification is useful in the biology research in that "they allow to identify the set of genes that are causally downstream of the perturbation agent" [8].

There are two key concept for this work, which are *Perturbation* (A.1.9) and *Control* (A.1.10)

First, a Support Vector Machine is trained on a manually curated dataset to classify "Perturbation" GSE; besides that Performance was maximised by the radial basis function kernel, no additional information is provided. As a result, the paper shows that the model was able to reach an Area Under The following process aims to pair each perturbed GSM with the corresponding control sample. A hierarchical clustering approach is used, based on Gower distance A.2.21 between tokenised GSM titles with a cut at height 0 (in other words it's probable that the approach simply couples titles in order of distance). Then, the same hierarchical clustering is performed on the "Characteristics" field and the method with the highest confidence (not specified what type of metric is utilized) is selected.

This approach leads to a sensitivity measure of **93.2%**.

As an additional step, users which utilize the software can manually check for correctness of the predictions providing a very accurate tool.

The downside of the work stands in a very complex analysis for a very specific case of analysis. The work is limited to the classification of a single type of experiment, thus provides useful data only for particular analysis.

3.2.4 CREEDS

As said in section 3.1.3, the major contribution of the study [36] is the manual crowd sourced annotation of samples, however, the work presents an interesting analysis of different Text Mining approaches to automatically classify text, using the result of the manual curation as dataset.

There are two different classification tasks embraced by the work, one is the binary classification of Series containing *gene*, *drug* or *disease* signatures (A.1.6) - which can contain both *perturbation* and *control* samples - the other is the classification of GSMs as *perturbation* (A.1.9) or *control* (A.1.10) samples inside a Series.

Classification of gene, drug and disease signatures:

Three binary classifiers were built, they used as input three different matrices which represented an embedding of the *Title*, *Summary* and *Keywords* sections of the GSE metadata. The embedding matrices were a result of a Wordnet Lemmatizer and Porter Stemming algorithm, followed by a TF-IDF representation and, subsequently, a Singular Value Decomposition to reduce dimensionality.

Models adopted included random forest, extra trees, support vector classifier and the XGBoost implementation of gradient boosting machines.

To measure the performance of the classification, three-fold cross-validation was applied to calculate the area under the ROC (A.2.14) curve, area under the Precision and Recall curve (A.2.15), Matthew's correlation coefficient (A.2.17) and F1 (A.2.16) score, with results showed in Table 3.2 results:

Class	AUROC	AUPRC	MCC	F1
Gene	0.9	0.9	0.70	0.80
Drug	0.87	0.87	0.60	0.74
Disease	0.81	0.79	0.5	0.64

Table 3.2: CREEDS performances in the classification of Series with gene, drug and diseases signatures

Classification of perturbation and control samples:

Another binary classifier was experimented in the task of classifying *perturbation* and *control* GSM - similarly to GEOracle 3.2.3.

The GEO fields used as input text were *Title*, *Description*, *Characteristics* and *Source Name*. The input text was represented as a vector in the vocabulary space representing the presence or absence of words in the given sentence. The classifier used for solving the problem was a Bagging of 20 multinomial Bernoulli Naive Bayesian classifiers after probability calibration with isotonic regression.

To measure the performance of the classification the same metrics adopted for the first classification task, with results showed in Table 3.3 results:

Metric	Value
AUROC	0.85
AUPRC	0.84
MCC	0.58
F-1	0.71

Table 3.3: CREEDS performances in the classification of perturbation and control Samples

Given that both tasks were trained and tested on human curated data, models for automated techniques should be expected to perform very well, with performances that almost reach the human ability. The machine learning models, do not seem to provide annotation with a quality comparable to the crowd-sourcing approach, this is why the work proposes the adoption - for future works - of active learning to improve the performances of the classifiers.

3.2.5 Onassis

Ontology Annotations and Semantic Similarity Sofware [9]. It is an R package which exploits NLP techniques, biomedical ontologies (from Open Biomedical Ontologies [30]) and R statistical frameworks to identify and relate biological entities in public repositories metadata files.

Onassis works on textual input data retrieved mainly from GEO or SRA. In the case of GEO data is taken through queries to GEOMetaDB, the fields for GSM that worked as input data are: *title*, *summary*, *source_name* *organism*, *characteristics* and *description*.

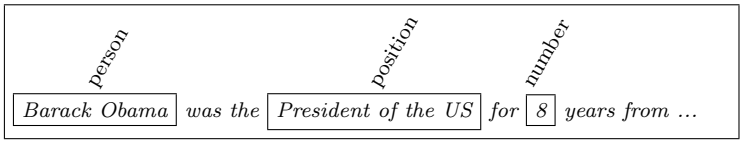
The NLP technique adopted for the extraction of entities is Named Entity Recognition consisting of identifying text spans that mention named entities, and classifying them into predefined categories (e.g, "Cell Line", "Tissue", "Sex", etc.).

After the named entities are extracted, the pipeline follows a semantic similarity measurement between distinct samples, to identify whether they express the same experiment. Specifically, the ontology, represented as a graph, can be traversed to calculate the semantic similarity between pairs (pairwise similarity) or groups (group-wise similarity) of concepts.

After that, Onassis exploits the semantic information extracted in previous steps to performs a semantically-driven statistical analysis directly on genomic data.

Let's bring the attention to the NLP step, namely NER. Typically, the extraction of named entities aims to identify classes such as "Person" or "Position":

Example of NER



In the case of extraction of information from GEO samples, the entities would belong to classes such as "Tissue" or "Organism". The tool used to extract entities is the Conceptmapper, a dictionary lookup tool.

Results show an accuracy of **0.8** in identification of Tissues and **0.9** in the identification of Diseases (no more information about the evaluation is given).

This lookup approach does not approaches well some of the problems cited in Table 2.2. In particular, its ability to handle the problems of *Adoption of synonyms*, *Abbreviations* and *New Information* mostly relies on the quality of the ontologies, but, most importantly, the approach totally lacks of the ability extract sequence of *Words scattered in paragraphs* and *Implicit information*, making any kind of evaluation biased to a dataset where the information is explicit in the input text.

State of the art in NER makes use of large pre-trained neural networks that have shown great performances in NLP tasks. Those new approaches would exploit the information hidden in entire sentences, being able to classify words based on context analysis and, thus, being able to extract entities which are not collected under pre-defined dictionaries. However a NER modeling of the problem is bonded to search for explicit information in text, without possibilities of inference.

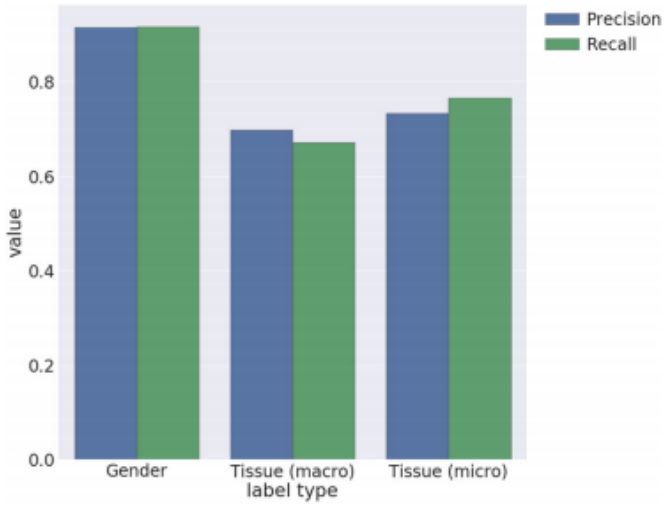


Figure 3.3: ALE performance for the classification based on gene expressions

3.3 Extraction from gene expressions

Besides the approaches that lead with metadata annotations, a totally different method has been adopted by a relevant number of studies, which is the extraction of metadata directly from gene expression data.

However, the approach presents low performances determined by the fact that data itself presents large data records, with high levels of noise.

ALE offers a good comparison between an NLP approach and the extraction from genomic data in that the authors developed both methods and tested them using the same test set. Results of the text-mining approach (which made use of Regular Expressions) are showed in Figure 3.2, while Figure 3.3 shows the results obtained through one-vs-rest (OVR - A.2.3) Logistic Regression classification approach, based on gene expressions.

Despite the naive approach of the text mining method, the performances showed by the classifier do not reach the ones obtained with Regular Expressions. In addition, the label "Age" could not be predicted, this is due to the difficulty to extract that label directly from the gene expressions.

Surely, a lot of attention is paid to this type of studies in the field of biology research. For example, many works ([39], [32]) aim to diagnose diseases from genomic samples; however they represents the arrival point of the research, while the proposed **task** is upstream of the problems faced by this type of approach.

It is straightforward that the models that the research aim to build for diseases diagnosis using gene expressions, can't be trained and evaluated on the labelled data produced by themselves.

3.4 Summary

In this chapter has been described the related work for the given **task**. It has been showed that the major contributions to the problem can be grouped in 3 categories: *Manual curation*, *Extraction from gene expressions* and *Natural Language Processing techniques*. The first appears practically unfeasible for a large scale application, but provides high quality data to build and test automatic techniques. The second seems to be the class of approaches that leads to the most promising results. Different classical NLP approaches were studied in the attempt of extracting information from GEO repository, from classical data mining approaches, to Named Entity Recognition. This chapter has shown that NLP techniques adopted so far don't overcome all of the problems related to a free-text metadata annotation and results showed that the performances were not comparable with human annotation. Moreover, most published studies cover few regulatory factors or genetic marks, often adopting very simple approaches and never leveraging the state of the art.

The third shows lack of applicability, given that the approach represents the goal of the biology research field. The majority of the target information is not deducible from gene expressions and the field of biology research needs more labelled data to understand gene patterns that represents certain types of traits.

This chapter has shown the need for a new approach and that NLP represents the best category of methodology to face the given task.

Chapter 4

Approach

This section is dedicated to describing in detail the new approach proposed in this work.

