NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

BSc THESIS

Development of machine learning software for the recognition of modifications in Greek legislation

Christos D. Skoulikas-Androutsos
Viktor T. Tsakalakis

Supervisor: **Manolis Koubarakis,** Professor

ATHENS

JUNE 2020

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη λογισμικού μηχανικής μάθησης για αναγνώριση τροποποιήσεων στην ελληνική νομοθεσία

Χρήστος Δ. Σκουλίκας-Ανδρούτσος
Βίκτωρ Θ. Τσακαλάκης

Επιβλέπων: **Μανόλης Κουμπαράκης,** Καθηγητής

ΑΘΗΝΑ

ΙΟΥΝΙΟΣ 2020

BSc THESIS

Development of machine learning software for the recognition of modifications in Greek legislation

Christos D. Skoulikas-Androutsos
S.N.: 1115201200164
Viktor T. Tsakalakis
S.N.: 1115201200182

SUPERVISOR: **Manolis Koubarakis,** Professor

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη λογισμικού μηχανικής μάθησης για αναγνώριση τροποποιήσεων στην ελληνική νομοθεσία

Χρήστος Δ. Σκουλίκας-Ανδρούτσος
Α.Μ.: 1115201200164
Βίκτωρ Θ. Τσακαλάκης
Α.Μ.: 1115201200182

ΕΠΙΒΛΕΠΟΝΤΕΣ: **Μανόλης Κουμπαράκης,** Καθηγητής

ABSTRACT

The ΝΟΜΟΘΕΣΙ@ projects aims to convert Greek legislation into a machine readable and queryable format. One important aspect that it is currently missing is the identification of legislative modifications and their semantic components. In this disseration we present an automated solution based on deep learning, in particular the BiLSTM architecture. Our model demonstrates remarkably good results, with a prediction accuracy reaching over 98% per lexical token.

SUBJECT AREA: Natural Language Processing
KEYWORDS: deep learning, neural networks, legislative modifications

ΠΕΡΙΛΗΨΗ

Το πρόγραμμα ΝΟΜΟΘΕΣΙ@ αποσκοπεί στην μετατροπή της ελληνικής νομοθεσίας σε μηχανικά αναγνώσιμη μορφή που επιδέχεται ερωτήσεις. Μία σημαντική λειτουργία που δεν υφίσταται ακόμη είναι η ταυτοποίηση των νομοθετικών τροποποιήσεων και των σημασιακών του μερών. Σε αυτή την πτυχιακή εργασία παρουσιάζουμε μία αυτοματοποιημένη λύση βασισμένη στην βαθιά μάθηση, συγκεκριμένα την αρχιτεκτονική BiLSTM. Δείχνει αξιοσημείωτα καλά αποτελέσματα, με ακρίβεια προβλέψεων να φτάνει το 98% ανά λεκτική μονάδα.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Επεξεργασία Φυσικής Γλώσσας

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: βαθιά μάθηση, νευρωνικά δίκτυα, νομοθετικές τροποποιήσεις

We would like to thank the ΝΟΜΟΘΕΣΙ@ team for their valuable help and support and especially Iosif Angelidis and Christos Papaloukas.

CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

The ΝΟΜΟΘΕΣΙ@ [1] platform is an initiative by a research team of the Department of Informatics and Telecommunications of the National and Kapodistrian University of Athens [1].The project aims to convert the issues of the Government Gazette of Greece (Φύλλα Εφημερίδας της Κυβερνήσεως, ΦΕΚ) into a machine-readable format and to publish them to a website, allowing semantic queries over the body of law of the Hellenic Republic. In this thesis we focus on modifications of laws in particular, and we use modern machine learning technologies to automate their recognition, categorisation and the annotation of their component parts [2] [3].

