**Decoding legislation for citizens: proof of concept of the “Retirement Law” use case**

State of the Art

**Outline**

[1. Introduction 3](#_Toc59616901)

[2. Data Structure and Transformation 6](#_Toc59616902)

[2.1 Techniques 6](#_Toc59616903)

[2.1.1 Knowledge Graphs 6](#_Toc59616904)

[2.1.2 Web Ontology Language (OWL) 9](#_Toc59616905)

[2.1.3 Decision Tree 11](#_Toc59616906)

[2.1.4 Probabilistic Latent Semantic Analysis 12](#_Toc59616907)

[2.2 Discussion 13](#_Toc59616908)

[3. Question Answering 15](#_Toc59616909)

[3.1 Techniques 15](#_Toc59616910)

[3.1.2 Bidirectional Encoder Representations 16](#_Toc59616911)

[3.2 Discussion 17](#_Toc59616912)

[4. Text Simplification 18](#_Toc59616913)

[4.1 Techniques 19](#_Toc59616914)

[4.1.1 Text Summarization 19](#_Toc59616915)

[4.1.2 Parts of speech extraction 19](#_Toc59616916)

[4.1.3 Semantic Representation 20](#_Toc59616917)

[4.1.4 Neural Text Simplification 21](#_Toc59616918)

[4.1.5 Bidirectional Encoder Representations 22](#_Toc59616919)

[4.1.6 OpenAI GPT-3 24](#_Toc59616920)

[4.2 Discussion 24](#_Toc59616921)

[5. Conclusion 26](#_Toc59616922)

[References 27](#_Toc59616923)

# 1. Introduction

The Portuguese Mint and Official Printing Office (INCM) is responsible for the production of goods and services that are essential to the functioning of the Portuguese State, such as the identity and travel documents, the minting of coins and the official publication of the Portuguese official Gazette (Diário da República), though which all citizens can become aware of the acts that govern the Portuguese society. The Electronic Official Journal (DRE) is published at DRE.pt as a public service, with universal and free access to all users. Since the 1st of July 2006 the acts published in this website have full legal value (Law nº 26/2006 of 30 June). Given its maturity and its potential of enhancement as a service tool for the citizens, the DRE has become a relevant object of interest on applying innovative techniques. The access to DRE allows the consultation of all Portuguese legislation, but with a complex navigation structure and the necessity of a technical and detailed interpretation of law texts. In order to take advantage of the potential and the challenges of the DRE, INCM is investing on research and experimentation targeting the service to the citizen, by making legislation texts more accessible and the DRE.pt law content easier to navigate. In this context, the INCM in partnership with others R&D organizations, started a case study on the Consumer Law, with the objective of develop a Natural Language Processing (NLP) tool capable of receiving as input a citizen’s questions and deliver as result an answer based on an interpretation of the Portuguese law.

The partnership between the Nova SBE Data Science Knowledge Center (DSKC) and INCM rises as an opportunity to enhance the experimentation in this field, by testing an agile, efficient and low-cost model in the Nova SBE’s new Master’s program in Business Analytics, with the goal to develop a new use case to serve as a proof of concept of the “Decode legislation” project, regarding the Retirement Statute. By applying cutting-edge technology in order to improve the research and navigation resources for the final user, DSKC and INCM are working on developing a system capable of answering questions that Portuguese citizens may have regarding the Portuguese Retirement Statute, making the law more accessible and interpretable. The final product should be able to answer queries such as “What is the minimum age to get my pension?” and "Am I eligible for early retirement?". This application may be a chatbot system, or an auto-generated frequently asked questions (FAQ) that responds to user queries.

The process will start with mapping the Retirement Statute to a data structure that will feed a law simplification application. The Retirement Statute contains all the legal content on the right to get retirement [1]. The statute is structured in 143 articles, with each article having several numbers. The text presented in each article is complex to interpret, often containing references to other articles, such as exceptions, conditions to consider, and extension of those articles. The references sometimes refer to external laws. This work presents the state of the art that will be the foundation to build annotations and a data structure for the law simplification application.

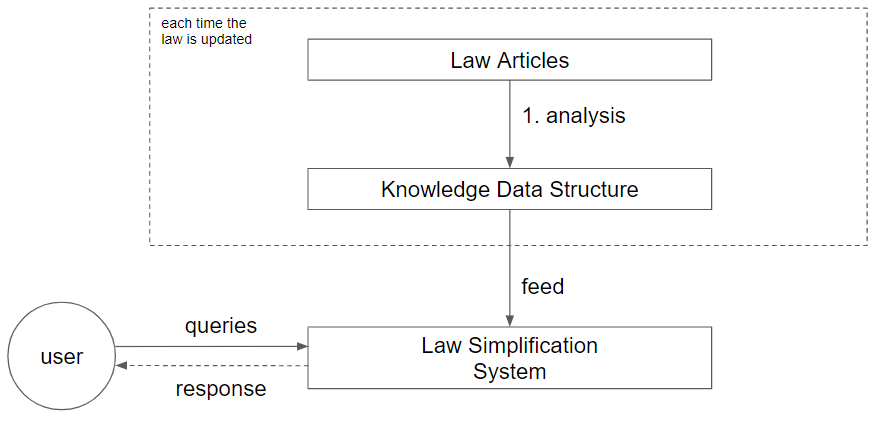


Figure 1: System Overview

In one year of queries made by users to the current law search website, approximately 7,000 queries were made about this statute, being 3,400 of them unique. The queries were selected from the queries database which contained 1,100,000 queries and the classification of queries to this statute were based on the match of any of the following sequence of characters: “reform”, “aposent” and “pens”.

The top 20 queries with the correspondent value were the following:

1. aposentação (195)

2. aposentações (154)

3. reforma (151)

4. aposentados (125)

5. reformas (120)

6. reforma antecipada (97)

7. lista de aposentados (76)

8. caixa geral de aposentações (71)

9. reformas antecipadas (69)

10. reforma militares (55)

11. lista aposentados (50)

12. pensões (40)

13. fundo de compensação trabalho (30)

14. pensão (25)

15. lista de aposentações (25)

16. reforma gnr (23)

17. estatuto da aposentação (23)

18. reformados (22)

19. pensão social regime não contributivo (21)

20. aposentaçoes (21)

The queries are objective searches based on keywords from the Retirement Statute such as “reformas antecipadas”, “idade da reforma”, and “valor mínimo das pensões da reforma”. On the current search engine, the result of these queries is the article numbers where the words match the content. This project aims to return more objective data with the possibility to ask questions to reach a simple and detailed text response.

The present document[[1]](#footnote-2) is divided into two main sections:

* **Data Structure and Transformation**, the final structure in which the law will be represented for the Law Simplification System (LSS);
* **Text Simplification**, processes to simplify the law text into smaller annotations or an entire text translation to a simpler text;

First, examples of already existing projects about each of the topics will be presented, followed by a more extensive array of concepts and techniques that might be useful for the scope of the project. Each section will then end with the discussion of the tools mentioned, analysing their main advantages and limitations. The ultimate goal is to be able to recommend the right methodology to pursue throughout the project.

# 2. Data Structure and Transformation

Since the original structure of the law is in text articles and free text is not a suitable data structure to feed the LSS, this section will cover the main concepts and technologies that are able to structure the law into a “machine-friendly” format.