To the best of knowledge, no previous work has made use of seq-to-seq models for automating the process of integrating experiment metadata. The previous section suggests the need for a novel approach able to overcome all highlighted problems. Our idea is to treat the problem of extracting metadata from unstructured text as a *machine translation* task. Instead of actually translating input sentences into another language, an original output format was chosen, i.e. a well structured list of attributes extracted from input text. The output format assumes considerable importance in that must be both human and machine readable in order to solve the problem. A dash-separated list of "key: value" pair was arbitrarily selected as output format:

Example of data format

```
Input: [Textual description of a sample]
Output: Cell Line: HeLa-s3 - Cell Type: Epithelium
        - Tissue Type: Cervix - Factor: BTAF1
```

The work is done leveraging the state of the art of Sequence-to-Sequence (seq-to-seq) models, comparing an LSTM + attention [15, 21] Neural Network (which represented the SOTA a few years ago) and a Transformers [33] based Language Model (A.2.8), i.e. OpenAI GPT-2 [25] (which has been proved to reach the top scores in most of the famous NLP tasks).

By approaching the task in this fashion, all the problems described in Table 2.2 could be overcome:

- The **adoption of synonyms** does not present an issue in that translation models do not search for specific textual patter in the input

text.

- Neural Networks can exploit context information to understand the meaning of **abbreviations**, which do not represent an obstacle to the task.

- The ability to extract information based on the textual context, allows the proposed models to locate sequence of target words that are **scattered in paragraphs**.

- The analysis of the context, allows Neural Networks to extract information even if it is **hidden** and not explicit in the input text.

- Neural Networks do not perform any kind of lookup in pre-defined ontologies, thus the presence of **new terms** does not present a particular issue, whenever the context provides a sufficient amount of information.

Moreover, the training phase of this kind of model is highly simplified, by treating each target value as a string, many pre-processing steps are avoided and different types of values can be considered all at once.

In addition to the two seq-to-seq models, a **Multi-Label Text Classifier** was trained and tested to offer a comparison for the first two experiments: one of the models which represents the state of the art for text classification tasks, **RoBERTa** [20], which is based on the powerful BERT Language Model [20].

4.1 Sequence to Sequence

Each training sample is composed of *input-output* pairs, where *input* corresponds to the textual description of a biological sample and *output* is a list of attribute-value pairs. Fig. 4.1 shows an example translation task: on the left, a metadata record from GEO repository (coming from a specific endpoint of GEO[1]), describing a human biological sample derived from dendritic cells infected with a particular type of tuberculosis; in the middle the GCM schema, where targeted attributes are squared in red; on the right the resulting output pairs (keys are underlined).

[1] `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM1565792`

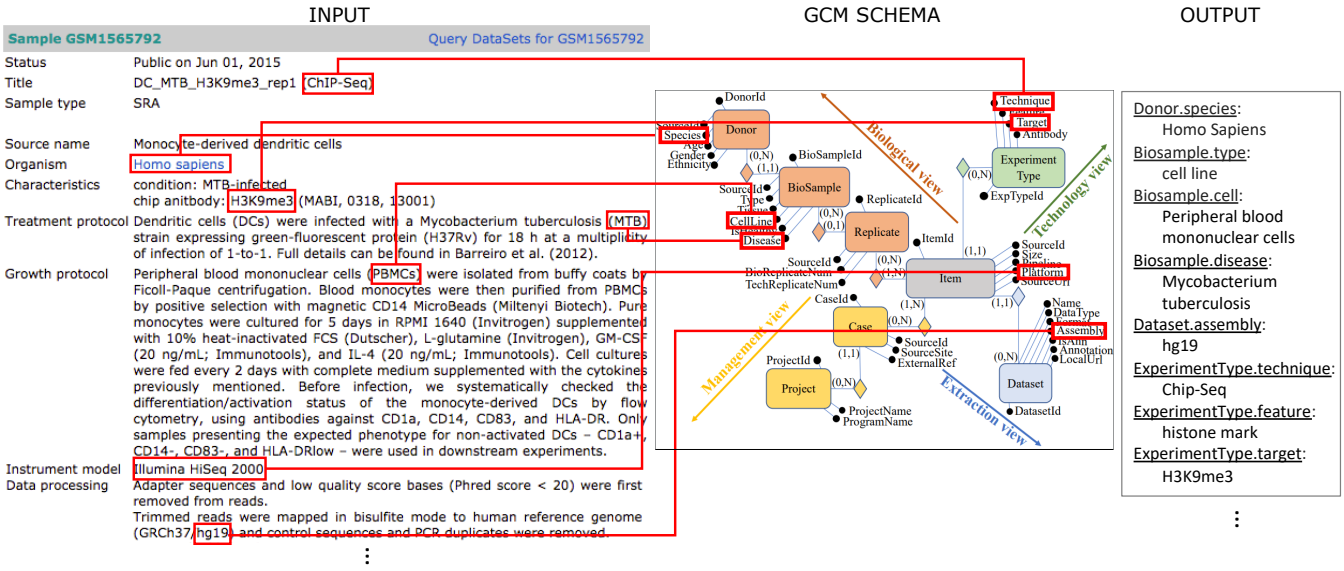


Figure 4.1: Example mapping task from input text into GCM, producing output pairs.

4.1.1 Input format

Input textual descriptions are from the most updated version, to date, of the SQLite database GEOmetaDB [41], extracting only the "Title", "Characteristics", and "Description" fields, which include information about the biological sample from the `gsm` table. The query performed to the database is:

```
SELECT * FROM gsm WHERE gsm in """ + str(tuple(ids))
```

Where the "ids" are the GSM corresponding to the samples for which the labels were available. Input was formatted by alternating a field name with its content and separating each pair with the dash "-" character:

Example of input format

```
Title: ... - Characteristics: ... - Description: ...
```

In this way, the model is allowed to learn possible information patterns, for example, information regarding "Cell Line" is often included in the "Title" section. Input underwent a text cleaning process which was done by executing the following steps:

- !@#$&*[]?\—‘˜_+” with spaces
- “\n” and “\t” removed

4.1.2 Output format

The *output* texts are dashed-separated sequences of “key: value” pairs:

Example of output format

```
Cell line: HeLa-S3 - Cell Type: Epithelium -
    Tissue Type: Cervix - Factor: DNase
```

The aim is to produce, as a result of the translation, well structured sentences, easily interpretable by humans and algorithms; the chosen structure allows to extract the desired attributes using simple, pre-defined, Regular Expressions; results show that both seq-to-seq models were able to learn the output shape after a few epochs of training.

In order to make the text readable by our models, both *input* and *output* text was tokenized with different methodologies, depending on the type of model they were fed into.

4.2 Multi Label Classification

Multi-label classification is a variant of the classification problem where multiple labels may be assigned to each instance.

Each distinct value of a given attribute, was considered to be an output class; thus, the output of a sample is an array of a one-hot-encoding of the target values, 1 for the correct ones, 0 for the others. Despite resolving a number of issues related to RE and NER, this approach has a strong requirement: each attribute must contain a finite number of values and each value must be previously known to be considered as a class. This makes it suitable for some attributes, for example: “Organism” (it only has a restricted set of declared possibilities in GEO database, such as Homo Sapiens, Mus Musculus, Drosophilae etc.) or “Age units” (it can range from “year” to “hours”). However, the approach is totally unfit for other attributes such as: “Age” (it can range from 0 to any positive number, e.g. 140 weeks, 2 years, 400 days) or “Sex”—intuitively it may represents a binary classification example, but GEO samples present a more complex scenario. Each GEO sample can be extracted from human cells or from other species; thus, possible values are not only “Male” and “Female”, but include “Unknown”, “Hermaphrodite”, “Mixed”, “None”, moreover samples can include more than one cell, this brings the “Sex” attribute to have a

infinite range of possible values (e.g. “Male, Female”, “Male, Male”, “Male, Male, Male ...”). This last problem can be attached to any of the target attributes.

For these reasons, Multi-Label Classifier offers an interesting comparison for the proposed seq-to-seq approach, while it is impossible to consider it as an acceptable solution for the given **task**.

4.3 Models

First, we present RoBERTa, which is used for Multi Label Classification as a comparison model. Then, LSTM and OpenAI GPT-2 are described in details. The two models represents the core of the work, being the chosen models for the Sequence-to-Sequence approach. Results are reported in the experiments in Chapter 6.

4.3.1 RoBERTa

RoBERTa [20] is an updated version of BERT [7] a pre-trained Language Model based on Transformer Encoder cells [33]. RoBERTa presents a few modifications both in the architecture and training sides, but the core of the model remains the same. The model is a stacking of Transformer Encoder cells [33], which exploits Self Attention (A.2.9). The model was pre-trained as a Masked Language Model (A.2.11) and have been proved to perform very well in numerous NLP tasks, as shown in Figure 4.2

Model	**data**	**bsz**	**steps**	**SQuAD** (v1.1/2.0)	**MNLI-m**	**SST-2**
RoBERTa						
with BOOKS + WIKI	16GB	8K	100K	93.6/87.3	89.0	95.3
+ additional data (§3.2)	160GB	8K	100K	94.0/87.7	89.3	95.6
+ pretrain longer	160GB	8K	300K	94.4/88.7	90.0	96.1
+ pretrain even longer	160GB	8K	500K	**94.6/89.4**	**90.2**	**96.4**

Figure 4.2: Performance of different size variations of RoBERTa on SQuAD [27] corpus, MNLI [38] corpus and SST-2 [31] corpus

For the classification task, the model presents a Dense layer on top of the Transformer stack. **Tokenization** exploits BytePair Encoding and was done using the default tokenizer provided by Rajapakse’s Simpletransformers repository[2].

The BERT section of the model is the actual pre-trained Language Model, which acts as an embedding of the input sentence, while the last

[2] `https://github.com/ThilinaRajapakse/simpletransformers`

layer gives the model the task-specific structure, being a dense layer with softmax activation function. The last layer presents, indeed, a number of neurons equal to the number of target classes. Given that each output node represents a single value, the toal number of target classes corresponds to the total number of distinct output values (E.G. for Cistrome dataset, the total number of distinct values is 2005)

Training is performed as a Multi Label classification with *CrossEntropy* (A.2.4) as loss function, each sample presented a number of target values equal to the number of classes that composed the dataset (4 for Cistrome, 15 for ENCODE). Each target array was a concatenation of one-hot-encoding of the different attributes.