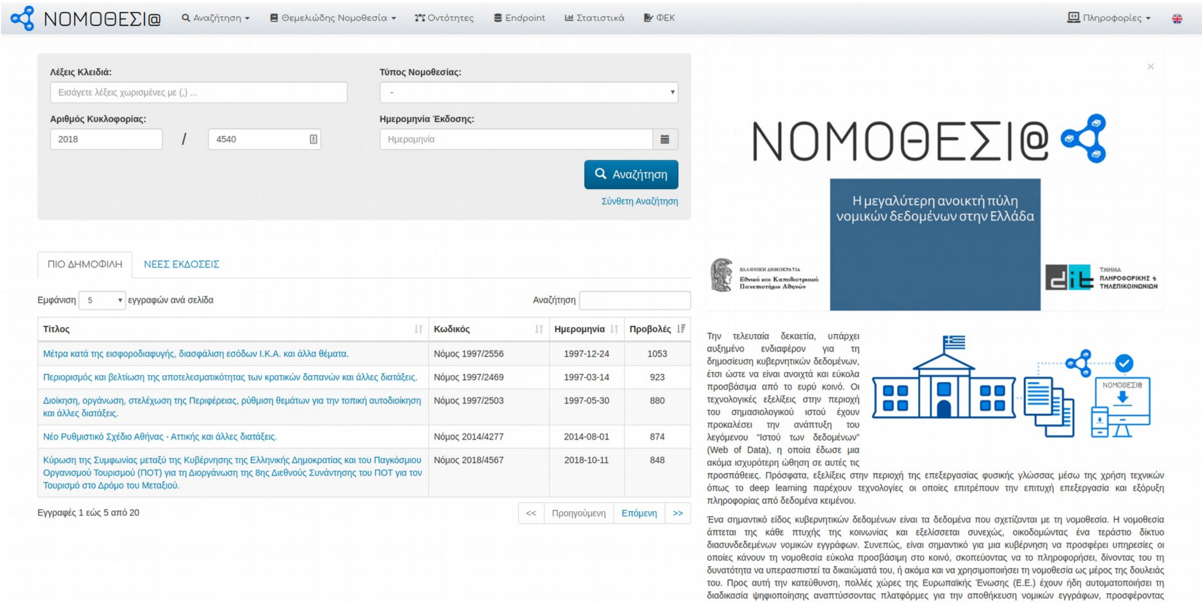

Illustration 1: Screenshot of the Nomothesi@ website

The necessity of encoding a nation’s body of law is urgent in this day and age. Making legislation directly available and accessible to all citizens facilitates their interactions with the state and their employers, encourages their participation in public affairs and reinforces the institution of democracy. Furthermore, having this legal information encoded in such a way as to be accessible through standardised APIs creates new options in the field of informatics in allowing for the development of applications and other tools based on an existing platform. Such apps can be fitted to any public need and cover each specific use case, compounding on the utility of the original platform. As an example, there are already several applications on the market which can offer automated legal aid to citizens and corporations of the United States of America.[2] [3] [4]

1 http://legislation.di.uoa.gr/
2 https://www.lawbot.info/
3 https://donotpay.com/
4 https://rossintelligence.com/

Illustration 2: Screenshot of the DoNotPay website, ‘the world’s first robot lawyer’

The European Union, in recognition of this rising need, has created the European Legislation Identifier (ELI) [5] , a system for digital distribution of European legislation in a standardised format so that they can be accessed, exchanged and reused within all member nations. ΝΟΜΟΘΕΣΙ@ inherits from the ontology developed for ELI and endeavors to bring the same benefits to Greek law as well [4].

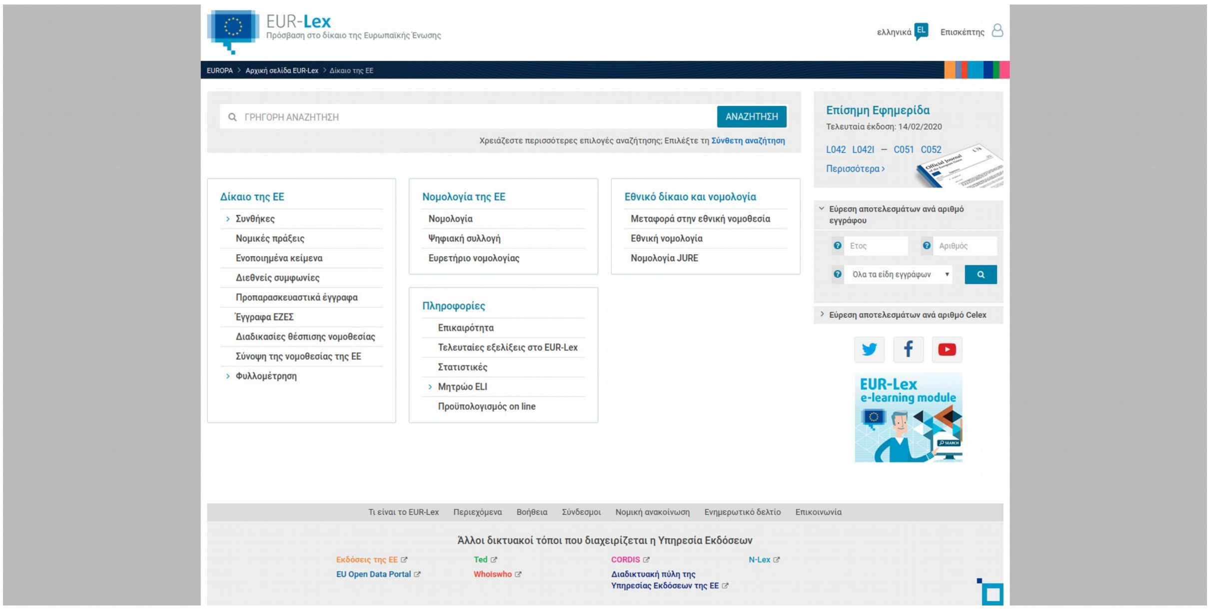

Illustration 3: Screenshot of the Eur-Lex website

5 https://eur-lex.europa.eu/eli-register/about.html?locale=el

Every system of law which is in active use has to evolve over time to keep up with changes in the goals, social norms or material conditions of the society it is written for. As such, the capability to modify established laws is required. In the greek legislative system, as well as in many other republics, there is no specific defined process to allow that. Instead, the sole mechanism for modification is the introduction of new legislation which dictates the way in which an existing law is changed. This makes the automatic recognition of legislative modifications a non-trivial problem, and necessitates the development of specialised software.

It is our goal in this dissertation to present one such automated system which can recognise and annotate salient component parts of modifications with a sufficiently large success rate for useful application. Our implementation is based on the machine learning method known as deep learning.

