UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE INFORMÁTICA

Predicting non-coding RNA function using Artificial Intelligence

David Alexandre da Costa Correia

Mestrado em Bioinformática e Biologia Computacional

Dissertação orientada por:

Doutor Hugo Filipe de Mesquita Costa Martiniano

Professor Doutor Francisco José Moreira Couto

2024

Acknowledgements

For Francisca, who cultivated the curiosity in me. These are the fruits of your labour.

I would like to thank my supervisor Hugo Martiniano for his availability, guidance and support.
I also thank Professor Francisco Couto for his teachings and guidance.

To my family, to João Pedro, Júlia and Joana, and to everyone who shares a piece of my heart, thank you.

Abstract

Non-coding RNAs (ncRNAs) represent the majority of human gene products and are involved in various important biological processes, being considered relevant disease biomarkers and therapeutic agents. However, there are few functional annotation databases dedicated to ncRNAs and information about these biomolecules remains sparsely distributed, mostly in the form of scientific research articles. It is then of pivotal importance to aggregate and summarize the existing information.

Natural Language Processing methods applied to text mining enable automatic information extraction and summarization from textual data. These techniques can be used to generate collections of annotated sentences expressing relations between entities, called relational corpora.

In this work, a text mining pipeline was implemented to generate a ncRNA-phenotype relational corpus (ncoRP) using Distant Supervision Relation Extraction (DSRE), consisting of 21,608 annotated articles, 2,835 unique ncRNAs, 1,118 unique phenotypes and 35,295 unique relations, with a precision of 0.761 and F1-score of 0.593, calculated through human validation. DSRE methods require a set of pre-documented relations to function, as such, a high-fidelity ncRNA-phenotype relation dataset, consisting of 214,300 unique relations, was created by the aggregation of five ncRNA-disease functional annotation databases. Then, both ncoRP and the relation dataset represent important contributions towards solving the problem with the sparseness of information about ncRNAs.

Large Language Models (LLMs) are an emerging type of language model, showing great capabilities in general task-solving through text generation, without the requirement of fine-tuning with large datasets. This benefit shows promise for applications in Relation Extraction (RE), when compared to data-intensive state-of-the-art deep learning methods. In this work, a LLM RE methodology is proposed and evaluated, achieving an F1-score of 0.978 by combining the RE task with a preceding sentence filtering task and applying prompting principles such as in-context learning and Chain-of-Thought self-explanation.

Keywords: non-coding RNAs, Relation Extraction, Text Mining, Distant Supervision, Large Language Models.

Resumo Alargado

RNAs não codificantes (ncRNAs) representam a maioria dos produtos génicos humanos e estão envolvidos numa gama extensa de processos biológicos importantes, como a síntese proteica, a regulação da expressão de outros produtos génicos e vários outros. Por isso, a disregulação da expressão destas biomoléculas conduz naturalmente à ocorrência e agravamento de diferentes patologias. Então, para além de biomarcadores de doença, ncRNAs servem também como agentes terapêuticos, nomeadamente através do aumento de alvos para fármacos pela modificação estrutural de RNAs mensageiros e proteínas. No entanto, tendo em conta a sua importância, ncRNAs estão ainda relativamente pouco estudados e documentados, existindo poucas bases de dados de anotação funcional. Isto porque, dada a sua complexidade, estudos biológicos específicos, que são dispendiosos, são necessários para estudar a sua função. Consequentemente, grande parte da informação disponível sobre ncRNAs está dispersa sob a forma de artigos de divulgação científica, o que torna difícil estar consciente de todo o conhecimento existente. É então essencial agregar e sumarizar toda a informação valiosa disponível, de modo a evitar a repetição de estudos dispendiosos.

Processamento de Linguagem Natural (PLN) é um ramo da Inteligência Artificial (IA) que procura fazer com que computadores compreendam linguagem humana de forma automática, tendo aplicações em extração e sumarização de informação a partir de grandes quandidades de texto, o que é chamado de prospeção de texto. Estes métodos podem então ser utilizados para obter coleções de frases que expressem relações entre entidades, denominadas de *corpora* relacionais. *Corpora*, para além de servirem como um método de agregação de informação, têm também aplicações importantes no desenvolvimento e treino de modelos de Aprendizagem Automática para várias tarefas no PLN.

Neste projeto, foi implementado um *pipeline* de prospeção de texto para gerar um *corpus* relacional ncRNA-fenótipo (ncoRP) através de Extração de Relações por Supervisão à Distância (ERSD). Técnicas de ERSD requerem um conjunto de relações pré-documentadas para as entidades em questão, já que funcionam baseado-se na simples aplicação do princípio que se segue: Se existe uma relação pré-documentada entre A e B, então uma frase que mencione A e B expressa necessariamente essa relação. Este tipo de algoritmo é então bastante útil sendo que consegue gerar grandes quantidades de frases anotadas sem necessitar de treino.

Para a aplicação de ERSD, um *dataset* de relações ncRNA-fenótipo foi construído através da agregação e processamento de cinco bases de dados de anotação funcional de ncRNA-doença. Com o objetivo de aumentar a qualidade das relações no *dataset*, cada ncRNA e fenótipo foi associado a um identificador

(ID) externo. Para os ncRNAs, foram utilizados IDs da RNACentral, uma base de dados de ncRNAs que associa cada sequência a um ID único, o que torna estes IDs uma boa forma uniforme de identificar cada ncRNA. Para os fenótipos, IDs da Human Phenotype Ontology (HPO) foram utilizados, sendo esta uma ontologia padrão para fenótipos humanos, onde estão organizados numa hierarquia lógica e estruturada, fornecendo assim um método para os identificar de forma uniforme. A atribuição de IDs a fenótipos é especialmente importante, sendo que as bases de dados utilizadas não representam as doenças contidas nas suas relações de forma padrão, estando estas maioritariamente sob a forma de descrições ambíguas ou redundantes. Para encontrar o termo da HPO que melhor representa cada descrição de doença, foi desenvolvido um *pipeline* baseado em semelhança de *embeddings* utilizando *SentenceTransformers* e *Facebook AI Similarity Search* (FAISS). Com o objetivo de minimizar o número de Falsos Negativos obtidos por ERSD, o dataset deve ser o mais completo possível, por isso, novas relações foram inferidas a partir de i) nomes alternativos de cada ncRNA e ii) parentes hierárquicos de cada fenótipo. No final, o *dataset* contém 214,300 relações ncRNA-fenótipo associadas a artigos científicos que as suportam.

Para a construção do ncoRP, estes artigos foram obtidos (utilizando o NCBI E-utilities) e processados de modo a obter frases isoladas. Nestas frases, foi efetuado o *Entity Recognition and Linking* (ERL) de ncRNAs e fenótipos para anotar as frases que mencionem estas entidades, utilizando a ferramenta merpy. Finalmente ERSD foi aplicada para identificar as relações expressas nas frases que mencionem pelo menos um par ncRNA-fenótipo. O ncoRP contém 21,608 artigos anotados, 2,835 ncRNAs únicos, 1,118 fenótipos únicos e 35,295 relações únicas, associados a uma precisão de 0.761 e a um *F1-score* de 0.593, tendo estas métricas sido obtidas através da validação de uma porção do *corpus* por parte de dez curadores humanos. O ncoRP e o *dataset* de relações são então contribuições importantes para a resolução do problema existente relacionado com a informação sobre ncRNAs.

A modelação de linguagem é uma área do PLN, na qual o objetivo principal é obter representações-máquina de linguagem humana - os chamados Modelos de Linguagem (ML) - que sejam capazes de prever corretamente sequências de texto que respondam ou completem de forma pertinente um *input* de texto. Apesar do pico de interesse atual em IA capaz de gerar texto, os primeiros modelos de linguagem já existem desde a década de 90, e foram estes que através do desenvolvimento gradual de novas tecnologias e da capacidade de processamento dos computadores, conduziram ao paradigma atual dos Grandes Modelos de Linguagem (GLMs).

Os Modelos de Linguagem Estatísticos foram os primeiros a surgir, e consistem na aplicação de conceitos estatísticos e probabilísticos para auxílio em tarefas específicas. O aparecimento dos Modelos Neuronais de Linguagem (MNLs), baseados em Redes Neuronais, permitiu obter modelos mais complexos com maiores quantidades de dados, capazes de fazer previsões baseadas na semelhança relativa de palavras (através de representações vetoriais). Ainda neste paradigma surgiram as Redes Neuronais Recorrentes, capazes de fazer previsões utilizando o contexto (isto é, as palavras envolventes) de cada palavra. No entanto o treino de MNLs requer computações necessariamente sequenciais, e portanto, não paralelizáveis, tornando-os ineficientes para sequências grandes de texto, o que limita a sua complexidade máxima possível. Com o aparecimento da arquitetura *Transformer* e, a partir desta, da *Bidirectional Encoder Representations from Transformers* (BERT), foi possível ultrapassar este problema, permitindo o treino dos chamados Modelos de Linguagem Pré-Treinados (MLPts), com maiores

quantidades de dados. Estes modelos demonstram flexibilidade para várias tarefas, mas requerem no entanto, *fine-tuning* para tarefas específicas para alcançar um bom desempenho nas mesmas. Com os avanços graduais no poder de processamento dos computadores, foi possível treinar MLPts cada vez mais complexos e com maiores quantidades de dados capazes de alcançar resultados cada vez melhores após cada incremento. Estes Grandes Modelos de Linguagem (GMLs) demonstram compreensão avançada de linguagem e capacidades emergentes para resolver várias tarefas, mesmo sem necessitar de *fine-tuning* específico, demonstrando-se assim úteis para o tipo de problema apresentado neste projeto.

Sendo assim, GMLs demonstram-se promissores para Extração de Relações (ER), já que não requerem o treino com as grandes quantidades de dados que os atuais modelos estado-da-arte requerem. Foi desenvolvida uma metodologia para ER utilizando GMLs que alcançou um *F1-score* de 0.978. Para isso, foi desenvolvida uma *prompt* para ER pela aplicação de princípios como *in-context learning* (que consiste na inclusão de exemplos da tarefa resolvida na *prompt*) e auto-explicação *Chain-of-Thought* (que consiste em pedir ao modelo que justifique a sua resposta). Depois, uma segunda *prompt* para a filtragem de frases sem conteúdo informativo foi também desenvolvida seguindo os mesmo princípios. A metodologia acima mencionada combina então estas duas prompts para prever cada relação.

Concluíndo, a criação de um *dataset* e *corpus* de relações ncRNA-fenótipo contribuiu para a mitigação dos problemas previamente mencionados, relacionados com a informação sobre ncRNAs. A partir deste *corpus* foi então possível o desenvolvimento de uma metodologia para ER utilizando GMLs, que alcançou resultados comparavéis (ou até superiores) com métodos estado-da-arte para esta tarefa.

Palavras Chave: RNAs não codificantes, Extração de Relações, Prospeção de Texto, Supervisão à Distância, Grandes Modelos de Linguagem

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, the motivation, objectives, methods, contributions and document structure for this work are presented.

1.1 Motivation

Non-coding RNA (ncRNA) is a broad term that encompasses all RNA molecules that do not encode proteins. ncRNAs are involved in a wide range of cellular processes, such as protein synthesis, regulation of gene product expression and other biological processes [1]. Non-coding regions constitute the majority of the human genome - according to the HUGO Gene Naming Committee (HGNC[1]), there are 19,393 documented protein-coding genes and 25,641 documented non-coding genes, whereas 9,307 of them represent ncRNAs.

The most trivial ncRNAs that come to mind might be ribosomal RNA (rRNA) and amino acid transport RNA (tRNA), both directly involved in the translation step of protein synthesis [1]. Other types of ncRNA, depending on their size, can be classified as micro RNA (miRNA), and small/long non-coding RNA (sncRNA or lncRNA) [1, 2].

miRNAs and lncRNAs play important roles of cytoplasmic regulation. The former is involved in signalling for repression or degradation of messenger RNA (mRNA), while the latter interacts with various regulatory elements of transcription, such as enhancers and even miRNAs [1]. Moreover, lncRNAs also take part in other regulatory processes, such as X chromosome silencing, genomic imprinting and chromatin modification [2].

Other less prominent types of ncRNA in research literature are small nucleolar RNA (snoRNA), circular RNA (circRNA), small interfering RNA (siRNA) and PIWI-interacting RNA (piRNA) [3].

With this wide range of functions, it is not surprising that dysregulation of ncRNA expression could be a cause of disease and its aggravation. Consequently, ncRNAs can serve not only as diagnostic biomarkers but also as therapeutic agents [1, 2]. This latter application shows promise in the sense that, due to

[1] https://www.genenames.org

their specificity to certain mRNAs or proteins, ncRNAs can alter these molecules' function and structure. Thus, RNA therapies could increase the number of druggable targets, enabling and enhancing other therapies [3].

However, disproportionately to their apparent importance, ncRNAs still remain poorly studied and documented, mainly due to their sheer complexity. It is difficult to understand unique function in the midst of the complex networks and biological pathways that ncRNAs are inserted [3]. Most of the times specific biological studies are required, which are expensive and time-consuming [2]. In addition and in consequence of this, even though there are ncRNA databases there are only a few dedicated to functional annotation [2], with the majority of information on ncRNAs still being dispersed in the form of scientific research articles, thus making it essentially impossible to be aware of its full extent by manual interpretation [4, 5]. Considering all this, it is of pivotal importance to aggregate and summarize the available information in a way that enables easy comprehension of existing associations, thus avoiding wasteful repetition of studies.

Natural Language Processing (NLP) is the sub-field of Artificial Intelligence (AI) that enables computers to automatically understand human language in its many nuances, and encompasses Text Mining methods such as Information Extraction (IE) and summarization [6, 7]. One application of such techniques is in the creation of relational corpora, which are collections of sentences annotated with found associations between entities there mentioned [8]. Thus, the creation of a corpus containing ncRNA-phenotype associations could be a step towards solving the existing problems with the sparseness of information on this topic. Moreover, relational corpora can be further used to develop and train Machine Learning (ML) models to perform various NLP tasks. Large Language Model (LLM) research is currently on the rise due to their complex understanding and generation of language [9], as so, further exploration on the potential of these models for this kind of problem may be of worth.

1.2 Objectives

Considering the aforementioned problems, three objectives were defined for this work:

1. Create a ncRNA-phenotype relation dataset through the aggregation of existing ncRNA functional annotation databases;
2. Implement a text mining pipeline for the creation of a ncRNA-phenotype relational corpus;
3. Explore emerging Large Language Models as an alternative or complement to conventional text mining methods.

The first and second objectives aim to solve the sparseness of ncRNA information, through its organization and summarization in a ncRNA-phenotype dataset and corpus. In the third, LLMs are studied on their potential in Relation Extraction tasks, as an alternative or complement to traditional methods.