E.G.

```
Cell Line: HeLa => [0,1,0, ...,0,0,0] (vec_1)
Cell Type: Epithelium => [0,0,0 ... 1,0,0] (vec_2)
Tissue Type: Cervix => [0,0,0,0,1,0,0 ...] (vec_3)
Factor: BTAF1 => [0,0 ... 0,1,0,0,0] (vec_4)

Target vector = vec_1 + vec_2 + vec_3 + vec_4
    '+' is the vector concatenation
```

Evaluation for RoBERTa is done taking into account that the number of target values is fixed for each of the two datasets. The process of concatenation is inverted to get the predictions, so each output vector is split into sub-vectors. Each sub-vector is a representation of the model prediction for each of the Dataset attributes (such as Cell Line or Cell Type). Hence, each sub-vector undergo an *argmax* function to get the predicted value.

E.G.

```
pred_vec => [0.213, 0.051, ..., 0.5]

vec_1 = pred_vec[:len(attr_1)]
vec_2 = pred_vec[len(attr_1):len(attr_2)]
vec_3 = pred_vec[len(attr_2):len(attr_3)]
vec_4 = pred_vec[len(attr_3):]

pred_cell_line = np.argmax(vec_1)
pred_cell_type = np.argmax(vec_2)
pred_tissue_type = np.argmax(vec_3)
pred_factor = np.argmax(vec_4)
```

4.3.2 LSTM with Attention

The model was composed of 2 Feed Forward Neural Networks the *encoder* and the *decoder* and exploits Luong Attention [21] mechanism.

The encoder is composed of an embedding layer plus an LSTM one, which provides hidden states to feed the attention mechanism for the decoding phase. The decoder is composed of an embedding layer, an LSTM one and 2 dense layers. Layer sizing for the two models are shown in Table 4.1 and Table 4.2. The reason behind the 2 dense layers is due to the Luong Attention mechanism. The output of the LSTM layer is concatenated with the Context vector, thus doubling the size of the vectors coming from the LSTM layer, hence, the first Dense layer (which exploits a "tanh" activation funtion) re-shapes the LSTM output to the LSTM size, while the second one (which uses no activation function, thus a linear function is applied) maps the output of the first dense layer to the size of the vocabulary, outputting the logits. Once the decoder has output the logits, an "argmax" of the last layer provides the predicted token for the output at each time step.

Layer	% Size
Embedding	256
LSTM	512

Table 4.1: Number of neurons per layer in the Encoder network

Layer	% Size
Embedding	256
LSTM	512
Tan_h	512
Dense	Vocab_size

Table 4.2: Number of neurons per layer in the Decoder network

The embedding layer of the Encoder is fed with a tokenized version of the input text and is executed once for each sample (batch of items). The decoding phase takes place iteratively, thus the output is generated token-by-token. At each "i" time step (which corresponds to each token), the embedding layer of the Decoder is fed with a tokenized version of the Output text starting from the "start" token ("<start>") reaching the "i"-th token. The Decoder is trained to generate the "i+1"-th token until the entire sequence has been generated, producing a termination token ("<end>"). The Decoder exploits the attention mechanism.

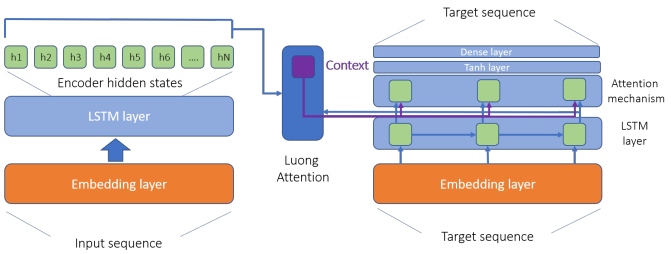


Figure 4.3: Encoder-Decoder structure with Luong Attention mechanism

The tokenization process for **LSTM** model was done using the default keras tokenizer (B.2) with the following parameter setting:

Parameter	Value
lower	True
split	' '
char_level	False

Table 4.3: Parameter settings for Keras tokenizer

For LSTM model only, a few additional text cleaning procedures preceded the tokenization process:

- '(' and ')' with ' (' and ') '
- '-' and '_' with ' - ' and ' _ '
- '=' with ''

This choice was determined by the different tokenization technique which identifies a token with each sequence of characters separated by a space, so, without performing those substitutions, each sequence of dash separated words would be otherwise identified as a single token (e.g. "RH_RRE2_14028_" would have been tokenized as a single word, but adopting substitutions it's possible to find instead "RH _ RRE2 _ 14028 _" becoming 6 different tokens), the same would have occurred for every word preceded by an open (or followed by a closed) bracket. Without those steps, the LSTM model wouldn't have the chance to look for specific tokens which could be the target ones, if they are dash separated or are adjacent to a bracket.

Training

The training phase is performed by learning conditioned probabilities of the "Next" token over the entire vocabulary, given the encoded input sequence, and the sequence of previous tokens, all exploiting a Luong Attention mechanism. So $p(\vec{X}_{n+1}|\vec{x}_n, \vec{x}_{n-1}, ...\vec{x}_{<start>}, \vec{x}_{input})$, where $p(\vec{X}_{n+1})$ is the prediction, $\vec{x}_n$ is the n-th token of the output sequence which starts with the

"<start>" token ($\vec{x}_{<start>}$) and $\vec{x}_{input}$ is the vector of the hidden states of the LSTM layer for the *input* sequence. Training is performed through minimization of the **CrossEntropy** (A.2.4) loss function, typically adopted for multi-class classification tasks, which perfectly fits the problem of prediction of tokens over the vocabulary space.

Evaluation

The evaluation phase is performed as follows, the model encodes the input sequence, the decoder generates the first prediction starting from the "¡start¿" token: $p(\vec{X}_1|\vec{x}_{<start>}, \vec{x}_{input})$, where $\vec{X}_1$ represents the 1st vocabulary vector, $\vec{x}_{input}$ is the encoded input vector and $\vec{x}_{<start>}$ is the first input token of the *decoder*. The n+1-th vector is *arg-maximized* and the result will be the first generated token ($\vec{x}_1$) , then the probabilities $p(\vec{X}_2|\vec{x}_1, \vec{x}_{<start>}, \vec{x}_{input})$ will be computed and the same process will be executed iteratively until the termination token which corresponds to the string "¡end¿" is generated. In the unlikely case of a generation that doesn't end (because the termination character is never generated), the generation is stopped after the output sequence reaches the maximum output length available in the dataset; however this case never happened. After the entire sequence is generated, the tokenized output sequence is turned back into text. After that, a simple search for Regular Expressions in the form "Key: .* [-—<end>]" will return a match for the given "key". If the predicted word (or group of words) corresponds to the target one, it is counted as a correct prediction, wrong otherwise.

4.3.3 OpenAI GPT-2

The second sequence-to-sequence model considered in this work is the more powerful OpenAI GPT-2 [25] model. It is a pre-trained Language Model which structure is based on Transformer Decoders [33].

The presentation paper defines it as a **Unsupervised Multi-Task Learner**, in that it has been proved to perform very well - overcoming the State of the Art in the majority of the cases - for a lot of NLP tasks with a zero-shot learning (A.2.12), as shown in Figure 4.4 (where **LAMBADA** [23] and **WikiText-2** are datasets for prediction of the next word; **CBT** [14] is the "Childrenś Book Text", a dataset to examine the performance of LMs on different categories of words: named entities, nouns, verbs, and prepositions; PPL (A.2.5) is the Perplexity, while ACC is the accuracy).

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**
1542M	**8.63**	**63.24**	**93.30**	**89.05**	**18.34**

Figure 4.4: Performance of different GPT-2 model sizes on some of the major NLP datasets [25]

The model structure is quite simple, it is a stack of multiple *Transformer Decoders* [33] cells, which exploit the **Masked Self-Attention** (A.2.10) mechanism.

Text generation is done in a similar fashion as Encoder-Decoder, a generation token-by-token, but, unlike LSTM model, text generation phase is not preceded by an encoding phase. This means that the model is not trained on input-output pairs, instead, it is trained on single sequences.

Given that, each sequence must include both the Input and the Output, thus, the model was trained on sentences composed of Input and Output pairs separated with the "=" character.

Example of GPT-2 training sentence:

```
[Input sentence] = Cell line: HeLa-S3 - Cell Type: Epithelium
- Tissue Type: Cervix - Factor: DNase $
```

Despite the model has been proven to perform well in NLP problems, the GEO **task** does not present a standard output structure, such as a simple prediction for the 'next token' or a summarization. The output format requires a precise structure, this is why a **finetuning** process is necessary to accomplish the task.

Finetuning

Finetuning is performed by learning conditioned probabilities of the "Next" token over the entire vocabulary, given the sequence of previous tokens.

Let's suppose that a certain "input = output" text is the following sequence of tokens: `13,51,23,555,123,1412,15`. The model will be trained to predict the probability $p(51|13)$, and then, at the next iteration the model will be trained to predict $p(23|51\&13)$ and so on.

Let's suppose that the token "555" corresponds to the "=" character which is used as a separator between input and output. This means that

the model will learn that after the token "555" will be highly probable that the following sequence of tokens will be the target one.

The cleaning process of the input sequence didn't include the removal of "=" characters because they are often used to denote some useful equality (e.g. "Cell Line = HeLa"), but given that in the vast majority of cases it is a separator between *input* and *output*, the conditioned probabilities will be built accordingly, so it will not be (and has been proved that it is not) a problem. Like in the case of LSTM, the loss function is the **CrossEntropy** (A.2.4), with **Adam Optimizer** and a learning rate of **0.001**, which is the default setting for GPT-2 in HuggingFace's [3] repository.