In this chapter we have explained the problem we are trying to solve and why a solution would be valuable. In the next chapter we shall define some fundamental concepts, summarise the work which has taken place in the field of computer science in solving similar problems and describe the technological framework on which our project will be based.

2. PRIOR WORK

2.1. Background

2.1.1. NLP

Our problem belongs to the general category of Natural Language Processing (NLP). NLP is the field concerning the interpretation of human speech and written text through mechanical means. Until the 1980s the methods used were based on sets of rules composed by human experts in linguistics, or statistical analysis. Following technological developments during that decade, the field of machine learning began to blossom and found broad application in NLP with great success.

2.1.2. Machine Learning

Machine learning is a subfield of artificial intelligence. The term refers to the development of systems capable of analysing data and solving problems without the need of predetermined algorithms leading to the solution. Based on the type of feedback that is supplied the learning process can be unsupervised, reinforced or supervised.

In unsupervised learning the system learns patterns in the input even though no explicit feedback is supplied. The most common unsupervised learning task is clustering: detecting potentially useful clusters of input examples.

In reinforcement learning the system learns from a series of reinforcements-rewards or punishments. For example the two points for a win at the end of a chess game tell the system that it did something right. But it is up to it to decide which actions prior to the reinforcement were most responsible for it.

In supervised learning the system is presented with some example input-output pairs and learns a function that maps from input to output. A common supervised learning task is classification e.g. classifying if an e-mail is spam or not.

2.1.3. Neural Networks

Neural Networks are a machine learning technology based on the structure of the artificial neuron, an abstract mathematical representation meant to broadly approximate the function of the biological neurons found in living beings. An artificial neuron has one or more weighted arithmetic inputs, on the combination of which it then applies an arbitrary activation function.

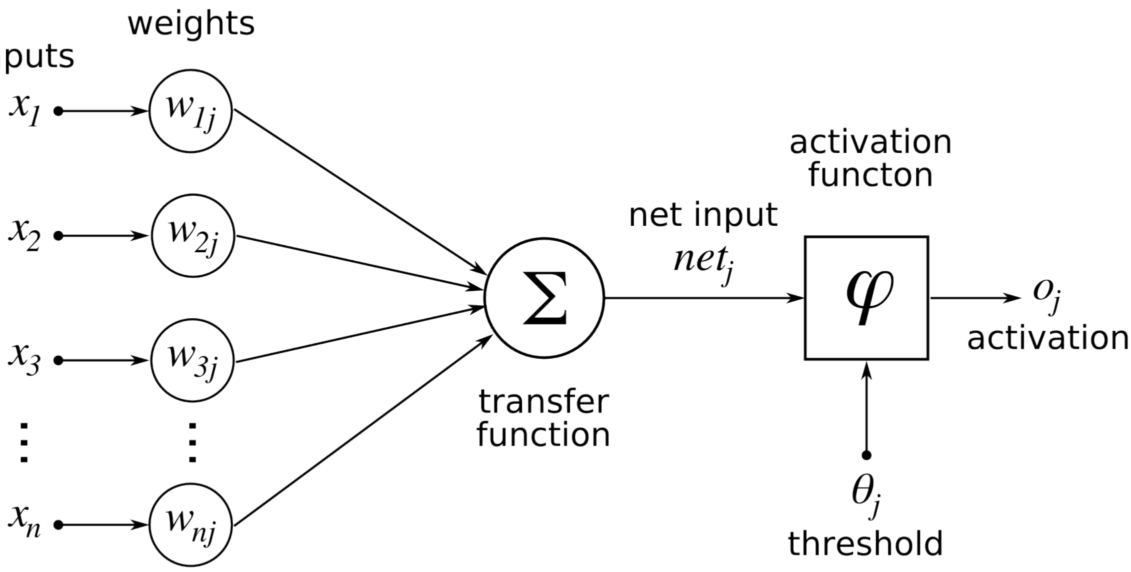

Illustration 4: An artificial neuron by the user Chrislb through Wikimedia Commons {CC BY-SA (http://creativecommons.org/licenses/by-sa/3.0/)}

The result of the activation function is the neuron’s output, which can then be multiplied by a corresponding weight and propagated as input into one or several other neurons. By stacking several layers of such artificial neurons into a network and manipulating the weight of each input through a training process we can create autonomous systems which can be used in a large range of problems.

2.1.4. Deep Neural Networks

In recent years, the increase in computing power has allowed for the development of large, complex and multilayered neural networks offering a qualitative shift in their potential and range of applications. These have come to be known as Deep Neural Networks (DNNs). In addition, the vast volume of data collected and disseminated through the internet which is now available has been a factor complementing the effectiveness of these networks, and has further contributed to the expansion of relevant areas of research. The two most commonplace types of deep neural networks are Convolutional Neural Networks (CNNs), which are mainly used in image processing, and Recursive Neural Networks (RNNs), which are particularly suited for processing discrete time signals such as natural language texts.