## 2.1 Techniques

### 2.1.1 Knowledge Graphs

Knowledge graphs transforming the law to a more structured format has been developed in several projects and for different applications. One of such project is the Lynx project (<https://www.lynx-project.eu/>).

The Lynx project aims to represent Compliance Services law in annotations, understanding the relations between conditions, subjects, actions, and variables. It maps them to (<https://www.lynx-project.eu/> a Legal Knowledge Graph (LKG) for Multilingual Compliance Services, that integrates heterogeneous compliance data sources including legislation, case law, standards, and other private contracts.

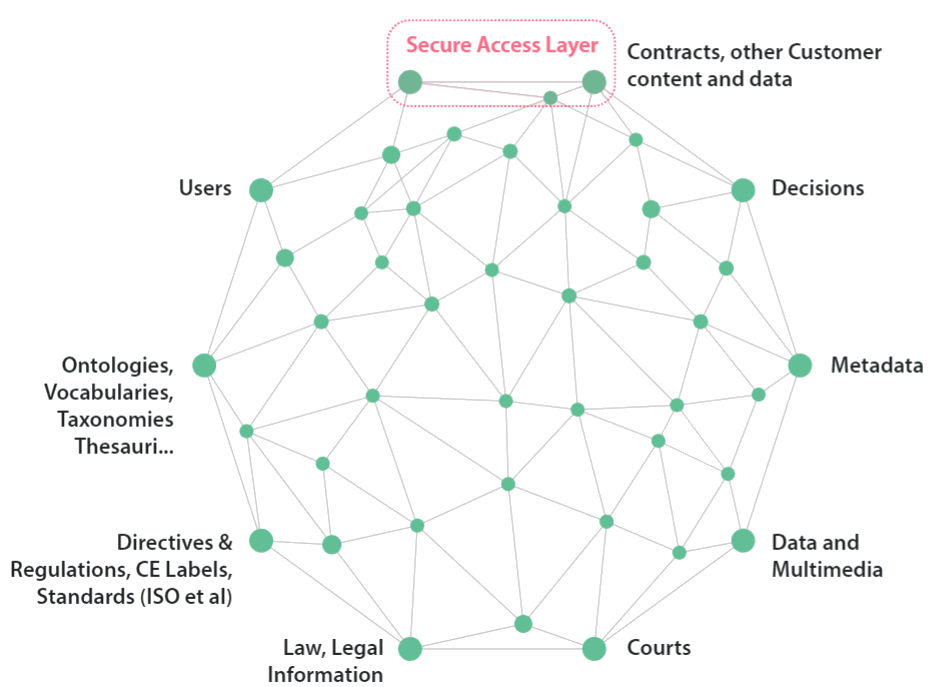


Figure 2: Legal Structure Knowledge Graph | Source: <https://www.lynx-project.eu/>

The project will identify the major legal resources necessary to provide exploitable compliance services. The referred legal knowledge graph is a collection of structured data and unstructured documents that are densely interlinked. These documents shall be variate in their language. Terminologies and language resources will also be included in order to provide multilingual services. Semantic Web technologies favour the distributed publication of documents and data and provide the technology to announce and traverse the links. [16].

Regarding the extraction of information from the legal text, the annotations [17] are performed automatically. The solution consists of the extraction of semantic information and a tagging to semantically annotate documents.

The extraction of modificatory provisions is illustrated as a three-step process. First, the relevant excerpts of text are retrieved, then parsed, and finally the resulting parse trees are mapped onto the appropriate semantic frame. [17]

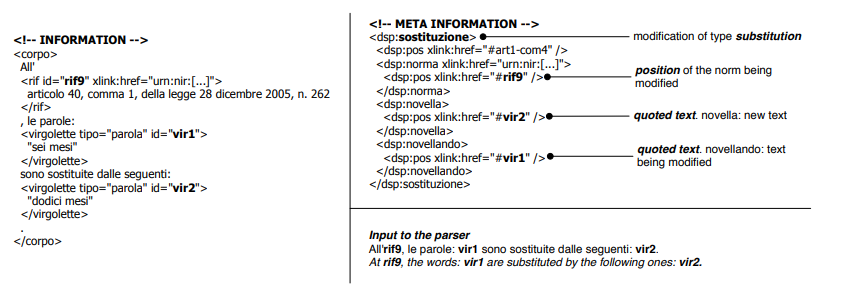


Figure 3: Annotations developed for the Italian law | Source: [17]

The knowledge graph (KG) represents a collection of interlinked descriptions of entities – real-world objects, events, situations, or abstract concepts – where:

* Descriptions have a formal structure that allows both people and computers to process them efficiently and unambiguously;
* Entity descriptions contribute to one another, forming a network, where each entity represents part of the description of the entities, related to it. [2]

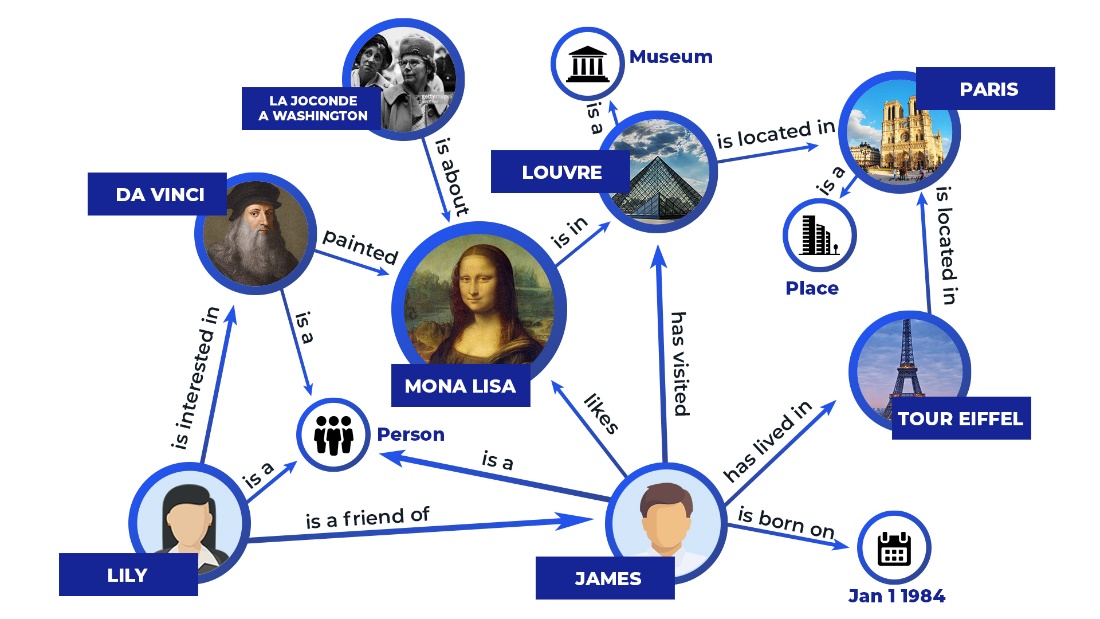


Figure 4: Example of Knowledge Graph | Source: <https://yashuseth.blog/2019/10/08/introduction-question-answering-knowledge-graphs-kgqa/>

Given the scope of the project, a Knowledge Graph could be used to connect articles given their references and similarities, to assign topics to each article or, in a more detailed way, to connect subject and condition inside articles. This data structure has the advantage to provide a good visualization of the connections, giving the ability to better identify a group of topics/problems in which the law is based on.