Evaluation

Evaluation is performed similarly to the LSTM model, GPT-2 outputs the probabilities over the entire vocabulary for a given input sequence, $p(\vec{x}_{n+1}|\vec{x}_n, \vec{x}_{n-1}...\vec{x}_1)$, where the list of tokens until 'n' is the input, the n-th token will correspond to the "=" character and the n+1-th vector will be a list of probabilities, one for each token in the vocabulary. The n+1-th vector is *arg-maximized* and the result will be the first generated token, then the probabilities $p(\vec{x}_{n+2}|\vec{x}_{n+1}, \vec{x}_n...\vec{x}_1)$ will be computed and the same process will be executed iteratively until the termination character ("$") is generated. As for LSTM, if the termination character is not generated, the process is stopped after the total length reaches 800 tokens. Similarly to LSTM, the tokenized output sequence is turned back into text. After that, a simple search for Regular Expressions in the form "Key: .* [-—$]" will return a match for the given "key". If the predicted word (or group of words) corresponds to the target one, it is counted as a correct prediction, wrong otherwise.

[3] `https://github.com/huggingface/transformers`

4.4 Summary

This chapter exposes the modeling of the approach chosen. In the first Section the general *Machine Translation* methodology in the GEO **task** is described focusing on how the proposed models would be able to face the list of given problems described in Table 2.2.

The second Section describes in details the textual input and output format chosen for the experiments, highlighting how the proposed output format enables to easily retrieve the values extracted by the models through Regular Expressions.

The third Section describes in details the models proposed, first the classifier used (RoBERTa), then the two Sequence-to-Sequence models, i.e. LSTM + attention and GPT-2, also explaining the training and evaluation mechanism for each of the tested models.

In the next Chapter there will be a detailed description of the Datasets that enabled the two experiments.

Chapter 5

Data

This chapter is dedicated to the description of the dataset used in this work, CISTROME and ENCODE.

They are two rich datasets which were selected - among all the others cited in Chapter 3 - for number of samples, reliability of the annotations and number of labels associated to samples.

5.1 Cistrome

Cistrome [40] is a project which collects a large number of genomic samples and place them in a structured, publicly available database. Collected samples are of three types of genomic experiments: DNA-seq, Chip-Seq and ATAC-seq; the characteristic that they share is the presence of the "Transcription Factors" (TF).

Samples were manually labeled with the corresponding TF examined in the experiment. The experiments were collected from three different source projects: GEO, ENDODE and Epigenomic Roadmap Project [1, 28, 6]. The project focuses on two sides of the collection of data, one being the *data* side and one being the *Metadata* side. The collection of data is followed by quality control pipelines which provide a good marker for research which focuses on Transcription Factors. However, for the purposes of this thesis, the metadata annotation will be the focus of this chapter. As mentioned before, the sources of data for the project are three, but the focus will be directed only on samples extracted from GEO archive. The data was collected exploiting two of the fields in GSM metadata files available in the web application (but not in the DB version), which contain structured information: "Library Strategy" and "Organism". The searched values were, "ChIP-Seq" and "DNase-seq" for *Library Strategy* and "Homo sapiens" and "Mus Musculus" for *Organism*. Since ATAC-Seq data is usually labeled as "OTHER" in *Library Strategy*, the identification of ATAC-seq data was done by matching the keywords in the GEO sample "Description"

field.

Each retrieved GSM was subject to manual annotation in order to provide 6 accurate labels: *Species*, *Cell Line*, *Cell Type*, *Tissue Type* and *Factor*. The resulting metadata collection is structured as a list of GSM identifier and the associated labels. From Cistrome DB we downloaded a total of **44.843** labeled samples. An input text for each of the samples was retrieved by querying the GEOMetaDB selecting all the tuples with a GSM that matched one of the Cistrome entries:

```
"SELECT * FROM gsm where gsm in (...)"
```

Where the first "gsm" represents the table to perform the query on (the others being "gse", "gpl" and many others not interesting for this work), the second "gsm" is the column which contains the ID and "(...)" is the list of GSM extracted from Cistrome. However, only 42.569 samples - of the Cistrome downloaded - had a correspondence in GEOMetaDB, so 2.274 samples were excluded from dataset. The analysis of missing values (annotated as 'None') shows percentages resumed in Table 5.1:

Attribute	**MV Percentage**
Cell Line	53%
Cell Type	19%
Tissue Type	30%
Factor	0%

Table 5.1: Per class Missing Values percentage

A manual inspection of the samples, showed that the "None" values, do not correspond to actual missing values, but instead can be interpreted as a valid attribute value. As an example, let's suppose that a certain GSM presents the label "Cell Line" = "None", the item hence represents a sample which does not belong to any particular Cell Line. The same is valid for *Cell Type* or *Tissue Type*. Another example is a "None" *Tissue Type*, which usually refers to *Stem Cells*, cells that do not belong to any particular tissue. This suggests that the type of missing values is hence Missing At Random (MAR - A.2.13) and, consequently, for this particular problem, they are considered to be a correct target value, interpreted as the string "None".

As can be noticed, the column *Factor* does not present "None" values, in fact, the Cistrome database collects only data for which a certain *Transcription Factor* is present. From this analysis it is possible to claim that the Cistrome database does not contain any missing values. An analysis of the values count for each attribute is shown in Table 5.2:

Attribute	Values count
Cell Line	1045
Cell Type	380
Tissue Type	249
Factor	1565

Table 5.2: Per class values count

The large number of "None" values for the *Cell Line* attribute, suggests that an analysis of Mean and Standard Deviation of the values count per class could be helpful to better understand the distribution of the values 5.3.

Attribute	Mean	Standard Deviation
Cell Line	40.73	697
Cell Type	112	614
Tissue Type	170	918
Factor	27	203

Table 5.3: Mean and Std. for values count per each attribute

The high Standard Deviation suggests that per class there must be some values that occur far more times than the others, an analysis of the mode values shown in Figure 5.4, highlights this trait.

Attribute	Mode value	% of occurrence
Cell Line	None	53%
Cell Type	None	19%
Tissue Type	None	30%
Factor	H3K27ac	11%

Table 5.4: Mode for each Cistrome attribute

As expected, for the first three classes, the mode is represented by "None" values, which - given the total number of distinct values - represent the large majority. While for *Factor* - which contains 1565 distinct values 5.2 - a the mode is represented by "H3K27ac". This suggests that good metrics for performance evaluation on this dataset will certainly be **Accuracy**, **Precision** and **Recall**. Given that the values for each attribute are unbalanced in number, Precision and Recall will be weighted according to total number of occurrences of each attribute value, in this way, the Precision for the "None" *Cell Line* - given that it represents 53% of the occurrences - will count for 0.53 of the total Precision for *Cell Line*.

5.2 ENCODE

The Encyclopedia of DNA Elements [28] is a consortium which - since 2007 - has collected a rich dataset of genomic data publicly available. Similarly to Cistrome, ENCODE collects data focusing on two sides, high-throughput genomic data and related metadata annotations. For this work, the focus is centered on the metadata section.

Metadata was collected exploiting many different techniques, including manual curation, making the repository one of the few genomic archives that is complete and accurate from a metadata point of view. Many different schemas are collected in the ENCODE repository, but only few of them were selected to extract metadata which had a correspondence with the GCM 2.2. (A list of the schemas is available at `https://www.encodeproject.org/profiles/`).

For example, biological information relative to a certain sample is available in the *biosample* ENCODE schema and many of its fields find a correspondence to the *Biological view* of the GCM 2.4. (Data related to experiments was downloaded through this [1] url-encoded query)

Resulting data was a collection of tuples with the following list of attributes 5.5:

[1] `https://www.encodeproject.org/report.tsv?type=Experiment&field=accession&field=dbxrefs&field=assay_term_name&field=assay_slims&field=target.label&field=assembly&field=biosample_ontology.term_name&field=lab.title&field=award.project&field=replicates.library.biosample.organism.scientific_name&field=replicates.library.biosample.life_stage&field=replicates.library.biosample.age&field=replicates.library.biosample.age_units&field=replicates.library.biosample.sex&field=replicates.library.biosample.donor.ethnicity&field=replicates.library.biosample.donor.health_status&field=replicates.library.biosample.biosample_ontology.classification&field=target.investigated_as&field=biosample_summary&field=description&field=replicates.library.biosample.description&field=replicates.antibody.antigen_description%20&limit=all`

Attribute	Description
External resources	Information about ID used by other repositories (including GEO)
Assay Name	Type of experiment (Chip seq, rna seq, dna seq....)
Assay Type	Super-category of experiment type
Target of Assay	For assays, such as ChIP-seq or RIP-seq, the name of the gene whose expression or product is under investigation for the experiment
Genome assembly	Genome assembly of reference
Biosample term name	Name of the biosample
Lab	Name of the labotory which performed the experiment
Project	Name of the project the biosample belongs to (including ENCODE)
Organism	Species of the biosample
Life stage	Such as "adult" or "embryonic"
Age	A string identifying the age
Age units	Unit measure of the age ("month", "year" etc.)
replicates.library. biosample.sex	Sex of the organism
library.biosample. donor.ethnicity	Ethnicity of the donor
replicates.library.biosample. donor.health_status	Brief description of the health status
replicates.library .biosample.biosample_ontology. classification	Classification of the biosample ontology
target.investigated_as	What the target was being investigated as within an assay
Biosample summary	A description of the biosample
Description	Another general description, not strictly related to the biosample
Submitter comment	Comment left by the submitter
replicates.library.biosample. description	Another description of the biosample

Table 5.5: Description of downloaded ENCODE attributes

The resulting file was a collection of **16.732** entries.

The majority of the attributes present in the file, contained values repre-

sented by a brief string, so it's plausible to suppose that they were filled only with the target relevant information, without textual noise. While **Description, Biosample summary** and **replicates.library.biosample.description** were a long plain text field, similar to the one found in GEO metadata files. These three attributes, were merged into one column named **Input** and were hence used as input text. A manual inspection of the **Submitter comment** column suggested that it could be dropped, because it did not contain information useful for the aim of this work. The column **External Resources** contained references to the same sample in a different archive and often a reference to the relative GSM was found.