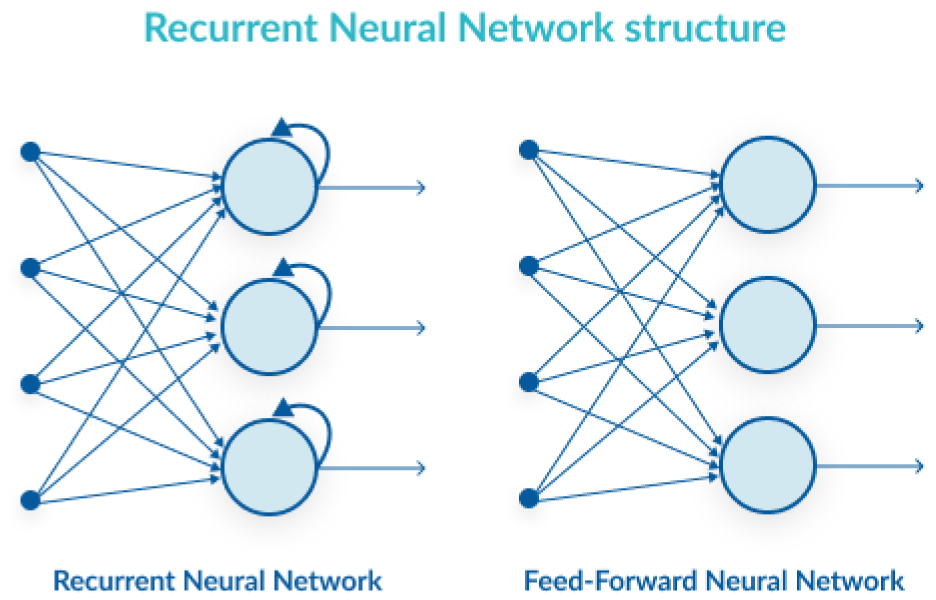

Illustration 5: A schematic of a recurrent neural network compared to a feed-forward neural network

2.2. Neural Network Concepts

2.2.1. LSTM

Long Short-Term Memory [5] is an RNN architecture that elegantly addresses the vanishing gradients problem, in which low gradient values prevent the information from flowing backwards and thus one or more neurons form training properly, by using “memory units”. These linear units have a self-connection of strength 1 and a pair of auxiliary “gating units” that control the flow of information to and from the unit. When the gating units are shut, the gradients can flow through the memory unit without alteration for an indefinite amount of time, thus overcoming the vanishing gradients problem. While the gates never isolate the memory unit in practice, this reasoning shows that the LSTM addresses the vanishing gradients problem in at least some situations, and indeed, the LSTM easily solves a number of synthetic problems with pathological long-range temporal dependencies that were previously believed to be unsolvable by standard RNNs [6]. LSTMs were also successfully applied to speech and handwritten text recognition [7].