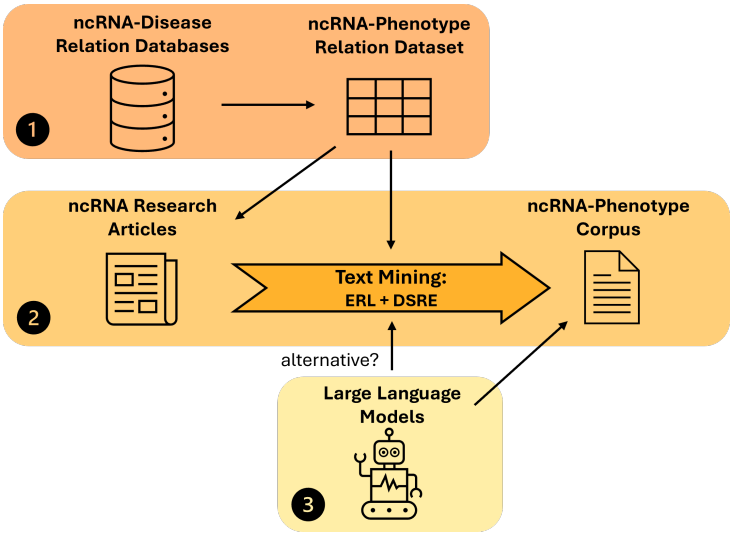

Figure 1.1: Visual representation of the project's objectives

1.3 Methodology

This work can be divided into three phases: 1) the creation of a ncRNA-phenotype relation dataset, 2) the creation of a ncRNA-phenotype corpus and 3) the development of a LLM-based Relation Extraction (RE) methodology.

In the first phase, five ncRNA-disease functional annotation databases (see 2.4.1) are processed and aggregated in a ncRNA-phenotype relation dataset. To guarantee uniformity and non-ambiguity, every relation in the datsaset is associated with a PubMed evidence article and all the ncRNAs and phenotypes are linked, respectively, to RNACentral and Human Phenotype Ontology identifiers. To perform the latter, an embedding similarity Entity Recognition and Linking pipeline is developed using Facebook AI Similarity Search (FAISS) and SentenceTransformers.

In the second phase, a text mining pipeline is implemented using Python programming language to 1) obtain scientific research articles that have information on ncRNAs, 2) extract sentences in these articles that mention both ncRNAs and clinically significant phenotypes and 3) identify relations between these mentioned entities. In the first step, the evidence articles from the ncRNA-phenotype relation dataset are downloaded and processed using NCBI's E-utilities. Then, in the second step, MER [10] is used to identify mentions to ncRNAs and phenotypes in the sentences of the articles. The last step is to perform Distant Supervision Relation Extraction (DSRE) to label the sentences as either Positive (if they express a relation between a ncRNA and phenotype) or Negative (if they do not express such a relation). In this step, the ncRNA-phenotype relation dataset is used as the source of pre-documented associations, required by DSRE methods. The corpus quality is then evaluated by distributing a random sample of annotated sentences through human curators.

In the third phase, the potential of LLMs for RE tasks is evaluated. To do that, an Ollama-based framework is implemented (using Python) to benchmark the performance of various LLMs in different scenarios. This evaluation is made by direct comparison between the LLM predictions and labelled instances, obtained from the human-curated sentences of the corpus.

1.4 Contributions

This work has essentially four contributions:

1. a ncRNA-phenotype relational corpus (ncoRP)
2. a ncRNA-phenotype relation dataset aggregating 5 functional annotation databases
3. a pipeline for the ERL of phenotypes in broad disease descriptions using embedding similarity
4. a methodology for RE using LLMs

The pipelines indicated in the first three contributions are easily adaptable to work with any other pair of entities, and the methodology indicated in the fourth contribution naturally works for any pair of entities. All the code developed for this work, ncoRP and the relation dataset are open-source and publicly available on GitHub[2].

1.5 Document Structure

This document is divided into six chapters and an appendix:

- **Chapter 1 - Introduction:** This present chapter, in which the motivation, the objectives, the employed methodologies and contributions of this work are described.
- **Chapter 2 - Related Work:** In which the key-concepts, methodologies, technologies and sources of information essential for the understanding of this work are described.
- **Chapter 3 - ncoRP: ncRNA-Phenotype Relational Corpus:** In which the pipelines for the creation of the ncRNA-phenotype relational dataset and corpus are described and evaluated.
- **Chapter 4 - Large Language Models for Relation Extraction:** In which the Large Language Model methodology for Relation Extraction is described and evaluated.
- **Chapter 5 - Case Study: Autism Spectrum Disorder:** In which it is described how the developed pipelines and methods can be applied to a specific case study.

[2] https://github.com/davidcoscor/ncRNA-AI

- **Chapter 6 - Conclusion:** In which the final discussion is made and prospects of future work are described.
- **Appendix A:** In which additional relevant information is presented.

Chapter 2

Related Work

In this chapter are presented the key-concepts and methods in Natural Language Processing (and more specifically in Text Mining and Language Modelling), essential for the understanding of this work. An overview on the used data- and knowledgebases is also given.

2.1 Natural Language Processing

Natural Language Processing (NLP) is a branch of artificial intelligence (AI) which aims to make computers understand human language. This is a task of increasing importance, in a time where the standard of scientific knowledge sharing is through written text [5]. Considering the overwhelming throughput of publications, it is impossible to manually read and organize all the available information, even for a specific topic of interest [5, 4, 11]. Developments in NLP solve this problem by enabling automatic Information Extraction (IE) and summarization [6], but also other applications such as machine translation, sentiment analysis and many others.

NLP can be divided into two main branches: Natural Language Understanding (NLU) or Linguistics and Natural Language Generation (NLG), respectively responsible for making a machine able to understand and generate language in the form of text.

- **Natural Language Understanding / Linguistics (NLU):** NLU deals with the meaning of language in all its components and degrees of complexity. From basic Morphology (responsible for prefixes, suffixes and word-level understanding, for processes known as tokenization and stemming), through Syntax (responsible for the role of words at sentence-level, with tasks such as Part-of-Speech (PoS) tagging e.g. identifying that ”jump” is a verb) and ending in complex Semantics and Pragmatics (responsible for understanding sentence and inter-sentence meaning, and to infer new information not explicitly mentioned) [6]. This is the branch of NLP responsible for text mining in large textual datasets.

- **Natural Language Generation (NLG):** NLG is the task of generating meaningful text in a human

language, based on an internal representation - model - of said language, obtained through previous NLU of large quantities of text. This process of generation is usually guided by a user input, that makes the model 1) identify the goal of the generation, 2) select the relevant information to answer the goal, 3) organize the information in a meaningful text output, 4) adjust this text to fit a particular language and tone and finally 5) output via written or voiced text [6]. This is the branch of NLP behind language modelling and text generation.

2.2 Text Mining

Text mining is a specific application of IE in the NLP of large quantities of text, with the final objective of obtaining an organized, computer-accessible representation of the information available in this text. In this context, ”information” is often thought of as a set of associations between entities, whereas an entity is an abstract representation of a concept such as ”a person” or ”a job”, or in biomedical text, ”a gene” or ”a disease” and so on.

There are various data structures that could be used to represent such associations, like i) knowledge graphs, where each association is represented by a link between two entity nodes or ii) relational corpora, which are collections of sentences annotated with the entities there mentioned and their respective association. Knowledge graphs can be used to infer new information by analysing existing links between nodes or through the development of graph machine learning models [11]. The latter, relational corpora, are essential to train Relation Extraction systems [8] and other NLP machine learning (ML) applications.

Text mining pipelines can be considered as divided into sub-tasks, usually those being Entity Recognition and Linking (which itself is composed of Named-Entity Recognition and Named-Entity Linking) and Relation Extraction.

2.2.1 Entity Recognition and Linking

Entity Recognition and Linking (ERL) is often the first step in NLP and text mining pipelines, and consists in the joint process of identifying mentions to named entities (NEs) in text and subsequently linking them to a knowledge-base entry [10]. These two sub-tasks are respectively called Named-Entity Recognition (NER) and Named-Entity Linking (NEL).

- **Named-Entity Recognition (NER):** NER consists in identifying mentions to NEs in text [10, 12]. There are various methods to perform NER, such as i) rule-based methods, ii) ML approaches (supervised, unsupervised and deep learning) [13], which are the current state-of-the-art [10] but also iv) dictionary lookup solutions, which are simpler to implement and use, but tend to be less effective [10]. Despite of the lesser efficiency of this latter methods, they are still useful when the task at hand does not require for the most efficient - yet complicated - method, this work is an example of such use-case, in which MER [10], a minimal dictionary lookup NER (and NEL) tool was employed (see 3.1.2).

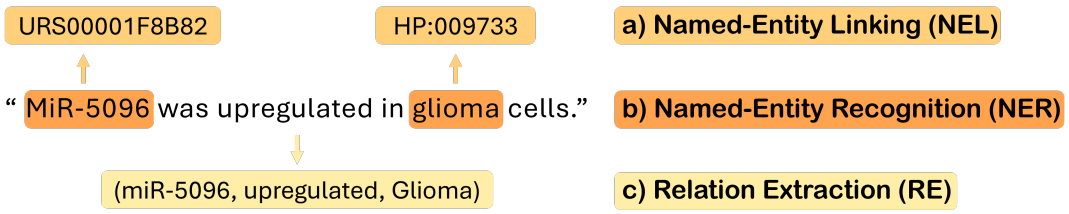

Figure 2.1: Visual representation of the Named-Entity Recognition, Named-Entity Linking and Relation Extraction tasks

- **Named-Entity Linking (NEL):** NEL, sometimes called Named-Entity Disambiguation, is the process of linking a previously recognized NE (be it from NER or other source) to an entry from a knowledge-base [10, 12, 14], e.g. the Human Phenotype Ontology for phenotype entities or the RNACentral for RNA entities. This process has the ultimate goal of avoiding redundancy, ambiguity and confusion in the final results, by guaranteeing that each recognized entity points to an unique identifier. NEL methods usually intertwine with NER and can be divided into i) rule-based, ii) graph-based, in which NE and knowledge base entries are organized in a graph to obtain the best links, iii) deep learning based [14], and also iv) dictionary-lookup based [10], in the likeness of those described for NER. These methods can be based in i) local approaches, in which entities are linked based solely on how they relate to knowledge-base entries (for example, in terms of semantic similarity) or in ii) global approaches, in which all entities in a unit of text are considered simultaneously, and impact each other's linked knowledge base entries [14].

2.2.2 Relation Extraction

Relation Extraction (RE), sometimes called relationship extraction is the process of identifying relations between NEs in text, and is an essential step in text mining and IE pipelines. In the context of RE, a relation can generally be defined as a tuple (**entity1**, **predicate**, **entity2**) [7], as exemplified in Figure 2.1. Depending on the approach, the predicate, that describes the interaction between the entities, can i) be limited to a predefined closed set of types, ii) be any verb that appears between the entities or iii) not be considered [11]. In this work, the latter approach is used, therefore the existence of a relation between two entities in a sentence simply means that they share a semantic interaction, be it positive (such as activation) or negative (such as inhibition).

Below are described the various methods employed for RE:

- **Pattern/Heuristic/Rule-based Methods:** This set of methods was the first to appear, and includes simpler approaches [4, 11] that may use hand-build patterns [7], semantic similarity, grammar and syntax to infer relations [11], namely with the use of Parse Trees [15] (hierarchical representations of the syntactic structure of a sentence, where leaf nodes represent words e.g. ”dog” or ”jumps”,

linking to inner nodes representing syntactic word categories e.g. ”noun” or ”verb”). These methods often rely on pre-made dictionaries of relation trigger-words (e.g. ”involved”), mode-of-action cues (e.g. ”activation”) and negation/speculation cues (e.g. ”not” or ”may”) [11].

- **Supervised Learning:** Supervised ML methods use a large amount of training data, such as corpora of annotated sentences, to train classifiers able to identify relations in other, previously not seen, sentences. There are two main types of supervised methods: i) kernel-based methods and ii) feature-based methods. In kernel-based methods, the training instances are transformed into a higher-dimensional space by a kernel function, and their pairwise similarities are computed and organized in a matrix, then an algorithm (usually a Support Vector Machine) finds a hyperplane that separates positive and negative instances, which then enables to predict on which side of this hyperplane a test instance would fit. On the other hand, in feature-based methods, each training instance is represented as a vector of features (such as PoS tags, parse trees or context words), then an algorithm learns the nuances on how these features affect the instance labels to then predict a test instance’s label.

- **Semi-supervised and Unsupervised Learning:** In semi-supervised learning, the labelled training data is generated automatically in an self-feeding iterative way, what is called bootstrapping, thus solving the necessity for large quantities of training data that exists in supervised learning. These algorithms use unlabelled data, which is readily available, to learn relation extraction patterns, requiring only a small amount of labelled data [7]. Unsupervised learning methods do not require any training data and often employ clustering algorithms, such as K-means or Hierarchical Clustering, to create groups of instances, which may offer insight on various aspects where they differ, such as their semantic relation type [7, 15].

- **Distant Supervision:** Distant supervision RE (DSRE) is based on the simple application of the following principle: If a relation between A and B is pre-documented in a knowledge base then a sentence that mentions both A and B expresses that relation. This results in a computationally fast method that does not require manually labelled training data. However these methods face two major problems: i) the defined principle does not apply to all sentences, because even if two entities are known to be related, a sentence that mentions these entities may not express that relation, thus resulting in false positive labels [11, 16] and ii) if a relation between two entities is not pre-documented, a sentence expressing a relation between them will always be labelled as negative, resulting in false negative labels. However these methods are still powerful as they are capable of creating a large amount of annotated instances that may be used in further development of complementary approaches. In this work, DSRE is applied in the creation of a ncRNA-phenotype relational corpus (see 3) that is then used in the development of a Large Language Model (LLM)-based RE methodology (see 4).

- **Deep Learning:** A perceptron is a basic unit of processing that, based on an input and in a set of

internal parameters (called weights) can either output a positive or negative signal. By creating a network of perceptrons, chained in multiple layers - a Multi-Layer Perceptron (MLP) - conventionally called Neural Network (because of the perceptron's resemblance to the biological neuron), it is possible to obtain an algorithm able to learn and store information from training data. The term "Deep Learning" originates from the use of Deep Neural Networks (DNNs), which include many layers of perceptrons, making them capable of learning complex tasks namely in NLP [7]. Various implementations of DNNs, such as Convolutional Neural Networks (which apply spatial filters to input data), Recurrent Neural Networks (which sequentially repeat learning operations), and Long short-term Memory (LSTM) (which can learn from larger sequences) have shown capacity in language modelling and NLP [17]. Then appeared the Transformer, based solely on self-attention mechanisms, which made the training more efficient and thus able to process larger quantities of input textual data [18]. From this architecture, appeared the Bidirectional Encoder Representations from Transformers (BERT) [19] models, whose internal representations encode text that comes both before and after each word, and upon being pre-trained with very large text datasets, show capabilities in general tasks and can be further trained on new, task/topic-specific data to learn its nuances - what is called fine-tuning. These models, specially topic-specific variations such as BioBERT and SciBERT, are the current state-of-the-art RE approaches [4, 7].