Of the 16.732 samples, 6.233 had a reference to the GSM, for those values, as input text, was used the concatenation of the GEO fields "Title" "Characteristics" and "Description", instead of the three ENCODE input fields. The analysis of missing values brought to the following results 5.6:

Attribute	**MV percentage**
Assay Name	0%
Assay Type	0%
Target of Assay	48%
Genome assembly	16%
Biosample term name	0%
Lab	0%
Project	0%
Organism	1%
Life stage	1%
Age	1%
Age units	32%
Sex	1%
Ethnicity	74%
Health_status	53%
Classification	1%
Investigated_as	48%

Table 5.6: Missing Values percentage for each ENCODE output class

The great variety of types of cells, makes the missing values count irrelevant in that, the related sample could be missing certain attributes for biological or experimental reasons. A manual inspection of a subset of entries with missing values suggested that they are Missing At Random (MAR - A.2.13) and, as happened for Cistrome, each of the target missing value was considered to be a correct target value, interpreted as the string "None".

However, some of the classes present a dominant occurrence of missing values, so this suggests to adopt the same performance evaluation metrics used for Cistrome: **Accuracy**, **Weighted Precision** and **Weighted Re-**

call (A.2.6).

5.3 Summary

This Chapter have described in detail the two dataset used for experiments: Cistrome and ENCODE. The first was a collection of 44.843 labeled samples taken from GEO. The samples belong to three different types of genomic experiments, Chip-Seq, ATAC-Seq and DNA-Seq. The second represents a collection of 16.732 samples taken from the ENCODE archive. The resulting dataset is a rich list of 15 attributes. Thanks to a reference to GEO identifier, it was possible to retrieve the GEO metadata description for 6.233 entries. For the remaining ones, a concatenation of three ENCODE attributes (which contained a plain text description similar to the one used in GEO) was used. A manual inspection of missing values suggested that they can be classified as Not Missing At Random in that they represents an actual lack of information for the given sample. An analysis of the mode for each attribute suggested that good performance measures to be considered for the experiments are: Accuracy, Weighted Precision and Weighted Recall.

Chapter 6

Experiments

In this section we will describe the sequence of experiments performed to investigate the **research questions** and, possibly, answer them. The models used for Machine Translation were Encoder-Decoder LSTM and OpenAI GPT-2. The baseline performances were obtained using a classification approach with RoBERTa model. Table 6.1 shows systems setup configurations.

Three different experiments were designed to validate the proposal. Experiment 1 and 2 allow to compare performances of the three analyzed models on different datasets (respectively, Cistrome with input from GEO and ENCODE with input both from GEO and ENCODE itself).

RoBERTa and GPT-2 were trained using a Tesla P100-PCIE-16GB GPU, while the LSTM model was trained on Google Colaboratory[1] B.3 with GPU accelerator.

Model	Batch size	Loss function	Tokenizer	Optimizer	LR	beta_1	beta_2	epsilon
RoBERTa	10	Cross Entropy	BPE	Adam	2e-4	0.9	0.999	1e-6
LSTM	64	Sparse Cross Entropy	keras	Adam	1e-3	0.9	0.999	1e-7
GPT-2	5	Cross Entropy	BPE	Adam	1e-3	0.9	0.999	1e-6

Table 6.1: Setup of the three different models for each experiment. BPE = Byte Pair Encoding; LR = learning rate

6.1 Experiment 1

In this experiment we evaluate the performances of the two sequence-to-sequence models (LSTM and GPT-2), comparing them to the standard Multi Label Classifier (RoBERTa) using samples taken from Cistrome dataset. The three models were subject to an Early Stopping method to avoid overfitting. Training was stopped if the loss computed for predictions on validation set stopped decreasing from one epoch to the next.

[1] `https://colab.research.google.com/`

6.1.1 Data processing

Data was split into a training set (80%), Validation set (10%) and Test set (10%). In addition to cleaning procedures, a different padding process was executed for the two sequence-to-sequence models. The Encoder-Decoder required Input-Output pairs which were respectively padded to the length of the maximum Input and Output. GPT-2 required single sentences which were padded to a maximum length of 500 characters, 222 sentences exceeded the maximum length and were excluded from datasets.

6.1.2 Results and comments

Model	# Epochs	Accuracy	Precision	Recall
RoBERTa	69	0.90	0.89	0.91
LSTM + Attention	15	0.62	0.65	0.62
GPT-2	47	0.93	0.93	0.93

Table 6.2: Precision and Recall were weighted for the number of occurrences of each attribute value

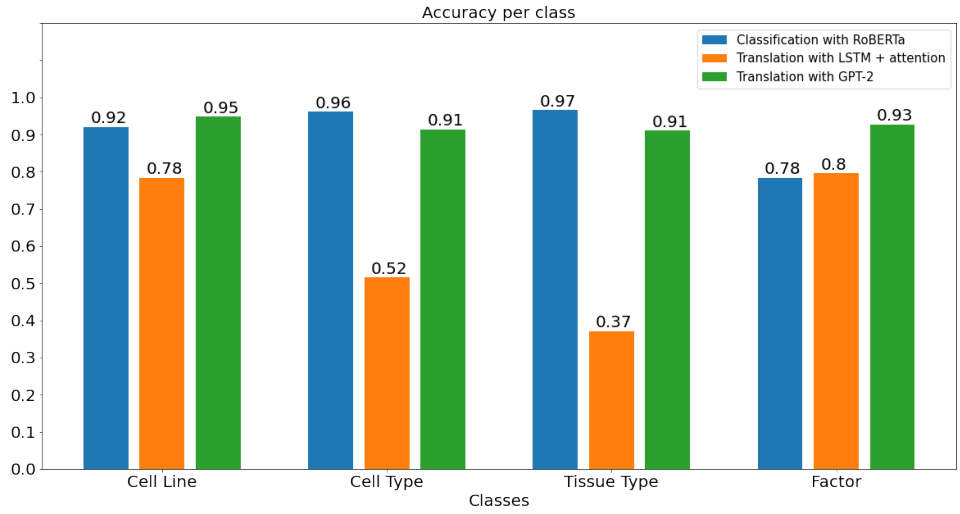


Figure 6.1: Per class accuracy of the three models for Experiment 1

Figure 6.1 and Table 6.2 show the performances of the three models. The overall performances obtained by GPT-2 is higher than LSTM and RoBERTa. As shown in Figure 6.1, the classifier seems to perform better for classes which contain a low number of distinct values i.e. *Cell Type* and *Tissue Type* (which contain 380 and 249 possible values), while for *Cell Line* and *Factor* (which contain 1045 and 1565 possible values) even the simple LSTM model can beat RoBERTa.

Another interesting observation emerges if the number of "None" values is taken in consideration (Table 5.1), the classes Cell Line, Cell Type and Tissue Type present a relevant percentage of "None", the weighted Precision and Recall analysis, however, shows high scores, despite the unbalance of values count; this implies that the models were able to correctly classify samples which lack of labels for certain classes. It is interesting to mention a **comparison with related works** that embraced similar types of information extraction from GEO samples: ALE 3.2.2 and Onassis 3.2.5. The first used a matching score between the input text and an ontology for *Tissues* reaching about 0.85 Micro Precision and 0.75 Micro Recall (A.2.7); results showed that the extraction of *Tissue* for **GPT-2** reached **0.93** Weighted Precision and **0.92** Weighted Recall (weighted average brings to the same results as micro average for this task). The second used NER with a lookup approach, reaching an Accuracy of about 0.8; results showed that the Accuracy reached by **GPT-2** is equal to **0.91**. It's important to mention that also the baseline RoBERTa classifier beat the ALE and Onassis approach. An additional fact - mentioned in Chapter 3 - gives even more strength to the results obtained, it must be highlighted indeed that scores reached by ALE in the *Tissue* extraction are obtained only on the 25 most frequent tissues and that Onassis approach is limited to cases where the information is present in the input text. Table 6.3 shows few predictions made by GPT-2 were the model was able to handle cases of inference, thus, cases which none of the previous works would have been able to classify.

Input Sentence	Output Sentence
Title: JC2379_T47D_PR_3hr_E2 r3 - Description: Cells were treated with progesterone (100nM) or R5020 (10nM) for 3hr and immunoprecipitated, and DNA extracted. - Characteristics: cell line: T47D; cell type: ER+ breast cancer cell line; condition: E2; chip antibody: PR Antibody (H-190); chip antibody vendor: Santa Cruz; chip antibody cat. #: sc-7208; chip antibody lot #: H2312	Cell Line: T47D - Cell Type: **Epithelium** - Tissue Type: **Mammary gland** - Factor: PR
Title: GM12878-H3K27ac-Rep2-Manual-SL45089 - Description: chip-antibody: mouse monoclonal H3K27ac, Wako 306-34849 - Characteristics: Homo sapiens	Cell Line: GM12878 - Cell Type: **B Lymphocyte** - Tissue Type: **Blood** - Factor: H3K27ac
Title: Nanog-ChIP - Description: cell line: Sim2-expressing cells; clone: A6; antibody: anti-Nanog (D2A3) XP (Cell Signaling #8822) - Characteristics: Mus musculus	Cell Line: None - Cell Type: **Embryonic Stem Cell** - Tissue Type: **Embryo** - Factor: NANOG

Table 6.3: Examples of GPT-2 translations for Cistrome. Bold labels are not explicit in the input text

The last case is particularly meaningful. Despite the presence of "cell line: Sim2-expressing cells", the target value for "Cell Line" is "None". Sim2 represents indeed a protein and not a cell line. This is a clear example where standard Regular Expressions would certainly have produced an erroneous prediction, but GPT-2 was able to output the correct value, which is indeed missing.

During the training of the sequence to sequence models it is possible to observe that GPT-2 is able to learn the output structure after the first epoch of training, while the LSTM model takes 2-3 epochs.

6.2 Experiment 2

This experiment mirrors the first one, but was performed on the ENCODE dataset. As for Experiment 1 6.1 the three models were subject to an Early Stopping method to avoid overfitting, training was stopped if the loss computed for predictions on validation set stopped decreasing.