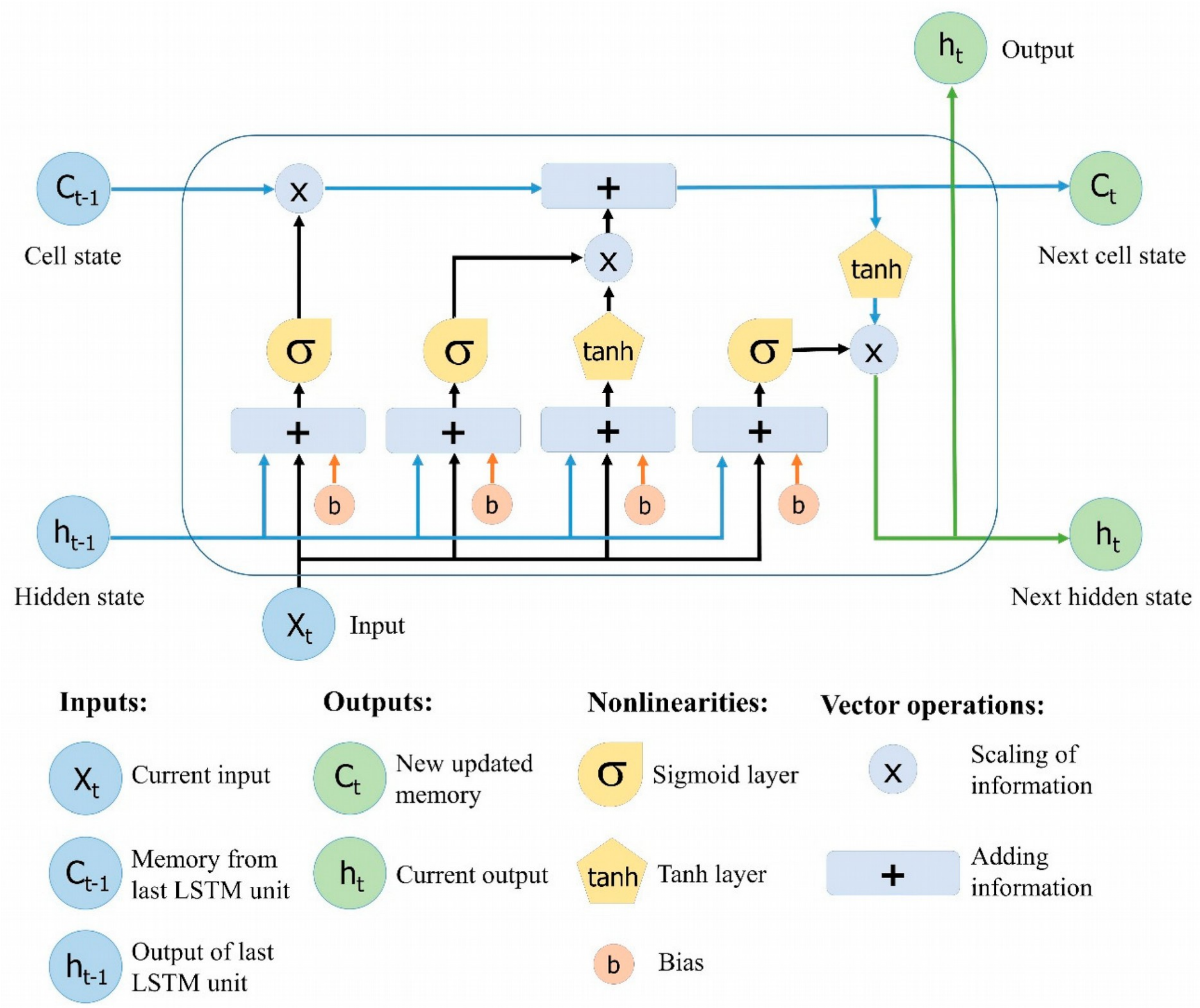

Illustration 6: A more detailed look into the structure of an LSTM-cell

2.2.2. BiLSTM

A bidirectional LSTM (BiLSTM) network is in practice composed of two different layers of LSTM units, each processing the input signal in a direction opposite to the other.

The BiLSTM architecture has had demonstrated success in tasks related to labelling sequences such as natural language text and are currently widely considered to be state of the art in the field [8] [9].

In this thesis, we will construct a BiLSTM-based deep neural network to fulfill our task, the annotation of the structural elements of Greek law modifications which we will specify in the following chapter.

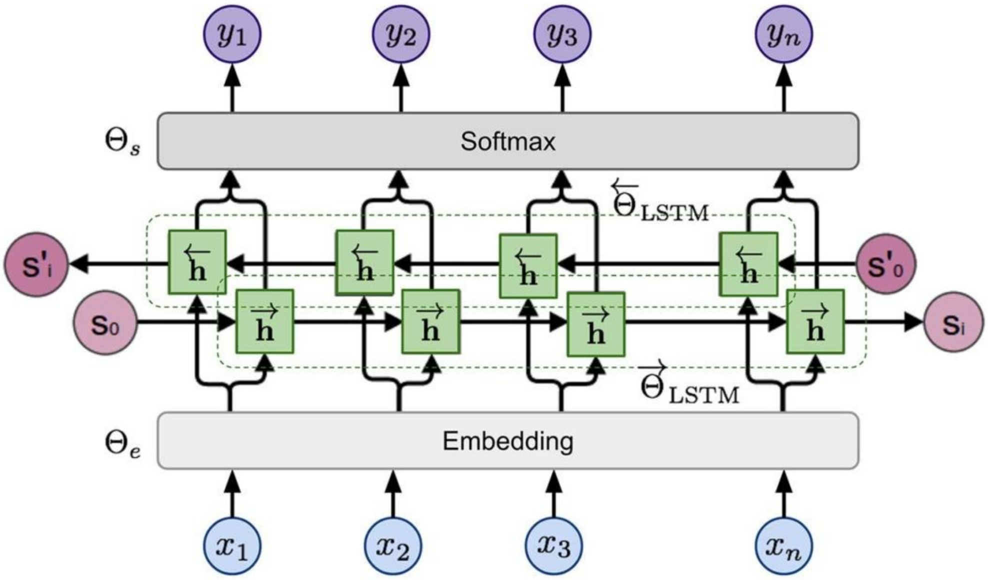

Illustration 7: An artificial neural network utilizing a BiLSTM layer

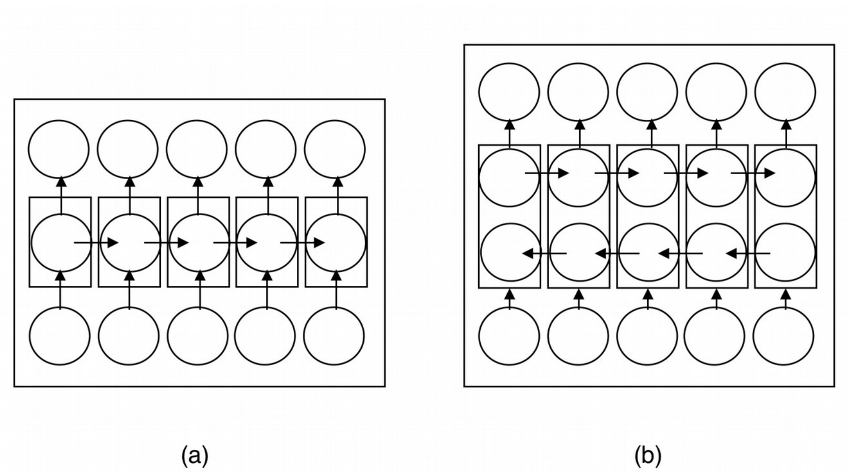

Illustration 8: (a) A unidirectional RNN (b) A bidirectional RNN

2.2.3. Loss Functions

Loss function is a type of objective function that is used during the training process to show the difference between the network's output and the desired one's. Typically a loss function acts as a distance function that compares the output of the network to the correct output which has been annotated earlier during the creation of the training set. Through the optimization algorithm the weights of the neural network are properly adjusted so that this distance is minimized.