2.3 Language Modelling

Language modelling (LM) is a sub-task in NLP (more specifically in NLG) and a branch of ML in which the main goal is to obtain computer representations of language - Language Models - capable of effectively predicting word sequences that would meaningfully complete or answer an input text query [9]. Despite of the current surge of interest in text-generative AI, early language models have been used and studied in NLP at least since the 1990s, and with the gradual improvements in computer processing power and development of new architectures capable of leveraging increasingly more data, enabled the Large Language Model (LLM) paradigm we observe today.

Below are described the Language Model architectures that preceded and led to LLMs:

1. **Statistical Language Models (SLMs):** The first language models to appear were based on the application of statistical and probabilistic assumptions such as the Markov assumption (that says the future state can be simply determined by the current state), that led to n-gram Models (the previous n words are used to predict the next word) [9], then Hidden Markov Models (HMMs) and Maximum Entropy Markov Models (MEMMs) (predictions based on underlying "hidden" states) and finally Conditional Random Fields (CRFs), all with applications in NLP tasks such as IE and (rule-based) RE methods [15].

2. **Neural Language Models (NLMs):** The application of NNs to LM made it possible to outperform SLMs by enabling the training of more complex models on more data [20]. The first models of

this kind (such as word2vec) used single-layer shallow NNs to encode vector representations of words [9, 17] - what are called embeddings - which enabled predictions based on word similarity and still have applications in NLP tasks such as ERL (as was applied in this work, see 3.1.1). Then RNNs, enabled the first contextual word representations [17], which enabled better predictions based on the context (i.e. preceding and proceeding words) surrounding each word. However, these early NLMs architectures' training is necessarily sequential and thus unparallelizable, making it unfeasible to work with large sequences, limiting the performance [18].

3. **Pre-trained Language Models (PtLMs):** With the advent of the Transformer architecture, and then BERT, it was possible to overcome the task-specific limitations of the existing NLMs. The training of these models is naturally parellelizable as it does not require sequential computations, relying entirely on self-attention (i.e. the representation of a sequence is constructed based only on different tokens of this same sequence) [18]. This enables the pre-training of complex task-agnostic models, very effective in various general NLP tasks. However, to achieve high performance in specific tasks, there is still a need of task-specific fine-tuning with large task-specific example datasets [9, 17].

2.3.1 Large Language Models

Throughout the years, as PtLMs were subsequently trained with more data and bigger parameter sizes, better performance was achieved at each increment [17]. These Large (pre-trained) Language Models (LLMs) began showing strong understanding of natural language and emergent complex general task-solving capabilities, namely i) instruction following, ii) in-context learning and iii) step-by-step reasoning [9]. This without requiring any task-specific fine-tuning, making them powerful for NLP tasks (such as RE) when in comparison to other data intensive state-of-the-art methods. However, it is of importance to mention that the exact influence of both model and data sizes in performance and emergent capabilities are still largely unstudied.

The main way of interacting with a LLM is through a prompt [9] - a textual stimulus that guides the model's text generation. Prompts can be questions, instructions or examples and their relative quality can have a very significant impact on the quality of the output. In general, the aspects that make a prompt effective can be specific for a model and/or task, making prompt design a difficult, often trial-and-error effort [21]. Nonetheless, there are some prompting principles that are generally accepted as being effective, some known to elicit the aforementioned emergent capabilities of LLMs:

- **Clearness of Instructions:** A good basic principle is to write non-ambiguous clear task instructions, where key-points such as i) the task goal, ii) input data format and iii) output format and restraints should be provided. In some LLMs, the use of specific formatting or delimiters (such as "###" to identify subsections) in the prompt can also boost performance [9]. However, the sole application of these general principles may not be enough, specifically for complex tasks that may

be difficult to understand i) without simplifying them into multiple smaller steps [22] or ii) without providing prior examples [17]. To help in these cases, principles like Chain-of-Thought (CoT) and In-Context Learning (ICL) prompting should be applied, respectively.

- **In-Context Learning (ICL):** ICL, also called few-shot learning, consists in evoking language understanding capabilities gained at the training stage during generation, by including input-output examples ("shots") in the prompt. In general, the effectiveness of ICL increases with the number of shots and model size, but with diminishing returns [17]. Other factors such as the use of specific examples, their ordering [23] and their origin (model-generated or human-generated) may also influence performance.
- **Chain-of-Thought (CoT):** CoT prompting consists in splitting an initially complex instruction into a set of smaller simpler tasks. This has proven effective especially on arithmetic reasoning tasks, but also on common-sense reasoning tasks [22]. One example of CoT is prompting a model to self-explain its reasoning for a certain task alongside the response to said task (effectively splitting the task into two steps). This method has shown to improve performance and reduce output noise [24].

2.4 Data Sources

In this section, it is given an overview on the knowledge- and databases used in this work. More specifically, in 2.4.1, the ncRNA-disease relation databases that were aggregated in the creation of the ncRNA-phenotype relation dataset (see 3.1.1).

- **Hugo Gene Nomenclature Committee (HGNC)[1]:** The HGNC is the responsible authority for approving human gene symbols and names of human loci, including protein coding genes, ncRNA genes and pseudo-genes [25]. In this work, HGNC was used to find alternative and previous names for ncRNAs documented in the databases described in 2.4.1.
- **Human Phenotype Ontology (HPO)[2]:** Generally, an ontology can be described as an objective structured representation of a group of concepts and the relations they share among them [26]. The HPO is a globally recognized standard ontology of human phenotypes, defined and organized in a logically structured hierarchy. It is readily computer-available and has applications in many life, medical and computer science fields such as gene-disease discovery and ML [27]. Currently, the HPO contains over 13,000 unique terms, arranged in a directed acyclic graph, in which each term is a more specific occurrence of its parent, and is associated with an unique identifier. This is especially useful in the context of this work as it provides an uniform way of labelling and referring to phenotypes, namely in the created ncRNA-phenotype relation dataset (3.1.1) and corpus (3.1.2).

[1] https://www.genenames.org
[2] https://hpo.jax.org

- **RNACentral**[3]**:** RNACentral is a comprehensive database of over 18 million ncRNA sequences, accommodating 44 different expert ncRNA databases, while offering a wide range of annotations to other resources such as Gene Ontology or Rfam. Thus, RNACentral aims to unify the available information about each unique ncRNA sequence under unique identifiers, enabling applications in a wide range of areas [28]. In the context of this work, RNACentral's unique identifiers were used to refer to ncRNAs in the created ncRNA-phenotype relation dataset (3.1.1) and corpus (3.1.2).

2.4.1 ncRNA-Disease Relational Databases

In this section are described the ncRNA-disease relational databases and resources that were aggregated in the creation of a ncRNA-phenotype relation dataset (see 3.1.1).

- **Human microRNA Disease Database (HMDD)**[4]**:** MiRNAs represent a class of ncRNAs with known involvement in gene post-transcriptional repression, thus representing important disease bio-markers [1, 29]. HMDD provides a web-based resource that enables the download and analysis of a dataset that contains 53,530 miRNA-disease associations extracted and curated from 51,215 scientific research articles. This was done by firstly recognizing miRNA and disease entities in articles using BERN2, a multi-task learning and language model based tool, followed by a manual curation on the existence relations between each miRNA-disease candidate pair [29].

- **lncRNA-Disease Database**[5]**:** LncRNAs are a class of ncRNA with a wide range of known function, mainly sprouting from chromatin, transcript and miRNA interactions. LncRNA dysfunction thus unsurprisingly lead to various diseases [1, 30]. The lncRNA-Disease Database includes 25,440 experimentally supported lncRNA-disease associations, covering 6,066 unique lncRNAs and 10,732 unique circRNAs found to be related to 566 unique diseases in 11,840 evidence scientific research articles. These results were obtained trough manual curation of an initial set of articles obtained through keyword search [30].

- **ncrPheno**[6]**:** ncrPheno is a comprehensive database containing 482,751 ncRNA-disease associations, that compiles information from 15 different databases (compiling 50,681 associations) as well as additional associations (432,070) inferred from disease parent-child relations from the Experimental Factor Ontology. ncRPheno covers associations for 14,494 unique ncRNAs, including miRNAs, lncRNAs, circRNAs, piRNAs and snoRNAs and for 3,210 disease phenotypes [31].

- **RIscoper**[7]**:** RIscoper is in fact a text mining tool for the recognition and extraction of relations involving RNAs from literature. However, the authors of RIscoper make available a corpus of

[3] https://rnacentral.org
[4] http://www.cuilab.cn/hmdd
[5] http://www.rnanut.net/lncrnadisease/
[6] http://liwzlab.ifr.fidt.top:61010/ncrpheno/ncrpheno.html
[7] http://www.rnainter.org/riscoper

Table 2.1: Results from a general binary classification task

Prediction	Ground Truth	Classification
Positive	Positive	TP
	Negative	FP
Negative	Positive	FN
	Negative	TN

14,292 RNA-disease associations extracted from 15,581 scientific research articles, through manual curation [32].

- **RNA-Disease[8]:** RNA-Disease is a database that contains over 3 million RNA-disease associations, extracted from RNA sequencing data, computationally predicted data and experimentally validated data. In the context of this work, only associations from the latter were considered (a total of 343,273 associations), that were obtained through manual validation of over 40,000 scientific research articles and integration with 23 other experimentally validated databases. Additionally, for uniformity the diseases were mapped to Disease Ontology (DO), Medical Subject Headings (MeSH) and Kyoto Encyclopedia of Genes and Genomes (KEGG) identifiers [33].

2.5 Evaluation Metrics

In this section are presented the evaluation metrics used throughout this work. For the evaluation of classification-like tasks (such as RE or ERL), Precision, Recall and F1-score were used, while in the evaluation of inter-curator agreement in the validation of the created ncRNA-phenotype corpus (see 3.2), Fleiss' Kappa was used.

- **Precision, Recall and F1-score:**

  The conventional way of evaluating classification tasks (such as RE), is to subject the classification algorithm to a ground-truth test set of instances with known labels. If the classification algorithm i) has the capacity of learning from instances, as is the case in ML models or ii) can be parameter-tuned, then this ground-truth test set should be hidden from the algorithm in this training / tuning stage.

  To evaluate the algorithm's performance in correctly predicting the labels for the test set instances, these predicted labels are compared to the known ground-truth test labels. By doing this, the number of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) classifications are counted, following the logic described in Table 2.1.

[8] http://www.rnadisease.org

To further weigh and interpret these results, three common metrics [7] are usually computed: 1) Precision (Eq. 2.1) gives the proportion of correctly labelled positives among positives classified by the algorithm; 2) Recall (Eq. 2.2) gives the proportion of correctly labelled positives among actual positives and 3) F1-score (Eq. 2.3) is the harmonic mean of Precision and Recall, giving the overall performance of the algorithm. All these metrics can range from 0 to 1, with values closer to 1 meaning a better performance of the classification algorithm, i.e. that it was generally able to produce more TP and TN compared to FP and FN.

$$Precision = \frac{TP}{TP + FP} \tag{2.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.2}$$

$$F1\text{-}score = \frac{2.Precision.Recall}{Precision + Recall} \tag{2.3}$$

- **Fleiss' Kappa:**

To evaluate the inter-curator agreement in relation to a task, the Fleiss' Kappa [34] - κ (Eq. 2.4) may be used. The Fleiss' Kappa measures how much the curators agree ($\bar{P}$) in relation to the agreement by chance (P_e). Also note that in Eq. 2.4, P_i represents the agreement of curators in relation to the i subject (N is the total number of subjects), and that p_j represents the number of subjects considered to belong to the j evaluation category by the curators (k is the total number of categories). The Fleiss' Kappa can range from -1 to 1, with i) negative values meaning the agreement was lower than expected by chance, ii) values equal to 0 meaning the agreement was the expected by chance and iii) positive values meaning the agreement was higher than expected by chance. In the context of this work, it was used to evaluate the inter-curator agreement in relation to how well the sentences of the ncRNA-phenotype corpus were labelled (see 3.2)

$$\bar{P} = \frac{1}{N}\sum_{i=1}^{N} P_i \ , \qquad P_e = \sum_{j=1}^{k} p_j^2 \ , \qquad \kappa = \frac{\bar{P} - P_e}{1 - P_e} \tag{2.4}$$

Chapter 3

ncoRP: ncRNA-Phenotype Relational Corpus

In general, a corpus consists of a large collection of topic-specific annotated text, organized in a computer-accessible format. These can have various applications, such as search engines and the training of ML models. In this work, a ncRNA-phenotype relational corpus - ncoRP - is produced in order to summarize the existing information about these important gene products, which comes mostly in the form of scientific research articles. The main goal is then to find sentences in literature that mention both a ncRNA and a human phenotype, and to classify them on the expression of any semantic relation between them.

To extract these relations, DSRE will be employed, which requires a set of known ncRNA-phenotype relations. Thus, the first step is to create a ncRNA-phenotype relation dataset, through the aggregation of existing databases.

By nature, distant supervision methods tend to create i) positive label noise, as the presence of two related entities in a sentence does not mean they share a relation in that sentence and ii) unbalanced negative-leaning labels, as positives require a relation to be pre-documented [16, 11]. To address the first mentioned problem, a sentence is only labelled as *Positive* if the relation is documented for the article from which this sentence originates, thus increasing the confidence of that sentence actually expressing a relation between the entities. And, to address the second problem, the number of pre-documented relations is maximized by the propagation of relations to both ncRNA aliases and phenotype ancestors.

3.1 Methods

Considering what was aforementioned, Python was used to implement a text mining pipeline for the creation of the ncoRP corpus, composed of the following four steps:

1. Generate a ncRNA-phenotype relation dataset

2. Obtain and process ncRNA scientific research articles
3. ERL of ncRNAs and phenotypes in article sentences
4. Distant supervision RE of ncRNA-phenotype relations

For eventual reproduction purposes, all input files were last updated in the 15th of May of 2024, to their current version at the time (download URLs available in the source code[1]).

3.1.1 Relation Dataset Creation

Each of the selected databases (2.4.1) represents ncRNA-Disease relations in a different format. Thus, to utilize the available information, it needs to be uniformly organized. To do this, a relation dataset was created in the form of a table with six columns: 1) RNACentral ID, 2) ncRNA name, 3) HPO ID, 4) Disease/Phenotype name/description, 5) Evidence Article PMIDs and 6) Database of Origin. Each row in the dataset can thus be treated as a relation between a ncRNA and a Phenotype.

As so, the creation of the relation dataset consisted of six steps:

1. **Process and merge databases:**

   In this step, each of the five databases were processed to fit the format of the relation dataset. While this step was relatively easy for ncrPheno, HMDD, lncRNA-Disease Database and RNADisease, it proved more challenging for RIscoper, due to the raw NER format of the files. HMDD, lncRNA-Disease Database and RNADisease relations were filtered to include only human entries. RNADisease is the only database that provides identifiers (from DO, MeSH and KEGG) for the diseases, as such, these IDs were linked to their HPO counterparts. Therefore, RNADisease relations were not processed in step 2.

2. **ERL of phenotypes in disease names/descriptions:**

   The selected databases (with the exception of RNADisease) do not provide an uniform representation of phenotypes, often simply using disease names, abbreviations or acronyms. As so, ERL of phenotypes solves this problem by i) assigning IDs to redundant descriptions (i.e. ”gastric cancer” and ”stomach neoplasm”) and ii) rejecting uninformative or ambiguous descriptions.