To build this kind of structure and retrieve the benefits associated with it, the standard engine used by the community, Neo4j, will be used. Neo4j is presented below.

#### Neo4j

Neo4j is a graph database engine, that is well-suited to irregular and complex structures such as the law, which can be used to connect data efficiently build knowledge graphs, among other uses. The engine allows the user to define nodes, with a specific type, and parameters, such as a node of type “Topic” with the parameters name and keywords. Each node can have one or multiple relations with other nodes, being the relationship also customizable.

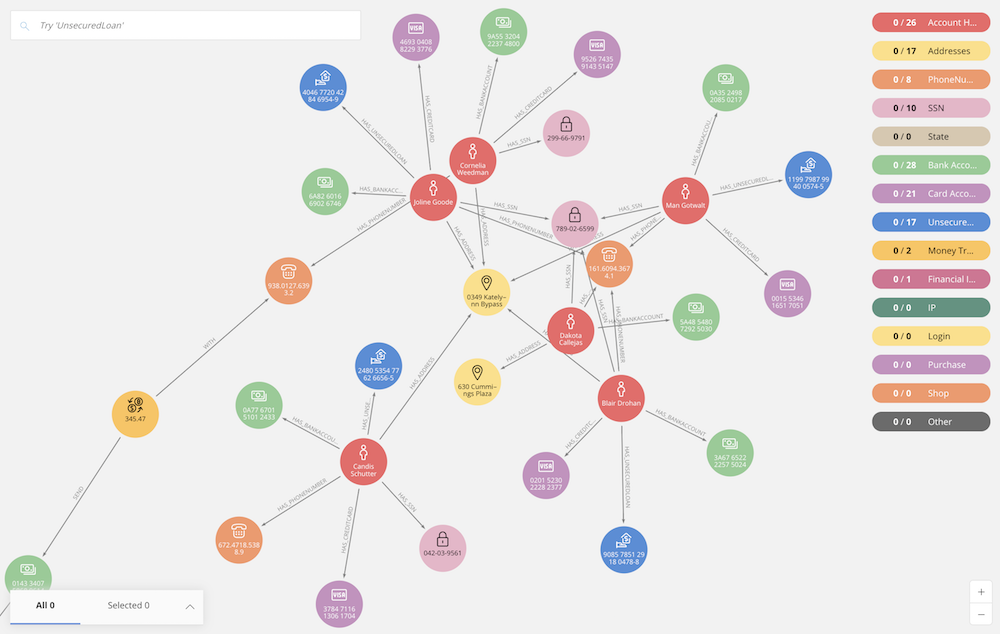


Figure 5: Neo4j Bloom application |Source: <https://neo4j.com/blog/neo4j-bloom-1-1-release/>

Neo4j has a visualization tool named Bloom (<https://neo4j.com/bloom/>) that enables us to visualize the database content without additional configurations. The tool offers a visualization of the nodes and relationships of the knowledge graph, being also able to perform queries to the data.

### 2.1.2 Web Ontology Language (OWL)

The development of an ontology is one of the techniques that can complement Knowledge Graphs in structuring free text. An Ontology captures the entities, their types (classes), class hierarchy, properties, and relationships between the entities. An Ontology has instances/individuals. The data is stored in the form of nodes and edges, i.e., each data point is a node, and the relationship becomes an edge that connects those nodes. An ontology is a way of showing the properties of a subject area and how they are related, by defining a set of concepts and categories that represent the subject [3].

An ontology usually deals with concepts, not instances of concepts. So, it could contain the relationship that a “cow” is owned by a “person” and that a “person” has a “profession”, but not the “fact” that the cow with registration number “B103245” is owned by a “farmer” with the name "John Doe”. In other words, an Ontology records the structure/schema whereas a Knowledge Graph captures the data. Moreover, an ontology has the power to enable restrictions/rules and reasoning [3].

An ontology could be represented in a knowledge graph. If an ontology is represented with a knowledge graph, it is possible to extend this knowledge graph with facts that are extracted from some source (textual or otherwise) and could help in the interpretation of the data.

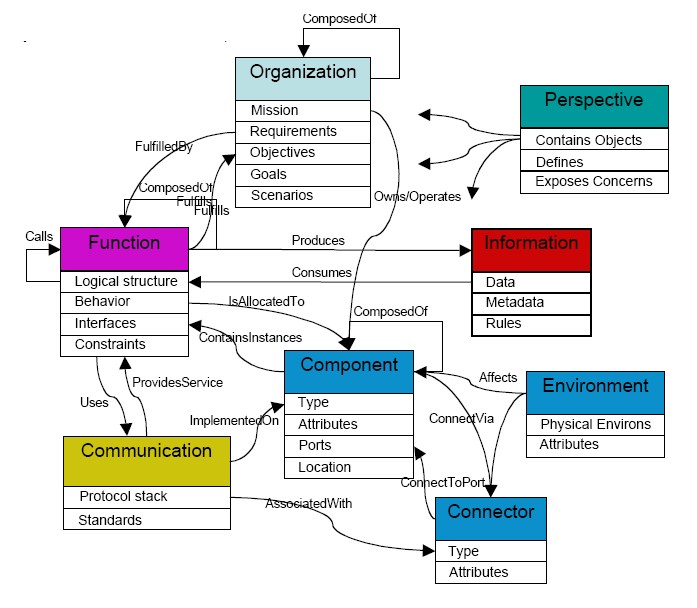


Figure 6: Example of ontology structure | Source: <https://en.wikipedia.org/wiki/Ontology_engineering>

The W3C Web Ontology Language (OWL) is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things, allowing the user to create an ontology structure.



Figure 7: Architecture of OWL 2 Ontology | Source: <https://www.w3.org/2007/OWL/draft/ED-owl2-overview-20090416/>

### 2.1.3 Decision Tree

A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including the chance of outcomes from events, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements. The nodes are conditions, and the two branches are the path to follow if the condition is true or false. The last nodes (leaves) are the results of the initial problem.

For instance, the tree below tries to predict the risk of a given user developing kidney stones, based on his or her characteristics.

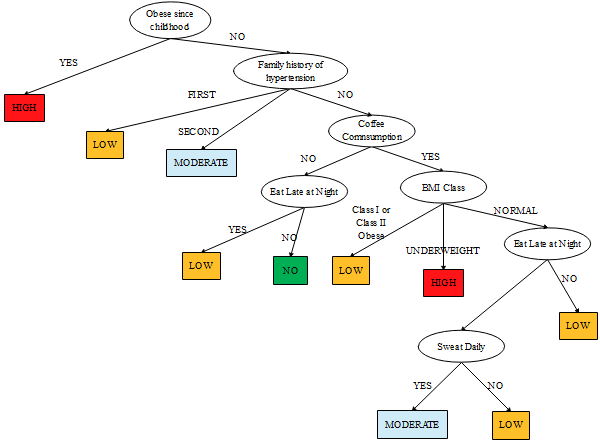


Figure 8: Decision tree for risk of kidney stones | Source: [4]

For the project, the decision tree conditions can be based on the keywords/topics matched with the processed user queries and also with the natural conditions that can be found in the law. The concept is to identify questions and problems that the application will solve and map them to the diverse solutions, such as text or a math formula, with conditions. For this structure to work each article should be analysed and inserted in a topic/problem (branch of the tree) with all the conditions identified along the articles that are part of this problem. The final leaves are the responses to the tree branch problem identified, given the possible inputs asked during the passage thought the tree branches.