2.2.4. Optimizers

Optimization algorithms (optimizers) help us minimize (or maximize) an objective function which is a function dependent on the various trainable parameters of the model. Such parameters are the weights associated with the neurons’ connections, the biases, the learning rate and the batch size. The way these parameters are changing differs for each optimization method. Gradient descent is a key element of this process. Three very important optimization algorithms are stochastic gradient descent [10], which is the one most commonly used, adam [11], which is an adaptive learning rate optimization algorithm that’s been designed specifically for training deep neural networks, and RMSprop, which is an unpublished optimization algorithm designed for neural networks, first proposed by Geoff Hinton [12].

2.2.5. Gradient descent

Gradient descent is one of the most popular algorithms to perform optimization and by far the most common way to optimize neural networks. At the same time, every state of the art Deep Learning library contains implementations of various algorithms to optimize gradient descent (e.g. lasagne’s [6] , caffe’s [7] , and keras’ [8] documentation). Gradient descent is a way to minimize an objective function parameterized by a model’s variables by updating those variables in the opposite direction of the gradient of the objective function. The learning rate determines the size of the steps we take in order to reach a (local) minimum.

There are three variants of gradient descent, which differ in how much data we use to compute the gradient of the objective function. Depending on the amount of data, we make a trade-off between the accuracy of the parameter update and the time it takes to perform an update [13] [14].

2.2.6. Batch gradient descent

The first variant, batch gradient descent (BGD), also known as vanilla gradient descent, computes the gradient of the cost function with respect to the parameters for the entire dataset. As we need to calculate the gradients for the whole dataset to perform just one

6 https://lasagne.readthedocs.io/en/latest/modules/updates.html
7 http://caffe.berkeleyvision.org/tutorial/solver.html
8 https://keras.io/api/optimizers/

update, batch gradient descent can be very slow and is intractable for datasets that do not fit in memory.

2.2.7. Stochastic gradient descent

Next is stochastic gradient descent (SGD), which in contrast to BGD, performs a parameter update for each training example. Batch gradient descent performs redundant computations for large datasets, as it recomputes gradients for similar examples before each parameter update. SGD does away with this redundancy by performing one update at a time. It is therefore usually much faster but because it performs frequent updates with a high variance the objective function fluctuates heavily. While batch gradient descent converges to the minimum of the basin the parameters are placed in, SGD's fluctuation, on the one hand, enables it to jump to new and potentially better local minima. On the other hand, this ultimately complicates convergence to the exact minimum, as SGD will keep overshooting. Still, it has been shown that when we slowly decrease the learning rate, SGD shows the same convergence behaviour as batch gradient descent, almost certainly converging to a local or the global minimum for non-convex and convex optimization respectively.

2.2.8. Mini-batch gradient descent

Last is mini-batch gradient descent (MGD) that takes the best of both worlds and performs an update for every mini-batch of a number of training examples. This way, it reduces the variance of the parameter updates, which can lead to more stable convergence; and can make use of highly optimized matrix optimizations common to state-of-the-art deep learning libraries that make computing the gradient with respect to a mini-batch very efficient. Common mini-batch sizes range between 32 and 256, but can vary for different applications. We used mini-batch gradient descent in training our model.

2.2.9. Backpropagation

Backpropagation is a very popular neural network learning algorithm because it is conceptually simple and computationally efficient. The simplest form of multilayer learning machine trained with gradient-based learning is simply a stack of modules, each of which implements a function $X_n = F_n(W_n, X_{n-1})$, where $X_n$is a vector representing the output of the module,W_n is the vector of tunable parameters in the module (a subset of W), and $X_{n-1}$is the module's input vector (as well as the previous module's output vector). The input $X_0$to the first module is the input pattern Z^p. If the partial derivative of $E^p$with respect to $X_n$is known, then the partial derivatives of $E^p$with respect to $W_n$and X_{n-1}can be computed using the backward recurrence

$$\frac{\partial E^p}{\partial W_n} = \frac{\partial F}{\partial W}(W_n, X_{n-1}) \frac{\partial E^p}{\partial X_n}$$

$$\frac{\partial E^p}{\partial X_{n-1}} = \frac{\partial F}{\partial X}(W_n, X_{n-1}) \frac{\partial E^p}{\partial X_n}$$

where $\frac{\partial F}{\partial W}(W_n, X_{n-1})$ is the Jacobian of F with respect to W evaluated at the point (W_n, X_{n-1}), and $\frac{\partial F}{\partial X}(W_n, X_{n-1})$ is the Jacobian of $F$with respect to X.

The Jacobian of a vector function is a matrix containing the partial derivatives of all the outputs with respect to all the inputs. When the above equations are applied to the modules in reverse order, from layer N to layer 1, all the partial derivatives of the cost function with respect to all the parameters can be computed. The way of computing gradients is known as back-propagation [15].

2.2.10. Evaluation metrics

When developing a neural network model one very important task is the process of evaluation. There are different kinds of metrics that can be utilized to evaluate a model. In order for this evaluation to be proper and unbiased the correct metrics must be chosen.

One such metric is called *Precision*, also known as *PPV (Positive Predicted Value)* and is defined as the fraction of relevant instances among the retrieved instances.

$$PPV=\frac{TP}{P}=\frac{TP}{TP+FP}$$

Intuitively it shows the ratio of correctly predicted positive observations (True Positives) to the total predicted positive observations. High precision value relates to the low false positive rate.