   To do this, a pipeline using SentenceTransformers[2] and Facebook AI Similarity Search (FAISS)[3] was implemented. The first, SentenceTransformers, is a Python module that enables the training and usage of text (and image) embedding BERT-based models, offering a collection of already pre-trained and ready-to-use models [35]. The second, FAISS, is a library (with a Python wrapper) that

[1] https://github.com/davidcoscor/ncRNA-AI
[2] https://www.sbert.net
[3] https://ai.meta.com/tools/faiss/

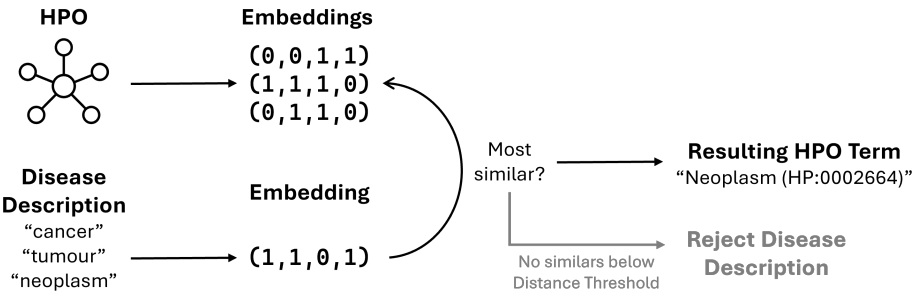

Figure 3.1: Visual representation of the Entity Recognition and Linking pipeline for phenotypes in disease descriptions

enables efficient similarity search of vectors (as are embeddings), using different distance metrics such as Euclidean, dot product and Cosine [36].

Therefore by using SentenceTransformers to create a collection of embeddings from the HPO in which each term name (and respective synonyms) is converted to an embedding associated with its HPO ID, it is possible to find the HPO term embedding most similar to an embedded disease description using FAISS. A distance threshold can also be set in order to filter descriptions that may be too dissimilar from any HPO term. The SentenceTransformers model used was "all-MiniLM-L6-v2", and a maximum euclidean distance threshold of 0.5 was imposed.

This method was manually evaluated in a random sample of 900 unique disease descriptions (from which 450 linked to HPO IDs), representing about 10% of the 8772 total unique disease descriptions, yielding a precision of 0.973, a recall of 0.883 and a F1-score of 0.926 and resulting in 4954 unique disease descriptions linked with HPO terms. More details regarding this evaluation are described in A.1.

3. **Propagate relations to ncRNA aliases:**

   To maximize the size of the dataset, and so, the number of positive labelled sentences, the existing relations of a ncRNA were propagated to all its documented alias names. To obtain the aliases for each ncRNA, the custom downloads tool of HGNC[4] was used to download a file containing the "Approved symbol", "Previous symbols" and "Alias symbols", filtering to obtain only entries representing the "non-coding RNA" locus group. Then, for each relation, if the ncRNA had aliases (i.e. "Previous symbols" and/or "Alias symbols"), the relation was propagated to each of them.

4. **NEL of ncRNAs:**

   In order to ensure that all ncRNAs names registered in the dataset represent actual ncRNA molecules, ncRNAs were linked to RNACentral IDs. The IDs were assigned through rule-based matching of

[4] https://www.genenames.org/download/custom/

Table 3.1: Statistics of the Relation Dataset

Statistic		Number
Relations	Total	214300
	Original	47798
	From phenotype ancestor propagation	166502
	Unique ncRNAs	4891
	Unique phenotypes	2413
	Unique articles	53427

the ncRNA name to the RNACentral mappings file. NcRNAs that failed to link to an ID were excluded from the dataset.

5. **Propagate relations to phenotype ancestors:**

   Again, to maximize the number of positive labels, the relations involving a phenotype were propagated to all the HPO ancestors of said phenotype. For example, relations involving ”Stomach cancer”, are propagated to ”Neoplasm of the stomach” and to all its other HPO ancestors until ”Phenotypic abnormality”, exclusively. This cut-off is applied because i) the term ”Phenotypic abnormality” is too general to have medical significance and ii) every original relation would be propagated to it, making it redundant.

6. **Removal of duplicates:**

   The last step consisted in removing all the duplicate rows. In the end, the relation dataset contains a total of 214,300 unique ncRNA-phenotype relations, involving 4,891 unique ncRNAs and 2,413 unique phenotypes. The statistics of the resulting relation dataset are summarized in Table 3.1.

3.1.2 Corpus Creation

The text mining pipeline for the creation of the corpus (represented in Figure 3.3) can be divided into three steps:

1. **Article Download and Processing:**

   As aforementioned, the relation dataset contains a column with all the evidence article PMIDs associated with each relation, as so, these articles are a good source of ncRNA informative text. In works of this kind, usually only the article abstracts are considered (as in [37]). However, in this work, ideally the whole text of the articles was mined for relations to maximize the amount of annotated sentences, because not all the possible relations mentioned in an article will be in the abstract.

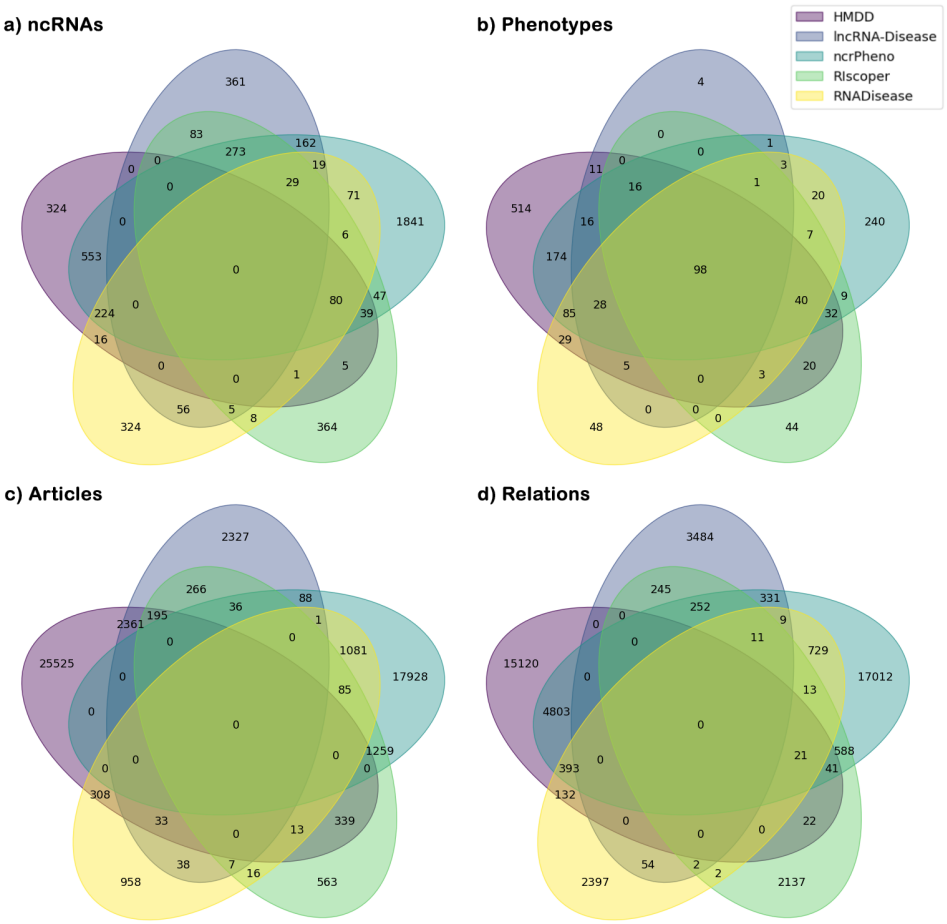

Figure 3.2: Overlap of the different databases in terms of a) unique ncRNAs, b) unique phenotypes, c) unique research articles and d) unique ncRNA-phenotype relations in the Relation Dataset

The first step was to use Biopython[5]'s NCBI E-utilities[6] wrapper to i) convert PMIDs into PubMed Central IDs (PMC IDs) and ii) download the articles in XML format. This format was the only found to enable the download of full-text articles from the PMC Open-Acess Subset. However, when a full-text version of an article was not available, the abstract was downloaded. A great number of articles could not be downloaded, either by not being available on PMC or other unknown reasons. By the end of this step, out of a total of 53,427 unique articles, 29,281 were successfully downloaded (23,869 full-text and 5,412 abstracts).

Then, ElementTree[7] was used to implement an XML parser to extract the text paragraphs of each article. Finally these paragraphs, if longer than 500 characters, were split into sentences of at least 50 characters, by cutting the text at each ". " (a period followed by a white space). These minimum paragraph and sentence sizes were established in order to avoid scientifically uninformative text (i.e. author names, acknowledgment section text, etc). The sentences were then post-processed to remove or replace characters that could cause eventual errors such as mid-sentence line change characters ("\n").

2. **ERL of ncRNAs and phenotypes in sentences:**

   MER [10] is a minimal dictionary lookup named-entity recognition and linking tool, that requires only a small set of text files (called a lexicon), containing the names and IDs of entities. Merpy[8] is its Python interface, enabling easy integration with the rest of the pipeline and offering lexicon managing functions. A small modification[9] had to be made for Merpy to work with entity names containing dashes ("-"), as is the case of ncRNAs.

   The first step was to create the ncRNA and phenotype lexicons. For the former, the ncRNA names and IDs were obtained from the RNACentral mappings file, on which a filter was applied to remove generic names such as "non-coding RNA" and names that would cause the wrong recognition of small words or letters, such as "TR" and "E". For the latter, the HPO terms and the synonym names of each term were added to the lexicon. Only the HPO terms descendant of "Phenotypic abnormality" were considered, as done in [37], in order to exclude terms with no clinical significance such as "localized" and "frequency".

   Then for each sentence, Merpy was executed twice, to recognize both the ncRNAs and phenotypes there mentioned, with sentences not mentioning at least one ncRNA-phenotype pair being rejected.

3. **Distant Supervision RE of ncRNA-phenotype relations:**

   The final stage in the creation of the corpus was the DSRE. To do this, for each sentence, if a

[5] https://biopython.org
[6] https://www.ncbi.nlm.nih.gov/books/NBK25501/
[7] https://docs.python.org/3/library/xml.etree.elementtree.html
[8] https://github.com/lasigeBioTM/merpy
[9] https://github.com/davidcoscor/merpy-ncrnas/tree/entity-name-dash-support

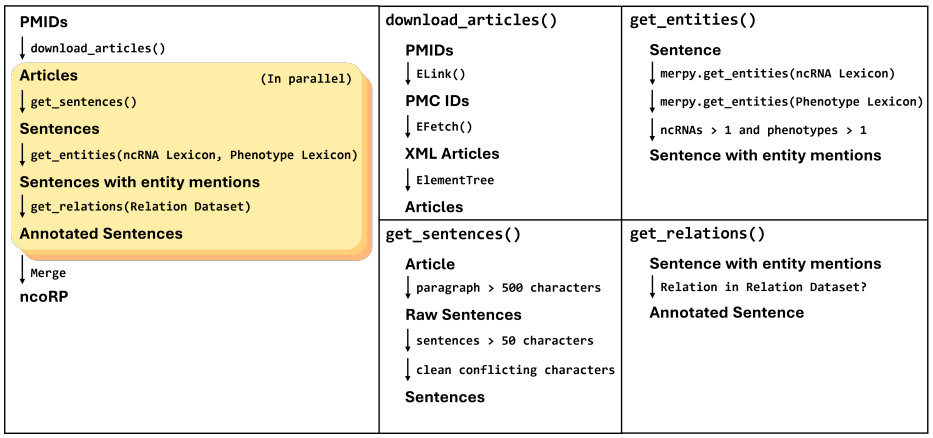

Figure 3.3: Visual representation of the implementation of the ncoRP creation pipeline

Table 3.2: Statistics for ncoRP

Statistic		Number
Annotated sentences		208539
Annotations	Total	400557
	Positives	84409
	Negatives	316148
Unique Relations	Total	35295
	Positives	3615
	Negatives	34147
Unique ncRNAs		2835
Unique phenotypes		1118
Annotated articles		21608

row reflecting the relation between the ncRNA-phenotype pair, for the article where the sentence originates was found in the relation dataset, then the sentence was labelled as a *Positive*, otherwise, it was labelled as a *Negative*.

The ncoRP corpus is available in two formats: i) a collection of JSON files, one for each annotated article, containing its annotated sentences and ii) a CSV file containing all the annotated sentences. Both ncoRP and the relation dataset were made publicly available[10]. Table 3.2 summarizes the statistics of ncoRP.

[10] https://github.com/davidcoscor/ncRNA-AI

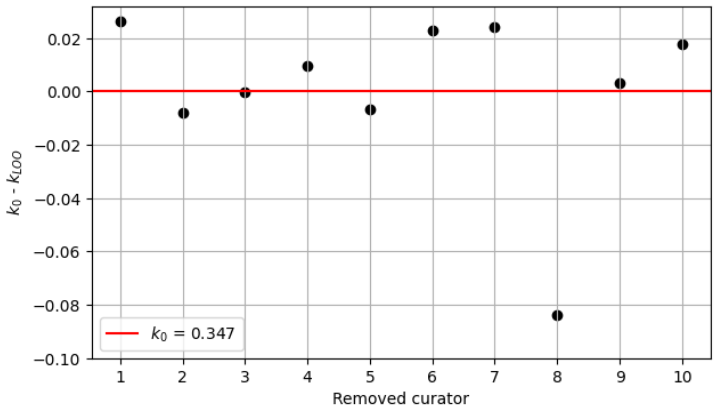

Figure 3.4: Differences between the baseline Fleiss' Kappa and each Leave-One-Out Fleiss' Kappa

3.2 Corpus Validation

To be aware of the general quality of ncoRP, a human validation was performed. To do that, random samples of 40 sentences (20 positives and 20 negatives) each were distributed among expert curators to be evaluated. In order to evaluate the fidelity of the curators' validation, out of the 40 sentences sent to each curator, 20 overlap sentences were the same for every curator. A total of 200 unique sentences were validated by 10 curators.

For each sentence, the curator had to choose one of three possible evaluations: *Correct* (when the sentence label correctly reflected the relation between the entities); *Incorrect* (on the contrary); or *Uncertain* (when due to ambiguity, error or any other reason, it was hard to attribute one of the other two evaluations).

The answers to the overlap sentences were used to calculate and analyse the Fleiss' Kappa measure in two different methods. Firstly, the baseline Fleiss' Kappa ($\kappa_\mathbf{0}$) was calculated, considering all the curators' answers, then Leave-One-Out Fleiss' Kappas ($\kappa_\mathbf{LOO}$) were calculated, each representing the Fleiss' Kappa if each curator was not considered. Figure 3.4 shows the differences between $\kappa_\mathbf{0}$ and each $\kappa_\mathbf{LOO}$, effectively representing the impact of each curator on $\kappa_\mathbf{0}$, and how much they differ from their peers. It is clear that curator 8 may be an outlier, and by excluding them, the final Fleiss' Kappa is 0.431 which represents a moderate inter-curator agreement.