### 2.1.4 Probabilistic Latent Semantic Analysis

Probabilistic Latent Semantic Analysis (PLSA) is a statistical technique for the analysis of co-occurrence of data and node relationships in a graph structure, which has applications in information retrieval and filtering in natural language processing.

PLSA is a technique of analysing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. The result is a collection of related documents.

Compared to standard latent semantic analysis which downsizes the occurrence tables (usually via a singular value decomposition), probabilistic latent semantic analysis is based on a mixture decomposition derived from a latent class model [5].

For instance, in the figure below, subreddits related to the 2016 US presidential campaign. The topics are on the triangle vertices and the subreddits are to a given distance relative to the vertices equal to the topic importance on the subreddit.

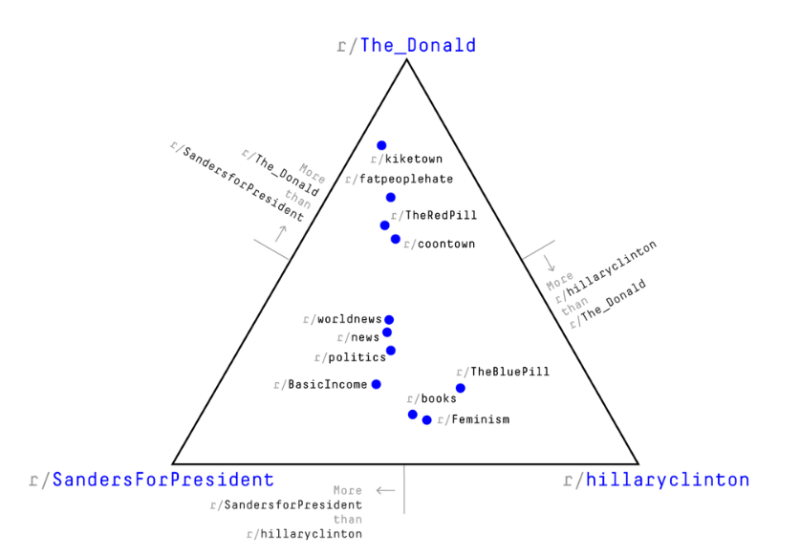


Figure 9: Comparing subreddits with Latent Semantic Analysis | Source: <https://www.r-bloggers.com/comparing-subreddits-with-latent-semantic-analysis-in-r/>

Documents are compared by taking the similarity between all the words, tuples and triples contained in the documents. Values close to 1 represent very similar documents while values close to 0 represent very dissimilar documents [5].

This technique can be used to group the article numbers. The classification into a group sets a view of the correlation of the article numbers. Great isolation of groups can turn the group number into an article number feature. The final application can then filter results based on a feature built with this technique.

## 2.2 Discussion

Although chatbots with similar legal use cases were not found, chatbots in the medical field based on Knowledge Graphs were considered to understand the interaction with the data structured.

[19] presents a chatbot with a so-called hybrid model, which consists of a knowledge graph and a text similarity model. Based on this chatbot framework, an online question-and-answer (QA) Healthcare Helper system for answering complex medical questions was built. The solution maintains a knowledge graph constructed from medical data collected from the Internet and also implements a novel text representation and similarity deep learning model, Hierarchical BiLSTM Attention Model (HBAM), to find the most similar question from a large QA dataset.

[20] proposes a knowledge-based conversational chatbot for answering medical questions. A relevant characteristic of this system is represented by the usage of Knowledge Graphs to formally represent textual inputs given by the user as well as templates of questions and, contextually, efficiently navigate and use the domain knowledge of interest to provide an answer. The architecture of the system is presented below.

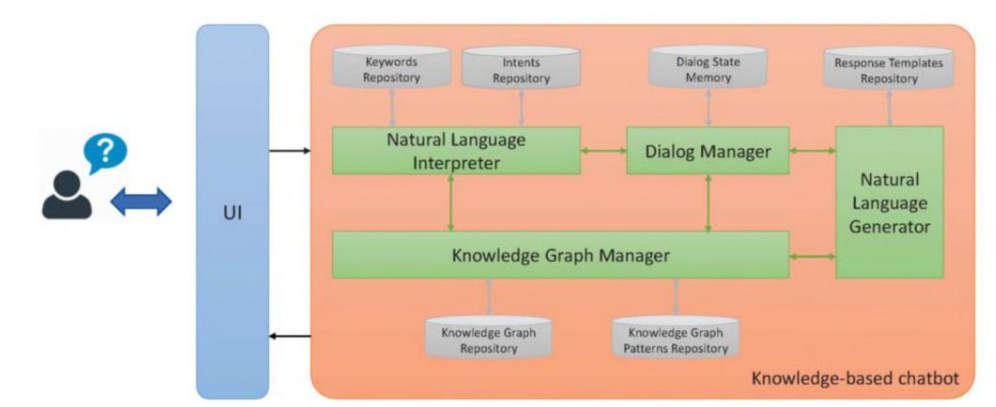


Figure 10: Components of a proposed conversational chatbot | Source: [20]

To efficiently handle the dialog, the system asks for missing information and generates more precise and contextualized responses. Exploiting this structuration of knowledge, the proposed chatbot can formally represent textual inputs given by the user as well as the templates of expected questions. This way, it can contextually and efficiently use the domain knowledge of interest to provide an answer [15].

Among other works a Knowledge Graph is a common piece in the system architecture, however, the technique to match the queries with the KG content differ, being common the use of deep learning techniques to perform this match or a content extraction with NLP techniques from the queries to match identical nodes in the KG.

# 

# 3. Question Answering

Question Answering (QA) is a field of Natural Language Processing and Computer Science with the aim to build a system which automatically answer questions produced by humans in a natural language.

There are 2 types of systems, Closed-Domain QA and Open-Domain QA.

**Closed-Domain QA** is about building systems that answer questions from a specific domain and the questions are usually restricted to be descriptive. For example, say we used a dataset of the Apollo mission with the documents about the moon rocks as the input to our closed-domain QA system, we could ask it *‘What is the composition of the lunar rocks?’ [24],* but we couldn’t ask it something like *‘Why do we dream?’.* The idea being that the closed-domain QA systems are exposed to much smaller datasets from which to extract the answer, which naturally restricts the range of questions that can be asked. The answers to this system are clearly present in the dataset.

**Open-domain** **QA** on the other hand deals with questions about nearly anything and can rely on general ontologies and world knowledge. However, these systems usually have much more data available from which to extract the answers. They can answer questions like *“What is the meaning of life?”*, *“Are we a simulation?”* (and other types of open questions) and are the types of systems that virtual assistants use [25].

Given the dataset and question types in the scope of this project, some techniques will be presented as potential methods for this work.

## 3.1 Techniques

**3.1.1 Dynamic Memory Networks**

Dynamic Memory Network (DMN) is a neural network-based framework for general question answering tasks. It achieves state-of-the-art performance in various QA and NLP tasks, and the modular architecture of the DMN allows it to implement various kinds of QA systems.

The architecture of an DMN with its five modules is presented in the following figure.