6.2.1 Data processing

Data was split into Trainset (80%), Validation set (10%) and Test set (10%). Same text cleaning and padding processes as Experiment 1 6.1 were adopted. 1174 items were excluded from dataset because they exceeded the maximum length imposed for the experiment.

6.2.2 Results and comments

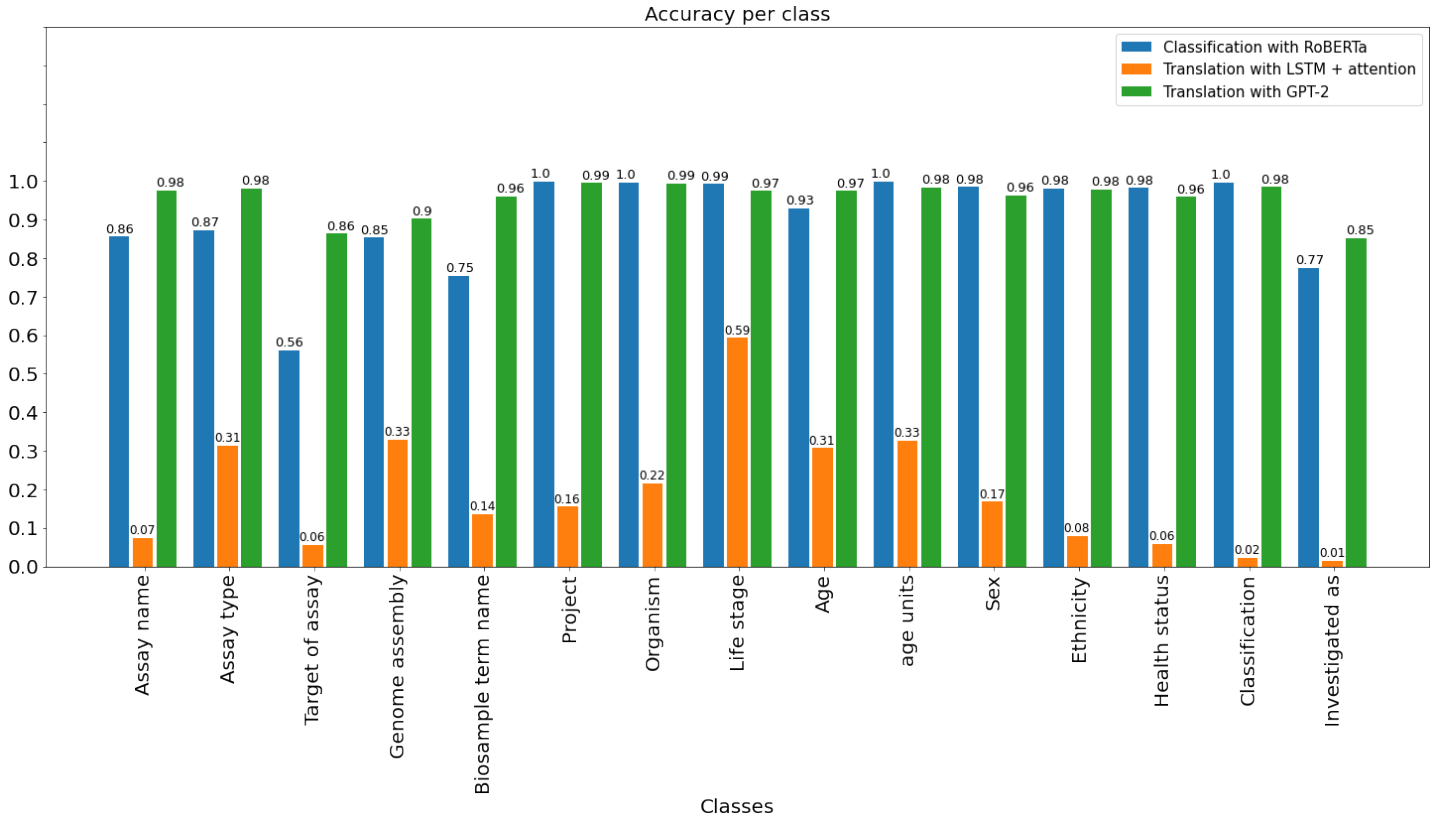


Figure 6.2: Per class accuracy of the three models for Experiment 2

Model	# Epochs	Accuracy	Precision	Recall
RoBERTa	71	0.90	0.89	0.90
LSTM + Attention	22	0.19	0.19	0.19
GPT-2	48	0.96	0.96	0.96

Table 6.4: Precision and Recall were weighted for the number of occurrences of each attribute value

Figure 6.2 and Table 6.4 show the performances of the three models. As for Experiment 1 6.1 the GPT-2 model seems to perform better than LSTM and RoBERTa.

The same behaviour of better performance for Translation models for attributes with larger number of distinct values seems to present even for this experiment. The attributes *Target of Assay* and *Biosample term name* are the ones that present the highest number of distinct values and GPT-2 far exceeded RoBERTa in terms of accuracy.

This experiments highlights how the simple LSTM + attention model is not able to perform well for a larger number of target attributes, at least with the tested model size.

The labels *Health Status* and *Ethnicity* presented several "None" values (74% and 53%), but both RoBERTa and GPT-2 were able to predict correctly almost the totality of samples, producing results with high Weighted Precision and Weighted Recall.

It is interesting - again - to mention a comparison with ALE 3.2.2 that embraced similar types of information extraction from GEO samples. ALE aimed to extract *Age* and *Sex* with poor results, often limiting the target age unit to Years or Months and the target Sex to only Male and Female; in particular, the work adopted Regular Expressions reaching about 0.95 Precision and Recall for the extraction of *Sex* and about 0.98 Precision, but 0.4 Recall on the extraction of *Age*, due to difficulties in matching the target age in the input text.

GPT-2 model reached **0.96** Precision and Recall in the extraction of *Sex* and **0.97** in the extraction of *Age*.

This experiment demonstrates that a translation approach can outperform the State Of The Art and it's able to handle a different number of non-standard cases as shown in Table 6.5.

Input Sentence	Output Sentence
DNase-seq on 91 day old male fetal human kidney tissue left kidney female embryo (59 days) and male embryo (91 day)	Assay name: DNase-seq - Assay type: DNA accessibility - Target of assay: None - Genome assembly: GRCh38 - Biosample term name: left kidney - Project: ENCODE - Organism: Homo sapiens - Life stage: embryonic - Age: 91,59 - Age units: day - Sex: male,female - Ethnicity: None - Health status: None - Classification: tissue - Investigated as: None
RNA Bind-n-Seq (RBNS) pulldown experiment against PUM1	Assay name: RNA Bind-n-Seq - Assay type: RNA binding - Target of assay: PUM1 - Genome assembly: hg19 - Biosample term name: cell-free sample - Project: ENCODE - Organism: None - Life stage: None - Age: None - Age units: None - Sex: None - Ethnicity: None - Health status: None - Classification: None - Investigated as: RNA binding protein
ChIP-seq against eGFP-efl-1 in L1 larva worms. strain YL418 (unc-119(ed3)III; vrIs65 [pGES-1::EFL-1::GFP FLAG:EFL-1 3'UTR, unc-119 (+)]) whole organism L1 larva genetically modified using bombardment	Assay name: ChIP-seq - Assay type: DNA binding - Target of assay: efl-1 - Genome assembly: ce11,ce10 - Biosample term name: whole organism - Project: modENCODE - Organism: Caenorhabditis elegans - Life stage: L1 larva - Age: unknown - Age units: None - Sex: mixed - Ethnicity: None - Health status: None - Classification: whole organisms - Investigated as: cofactor,transcription factor
datatype: DnaseSeq; datatype description: DNaseI HS Sequencing; labversion: baseAlignCounts.pl v 1,fseq v 1.84, iff_generic_male,p-value cutoff: 0.05; replicate: 1,2; cell: Colo829; cell organism: human; cell description: malignant melanoma; cell karyotype: cancer; cell sex: M; labversion: baseAlignCounts.pl v 1 Duke human Colo829 DNase-seq COLO829 malignant melanoma	Assay name: DNase-seq - Assay type: DNA accessibility - Target of assay: None - Genome assembly: GRCh38,hg19 - Biosample term name: COLO829 - Project: ENCODE - Organism: Homo sapiens - Life stage: adult - Age: 45 - Age units: year - Sex: male - Ethnicity: Caucasian - Health status: metastatic melanoma - Classification: cell line - Investigated as: None

Table 6.5: Examples of GPT-2 translations for ENCODE

The first example is a case in which the model handles a double parameter prediction, two cells were present in the sample and the prediction for *Age* presents the two values, 91 and 59. The second example is a case

for which the majority of the attributes were neither present, nor deducible from the input text. The model is able to tell correctly whenever the text lacks of information. The third and fourth example are cases of imputing from a messy input text. In the third it is possible to observe how the model can infer - from previous knowledge - that L1 cells belong to a 'Caenorhabditis elegans' and can tell the multiple Genome Assembly of reference, plus it's able to filter the relevant Target Of Assay from a noisy sequence of dash-separated characters. In the fourth case it is possible to observe the capability of the model to deduce the age from the COLO829 cell line.

Of note, with the prediction of the Health Status, the output is not a simple transcription of a portion of the input - otherwise the output would have been simply "malignant melanoma" - but GPT-2 adds the detail of "metastatic". During the training of the sequence to sequence models it's possible to observe that GPT-2 is able to learn the output structure after the first epoch of training, while the LSTM model takes 4 epochs.

6.3 Summary

Results showed that in the case of extraction of a limited number of output labels, both sequence to sequence models reach high performances. As expected, GPT-2 outperforms the LSTM model significantly, but it's remarkable how good are the performances of the LSTM model if compared with previous work especially given the simplicity of the model. The machine translation approach has been shown to beat both state-of-the-art and the baseline classifier - which represents one of the best performing Language Models for NLP tasks - for the given **task**, in particular when the target attribute contains a large number of possible values. Both translation models have shown the ability to understand the output structure after a few epochs of training. GPT-2 has shown some remarkable abilities in *deduction* from previous knowledge, whenever the information was hidden in the input text; understanding cases of *double cell experiments*, handling cases of a *misleading input text*.