Considering the curators' evaluated sentences (180 in total, excluding curator 8), a precision of 0.761, a recall of 0.468 and an F1-score of 0.593 were obtained for the corpus. These results are summarized in Table 3.3.

Table 3.3: Evaluation results and metrics for ncoRP

Evaluations					**Metrics**				
TP	FP	FN	TN	Uncertain	Precision	Recall	F1-score	Uncertain %	Fleiss' Kappa
54	17	57	14	38	0.761	0.468	0.593	21.1	0.431

3.3 Discussion

The obtained ncRNA-phenotype relation dataset successfully aggregates the five selected databases (described in 2.4.1) in an uniform format. On Figure 3.2, it is visible that there is small overlap between the databases, meaning that the inclusion of every database is pertinent. The attribution of unique identifiers to both ncRNAs and phenotypes guarantees the quality and non-ambiguity of the relations by forcing the rejection of wrongly registered, redundant or uninformative entries from the original databases. In Table 3.1, it is noticeable that the relations resulting from propagation to phenotype ancestors represent a great majority, meaning that this step significantly contributed to the completion of the dataset. However, in the ERL of phenotypes in disease descriptions, despite the high precision (0.973) of the method, some HPO Terms were wrongly attributed (statistically, about 3%), with this error being further amplified in the ancestor propagation. To mitigate this, a more restrictive smaller distance threshold could have been used, but in trade-off with a smaller number of disease descriptions with linked HPO Terms, and thus, relations. In perspective (referring to Table A.2), an euclidean distance threshold of 0.1, the only evaluated with a precision of 1.000, resulted in only 1,428 links, while the chosen euclidean distance threshold of 0.5 resulted in 4,954.

The produced ncRNA-phenotype corpus - ncoRP - contains more than 200,000 annotated sentences from a total of 21,608 scientific research articles concerning ncRNAs, with acceptable metrics (precision of 0.761 and F1-score of 0.593) considering the used method. However, based on the moderate inter-curator agreement (Fleiss' Kappa of 0.431) and the fact that only about 0.1% of the corpus was evaluated, these metrics may not represent it fully, but are still a good representation. In the corpus, the great majority of sentences were labelled as *Negative*, as was expected with DSRE, which was also aggravated by the imposed article-specific restriction. Thus, by lifting this restriction, more *Positive* sentences could have been obtained, but possibly resulting in more FP instances. The obtained high FP and FN counts are mostly caused by the aforementioned problems inherent to DSRE and the high number of *Uncertain* sentences (21.1%) are caused by i) the method used to individualize sentences (resulting in some getting too long and complex to understand), ii) errors in the ERL of entities in the sentences (despite the filters used, some words were wrongly identified as entities that share the same name) and iii) that some sentences are in fact inherently uncertain in their meaning. Furthermore, this high number of *Uncertain* sentences, may have also hindered the observed inter-curator agreement.

In the next chapter, it is described how the validated subset of ncoRP was used to implement a LLM-based RE methodology aiming to solve the mentioned problems, which could be used as an alternative

to DSRE to produce an improved version of ncoRP.

Chapter 4

Large Language Models for Relation Extraction

Currently, the best performing state-of-the-art RE methods are mostly based on pre-trained language models (PtLMs). However, the performance of these models on specific tasks is very dependant on their fine-tuning using large amounts of pre-labelled training data. Recently, Large Language Models (LLMs) - large PtLMs trained on very large datasets - have shown great capabilities in general task-solving, without requiring further fine-tuning for specific tasks. As such, in this chapter LLMs are evaluated in their potential for RE, with emphasis on how the application of prompting principles can leverage even relatively small models to state-of-the-art performance.

4.1 Methods

As previously done in this work, RE was treated as a binary classification task in which i) a *Positive* is obtained if two entities share any semantic relation in a sentence and ii) a *Negative* is obtained otherwise. The main challenge in applying LLMs to RE stems from the fact that these models' output formats are unpredictable. For example, to the prompt "Is there a relation between A and B in this sentence?", there are countless possible ways a LLM could respond. Thus, it is essential to guide the LLM to respond in an uniform (enough) format, that can be automatically processed by a response parsing function. Once this was achieved, the impact of the number of in-context examples ("shots") in performance was analysed.

To evaluate LLMs for RE, an Ollama[1]-based Python framework was implemented. Ollama is an open-source framework, available (not exclusively) as a Python library, that enables easy download and interaction with various pre-trained LLMs. The implemented framework wraps Ollama to ease the handling and evaluation of LLMs in binary classification tasks (such as RE), namely by enabling i) easy formatting of prompts to include in-context examples, ii) parsing responses through custom response

[1] https://ollama.com

parsing functions, iii) generation of in-context examples from LLM predictions and iv) automatic scoring and logging of LLM performance, by comparison to ground truth labels.

The validation of ncoRP revealed the existence of a major number of *Uncertain* sentences, which apparently have no information content or are too complex for it to be graspable. These sentences are thus counterproductive in corpora, resulting in noise when used as training data. With LLMs being very competent in language understanding, they could have the potential to identify and filter out these sentences automatically. As such, this ”uncertain filtering” paradigm is also analysed in this chapter.

The following LLMs, all available through Ollama, were tested: Llama3 (8B and 70B), Phi3 (3B and 14B), Mixtral (8x7B and 8x22B), Gemma (7B) and Gemma2 (9B). All these models are considered relatively small LLMs, with the exception of the larger versions of Llama3, Phi3 and Mixtral, which were included for comparison.

4.1.1 Evaluation

With the validation of ncoRP (see 3.2), by the direct analysis of the curators' evaluations, it was possible to produce a ground-truth dataset containing high confidence labels (e.g. if a *Negative* sentence was evaluated as *Incorrect*, its label is in fact *Positive*), in which *Uncertain* and outlier curator instances were not included. Then, to evaluate the performance of a model when predicting an instance, its prediction is compared to the high confidence label for that instance.

This ground-truth dataset was divided into three subsets[2]: 1) Train, used to produce examples for in-context learning; 2) Test, used to select the best performing method (combining a model, prompt and number of examples) and 3) Validation, used as a final evaluation of the method deemed the best. This was done in order to i) ensure that the iterative method design/test process was not contaminated by prediction bias from in-context examples and to ii) confirm that the performance of the found best method was not due to test instance bias.

4.1.2 Prompt Design and Response Handling

The process of designing a prompt that instructs a LLM for RE while ensuring the outputs can be automatically processed can be generally expressed as the iteration of the following steps:

1. Design a prompt that conveys the RE task and output format
2. Design a parsing function to handle the response
3. Observe and evaluate the LLM responses

[2]In truth, this process was not so linear. First, the in-context examples were generated and curated independently prior to the existence of any subsets, with the Train subset being created based on the instances present in the final examples (and the other subsets with the remaining instances). Second, the ncoRP validation was still ongoing when the work described in this chapter started being executed, which resulted in instances being added to the Validation subset after the initial split

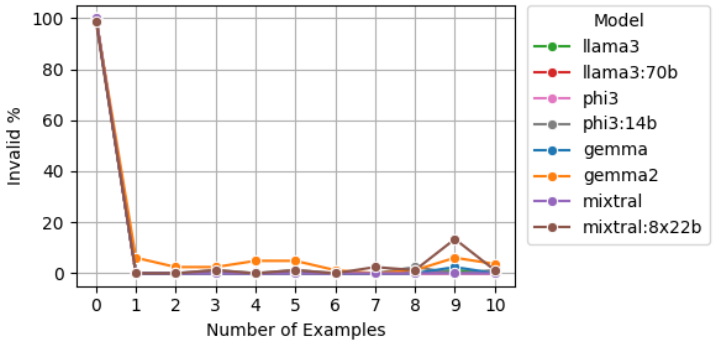

Figure 4.1: Analysis of the impact of the number of examples in the percentage of invalid LLM responses. Note: this was a preliminary analysis prior to what is described in 4.1.3, and the example set here used was not the final example set.

4. Repeat, making adjustments in 1. and 2.

In this stage, it was noticed that the inclusion of in-context examples in the prompt drastically reduced the number of invalid model responses (see Figure 4.1), making them essential. Furthermore, the self-explanation CoT principle was also included, as it has shown to increase performance, but also give insight on the reasoning behind a prediction, which assisted in the prompt design.

Once the final prompt (Figure 4.2) and parsing function combination was achieved, the impact of the number of shots on performance was further analyzed.

4.1.3 Number of Shots Analysis

The examples used in in-context learning were generated by firstly querying Llama3 with two (one *Positive* and one *Negative*) manually crafted examples, using the final prompt, to obtain a large set of model-generated responses. Then, the responses with correct predictions were manually curated into a set of ten final examples (five *Positive* and five *Negative*).

A maximum of ten shots were studied because, despite higher numbers of shots having shown to lead to better performance [17], this increase is throttled in smaller models, as are those studied in this work. Additionally, the shots were always the same and applied in *Positive-Negative* pairs (e.g. if four shots were used, the same two pairs of examples would always be used). This was done in order to obtain comparable results for different models, guaranteeing that these were not influenced by the content of the examples.

Each model and number of shots combination was evaluated in 4 duplicate runs. The results of this analysis are shown in Figure 4.4, with the best found method being 14B-Phi3 with 4 shots achieving an F1-score of 0.855. However, it is also clear that the number of shots appears to have little impact on performance. Table A.3 displays the complete results of this analysis.

a) Relation Extraction Prompt:
Identify if there is an explicit relation between "{e1}" and "{e2}" in the following sentence: "{sentence}".
You must provide an explanation for your answer. **Self-explanation CoT**
Your response should be a JSON object with two fields: "relation" and "explanation".

{examples} **In-context Learning**

Identify if there is an explicit relation between "{e1}" and "{e2}" in the following sentence: "{sentence}".

b) Uncertain Filtering Prompt:
Do you think the following sentence conveys information about the relation between "{e1}" and "{e2}" in a clear way? Sentence: "{sentence}“
You must also explain the reasoning behind your answer.
Your response should be a JSON object with two fields: “answer" and "explanation".

{examples}

Do you think the following sentence conveys information about the relation between "{e1}" and "{e2}" in a clear way? Sentence: "{sentence}“

Figure 4.2: Prompts used for a) Relation Extraction and b) filtering of uncertain sentences

4.1.4 Uncertain Sentence Filtering

Although curator-labelled *Uncertain* sentences were removed from the ground-truth dataset used on this chapter's work, it is worth to note that ”uncertain” is essentially a subjective label and so, what a curator considers uncertain, might not be uncertain for another. Then, it could also be the case that a LLM could consider uncertain a sentence that had not been previously labelled as such, thus making the prediction of a relation in that sentence harder or outright impossible, thus leading to decrease in general performance. As such, the filtering of uncertain sentences has two considerable benefits: i) removing unproductive sentences and ii) improving the general RE performance.

Then, to identify sentences a LLM would consider uncertain, the prompt presented in Figure 4.2 was designed, by making small alterations to the prompt used for RE. But note that instead of asking ”Is the sentence uncertain?”, which could still lead to conflicting results (because what is intended by ”uncertain” may not be clear), it is asked ”Does the sentence convey information well?”, which achieves the same goal whilst being clearer.

With this, the methodology presented in Figure 4.3 was constructed, which consists in querying an LLM in two separate stages: first to 1) identify if a sentence is not uncertain, and only if that is the case, 2) identify if it contains a relation between the entities.

To evaluate the performance of this methodology, an analysis comparable to what was done in 4.1.3 was executed. The in-context examples used in this stage were generated following the same previously described method, but using the prompt described in this section. However, it was observed that the prompt used to filter uncertain sentences resulted in a high number of invalid responses in some models, especially for higher numbers of shots. That being the case, these models' performances are not comparable to the rest. The results are presented in Figure 4.4, with the best found method being Phi3 with

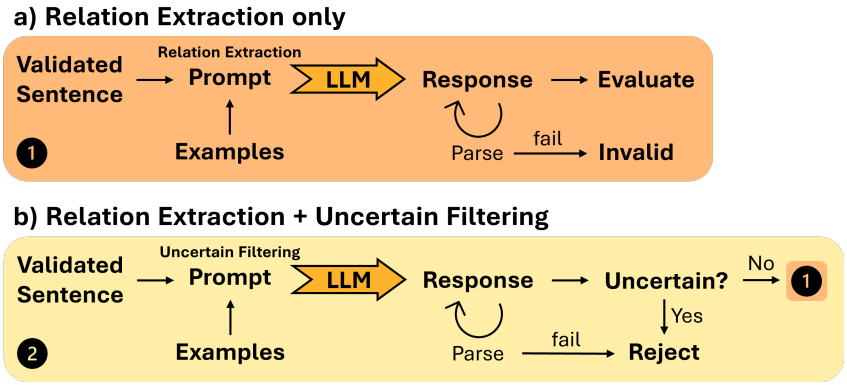

Figure 4.3: Visual representation of the Large Language Model Relation Extraction methodologies: a) Relation Extraction only and b) Relation Extraction preceded by uncertain sentence filtering

10 shots. Table A.4 displays the complete results of this analysis. This best method was then evaluated on the Validation subset, achieving an F1-score of 0.978, rejecting on average 15.5% of sentences as uncertain.

4.2 Discussion

A good method to obtain reliable uniform LLM RE responses was achieved by combining i) a prompt that instructed the model to produce a specific format (JSON), while including in-context examples with ii) a response parsing function, solving one of the main challenges of applying LLMs to close-ended tasks like RE. This enabled the automatic evaluation of responses with a ground-truth dataset (constructed from the validation results of ncoRP), which would have not been possible due to the diversity of possible responses these models normally generate. However, the employed process of prompt design could be improved as it consisted mainly of trial-and-error of different prompts, which made it difficult to comparatively evaluate their quality. As such, a more methodical approach should have been used.