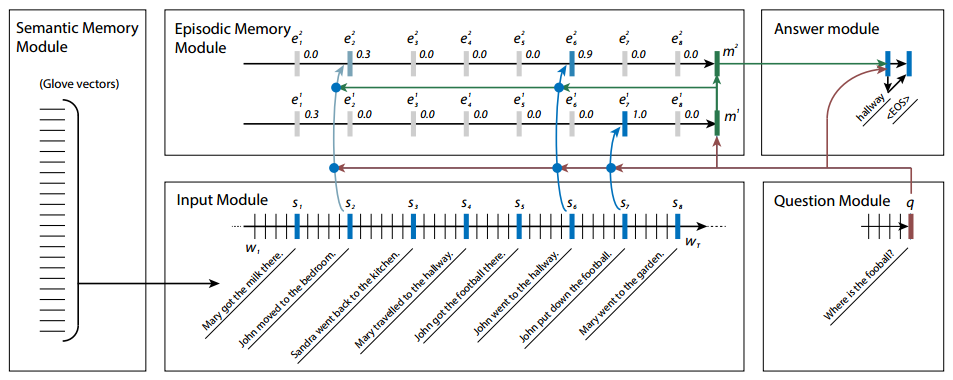


Figure 11: Dynamic Memory Network architecture | Source: [26]

The five modules of the DMN are: Semantic Memory Module, Input Module, Question Module, Episodic Memory Module and Answer Module.

The **semantic memory module** transforms the words in numbers. For the most part machine learning algorithms can only work with numbers and it is the job of the Semantic Memory Module to provide a mathematical representation of the input data, this is called word vectorization/word embedding.

The **input module** provides an embedded representation of the context. This means that the meanings of the words and sentences are captured by fixed-length vectors, called ‘facts’. A ‘fact’ is defined as an embedded representation of a sentence in a context. So if our context has 4 sentences, the Input Module will output 4 facts.

Similarly, to the Input Module, the **question module** takes in questions and returns a question embedding vector. Given that questions are usually one sentence long, there is only one question embedding vector.

The **episodic memory module** conducts multiple passes over the input data. On each pass, sentence embedding from the input module are fed as input to the neural network in the episodic memory module. Here each sentence embedding is assigned a weight corresponding to its relevance to the question being asked.

Finally, the **answer module** generates an answer given the final memory output vector from the episodic memory module. Once again there are many kinds of answer modules implementations. For single-word answers, a single layer feed-forward neural net is a viable option.

It is the modularity that allows the DMN to be used as a framework for general Question Answering.

### 3.1.2 Bidirectional Encoder Representations

Bidirectional Encoder Representations (BERT), for the Question Answering task, takes the input question and passage as a single packed sequence. The input embeddings are the sum of the token embeddings and the segment embeddings. The input is processed in the following way before entering the model:

1. Token embeddings: A [CLS] token (to classify) is added to the input word tokens at the beginning of the question and a [SEP] token (to separate sentences) is inserted at the end of both the question and the paragraph.
2. Segment embeddings: A marker indicating Sentence A or Sentence B is added to each token. This allows the model to distinguish between sentences. In the below example, all tokens marked as A belong to the question, and those marked as B belong to the paragraph.

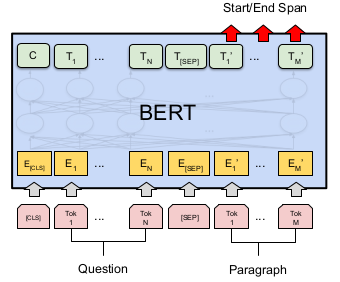


Figure 12: Question Answering with BERT | Source: [27]

To fine-tune BERT for a Question-Answering system, it introduces a start vector and an end vector. The probability of each word being the start-word is calculated by taking a dot product between the final embedding of the word and the start vector, followed by a softmax over all the words. The word with the highest probability value is considered [27].

## 3.2 Discussion

Question answering is very dependent on a good search corpus, without documents containing the answer, there is little any question answering system can do. It thus makes sense that larger collection sizes generally lend well to better question answering performance, unless the question domain is orthogonal to the collection. The notion of data redundancy in massive collections, such as the web, means that nuggets of information are likely to be phrased in many different ways in differing contexts and documents.

Techniques such as BERT [27] and DMN [26] for question answering are relatively recent and represent the state-of-the-art of this field, a field which has not the problem solved [28].

On this project, the dataset is small – one law statute – and the questions to this statute are objective. Having a more controlled model such as a Closed-Domain QA with simpler techniques may outperform the cutting-edge neural network-based techniques. Considering works with a similar task and using the Portuguese language, [29] uses an architecture with five major tasks:

1. indexing process, an off-line procedure by which a set of target documents is parsed in order to collect information in index files;
2. question analysis, since indexation is performed off-line, the question analyser is indeed the first module of our system. It receives as input a NL question submitted by the user, that is first lemmatized and morphologically disambiguated;
3. document retrieval, after analysing the question, a query is submitted to the index files using as search keys the pivot lemmas, their heads of derivation, their synonyms, the ontological domains and the question categories;
4. sentence retrieval, this module receives as input a set of documents, whose sentences that match the pivots are marked.
5. answer extraction, this module as input a set of scored sentences presumably containing answers and, in the end, outputs the answer with the highest score.

# 4. Text Simplification

This section presents the concepts and technologies used to simplify free text. Text Simplification is the task of reducing the complexity of the vocabulary and sentence structure of the text while retaining its original meaning, to improve readability and understanding. Simplification has a variety of important societal applications, for example increasing accessibility for those with cognitive disabilities such as aphasia, dyslexia, and autism, or non-native speakers and children with reading difficulties [6]. The text that will be used in this project is a Portuguese legal text, which is formally written and can be very complex. Below we present an example of the text:

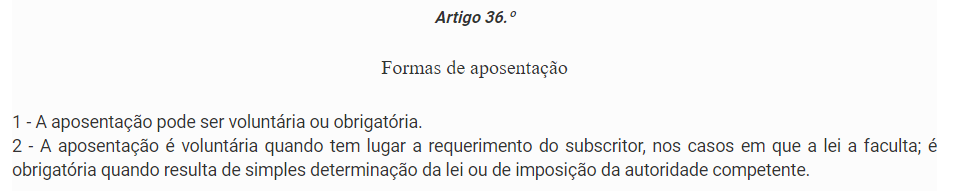


Figure 13: Example of a legal text structure to analyse | Source: <https://dre.pt/web/guest/legislacao-consolidada/-/lc/134309077/202008141636/73825467/diploma/indice?q=498%2F72>

Given the complexity of the text, we will review tools such as Text Summarization, Parts of Speech extraction, Semantic Representation, Neural Text Simplification, Bidirectional Encoder Representations and OpenAI GPT-3 to simplify it, allowing us to present legal simplified text based on article numbers to a user of the output application.

## 4.1 Techniques

### 4.1.1 Text Summarization

One way to simplify text is to summarize it. While text summarization can highlight the information that a user is looking for, the corresponding loss of information may make it unsuitable for legal text if specific information such as the case of legal exceptions is removed, but it can be beneficial to highlight the useful article content.