The next metric that we will utilize is called *Recall* also known as *Sensitivity* or *TPR (True Positive Rate)* and is defined as the fraction of relevant elements retrieved over the total amount of relevant elements.

$$TPR=\frac{TP}{TP+FN}$$

Intuitively it shows the ratio of the correctly predicted positive observations (True Positives) to the total relevant observations.

Another metric that combines the aforementioned is called *F-measure* or *F1-score* and is defined as a harmonic mean of precision and recall.

$$F_1=2*\frac{precision*recall}{precision+recall}=\frac{2*TP}{2*TP+FN+FP}$$

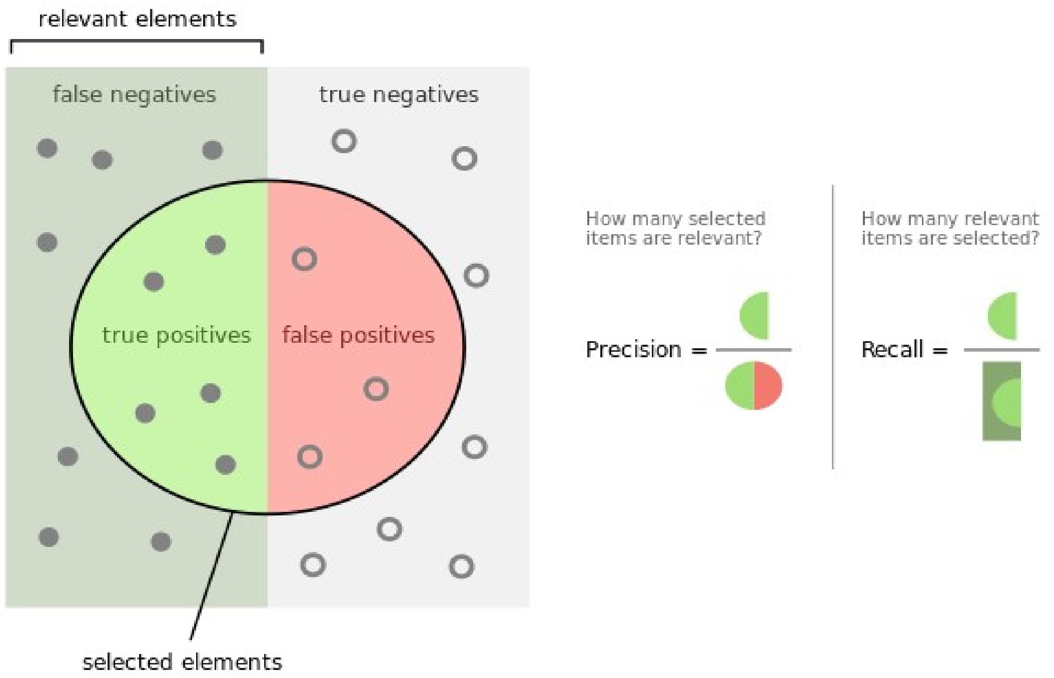

Illustration 9: A schematic depiction of precision and recall by the user Walber through Wikimedia Commons {CC BY-SA (https://creativecommons.org/licenses/by-sa/4.0)}

2.3. Pre-processing

2.3.1. Tokenization

Tokenization [16],in the context of NLP, is the process of separating a string of text into tokens in order to enable the processing of that text by other software, such as a neural network. The exact rules according to which the tokenization is done depends on the relevant task as well as the input data. They might include rules such as disregarding punctuation, normalising all letters to lowercase and substituting words which appear as different variants into one standard spelling.
An example of tokenization is the following:
Input: *"The urge to destroy is also a creative urge."*
Output: the urge to destroy is also a creative urge

2.3.2. Word embeddings

A word embedding can be any representation of words which is designed so that words with similar meaning should also have similar representations. An algorithm to create word embeddings aims to achieve as compact of a representation as possible while the maximum amount of semantic information is carried through. Words are typically converted into vectors within a multidimensional space to be fed as matrices of floating point numbers, also known as tensors, to a DNN.

2.3.3. Word2Vec

One such algorithm which is commonly used is Word2Vec, developed by researchers for Google in 2013 [17] [18].

It uses shallow neural networks, usually three layers deep, which are trained on a large volume of natural language texts. The first layer, the input, is fed with a simple representation of each word in the text. Desired network output is a neighboring keyword (or more). This is achieved by tweaking the weights of the hidden level during the training process. Finally, we keep the hidden level that represents a vector space within which words with greater conceptual affinity are closer.