The impact of in-context learning in LLM RE was studied by analysing the impact of the number of shots in performance (see Figure 4.4), with two main results arising. First, larger model sizes do not seem to increase performance, at least for the numbers of shots tested. Perhaps better performance on large models could be achieved with larger numbers of shots, that would make use of their larger context windows. Second, the number of shots does not seem to have a clear impact on performance in the RE single prompt method, but higher numbers of shots seem to lead to better performance when combining the RE and uncertain filtering prompts. A possible explanation could be that more examples improve the performance of the uncertain filtering step, leading to more uncertain sentences being filtered, thus improving the performance in the RE step by making the model predict relations only in sentences it deems understandable. The results support this explanation in the sense that the best performing method

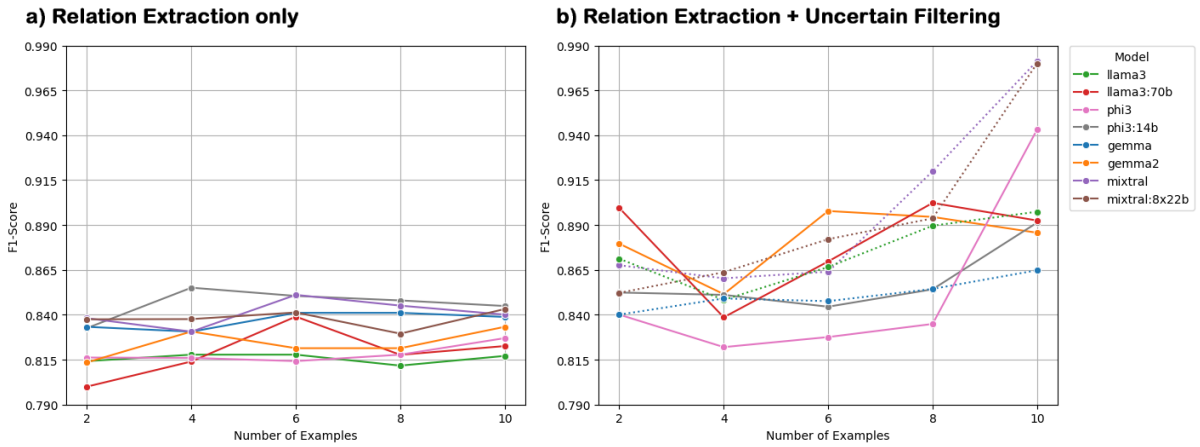

Figure 4.4: Large Language Model Relation Extraction performance analysis for different models and numbers of shots in a) a single Relation Extraction prompt and b) a Relation Extraction prompt preceded by an uncertain filtering prompt. Dotted lines represent models that resulted in more than 10% of invalid responses to the uncertain filtering prompt.

also has the most percentage of sentences being filtered as uncertain (see Table A.4). However, the obtained results could also be caused by the specific examples used and their order, and although it would be worthwhile to study different (possibly randomly selected/ordered) examples it would increase the complexity of the study beyond the scope of this work.

The best-found method combined the model Phi3 with ten shots, using the combined uncertain filtering and RE prompts, achieving an F1-score of 0.978, which outperforms RE state-of-the-art methods. However, a more thorough prompt design should have been applied for the uncertain sentence filtering prompt, as it resulted in a high number of invalid LLM responses in some models. Furthermore, the uncertain filtering prompt was only analysed in association with RE and its stand-alone performance (i.e. how good in fact it is at identifying uncertain sentences) was not fully examined as even manual analysis would be challenging due to the subjective nature of what constitutes an uncertain sentence.

To conclude, despite the simplicity of the methods presented in this chapter, based mostly on prompting principles (namely in-context learning and CoT self-explanation) applied on small LLMs, very promising results were obtained, leaving very much apparent the potential of LLMs for RE, especially considering that these models do not require pre-existing labelled training data.

Chapter 5

Case Study: Autism Spectrum Disorder

To further demonstrate the functionality of the developed pipelines and methods, in this chapter a short case study of Autism Spectrum Disorder (ASD) is performed. ASD is a brain-based neurodevelopmental disorder characterized by a heterogeneous set of behavioural hindrances in communication, social interaction and interests [38].

5.1 Methods

In the likeness of what was previously done in the creation of ncoRP (see 3.1.2), the implemented pipeline is composed of three steps:

1. Download and process articles mentioning ASD
2. ERL of ncRNAs and ASD-related phenotypes in article sentences
3. LLM RE of ncRNA-ASD relations

In the first step, to obtain scientific research articles encompassing ASD, the following query was used on PMC: *("Autism Spectrum Disorder" OR "ASD") AND "oa full text xml"[Filter]*, which yielded 47,905 results. The *Filter* statement in the query restricts the search to include only articles whose full-text is available to be downloaded in the XML format. These articles were then downloaded and processed following the same pipeline as described in 3.1.2 (skipping the PMID to PMC ID conversion step), which resulted in 47,756 articles.

Then, in the likeness of what was previously done in 3.1.2, Merpy was used for the ERL of ncRNAs and (ASD-related) phenotypes in the sentences of the articles. To do this, a new small ASD-related phenotype lexicon was created containing the HPO Terms related to ASD (and their synonyms), which are: 1) Autistic behaviour (HP:0000729), 2) Autism (HP:0000717) and 3) Autism with high cognitive abilities (HP:0000753). The ncRNA lexicon used was the same as in 3.1.2. By the end of this step, a total of 10,379 sentences mentioning at least one ncRNA and one ASD-related phenotype were found.

Table 5.1: Statistics for the ASD Case Study

Statistic		Number
Annotated sentences		1125
Annotations	Total	1403
	Positives	1373
	Negatives	29
Unique Relations	Total	311
	Positives	307
	Negatives	21
Unique ncRNAs		258
Annotated articles		551

Table 5.2: The 10 most found ncRNAs in positive relations with ASD-related phenotypes

#	ncRNA	#Positive Relations	(%)	#Articles	#Sentences
1	BDNF	508	(37.00)	205	463
2	miR-137	102	(7.43)	29	76
3	IL1RAPL1	52	(3.79)	24	36
4	miR-1290	21	(1.53)	4	16
5	miR-146a	19	(1.38)	11	18
6	miR-132	18	(1.31)	13	16
7	NHIP	17	(1.24)	2	14
8	miR-320a	14	(1.02)	11	15
9	THRIL	11	(0.80)	8	11
10	HOTAIR	11	(0.80)	6	7

Finally, following the best-found method for LLM RE (see 4.1.4), Phi3 with 10 shots was queried to identify uncertain sentences, which resulted in a total of 7,224 sentences being rejected as uncertain. Then the model was queried with the RE prompt to identify relations in the remaining sentences, resulting in a total of 3,114 annotations. However, a great part of these annotations resulted from errors/conflicts in the EL stage, mentioning words that, despite sharing a name with a ncRNA in the context of the sentences annotated, they did not represent the ncRNA. Thus, a post-filtering[1] of these annotations was done, resulting in 1,403 final annotations (refer to Table 5.1 for the complete statistics of this study).

Table 5.2 shows the ncRNAs most found in positive relations with ASD-related phenotypes.

[1] A description of the filtered words can be found in A.3

5.2 Discussion

Analysing the results of Table 5.2, it is confirmed that each of the mentioned ncRNAs shares in fact a relation with ASD.

The brain derived neurotrophic factor (BDNF) antisense RNA, was the ncRNA gene found to be most related to ASD in this study, appearing in 37% of the found positive relations. Neurotrophic factors are key in the healthy development of neurons, with BDNF having known involvement in the differentiation of dopamine-producing neurons in the developing brain [39].

Then, miR-137 was the second-most ncRNA found to be related to ASD, appearing in 7.43% of the positive relations. miR-137 is a microRNA with high expression in the hippocampus and cortical brain regions and lower expression in the cerebellum and brain stem, it was found to regulate neuronal cell proliferation, differentiation, maturation and dendritic development. As such, miR-137 appears to be related to various psychiatric disorders, including ASD, namely by the targeting of ASD candidate genes such as RORa (encodes a ligand-dependent orphan nuclear receptor, which itself targets other ASD candidate genes), SHANK3 (involved in synaptic formation and function) and NRXN1 (an ASD risk factor) [40]. Furthermore, according to SFARI Gene[2] (a database of genes implicated in ASD, funded by the Simons Foundation Autism Research Initiative - SFARI) miR-137 is considered a strong (score 2) ASD candidate gene.

Interleukin-1 receptor accessory protein-like 1 (IL1RAPL1) antisense RNA appeared in 3.79% of positive relations. IL1RAPL1 was found to be associated with intellectual disability and ASD, being involved in the regulation of neuron synapses [41].

And the other found ncRNAs are also all involved in ASD in various ways: miR-1290 and miR-146a is involved in neuronal proliferation and differentiation and maintenance [42, 43], miR-132 was found to be related to Fragile X syndrome (FXS), which represents the most common monogenic cause of ASD [44], NHIP is involved in synaptic regulation [45], miR-320a was found to be related to maternal stress during pregnancy, which is itself related to ASD risk [38], THRIL is a lncRNA involved in the regulation of tumour necrosis factor-α (TNFα), itself related to ASD [46] and finally, HOTAIR is involved in immune response regulation by NF-kB-induced cytokine expression, with this signalling pathway having been found to be related to ASD development [47].

This study proves the usefulness of the developed pipelines in information summarization. However the fact that a post-filtering of the annotations was required, which itself led to a large number of them being rejected indicates that the ERL pipeline needs to be adjusted to deal with these kinds of conflicts. Furthermore, annotations involving ncRNA genes that share a name with protein-coding genes (such as BDNF or IL1RAPL1), might reflect a relation involving the protein-coding gene instead of the ncRNA gene. Perhaps it would be worthwhile to review the methods used to create the ncRNA lexicon or to study more complex ERL methods.

[2] https://gene.sfari.org

Chapter 6

Conclusion

NcRNAs represent the majority of human gene products and are involved in a large set of important biological processes. Their dysregulation is associated with the origin and aggravation of various diseases, making them very relevant disease biomarkers and therapeutic agents. Despite their apparent importance, there is a lack of ncRNA functional annotation databases and information on these biomolecules is still sparsely distributed mainly in the form of scientific research articles. This coupled with the growing throughput of scientific text publications, makes it impossible to manually read and be aware of all the existing information.

This work demonstrates how AI can be employed, mainly through NLP techniques ranging different levels of complexity, to automatically extract and summarize the information available in these large textual datasets. First, a ncRNA-phenotype relation dataset was created, aggregating information from five ncRNA-disease functional annotation databases in a set of 214,300 unique relations. This enabled the creation of a ncRNA-phenotype relational corpus (ncoRP) through DSRE comprising 35,295 unique relations with a precision of 0.761 and a F1-score of 0.593. Both these contributions aim to mitigate the problem with the ncRNA sparseness of information, and the developed pipelines can be easily adapted to be applied to general information extraction and summarization.

Currently, the AI scene is dominated by the LLM paradigm, for their complex understanding of written instructions and capacity to answer in coherent and meaningful generated text. In this work, the potential of LLMs for RE (and NLP tasks in general) is shown, giving insight on how these powerful models can be used beyond day-to-day tasks. As such, a LLM-based RE methodology was developed, making use of simple prompting principles, such as in-context learning and CoT self-explanation, to leverage LLM emergent capabilities. This method chains i) an uncertain sentence filtering prompt with a ii) RE prompt to predict if two entities share a relation in a sentence, yielding a very competent F1-score of 0.978, outperforming state-of-the-art deep learning RE methods, without requiring fine-tuning with large (often manually) labelled datasets. Despite the possibility of these results appearing overly optimistic (as they can be inflated due to the in-context examples used), they still express how powerful LLMs can be in helping to solve the aforementioned problems with ncRNA and general scientific information.

6.1 Future Work

This work shows the usefulness of text mining in information extraction and summarization, through its application on ncRNAs. As such, it would be insightful to apply the established pipelines to different biomedical entity pairs, such as ”gene-drug” or ”protein-protein”. Furthermore, the development of a software tool capable of automatically annotating a given set of scientific research articles with relations involving any pair of entities stands as an interesting project.

The results obtained for LLMs, even employing simple prompting principles on small models, are already promising, as such, the application of more complex LLM approaches could further increase performance. Namely the use of i) larger models, ii) task-specific weight fine-tuning (which could be done using training data generated by these simpler approaches) or iii) refined in-context example generation/selection/ordering, are all topics that deserve further exploration.

References

[1] J. Niderla-Bielińska, E. Jankowska-Steifer, and P. Włodarski, "Non-coding rnas and human diseases: Current status and future perspectives," *International Journal of Molecular Sciences*, vol. 24, 7 2023. 1, 14

[2] J. Zhang, S. Zou, and L. Deng, "Gene ontology-based function prediction of long non-coding rnas using bi-random walk 06 biological sciences 0601 biochemistry and cell biology," *BMC Medical Genomics*, vol. 11, 11 2018. 1, 2

[3] T. Loganathan and G. P. D. C, "Non-coding rnas in human health and disease: potential function as biomarkers and therapeutic targets," *Functional and Integrative Genomics*, vol. 23, 3 2023. 1, 2

[4] D. F. Sousa and F. M. Couto, "K-ret: knowledgeable biomedical relation extraction system," *Bioinformatics*, vol. 39, 4 2023. 2, 7, 9, 11

[5] A. Lamurias, L. A. Clarke, and F. M. Couto, "Extracting microrna-gene relations from biomedical literature using distant supervision," *PLoS ONE*, vol. 12, 3 2017. 2, 7

[6] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: state of the art, current trends and challenges," *Multimedia Tools and Applications*, vol. 82, 2023. 2, 7, 8

[7] K. Detroja, C. K. Bhensdadia, and B. S. Bhatt, "A survey on relation extraction," *Intelligent Systems with Applications*, vol. 19, 9 2023. 2, 9, 10, 11, 16

[8] D. Sousa, A. Lamurias, and F. M. Couto, "A silver standard corpus of human phenotype-gene relations," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, 3 2019. 2, 8

[9] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen, "A survey of large language models," 3 2023. 2, 11, 12

[10] F. M. Couto and A. Lamurias, "Mer: A shell script and annotation server for minimal named entity recognition and linking," *Journal of Cheminformatics*, vol. 10, 12 2018. 3, 8, 9, 22

[11] N. Milošević and W. Thielemann, "Comparison of biomedical relationship extraction methods and models for knowledge graph creation," *Journal of Web Semantics*, vol. 75, 1 2023. 7, 8, 9, 10, 17

[12] X. Ling, S. Singh, and D. S. Weld, "Design challenges for entity linking," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 315–328, 2015. 8, 9

[13] J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recognition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, pp. 50–70, 1 2022. 8

[14] P. Ruas and F. M. Couto, "Nilinker: Attention-based approach to nil entity linking," *Journal of Biomedical Informatics*, vol. 132, 8 2022. 9

[15] Z. Nasar, S. W. Jaffry, and M. K. Malik, "Named entity recognition and relation extraction: State-of-the-art," *ACM Computing Surveys*, vol. 54, 4 2021. 9, 10, 11

[16] K. Zhou, Q. Qiao, Y. Li, and Q. Li, "Improving distantly supervised relation extraction by natural language inference," 2023. 10, 17

[17] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 5 2020. 11, 12, 13, 29

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Łukasz Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 2017-December, 2017. 11, 12

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 6 2019. 11

[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 1 2013. 11

[21] J. D. Zamfirescu-Pereira, R. Y. Wong, B. Hartmann, and Q. Yang, "Why johnny can't prompt: How non-ai experts try (and fail) to design llm prompts," *Conference on Human Factors in Computing Systems - Proceedings*, 4 2023. 12

[22] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," 1 2022. 13

[23] S. Wadhwa, S. Amir, and B. C. Wallace, "Revisiting relation extraction in the era of large language models," 2023. 13

[24] J. Zhao, Z. Yao, Z. Yang, and H. Yu, "Self-explain: Teaching large language models to reason complex questions by themselves," 12 2023. 13

[25] R. L. Seal, B. Braschi, K. Gray, T. E. Jones, S. Tweedie, L. Haim-Vilmovsky, and E. A. Bruford, "Genenames.org: the hgnc resources in 2023," *Nucleic acids research*, vol. 51, pp. D1003–D1009, 1 2023. 13