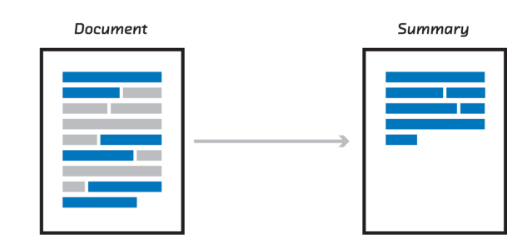


Figure 14: Text summarization | Source: <https://mc.ai/text-summarization/>

There are two types of summarization, extractive, and abstractive summarization.

Extractive methods attempt to summarize articles by selecting a subset of words that retain the most important points. This approach weights the important part of sentences and uses it to form the summary. Different algorithms and techniques are used to define weights for the sentences and further rank them based on importance and similarity among each other.

Abstractive methods select words based on semantic understanding, even if those words did not appear in the source documents. They interpret and examine the text using advanced natural language techniques in order to generate a new, shorter text that conveys the most critical information from the original text.

It is closer to the way human reads and then summarizes I a text using their own words [7][8].

Models for abstractive summarization fall under deep learning techniques. The recent developments are being made by Facebook AI research [9], Google Brain [10], and IBM Watson [11].

### 4.1.2 Parts of speech extraction

Parts of speech extraction is a technique where each word (often called *token*) of a sentence or many sentences is labelled, usually with grammatical descriptions, such as Noun, Adjective, Adverb. They can often get quite specific, also distinguishing, for example, between types of nouns (proper nouns etc).

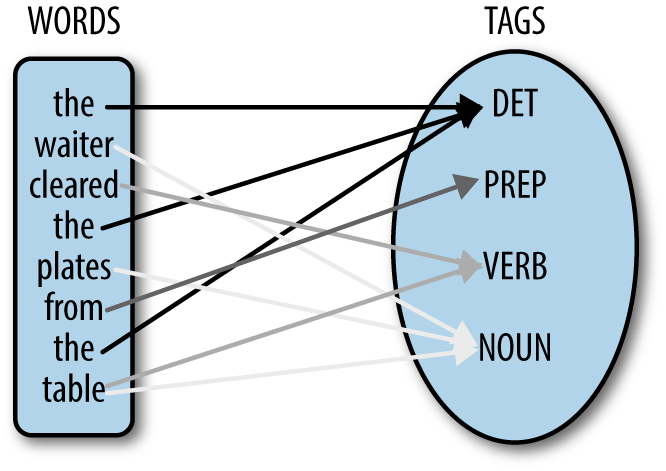


Figure 15: Example of parts of speech extraction | Source: <https://learn.g2.com/text-mining>

It is possible to then use these descriptions of the tokens as input to a model or to filter the tokens to extract only the interesting parts and build automatic annotations [12].

This technique can be used to build annotations for each article number. Given annotations such as conditions, events, variable declarations, and actions, tagging the article number’s words and defining a list of common structures for each annotation is achievable to build a parser that identifies these annotations.

### 4.1.3 Semantic Representation

UCCA (Universal Cognitive Conceptual Annotation) is a semantic annotation scheme rooted in typological and cognitive linguistic theory. It aims to represent the main semantic phenomena in the text, abstracting away from syntactic forms. UCCA has been shown to be preserved across translations and has also been successfully used for the evaluation of machine translation and for the evaluation of Text Simplification and grammatical error correction. Formally, UCCA structures are directed acyclic graphs whose nodes (or units) correspond either to the leaves of the graph or to several elements viewed as a single entity according to some semantic or cognitive consideration. [13]

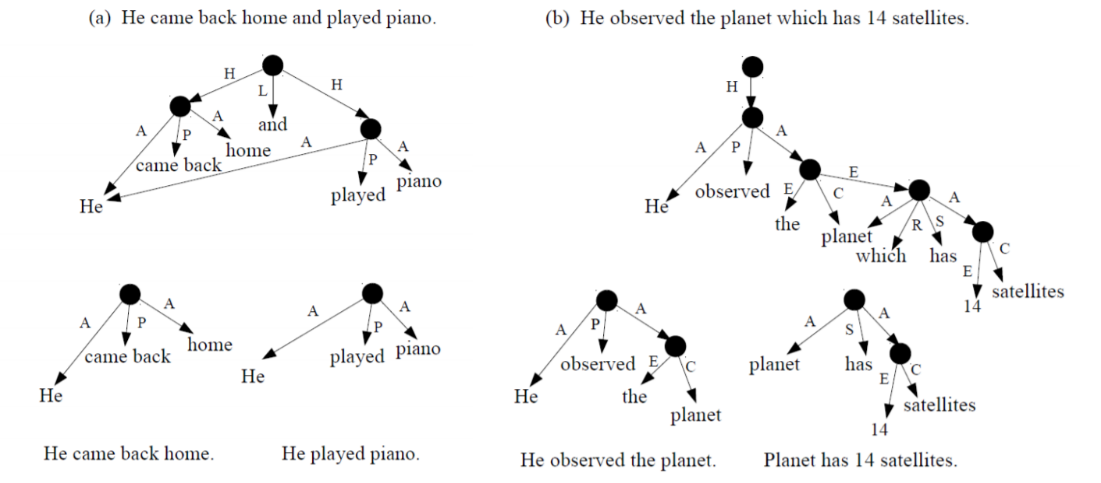


Figure 16: In both cases, the original sentence, the semantic parse, the extracted Scenes with the required modifications, and the output of the rules are presented top to bottom. The UCCA categories used are: Parallel Scene (H), Linker (L), Participant (A), Process/State (P/S), Center (C), Elaborator (E), Relator (R) | Source: [13]

On [13] for generating UCCA’s structures was used a parser for UCCA [14]. The parser uses an expressive set of transitions, able to support all structural properties required by the UCCA scheme. Its transition classifier is based on a neural network that receives a Bidirectional Long Short-Term Memory (BiLSTM) encoding of elements in the parser, given word embeddings – (where words or phrases from the vocabulary are mapped to [vectors](https://en.wikipedia.org/wiki/Vector_(mathematics))) and other features.

To replicate this method on the project the UCCA categories should be defined for the Portuguese language. This semantic representation would have two main results, the first is to give a simple response to the user of the application if is required such detailed text, the second is to simplify the results of other techniques such as Parts of Speech.

### 4.1.4 Neural Text Simplification

Neural Text Simplification (NTS) is the simplification of text with the usage of neural networks, also called neural to sequence models. The architecture of neural networks used is Recurrent Neural Networks (RNN), an architecture that has nodes not only connected from the previous layer to the next one but also with nodes in the same layer, creating a loop. These loops allow the network to use information from previous passes, which acts as memory. Inside the RNN types of architecture, the commonly used for text analysis is Long Short-Term Memory (LSTM). Neural sequence to sequence models has been successfully used in many applications, from speech and signal processing to text processing or dialogue systems. Neural machine translation is a particular type of sequence-to-sequence model that recently attracted a lot of attention from industry and academia, especially due to the capability to obtain state-of-the-art results for various translation tasks. Unlike classical statistical machine translation (SMT) systems, neural networks are being trained end-to-end, without the need to have external decoders, language models, or phrase tables. The architectures are relatively simpler and more flexible, making possible the use of character models or even training multilingual systems in one go. [15]

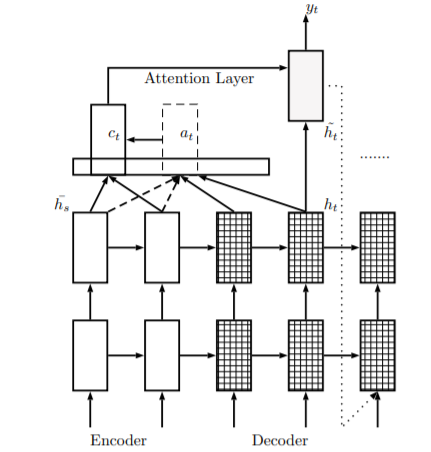


Figure 17: Example of a sequence-to-sequence model architecture | Source: [15]

Automated text simplification (ATS) systems, such as NTS, are meant to transform original texts into different (simpler) variants which would be understood by wider audiences and more successfully processed by various NLP tools. In the last several years, great attention has been given to addressing ATS as a monolingual machine translation problem translating from “original” to “simple” sentences [15].