The answers to the **Research questions** would then be:

- Sequence-to-Sequence models can provide structured information from unstructured text, in particular, the LSTM model was able to output structured text after 3-4 epochs of training, while GPT-2 needed only one epoch.

- Sequence-to-Sequence models can extract correct information from plain text overcoming the problems cited in Table 2.2; indeed, **GPT-2** reached **0.93** and **0.96** of accuracy on Experiment 1 and 2, respectively.

- Sequence-to-Sequence models can beat approaches tried in previous work.

  Experiment 1 showed that in the extraction of *Tissue*, Precision and Recall for **GPT-2** were **0.93** and **0.92**; while for ALE were 0.85 and 0.7. Accuracy reached by Onassis in the same task was about 0.8, while **GPT-2** reached **0.91**.

  Experiment 2 showed that **GPT-2** reached Precision and Recall both equal to **0.96** for the extraction of *Sex* and **0.97** for *Age*; while ALE reached about 0.95 Precision and Recall for *Sex* and 0.98 Precision, but 0.4 Recall on *Age*.

- Sequence-to-Sequence Language Models beat other types of approaches. Specifically **GPT-2** beat RoBERTa classifier in terms of Accuracy, Weighted Precision and Weighted Recall:

  0.93 VS 0.90 - **0.93** VS 0.89 - **0.93** VS 0.91 for Experiment 1;

  0.96 VS 0.90 - **0.96** VS 0.89 - **0.96** VS 0.90 for Experiment 2.

Chapter 7

Conclusions

This thesis evaluates the performances of Sequence to Sequence models in the task of extraction of structured metadata from plain text have been evaluated.

The **task** proposed at the beginning was to *extract all the information available in GSM metadata file in a structured form.* Chapter 2 of this work first describes the GEO repository, then the target GCM and, subsequently, it exposes the given task.

The following sections expose the main problems that emerge from a plain text description of biological samples, which are: adoption of synonyms, abbreviations, scattering of words in paragraphs, presence of hidden information and the rapid development of new knowledge in the biology field.

Given the problems mentioned above, the main research questions posed were *Can Sequence-to-Sequence models provide structured information from unstructured plain text? Can Sequence-to-Sequence models extract correct biological information from plain text overcoming the problems in Table 2.2? How do Sequence-to-Sequence models perform, in relation with other approaches?.*

Chapter 3 describes important previous contributions to the task. It has been showed that the three main existing categories of approaches deal with the extraction of a few factors or genetic marks, providing poor results and showed structural inabilities when tackling the problems cited above. The need for a new methodology was clarified, hence the Sequence-to-Sequence approach was presented.

The following Chapter reports the methodologies adopted to deal with the given **task**. In particular it describes that the approach reported in this work is the training of two Sequence-to-Sequence models, an LSTM + Attention layer and the GPT-2, with input-output pairs, where the input was an unstructured description of samples, while the output presents a fixed structure of "key: value" pairs, in order to learn the model to generate

sentences easily accepted by Regular Expressions. The output sentences were built from a list of target labels taken from two repositories: Cistrome and ENCODE. From the first archive we collected 44.843 annotations of four labels for GEO samples, from the second 23 downloaded 16.732 entries with a total of 15 labels.

The two datasets allows the setting of two experiments, which aimed to compare the seq-to-seq approach with a Multi-Label Classifier chosen as baseline i.e. RoBERTa, an evolution of the BERT language model.

Results showed that:

- Sequence-to-Sequence models can provide structured information from unstructured text.
- Sequence-to-Sequence models can extract correct information from plain text overcoming the problems cited in Table 2.2.
- Sequence-to-Sequence models can beat approaches tried in previous work.
- Sequence-to-Sequence Language Models beat other types of approaches. Specifically **GPT-2** beat RoBERTa classifier in terms of Accuracy, Weighted Precision and Weighted Recall.

Bibliography

[1] Barrett, Tanya and Wilhite, Stephen E. and Ledoux, Pierre and Evangelista, Carlos and Kim, Irene F. and Tomashevsky, Maxim and Marshall, Kimberly A. and Phillippy, Katherine H. and Sherman, Patti M. and Holko, Michelle and Yefanov, Andrey and Lee, Hyeseung and Zhang, Naigong and Robertson, Cynthia L. and Serova, Nadezhda and Davis, Sean and Soboleva, Alexandra. NCBI GEO: archive for functional genomics data sets—update. *Nucleic Acids Research*, 41(D1):D991–D995, 11 2012.

[2] Anna Bernasconi, Alessandro Campi, and Marco Masseroli. Conceptual modeling for genomics: Building an integrated repository of open data. pages 325–339, 10 2017.

[3] Anna Bernasconi, Arif Canakoglu, and Stefano Ceri. From a conceptual model to a knowledge graph for genomic datasets. In *International Conference on Conceptual Modeling*, pages 352–360. Springer, 2019.

[4] Anna Bernasconi, Stefano Ceri, Alessandro Campi, and Marco Masseroli. Conceptual modeling for genomics: building an integrated repository of open data. In *International Conference on Conceptual Modeling*, pages 325–339. Springer, 2017.

[5] Arif Canakoglu, Anna Bernasconi, Andrea Colombo, and Marco Masseroli. Genosurf: metadata driven semantic search system for integrated genomic datasets. *Database : the journal of biological databases and curation*, 2019, 01 2019.

[6] Lisa Helbling Chadwick. The nih roadmap epigenomics program data resource. *Epigenomics*, 4(3):317–324, 2012. PMID: 22690667.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[8] Djordje Djordjevic, Yun Xin Chen, Shu Lun Shannon Kwan, Raymond W. K. Ling, Gordon Qian, Chelsea Y. Y. Woo, Samuel J. Ellis, and

Joshua W. K. Ho. Georacle: Mining perturbation experiments using free text metadata in gene expression omnibus. *bioRxiv*, 2017.

[9] Eugenia Galeota, Kamal Kishore, and Mattia Pelizzola. Ontology-driven integrative analysis of omics data through Onassis. *Scientific reports*, 10(1):703, January 2020.

[10] Cory B Giles et al. ALE: Automated Label Extraction from GEO metadata. *BMC Bioinformatics*, 18(14):509, 2017.

[11] Gregory Gundersen, Matthew Jones, and Avi Ma'ayan. Geo2enrichr: A google chrome extension to extract gene sets from the gene expression omnibus and analyze these lists for common biological functions. volume 75, 11 2015.

[12] Zhengyu Guo, Boriana Tzvetkova, Jennifer M Bassik, Tara Bodziak, Brianna M Wojnar, Wei Qiao, Md A Obaida, Sacha B Nelson, Bo Hua Hu, and Peng Yu. Rnaseqmetadb: a database and web server for navigating metadata of publicly available mouse rna-seq datasets. *Bioinformatics*, 31(24):4038–4040, 2015.

[13] Dexter Hadley, James Pan, Osama El-Sayed, Jihad Aljabban, Imad Aljabban, Tej D Azad, Mohamad O Hadied, Shuaib Raza, Benjamin Abhishek Rayikanti, Bin Chen, et al. Precision annotation of digital samples in ncbi's gene expression omnibus. *Scientific data*, 4:170125, 2017.

[14] Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*, 2015.

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[16] Eurie L. Hong, Cricket A. Sloan, Esther T. Chan, Jean M. Davidson, Venkat S. Malladi, J. Seth Strattan, Benjamin C. Hitz, Idan Gabdank, Aditi K. Narayanan, Marcus Ho, Brian T. Lee, Laurence D. Rowe, Timothy R. Dreszer, Greg R. Roe, Nikhil R. Podduturi, Forrest Tanaka, Jason A. Hilton, and J. Michael Cherry. Principles of metadata organization at the ENCODE data coordination center. *Database*, 2016, 03 2016. baw001.

[17] L. KARTTUNEN, J-P. CHANOD, G. GREFENSTETTE, and A. SCHILLE. Regular expressions for language engineering. *Natural Language Engineering*, 2(4):305–328, 1996.

[18] Rasko Leinonen, Hideaki Sugawara, and on behalf of the International Nucleotide Sequence Database Collaboration Shumway, Martin. The

Sequence Read Archive. *Nucleic Acids Research*, 39(suppl_1):D19–D21, 11 2010.

[19] Jin Li, Ching-San Tseng, Antonio Federico, Franjo Ivankovic, Yi-Shuian Huang, Alfredo Ciccodicola, Maurice S Swanson, and Peng Yu. Sfmetadb: a comprehensive annotation of mouse rna splicing factor rna-seq datasets. *Database*, 2017, 2017.

[20] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[21] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025, 2015.

[22] Shenglin Mei, Qian Qin, Qiu Wu, Hanfei Sun, Rongbin Zheng, Chongzhi Zang, Muyuan Zhu, Jiaxin Wu, Xiaohui Shi, Len Taing, Tao Liu, Myles Brown, Clifford A. Meyer, and X. Shirley Liu. Cistrome Data Browser: a data portal for ChIP-Seq and chromatin accessibility data in human and mouse. *Nucleic Acids Research*, 45(D1):D658–D662, 10 2016.

[23] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. *CoRR*, abs/1606.06031, 2016.

[24] Lisa Posch, Maryam Panahiazar, Michel Dumontier, and Olivier Gevaert. Predicting structured metadata from unstructured metadata. *Database*, 2016:baw080, 2016.

[25] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[26] Erhard Rahm and Philip Bernstein. A survey of approaches to automatic schema matching. *VLDB J.*, 10:334–350, 12 2001.

[27] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, 1606.05250, 2016.

[28] Consortium ENCODE. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, 2012.

[29] Ida Schomburg, Antje Chang, Sandra Placzek, Carola Söhngen, Michael Rother, Maren Lang, Cornelia Munaretto, Susanne Ulas, Michael Stelzer, Andreas Grote, Maurice Scheer, and Dietmar Schomburg. BRENDA in 2013: integrated reactions, kinetic data, enzyme function data, improved disease classification: new options and contents in BRENDA. *Nucleic Acids Research*, 41(D1):D764, 2013.