[26] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, vol. 5, pp. 199–220, 1993. 13

[27] M. A. Gargano, N. Matentzoglu, B. Coleman, E. B. Addo-Lartey, A. V. Anagnostopoulos, J. Anderton, P. Avillach, A. M. Bagley, E. Bakštein, J. P. Balhoff, G. Baynam, S. M. Bello, M. Berk, H. Bertram, S. Bishop, H. Blau, D. F. Bodenstein, P. Botas, K. Boztug, J. Čady, T. J. Callahan, R. Cameron, S. J. Carbon, F. Castellanos, J. H. Caufield, L. E. Chan, C. G. Chute, J. Cruz-Rojo, N. Dahan-Oliel, J. R. Davids, M. Dieuleveult, V. Souza, B. B. de Vries, E. Vries, J. R. DePaulo, B. Derfalvi, F. Dhombres, C. Diaz-Byrd, A. J. Dingemans, B. Donadille, M. Duyzend, R. Elfeky, S. Essaid, C. Fabrizzi, G. Fico, H. V. Firth, Y. Freudenberg-Hua, J. M. Fullerton, D. L. Gabriel, K. Gilmour, J. Giordano, F. S. Goes, R. G. Moses, I. Green, M. Griese, T. Groza, W. Gu, J. Guthrie, B. Gyori, A. Hamosh, M. Hanauer, K. Hanušová, Y. He, H. Hegde, I. Helbig, K. Holasová, C. T. Hoyt, S. Huang, E. Hurwitz, J. O. Jacobsen, X. Jiang, L. Joseph, K. Keramatian, B. King, K. Knoflach, D. A. Koolen, M. L. Kraus, C. Kroll, M. Kusters, M. S. Ladewig, D. Lagorce, M. C. Lai, P. Lapunzina, B. Laraway, D. Lewis-Smith, X. Li, C. Lucano, M. Majd, M. L. Marazita, V. Martinez-Glez, T. H. McHenry, M. G. McInnis, J. A. McMurry, M. Mihulová, C. E. Millett, P. B. Mitchell, V. Moslerová, K. Narutomi, S. Nematollahi, J. Nevado, A. A. Nierenberg, N. N. Čajbiková, J. I. Nurnberger, S. Ogishima, D. Olson, A. Ortiz, H. Pachajoa, G. P. Nanclares, A. Peters, T. Putman, C. K. Rapp, A. Rath, J. Reese, L. Rekerle, A. M. Roberts, S. Roy, S. J. Sanders, C. Schuetz, E. C. Schulte, T. G. Schulze, M. Schwarz, K. Scott, D. Seelow, B. Seitz, Y. Shen, M. N. Similuk, E. S. Simon, B. Singh, D. Smedley, C. L. Smith, J. T. Smolinsky, S. Sperry, E. Stafford, R. Stefancsik, R. Steinhaus, R. Strawbridge, J. C. Sundaramurthi, P. Talapova, J. A. Castano, P. Tesner, R. H. Thomas, A. Thurm, M. Turnovec, M. E. van Gijn, N. A. Vasilevsky, M. Vlčková, A. Walden, K. Wang, R. Wapner, J. S. Ware, A. A. Wiafe, S. A. Wiafe, L. D. Wiggins, A. E. Williams, C. Wu, M. J. Wyrwoll, H. Xiong, N. Yalin, Y. Yamamoto, L. N. Yatham, A. K. Yocum, A. H. Young, Z. Yüksel, P. P. Zandi, A. Zankl, I. Zarante, M. Zvolský, S. Toro, L. C. Carmody, N. L. Harris, M. C. Munoz-Torres, D. Danis, C. J. Mungall, S. Köhler, M. A. Haendel, and P. N. Robinson, "The human phenotype ontology in 2024: phenotypes around the world," *Nucleic Acids Research*, vol. 52, pp. D1333–D1346, 1 2024. 13

[28] B. A. Sweeney, A. I. Petrov, C. E. Ribas, R. D. Finn, A. Bateman, M. Szymanski, W. M. Karlowski, S. E. Seemann, J. Gorodkin, J. J. Cannone, R. R. Gutell, S. Kay, S. Marygold, G. D. Santos, A. Frankish, J. M. Mudge, R. Barshir, S. Fishilevich, P. P. Chan, T. M. Lowe, R. Seal,

E. Bruford, S. Panni, P. Porras, D. Karagkouni, A. G. Hatzigeorgiou, L. Ma, Z. Zhang, P. J. Volders, P. Mestdagh, S. Griffiths-Jones, B. Fromm, K. J. Peterson, I. Kalvari, E. P. Nawrocki, A. S. Petrov, S. Weng, P. Bouchard-Bourelle, M. Scott, L. M. Lui, D. Hoksza, R. C. Lovering, B. Kramarz, P. Mani, S. Ramachandran, and Z. Weinberg, "Rnacentral 2021: Secondary structure integration, improved sequence search and new member databases," *Nucleic Acids Research*, vol. 49, pp. D212–D220, 1 2021. 14

[29] C. Cui, B. Zhong, R. Fan, and Q. Cui, "Hmdd v4.0:ã database for experimentally supported human microrna-diseaseãssociations," *Nucleic Acids Research*, vol. 52, pp. D1327–D1332, 1 2024. 14

[30] X. Lin, Y. Lu, C. Zhang, Q. Cui, Y. D. Tang, X. Ji, and C. Cui, "Lncrnadisease v3.0: an updated database of long non-coding rna-associated diseases," *Nucleic Acids Research*, vol. 52, pp. D1365–D1369, 1 2024. 14

[31] W. Zhang, G. Yao, J. Wang, M. Yang, J. Wang, H. Zhang, and W. Li, "ncrpheno: a comprehensive database platform for identification and validation of disease related noncoding rnas," *RNA Biology*, vol. 17, pp. 943–955, 7 2020. 14

[32] H. Zheng, L. Xu, H. Xie, J. Xie, Y. Ma, Y. Hu, L. Wu, J. Chen, M. Wang, Y. Yi, Y. Huang, and D. Wang, "Riscoper 2.0: A deep learning tool to extract rna biomedical relation sentences from literature," *Computational and Structural Biotechnology Journal*, vol. 23, pp. 1469–1476, 12 2024. 15

[33] J. Chen, J. Lin, Y. Hu, M. Ye, L. Yao, L. Wu, W. Zhang, M. Wang, T. Deng, F. Guo, Y. Huang, B. Zhu, and D. Wang, "Rnadisease v4.0: an updated resource of rna-associated diseases, providing rna-disease analysis, enrichment and prediction," *Nucleic Acids Research*, vol. 51, pp. D1397–D1404, 1 2023. 15

[34] J. L. Fleiss, "Measuring nominal scale agreement among many raters," *Psychological Bulletin*, vol. 76, pp. 378–382, 1971. 16

[35] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 11 2019. 18

[36] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019. 19

[37] Şenay Kafkas, S. Althubaiti, G. V. Gkoutos, R. Hoehndorf, and P. N. Schofield, "Linking common human diseases to their phenotypes; development of a resource for human phenomics," *Journal of Biomedical Semantics*, vol. 12, 12 2021. 20, 22

[38] L. H. Cui, W. R. Du, N. Xu, J. Y. Dong, B. J. Xia, J. Y. Ma, R. T. Yan, L. Y. Wang, and F. M. Feng, "Impact of micrornas in interaction with environmental factors on autism spectrum disorder: An exploratory pilot study," *Frontiers in Psychiatry*, vol. 12, 10 2021. 33, 35

[39] V. Bryn, B. Halvorsen, T. Ueland, J. Isaksen, K. Kolkova, K. Ravn, and O. H. Skjeldal, "Brain derived neurotrophic factor (bdnf) and autism spectrum disorders (asd) in childhood," *European Journal of Paediatric Neurology*, vol. 19, pp. 411–414, 7 2015. 35

[40] E. Mahmoudi and M. J. Cairns, "Mir-137: An important player in neural development and neoplastic transformation," 1 2017. 35

[41] T. Hayashi, T. Yoshida, M. Ra, R. Taguchi, and M. Mishina, "Il1rapl1 associated with mental retardation and autism regulates the formation and stabilization of glutamatergic synapses of cortical neurons through rhoa signaling pathway," *PLoS ONE*, vol. 8, 6 2013. 35

[42] D. Moore, B. M. Meays, L. S. Madduri, F. Shahjin, S. Chand, M. Niu, A. Albahrani, C. Guda, G. Pendyala, H. S. Fox, and S. V. Yelamanchili, "Downregulation of an evolutionary young mir-1290 in an ipscderived neural stem cell model of autism spectrum disorder," *Stem Cells International*, vol. 2019, 2019. 35

[43] L. S. Nguyen, J. Fregeac, C. Bole-Feysot, N. Cagnard, A. Iyer, J. Anink, E. Aronica, O. Alibeu, P. Nitschke, and L. Colleaux, "Role of mir-146a in neural stem cell differentiation and neural lineage determination: Relevance for neurodevelopmental disorders," *Molecular Autism*, vol. 9, 6 2018. 35

[44] R. R. Couto, F. Kubaski, M. Siebert, T. M. Félix, A. C. Brusius-Facchin, and S. Leistner-Segal, "Increased serum levels of mir-125b and mir-132 in fragile x syndrome: A preliminary study," *Neurology: Genetics*, vol. 8, 12 2022. 35

[45] Y. Zhu, J. A. Gomez, B. I. Laufer, C. E. Mordaunt, J. S. Mouat, D. C. Soto, M. Y. Dennis, K. S. Benke, K. M. Bakulski, J. Dou, R. Marathe, J. M. Jianu, L. A. Williams, O. J. G. Fugón, C. K. Walker, S. Ozonoff, J. Daniels, L. P. Grosvenor, H. E. Volk, J. I. Feinberg, M. D. Fallin, I. Hertz-Picciotto, R. J. Schmidt, D. H. Yasui, and J. M. LaSalle, "Placental methylome reveals a 22q13.33 brain regulatory gene locus associated with autism," *Genome Biology*, vol. 23, 12 2022. 35

[46] J. Xie, L. Huang, X. Li, H. Li, Y. Zhou, H. Zhu, T. Pan, K. M. Kendrick, and W. Xu, "Immunological cytokine profiling identifies tnf-α as a key molecule dysregulated in autistic children," *Oncotarget*, vol. 8, pp. 82390–82398, 2017. 35

[47] M. Safari, R. Noroozi, M. Taheri, and S. Ghafouri-Fard, "The rs12826786 in hotair lncrna is associated with risk of autism spectrum disorder," *Journal of Molecular Neuroscience*, vol. 70, pp. 175–179, 2 2020. 35

Appendix A

Extra Information

This appendix presents additional information about the work.

A.1 Evaluation of the Entity Recognition and Linking of Phenotypes

In this section, additional information about the ERL of phenotypes in disease descriptions (see 3.1.1) is presented. Namely i) the defined criteria for the evaluation (Table A.1), ii) the equations for the euclidean (A.1) and cosine (A.2) distance metrics and iii) the comparison between different distance metrics and thresholds (Table A.2).

$$d_{Euclidean}(A, B) = \sqrt{(A_1 - B_1)^2 + (A_2 - B_2)^2 + ... + (A_n - B_n)^2} \tag{A.1}$$

$$d_{Cosine}(A, B) = 1 - \frac{A_1 \cdot B_1 + A_2 \cdot B_2 + ... + A_n \cdot B_n}{\sqrt{A_1^2 + A_2^2 + ... + A_n^2} \cdot \sqrt{B_1^2 + B_2^2 + ... + B_n^2}} \tag{A.2}$$

A.2 Large Language Models for Relation Extraction

In this section, additional results referent to Chapter 4 are presented. Namely the complete results of the number of shots analysis on different models, using a single RE prompt (Table A.3) and two uncertain filtering and RE prompts (Table A.4).

Table A.1: Method used to label TP, TN, FP and FN in the ERL of HPO Terms from disease descriptions

HPO Term?	Case	Classification
Yes	HPO Term fits description	TP
	HPO Term does not fit description	FP
No	There is an HPO Term that could fit description	FN
	There is no HPO Term that could fit description / Bad description	TN

Table A.2: Preliminary performance evaluation of different distance metrics and thresholds in random samples of 200 unique disease descriptions (100 linked with HPO terms). [1]Number of total unique disease descriptions that linked to HPO terms (out of the total 8772 unique disease descriptions); [2]was further evaluated in a bigger random sample of 900 unique disease descriptions; [3]resulted in less than 100 linked HPO terms and therefore could not be evaluated

Distance	Threshold	Precision	Recall	F1-score	Total Links[1]
Euclidean	0.1	1.000	0.617	0.763	1428
	0.3	0.990	0.692	0.815	2942
	0.5[2]	0.950	0.856	**0.900**	4954
	0.7	0.850	0.944	0.895	6656
	0.9	0.800	1.000	0.889	7867
Cosine	0.1[3]	-	-	-	0
	0.3[3]	-	-	-	1
	0.5	0.000	0.000	0.000	458
	0.7	0.300	0.268	0.283	2092
	0.9	0.590	0.401	0.478	6729

Table A.3: Complete results of the Large Language Model Relation Extraction performance analysis of different models and numbers of shots across 4 duplicates