### 4.1.5 Bidirectional Encoder Representations

Bidirectional Encoder Representations from Transformers (BERT) developed by Google [22] is a language representation model with a unique neural network architecture. A neural network like BERT needs to be pretrainned. Pretraining prior to BERT was limited to word embeddings that mapped each word to a vector with some aspects of its meaning. For example, "apple” to “green”, “red”, “fruit”, or even “seed”. The embeddings are trained on a massive unlabelled set of text, with all English Wikipedia, then stored in a library for use in a model to recognize sentiment, allowing for models to achieve the knowledge from larger datasets without the time consumption.

Word embedding models are, in general, not very powerful. Available word embeddings are trained on very shallow language modelling tasks therefore, word embeddings are unable to capture combinations of words and context.

Language modelling is gathering the probability of a word occurring in a certain context for instance, “it’s raining dogs and \_\_\_”, a language model would output "cat" with the highest probability. Language models are usually trained in the manner we read, from left to right. They are given a sequence of words and must predict the next word.

For example, if the network is given the starter “She walked her”, it may construct the remaining portion to form the sentence “She walked her dog”. This way, language models get a feel for how the text is written and is particularly helpful when generating sentences.

Some more complex language models such as bidirectional LSTM learn to read forward and backward; that is, being able to predict ‘dog’ in “She walked her…?” and being able to predict ‘She’ in “dog her walked…?”. While this does strengthen the models’ familiarity with the text, it cannot recognize the significance of words that require context before and after the word.

However, BERT provides a new way of a modelling language. There is no need to train language models from left to right when there is no need to generate sentences, so instead of predicting the next word after a sequence of words like standard language models, BERT randomly covers words in a sentence and predicts them.

This way, instead of only predicting the next word based on the previous context, BERT learns more complex aspects of the language. [21].

BERT could be used for lexical simplification as shown in [23], this type of model would work given a text sentence and a word candidate to be replaced, the model would suggest their simpler alternatives of equivalent meaning as sown in Figure 16.

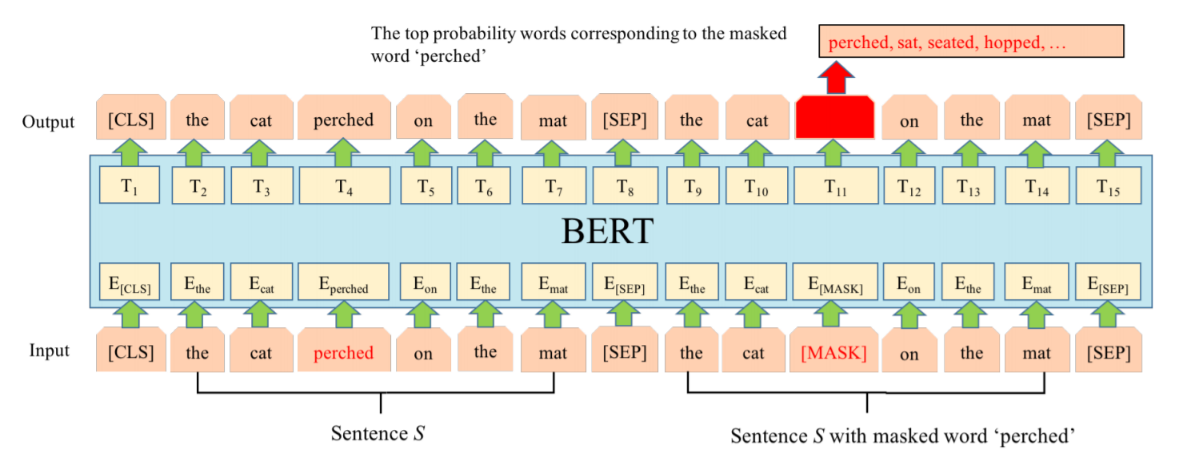


Figure 18: Substitution generation of BERT-LS for the target complex word prediction. The input text is ”the cat perched on the mat” with complex word ”perched”. [CLS] and [SEP] are two special symbols in BERT, where [CLS] is added in front of each input instance and [SEP] is a special separator token | Source: [23]

To use this technique, the model would need to be trained with the Portuguese language and the main goal would be to replace complex words in each sentence with their simpler alternatives of equivalent meaning.

### 4.1.6 OpenAI GPT-3

GPT-3, the third generative pre-trained transformer from OpenAI, is the largest natural language processing (NLP) transformer release to date, being a neural network with 175 billion parameters (the values that a neural network tries to optimize during training).

It consists of a tool to interpret text. It can translate complex text to a simpler one and respond to questions about the trained content.

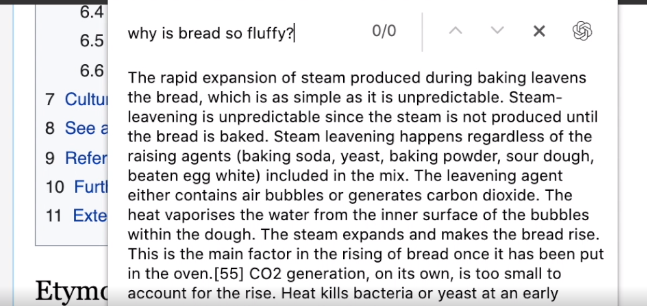


Figure 19: Example question and response to the GPT-3 model | Source <https://openai.com/blog/openai-api/>

This API provides a general-purpose “text in, text out” interface, allowing users to try it on virtually any English language task. It represents the best state of the art in the NLP field and, as so, it is a paid service. Furthermore, it only works for the English language at the moment (<https://openai.com/blog/openai-api/>). The model is not perfect and, in some cases, makes mistakes returning a bad output. GTP-3 makes the argument that many errors in GPT-3’s output can be fixed by fine-tuning the prompt.

## 4.2 Discussion

The first text simplification technique combining a structure which extracts the meaning of words (semantic structures) and neural machine translation is presented by [13]. The work shows that it outperforms existing lexical and structural systems. It uses the Universal Conceptual Cognitive Annotations (UCCA) which must be defined to each language, which in this case should be necessary to Portuguese.

The setup of the work has a Corpus, English Wikipedia text; a Semantic component, a parser trained on the UCCA-annotated wiki corpus; a Neural component; a comparison system of the results and the original text and the reporting of human and automatic evaluation.

The text simplification model training phase can be done using supervised or unsupervised learning. Supervised learning is a technique in which the model already knows the output data, the result of the simplified text. Unsupervised learning does not require the expected output to learn how to achieve the output, on text simplification models means that is not necessary to have the simplified text result for a complex text.