[30] Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, William Bug, Werner Ceusters, Louis Goldberg, Karen Eilbeck, Amelia Ireland, Christopher Mungall, Neocles Leontis, Philippe Rocca-Serra, Alan Ruttenberg, Susanna-Assunta Sansone, Richard Scheuermann, Nigam Shah, Patricia Whetzel, and Suzanna Lewis. The obo foundry: Coordinated evolution of ontologies to support biomedical data integration. *Nature biotechnology*, 25:1251–5, 12 2007.

[31] R. Socher, A. Perelygin, J.Y. Wu, J. Chuang, C.D. Manning, A.Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. *EMNLP*, 1631:1631–1642, 01 2013.

[32] Sara Tarek, Reda [Abd Elwahab], and Mahmoud Shoman. Gene expression based cancer classification. *Egyptian Informatics Journal*, 18(3):151 – 159, 2017.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv e-prints*, page arXiv:1706.03762, Jun 2017.

[34] Zichen Wang, Alexander Lachmann, and Avi Ma'ayan. Mining data and metadata from the gene expression omnibus. *Biophysical reviews*, 11(1):103–110, 2019.

[35] Zichen Wang, Alexander Lachmann, and Avi Ma'ayan. Mining data and metadata from the gene expression omnibus. *Biophysical reviews*, 11(1):103–110, 2019.

[36] Zichen Wang, Caroline D Monteiro, Kathleen M Jagodnik, Nicolas F Fernandez, Gregory W Gundersen, Andrew D Rouillard, Sherry L Jenkins, Axel S Feldmann, Kevin S Hu, Michael G McDermott, et al. Extraction and analysis of signatures from the gene expression omnibus by the crowd. *Nature communications*, 7(1):1–11, 2016.

[37] Weinstein, John N and others. The Cancer Genome Atlas pan-cancer analysis project. *Nature Genetics*, 45(10):1113–1120, 2013.

[38] Adina Williams, Nikita Nangia, and Samuel Bowman. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[39] Sitan Yang and Daniel Q. Naiman. Multiclass cancer classification based on gene expression comparison. *Statistical Applications in Genetics and Molecular Biology*, 13(4):477 – 496, 2014.

[40] Rongbin Zheng, Changxin Wan, Shenglin Mei, Qian Qin, Qiu Wu, Hanfei Sun, Chen-Hao Chen, Myles Brown, Xiaoyan Zhang, Clifford A Meyer, and X Shirley Liu. Cistrome Data Browser: expanded datasets and new tools for gene regulatory analysis. *Nucleic Acids Research*, 47(D1):D729–D735, 11 2018.

[41] Yuelin Zhu, Sean Davis, Robert Stephens, Paul S. Meltzer, and Yidong Chen. GEOmetadb: powerful alternative search engine for the Gene Expression Omnibus. *Bioinformatics*, 24(23):2798–2800, 11 2008.

Appendix A

Glossary

A.1 Biology

A.1.1 Micro array

Set of microscopic DNA probes attached to a solid surface such as glass, plastic, or silicon chips forming an array. These arrays allow to compare the gene expression profile of individual patient having a disease with that of a healthy one (also known as Case–control study) to identify which genes are involved in the disease.

A.1.2 Sequencing

Process for determining the exact primary structure of a biopolymer, i.e. the order of the bases in the case of a Nucleic acid(DNA or RNA). Several strategies have been developed in the last decades. One of the mostly used is the chain termination method or Sanger method. In the last few years, however, due to the reduction of costs and the time of the overall process, the next generation sequencing methods are the mostly used currently.

A.1.3 Next-generation-sequencing

Set of technologies that allow the parallel sequencing of millions of DNA fragments. Thereby performing analysis of multiple genes simultaneously, with a diagnostic yield higher than traditional sequencing, with drastically short time of genetic analysis.

A.1.4 Phenotype and Genotype

The complex of the visible characteristics of an individual and which are the result of the interaction between the genetic heritage and external factors is called fenotype. Is is the opposite concept to the genotype which is the

totality of the genes present in the genome or the genes involved in the determination of a single phenotypic trait.

A.1.5 Gene Expression

Series of events that since the activation of the transcription of a gene, lead to production of the corresponding protein.

A.1.6 Gene signature

Gene expression pattern associated with a particular cellular phenotype or with a precise prognosis in medicine.

A.1.7 ChIPSeq

Chromatin immunoprecipitation followed by sequencing. Experimental technique used in the analysis of interactions between DNA and proteins and for the study of epigenetic alterations of histones or basic proteins (i.e. promoters or enhancers) that model the structural component of chromatin. This step is then followed by sequencing to identify the order of nucleotides or aminoacids

A.1.8 Genome Assembly

Nucleic acid sequences of DNA, assembled by scientists as a representative example of a species' group of genes.

A.1.9 Perturbation

Functional investigation of the mammalian genome able to reveal how genetic alterations lead to changes in phenotype.

A.1.10 Case-Control study

A case-control study compares two groups retrospectively. Generally, individuals who have developed a disease (treatment or case) could be compared to a group of individuals who have not developed it (control). The researcher will observe if there are differences between the two groups in their previous exposure to possible risk factors.

A.2 Machine Learning

A.2.1 Schema Matching

Basic problem in many database application domains, such as data integration, which takes two schemas as input and produces a mapping between elements of the two schemas that correspond semantically to each other.

A.2.2 Regular Expressions

A regular expression (regex or regexp for short) is a special text string for describing a search pattern.

A.2.3 One Versus Rest

Classification strategy for multi class tasks. A binary classifier for each output class is built each of which is trained to identify the elements belonging to the target class and the elements belonging to any of all the remaining classes.

A.2.4 CrossEntropy

Formula to compute loss for multi class classification tasks for each observation:
$-\sum_{c=1}^{M} y_{i,c} log(p_{i,c})$.
Where:

- i is the observation
- $y_{i,c}$ binary indicator (0 or 1) if class label c is the correct classification for observation i
- M is the total number of classes
- p is the predicted probability observation i is of class c

A.2.5 Perplexity

Metric to measure how well a probability distribution predicts a sample.

A.2.6 Weighted Precision and Recall

Used in the case of a multi-class classification task. The Precision and Recall values are computed for each possible label and then, each resulting value is weighted for the number of occurrences of each label.

A.2.7 Micro Precision and Recall

Used in the case of a multi-class classification task. First, the number of TP, FP and FN is computed per each class, then the results are summed up and Precision and Recall values are computed on the overall number of TP, FP and FN.

A.2.8 Language Model

Model able to estimate a probability distribution over sequence of words, i.e. $p(x_1, x_2, [...], x_N)$ where $x_1, x_2[...]$ represents words.

A.2.9 Self Attention

Technique used in Neural Networks whether a certain layer exploits the mechanism of the attention [33] applied on the same input sequence that the network is working on.

A.2.10 Masked Self-Attention

Technique used in modern Language Models to make them exploit the mechanism of Self Attention, but placing to 0 some of the values of the attention vector.

A.2.11 Masked Language Model

Language Model trained by hiding to the model some of the words of an input sequence and let the model estimate the probability of the hidden word.

A.2.12 Zero shot learning

Task where the outcome of the models is evaluated without providing any label to the model in the training phase.

A.2.13 Missing at Random

Type of missing values which distribution depends on other variable. This is the case of the missing values for Cistrome and ENCODE: e.g. "Ethnicity" label can't exist for all the Species different from "Human". e.g. "Cell Type" label can't exist for Embrionic Stem Cells

A.2.14 ROC curve

Type of graph used for classification tasks that plots the True Positive Rate against the False Positive Rate. The more the value of the Area Under

Curve is near 1 (or 0), the more the classifier is able to perform. A random classifier will have an AUC of 0.5.

A.2.15 Precision and Recall Curve

Plot used - in classification tasks - to plot the precision (y-axis) and the recall (x-axis) for different thresholds. Same analysis of the AUC as for the ROC curve is applied here, the closest the AUC is to 0.5, the worse the classifier is.

A.2.16 F1-score

Formula to compute the harmonic mean between Precision and Recall: $2\frac{precision \cdot recall}{precision+recall}$ highest values for F-1 implies good balancing and high values for precision and recall.

A.2.17 Matthew's correlation coefficient

Binary classifications quality measure. $\frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

A.2.18 TF-IDF

Term frequency–inverse document frequency. TF-IDF is a function of a word which aims to weight the searched term for its relevance. It's the product of two terms, TF and IDF. TF is the frequency of a certain word in a document, the more a word is present in a certain document, the more it will be frequent, thus the more it will be important. IDF is the natural logarithm of the fraction of the total number of documents and the documents containing a certain word, if a word appears in several documents, this means that it will have less importance. E.G. the word "the" will be less relevant than the word "stochastic".

A.2.19 SVM

Support Vector Machine. One of the most used supervised classifiers which works by finding an hyperplane that better separates data points in the input feature space. Points that are closest to the hyperplane are called Support Vectors, the objective of the SVM is to maximize the distance between the hyperplane and the Support Vectors.

A.2.20 LDA

Linear Discriminant Analysis. It's a dimensionality reduction technique used in classification tasks. LDA does it by projecting the data into an hyperplane with smaller dimension w.r.t. the original input feature space. The objective

is to minimize the variance and maximize the distance between the means of the different classes.

A.2.21 Gower Distance

Distance measure that can be used to calculate distance between two entity whose attribute has a mixed of categorical and numerical values.

Appendix B

Tools and platforms

B.1 Python 3.7

Language used for the entire research process, including data analysis, data processing, model building, experiment settings, results.

B.2 Tensorflow 2.1

Machine learning framework used for building the LSTM + attention models, in particular Keras package was used.

B.3 Google Colaboratory

Free online platform that allows the development of jupyter notebooks; moreover it allows the free use (for a limited time) of GPUs. Used to run experiments for the LSTM + attention model.

B.4 Pytorch

Machine learning framework used for the usage of RoBERTa and GPT-2 models.

B.5 SQLite

Software for Relational Database Management Systems used to perform queries on the GEOMetaDB database.