(Model, N-shots)	Precision	Recall	F1-score	Invalid %	Pred Time
('llama3', 2)	0.687 ± 0.000	1.000 ± 0.000	0.814 ± 0.000	0.000 ± 0.000	1.597 ± 0.036
('llama3', 4)	0.692 ± 0.009	1.000 ± 0.000	0.818 ± 0.006	0.000 ± 0.000	1.495 ± 0.022
('llama3', 6)	0.692 ± 0.005	1.000 ± 0.000	0.818 ± 0.004	0.000 ± 0.000	1.647 ± 0.029
('llama3', 8)	0.691 ± 0.006	0.984 ± 0.009	0.812 ± 0.006	0.000 ± 0.000	1.757 ± 0.037
('llama3', 10)	0.699 ± 0.006	0.984 ± 0.018	0.817 ± 0.009	0.000 ± 0.000	1.756 ± 0.065
('llama3:70b', 2)	0.726 ± 0.006	0.891 ± 0.000	0.800 ± 0.004	0.000 ± 0.000	6.183 ± 0.030
('llama3:70b', 4)	0.757 ± 0.004	0.880 ± 0.019	0.814 ± 0.010	0.000 ± 0.000	5.731 ± 0.037
('llama3:70b', 6)	0.761 ± 0.007	0.935 ± 0.000	0.839 ± 0.004	0.000 ± 0.000	6.979 ± 0.051
('llama3:70b', 8)	0.748 ± 0.002	0.902 ± 0.011	0.818 ± 0.006	0.000 ± 0.000	7.272 ± 0.036
('llama3:70b', 10)	0.752 ± 0.004	0.908 ± 0.009	0.823 ± 0.002	0.000 ± 0.000	8.274 ± 0.021
('phi3', 2)	0.689 ± 0.003	1.000 ± 0.000	0.816 ± 0.002	0.746 ± 0.746	1.241 ± 0.101
('phi3', 4)	0.689 ± 0.004	1.000 ± 0.000	0.816 ± 0.003	0.000 ± 0.000	1.254 ± 0.111
('phi3', 6)	0.687 ± 0.000	1.000 ± 0.000	0.814 ± 0.000	0.000 ± 0.000	1.442 ± 0.142
('phi3', 8)	0.692 ± 0.005	1.000 ± 0.000	0.818 ± 0.004	0.000 ± 0.000	1.605 ± 0.067
('phi3', 10)	0.705 ± 0.005	1.000 ± 0.000	0.827 ± 0.003	0.000 ± 0.000	1.759 ± 0.187
('phi3:14b', 2)	0.713 ± 0.005	1.000 ± 0.000	0.833 ± 0.004	0.000 ± 0.000	2.569 ± 0.735
('phi3:14b', 4)	**0.750 ± 0.005**	**0.995 ± 0.009**	**0.855 ± 0.005**	**0.000 ± 0.000**	**2.107 ± 0.249**
('phi3:14b', 6)	0.746 ± 0.014	0.989 ± 0.011	0.851 ± 0.010	0.000 ± 0.000	2.749 ± 0.106
('phi3:14b', 8)	0.736 ± 0.013	1.000 ± 0.000	0.848 ± 0.009	0.000 ± 0.000	2.796 ± 0.112
('phi3:14b', 10)	0.741 ± 0.015	0.983 ± 0.010	0.845 ± 0.011	1.492 ± 0.000	3.561 ± 0.809
('gemma', 2)	0.726 ± 0.000	0.978 ± 0.000	0.833 ± 0.000	0.000 ± 0.000	1.095 ± 0.007
('gemma', 4)	0.725 ± 0.002	0.973 ± 0.009	0.831 ± 0.005	0.000 ± 0.000	1.069 ± 0.006
('gemma', 6)	0.738 ± 0.000	0.978 ± 0.000	0.841 ± 0.000	0.000 ± 0.000	1.314 ± 0.006
('gemma', 8)	0.738 ± 0.000	0.978 ± 0.000	0.841 ± 0.000	0.000 ± 0.000	1.388 ± 0.006
('gemma', 10)	0.734 ± 0.004	0.978 ± 0.000	0.839 ± 0.003	0.746 ± 0.746	1.469 ± 0.008
('gemma2', 2)	0.694 ± 0.002	0.984 ± 0.009	0.814 ± 0.005	0.000 ± 0.000	1.505 ± 0.008
('gemma2', 4)	0.711 ± 0.005	1.000 ± 0.000	0.831 ± 0.003	0.000 ± 0.000	1.484 ± 0.019
('gemma2', 6)	0.697 ± 0.000	1.000 ± 0.000	0.821 ± 0.000	0.000 ± 0.000	1.794 ± 0.012
('gemma2', 8)	0.697 ± 0.000	1.000 ± 0.000	0.821 ± 0.000	0.000 ± 0.000	1.896 ± 0.006
('gemma2', 10)	0.726 ± 0.000	0.978 ± 0.000	0.833 ± 0.000	0.000 ± 0.000	1.995 ± 0.012
('mixtral', 2)	0.746 ± 0.000	0.957 ± 0.000	0.838 ± 0.000	0.000 ± 0.000	2.724 ± 0.062
('mixtral', 4)	0.725 ± 0.002	0.973 ± 0.009	0.831 ± 0.005	0.000 ± 0.000	2.245 ± 0.029
('mixtral', 6)	0.753 ± 0.005	0.978 ± 0.000	0.851 ± 0.004	0.000 ± 0.000	2.677 ± 0.022
('mixtral', 8)	0.744 ± 0.006	0.978 ± 0.000	0.845 ± 0.004	0.000 ± 0.000	2.877 ± 0.019
('mixtral', 10)	0.749 ± 0.006	0.957 ± 0.000	0.840 ± 0.004	0.000 ± 0.000	3.000 ± 0.043
('mixtral:8x22b', 2)	0.723 ± 0.005	0.995 ± 0.009	0.838 ± 0.004	0.373 ± 0.646	36.999 ± 0.369
('mixtral:8x22b', 4)	0.736 ± 0.012	0.973 ± 0.010	0.838 ± 0.011	1.119 ± 0.646	37.820 ± 1.765
('mixtral:8x22b', 6)	0.748 ± 0.002	0.962 ± 0.010	0.841 ± 0.004	0.746 ± 1.293	44.721 ± 1.162
('mixtral:8x22b', 8)	0.736 ± 0.014	0.951 ± 0.009	0.830 ± 0.011	0.746 ± 0.746	46.355 ± 1.543
('mixtral:8x22b', 10)	0.768 ± 0.010	0.935 ± 0.000	0.843 ± 0.006	1.119 ± 0.646	52.772 ± 7.572

Table A.4: Complete results of the Large Language Model Relation Extraction performance analysis of different models and numbers of shots, with a preceding uncertain filtering prompt across 4 duplicates. [1]Percentage of sentences rejected as uncertain by the uncertain filtering prompt;[2]Percentage of invalid responses to the uncertain filtering prompt; In italic are the methods that resulted in more than 10% of invalid responses to the uncertain filtering prompt

(Model, N-shots)	Precision	Recall	F1-score	Invalid %	Pred Time	Uncertain%[1]	Invalid % in UF[2]
('llama3', 2)	0.772 ± 0.023	1.000 ± 0.000	0.871 ± 0.014	0.000 ± 0.000	1.638 ± 0.020	18.284 ± 2.208	0.000 ± 0.000
('llama3', 4)	0.737 ± 0.002	1.000 ± 0.000	0.848 ± 0.001	0.000 ± 0.000	1.537 ± 0.006	9.328 ± 0.646	0.000 ± 0.000
('llama3', 6)	0.765 ± 0.011	1.000 ± 0.000	0.867 ± 0.007	0.000 ± 0.000	1.640 ± 0.030	15.672 ± 0.746	0.000 ± 0.000
('llama3', 8)	0.801 ± 0.012	1.000 ± 0.000	0.890 ± 0.007	0.000 ± 0.000	1.682 ± 0.025	26.866 ± 2.111	2.612 ± 1.238
('llama3', 10)	*0.814 ± 0.026*	*1.000 ± 0.000*	*0.897 ± 0.016*	*0.000 ± 0.000*	*1.652 ± 0.028*	*35.821 ± 2.111*	*17.537 ± 3.054*
('llama3:70b', 2)	0.834 ± 0.012	0.976 ± 0.000	0.899 ± 0.007	0.000 ± 0.000	6.542 ± 0.562	24.627 ± 0.746	0.000 ± 0.000
('llama3:70b', 4)	0.801 ± 0.013	0.880 ± 0.010	0.839 ± 0.011	0.000 ± 0.000	5.693 ± 0.065	11.567 ± 1.238	0.000 ± 0.000
('llama3:70b', 6)	0.823 ± 0.003	0.923 ± 0.011	0.870 ± 0.007	0.000 ± 0.000	6.863 ± 0.090	16.791 ± 0.646	0.000 ± 0.000
('llama3:70b', 8)	0.839 ± 0.011	0.977 ± 0.000	0.902 ± 0.006	0.000 ± 0.000	7.158 ± 0.058	23.134 ± 1.669	0.000 ± 0.000
('llama3:70b', 10)	*0.847 ± 0.007*	*0.943 ± 0.011*	*0.892 ± 0.001*	*0.000 ± 0.000*	*7.866 ± 0.032*	*20.149 ± 0.746*	*0.000 ± 0.000*
('phi3', 2)	0.724 ± 0.003	1.000 ± 0.000	0.840 ± 0.002	0.000 ± 0.000	1.241 ± 0.149	8.209 ± 1.293	0.000 ± 0.000
('phi3', 4)	0.698 ± 0.009	1.000 ± 0.000	0.822 ± 0.006	0.000 ± 0.000	1.171 ± 0.091	3.731 ± 1.293	0.000 ± 0.000
('phi3', 6)	0.706 ± 0.009	1.000 ± 0.000	0.828 ± 0.006	0.000 ± 0.000	1.347 ± 0.140	4.851 ± 1.238	0.000 ± 0.000
('phi3', 8)	0.717 ± 0.010	1.000 ± 0.000	0.835 ± 0.007	0.000 ± 0.000	1.554 ± 0.055	13.060 ± 0.646	0.000 ± 0.000
('phi3', 10)	**0.892 ± 0.023**	**1.000 ± 0.000**	**0.943 ± 0.013**	**0.000 ± 0.000**	**1.899 ± 0.261**	**41.418 ± 2.866**	**0.000 ± 0.000**
('phi3:14b', 2)	0.743 ± 0.005	1.000 ± 0.000	0.852 ± 0.004	0.000 ± 0.000	2.022 ± 0.079	7.463 ± 1.828	0.000 ± 0.000
('phi3:14b', 4)	0.741 ± 0.006	1.000 ± 0.000	0.851 ± 0.004	0.000 ± 0.000	1.933 ± 0.031	4.104 ± 0.646	0.000 ± 0.000
('phi3:14b', 6)	0.744 ± 0.020	0.977 ± 0.016	0.845 ± 0.018	0.000 ± 0.000	2.712 ± 0.085	10.821 ± 3.231	0.000 ± 0.000
('phi3:14b', 8)	0.746 ± 0.013	1.000 ± 0.000	0.854 ± 0.008	0.000 ± 0.000	3.446 ± 1.027	14.552 ± 2.208	0.746 ± 1.293
('phi3:14b', 10)	0.808 ± 0.010	0.994 ± 0.011	0.891 ± 0.006	1.942 ± 0.019	3.155 ± 0.210	23.134 ± 0.746	0.373 ± 0.646
('gemma', 2)	0.736 ± 0.005	0.978 ± 0.000	0.840 ± 0.004	0.000 ± 0.000	1.135 ± 0.039	8.955 ± 0.000	0.000 ± 0.000
('gemma', 4)	0.738 ± 0.000	1.000 ± 0.000	0.849 ± 0.000	0.000 ± 0.000	1.071 ± 0.013	8.955 ± 0.000	0.000 ± 0.000
('gemma', 6)	0.736 ± 0.004	1.000 ± 0.000	0.848 ± 0.003	0.000 ± 0.000	1.326 ± 0.009	9.328 ± 0.646	0.000 ± 0.000
('gemma', 8)	0.746 ± 0.000	1.000 ± 0.000	0.854 ± 0.000	0.000 ± 0.000	1.393 ± 0.011	11.940 ± 0.000	2.985 ± 0.000
('gemma', 10)	*0.762 ± 0.004*	*1.000 ± 0.000*	*0.865 ± 0.003*	*0.000 ± 0.000*	*1.456 ± 0.010*	*19.403 ± 1.055*	*10.448 ± 1.055*
('gemma2', 2)	0.801 ± 0.007	0.975 ± 0.000	0.880 ± 0.004	0.000 ± 0.000	1.601 ± 0.064	25.373 ± 1.055	0.000 ± 0.000
('gemma2', 4)	0.742 ± 0.009	1.000 ± 0.000	0.852 ± 0.006	0.000 ± 0.000	1.492 ± 0.012	12.687 ± 1.293	0.000 ± 0.000
('gemma2', 6)	0.815 ± 0.009	1.000 ± 0.000	0.898 ± 0.005	0.000 ± 0.000	1.793 ± 0.020	23.508 ± 0.646	0.000 ± 0.000
('gemma2', 8)	0.809 ± 0.005	1.000 ± 0.000	0.894 ± 0.003	0.000 ± 0.000	1.890 ± 0.013	25.746 ± 1.939	0.000 ± 0.000
('gemma2', 10)	0.811 ± 0.002	0.975 ± 0.000	0.886 ± 0.001	0.000 ± 0.000	1.980 ± 0.001	26.492 ± 0.646	1.492 ± 0.000
('mixtral', 2)	0.782 ± 0.006	0.975 ± 0.000	0.868 ± 0.004	0.000 ± 0.000	2.908 ± 0.049	23.508 ± 1.238	0.000 ± 0.000
('mixtral', 4)	0.755 ± 0.010	1.000 ± 0.000	0.860 ± 0.006	0.000 ± 0.000	2.213 ± 0.053	20.895 ± 1.055	0.000 ± 0.000
('mixtral', 6)	0.761 ± 0.005	1.000 ± 0.000	0.864 ± 0.003	0.000 ± 0.000	2.680 ± 0.044	19.030 ± 1.238	1.492 ± 0.000
('mixtral', 8)	0.852 ± 0.009	1.000 ± 0.000	0.920 ± 0.005	0.000 ± 0.000	2.848 ± 0.014	31.716 ± 0.646	4.851 ± 1.238
('mixtral', 10)	*0.963 ± 0.001*	*1.000 ± 0.000*	*0.981 ± 0.001*	*0.000 ± 0.000*	*2.917 ± 0.017*	*59.328 ± 1.627*	*24.627 ± 1.669*
('mixtral:8x22b', 2)	0.743 ± 0.009	1.000 ± 0.000	0.852 ± 0.006	0.439 ± 0.760	36.915 ± 0.232	14.179 ± 1.669	0.373 ± 0.646
('mixtral:8x22b', 4)	0.764 ± 0.017	0.994 ± 0.010	0.864 ± 0.014	0.000 ± 0.000	38.038 ± 1.877	13.806 ± 1.939	0.373 ± 0.646
('mixtral:8x22b', 6)	0.789 ± 0.009	1.000 ± 0.000	0.882 ± 0.006	0.962 ± 1.665	46.574 ± 3.332	21.642 ± 1.669	1.866 ± 1.627
('mixtral:8x22b', 8)	0.812 ± 0.020	0.994 ± 0.011	0.894 ± 0.015	0.532 ± 0.921	44.984 ± 1.082	28.731 ± 2.866	5.597 ± 2.866
('mixtral:8x22b', 10)	*0.961 ± 0.027*	*1.000 ± 0.000*	*0.980 ± 0.014*	*0.000 ± 0.000*	*45.864 ± 2.759*	*62.687 ± 1.828*	*22.015 ± 1.627*

A.3 Autism Spectrum Disorder Case Study Annotation Post-Filtering

In this section are described the words that caused conflicting annotations in the ASD case study, and ended up being filtered out.

- **”Air”, ”Top”, ”Fast”, ”Particle”, ”Dams”, ”Cardinal”, ”Digit”, ”Laser”, ”Storm”, ”Tag”:** General common words that can generally appear in sentences;
- **”GST”:** Often found to refer to glutathione S-transferase, a group of enzymes;
- **”16S rRNA”:** Often found to relate to anti-shine Delgarno sequences instead of Autism Spectrum Disorder;
- **”TRN”:** Often found to refer to the thalamic reticular nucleus;
- **”CNTNAP2”:** Was found to be linked to an incorrect ID (URS000040D83C, representing snoRNA-U3), by the European Nucleotide Archive[1].

[1] https://www.ebi.ac.uk/ena/browser/view/Non-coding:AC083849.6:28401..28610:ncRNA