Performing text simplification without the extraction of semantic knowledge in the text is shown in [18] and presents the first attempt towards unsupervised neural text simplification that relies only on unlabelled text corpora, text without the label corresponding to the expected result. The core framework is composed of an encoder and a pair of attentional-decoder. The framework is trained using unlabelled text collected from English Wikipedia. The analysis (both quantitative and qualitative involving human evaluators) on public test data shows that the proposed model can perform text-simplification at both lexical and syntactic levels, competitive to existing supervised methods. It also outperforms viable unsupervised baselines. Adding a few labelled pairs helps improve the performance further. [18]

# 5. Conclusion

The process of building a chatbot system or an auto-generated FAQ for the simplification of the law can be divided into three main phases.

The first is to map the law into a data structure. Since the law has multiple references and complexity, a graph or tree-based structure seems to be best suited. A Knowledge Graph that contains conditions and references to other articles of the law is a solution that satisfies the requirements for the structure of the data.

Legal text can be mapped to various structures, although mainly in a graph structure, this structure has already converged to an optimal state when it comes to structure nodes and relationships. The construction of this data structure can be done using a manual interpretation and analysis of the law or using text mining methods such as topic modelling. The fact that the law is rich in references and written in Portuguese makes it more difficult to automate the text interpretability task.

Question Answering and Text Simplification are fields in constant development. The most recent solutions use deep learning techniques for both TS and QA, and they require a large amount of data, BERT used the English Wikipedia for training, which makes it difficult to use these techniques in the project to be developed given the size disparity for training with legal text data in Portuguese. To perform QA, neural network-based architectures should be tested, however, and considering the data available for the project, an architecture with more traditional techniques may outperform the new cutting-edge methods. To perform Text Simplification on Portuguese legal documents, a solution to consider is a combination of techniques, by combining semantic representation with traditional NLP techniques. Deep learning techniques are not excluded, but they are dependent on expensive resources such as large amounts of data and processing power.

TS can be seen as an optional feature for the system, however, the presentation of text simplification for the final user can be a useful feature to introduce, mainly for questions that do not have direct answers and are more dependent on the original law text. Related work shows that the most recent developments in text simplification use Neural Text Simplification (NTS), which consists of recurrent neural networks with large and complex architectures that require large resources, making it a task developed mainly by big companies or research groups. Text Simplification does not depend only on NTS and because the law articles are fundamentally short but complex, combining techniques such as semantic representation and text simplification, with the identification of complex terms and their replacement with synonymous defined by the lawmakers, makes a viable solution.

As an automatic validation task, questions should be selected from the current search engine's question history and have their solution/answer attributed, after which it is possible to create a model evaluation metric.

In the first delivery of the chat system, asking the end-users if the result was useful will certainly be an excellent policy to implement in order to retrain the model.

# References

1. “Estatuto da Aposentação Decreto-Lei n.º 498/72”, consulted on August 19th in <https://dre.pt/web/guest/legislacao-consolidada/-/lc/34576175/view?q=498/72>
2. Stokman, F. N., & de Vries, P. H. (1988). Structuring knowledge in a graph. In *Human-computer interaction* (pp. 186-206). Springer, Berlin, Heidelberg.
3. McGuinness, D. L., & Van Harmelen, F. (2004). OWL web ontology language overview. *W3C recommendation*, *10*(10), 2004.
4. Oladeji, F. A., Idowu, P. A., Egejuru, N., Faluyi, S. G., & Balogun, J. A. (2019). Model for Predicting the Risk of Kidney Stone using Data Mining Techniques.
5. Hofmann, T. (2013). Probabilistic latent semantic analysis. *arXiv preprint arXiv:1301.6705*.
6. Martin, L., Fan, A., de la Clergerie, É., Bordes, A., & Sagot, B. (2020). Multilingual Unsupervised Sentence Simplification. *arXiv preprint arXiv:2005.00352*.
7. Bharti, S. K., & Babu, K. S. (2017). Automatic keyword extraction for text summarization: A survey. *arXiv preprint arXiv:1704.03242*.
8. Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). Text summarization techniques: a brief survey. *arXiv preprint arXiv:1707.02268*.
9. Facebook AI Research, “Making Transformer networks simpler and more efficient” consulted on August 19th in <https://ai.facebook.com/blog/making-transformer-networks-simpler-and-more-efficient/>
10. Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104-3112).
11. Nallapati, R., Zhou, B., Gulcehre, C., & Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.
12. Santorini, B. (1990). Part-of-speech tagging guidelines for the Penn Treebank Project.
13. Sulem, E., Abend, O., & Rappoport, A. (2018). Simple and effective text simplification using semantic and neural methods. *arXiv preprint arXiv:1810.05104*.
14. Hershcovich, D., Abend, O., & Rappoport, A. (2017). A transition-based directed acyclic graph parser for UCCA. *arXiv preprint arXiv:1704.00552*.
15. Nisioi, S., Štajner, S., Ponzetto, S. P., & Dinu, L. P. (2017, July). Exploring neural text simplification models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 85-91).
16. Montiel-Ponsoda, E., Gracia, J., & Rodríguez-Doncel, V. (2018). Building the legal knowledge graph for smart compliance services in multilingual Europe. In *CEUR workshop proc.* (No. ART-2018-105821).
17. Lesmo, L., Mazzei, A., & Radicioni, D. P. (2009, June). Extracting semantic annotations from legal texts. In *Proceedings of the 20th ACM conference on Hypertext and hypermedia* (pp. 167-172).
18. Surya, S., Mishra, A., Laha, A., Jain, P., & Sankaranarayanan, K. (2018). Unsupervised neural text simplification. *arXiv preprint arXiv:1810.07931*.
19. Bao, Q., Ni, L., & Liu, J. (2020, February). HHH: An Online Medical Chatbot System based on Knowledge Graph and Hierarchical Bi-Directional Attention. In *Proceedings of the Australasian Computer Science Week Multiconference* (pp. 1-10).
20. Minutolo, A., Esposito, M., & De Pietro, G. (2017, September). A Conversational Chatbot Based on Kowledge-Graphs for Factoid Medical Questions. In *SoMeT* (pp. 139-152).
21. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).
23. Qiang, J., Li, Y., Zhu, Y., Yuan, Y., & Wu, X. (2019). A Simple BERT-Based Approach for Lexical Simplification. *arXiv preprint arXiv:1907.06226*.
24. Woods, W. A., & WA, W. (1977). Lunar rocks in natural English: Explorations in natural language question answering.
25. Chen, D., Fisch, A., Weston, J., & Bordes, A. (2017). Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.
26. Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., ... & Socher, R. (2016, June). Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning* (pp. 1378-1387).
27. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
28. Soares, M. A. C., & Parreiras, F. S. (2020). A literature review on question answering techniques, paradigms and systems. *Journal of King Saud University-Computer and Information Sciences*, *32*(6), 635-646.
29. Amaral, C., Figueira, H., Martins, A., Mendes, A., Mendes, P., & Pinto, C. (2005, September). Priberam’s question answering system for Portuguese. In *Workshop of the Cross-Language Evaluation Forum for European Languages* (pp. 410-419). Springer, Berlin, Heidelberg.

1. Draft document. [↑](#footnote-ref-2)