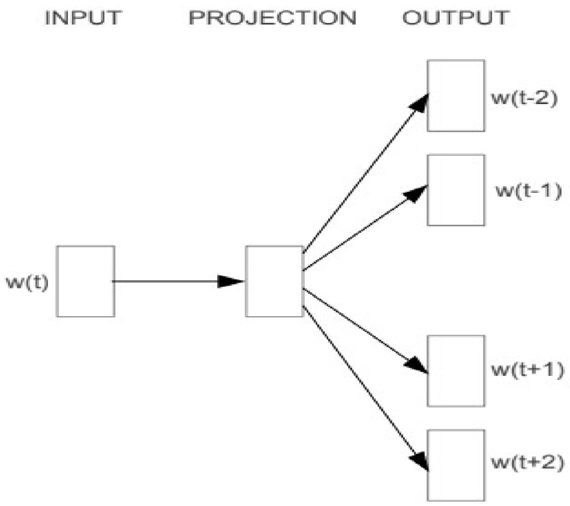

Illustration 10: The word2vec (skip-gram) architecture with window size n=2

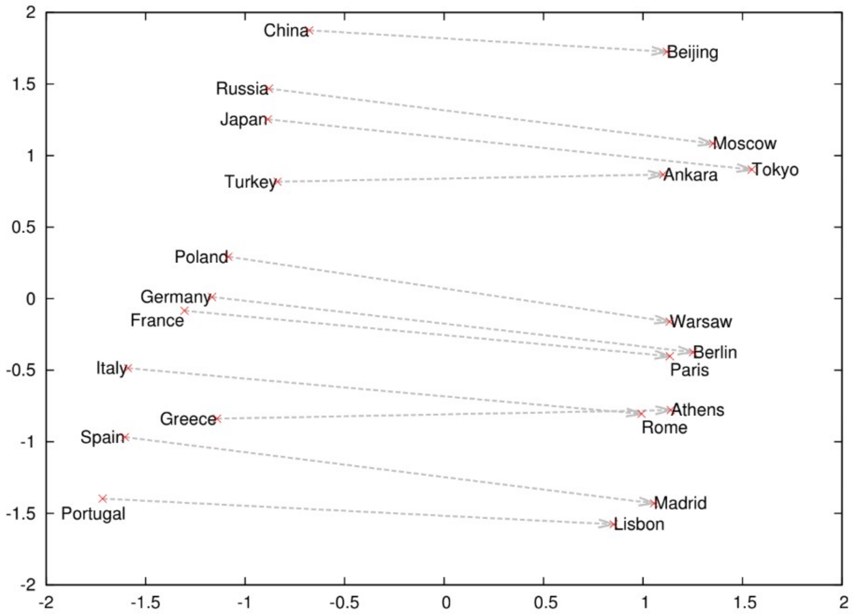

Illustration 11: Country and Capital Vectors projected by PCA

In this chapter we have introduced some concepts and used them to describe the prior work done in the area concerning our problem, on which we will be basing our own work. In the following chapter we begin to present our work, starting with the exact definition of our task.

3. TASK DEFINITION

The first step in solving the problem was to define a conceptual model for the modifications. We studied a large volume of Government Gazettes (issues A) in terms of their structure and content. The aim was to collect examples of legislative modifications and to define a categorization of them. We came up with the following categories:

Addition: a law is amended by introducing a new text, without otherwise altering the existing text.
Deletion: a law is amended by removing part of the existing text.
Replacement: a law is amended by removing part of the existing text and simultaneously inserting a new text.
Renumbering: a law is amended by changing the numbering of its parts.

In addition, we identified some structural elements that were common to many of the examples and contained information that we wanted to encode. In each one we will assign a label, and these are the ones that our neural network will be trained to recognize.

One of the observations we made while studying the examples was that any modification would necessarily be made on a single law, the Government Gazette of which may or may not be explicitly referred to the text of the modification. The division of a law is typically applied in articles at the first level, and then in paragraphs, but it shows great variability and does not seem to follow universal rules. For this reason, we came up with two specific labels that note the law and the Government Gazette number (**target law** and **target gazette**). Any other information on the part of the law to which the modification applies (e.g. article, paragraph, case, phrase) is included in a third label, **target fragment**.
In cases where the modification introduces a new part to the structure of the law (addition, renumbering) this is noted as **new fragment**. When a new text is added (addition, replacement) it is marked as **modification body**.
Finally, we defined the **modification type** label, which will mark the words in the text of the modification that indicate in which category it belongs to. Although it appears as one label in our conceptual model and at the annotation phase we used it as such in the next stage of processing our data we replaced it with five separate labels stating the five modifications categories (addition, deletion, replacement, renumbering, date change). We did this aiming at the additional useful value of automatic category recognition, and considering that this expansion would have little impact on our neural network.

To mark the individual details of the modifications that we want our neural network to recognize, we defined the following labels:

target gazette : labels the Government Gazette in which the law to be modified was published.
target law : labels the law to be modified
target fragment : labels the structural part of the law that is modified
new fragment : labels a new part of the law introduced by the modification
modification type : labels the category of the modification
modification body : labels the new text introduced by the modification

Moving on to the next phase of our work we set some rules for the annotation process.

- For the target gazette label, we do not annotate the parentheses that usually enclose it.
- We do not annotate grammatical articles unless they appear in the middle of a label and the words to the left and right of the article must be annotated (e.g. in the passage "Η περίπτωση 13 της παραγράφου 12 …", "της" is annotated but "Η" is not).
- We do not include punctuation marks when they appear at the end of a label, except in cases such as quotation marks in replacements, which are part of the structure of the modification.

3.1. Examples

To better demonstrate the labels and the way we applied them to the text we now cite some annotated examples of each modification category. Because the text we worked on is written in Greek language the examples are cited in Greek.

3.1.1. Addition

Paragraph addition
http://legislation.di.uoa.gr/eli/law/2011/3986/article/17/paragraph/2
Στο άρθρο 39 του ν. 3105/2003 (Α΄ 29) προστίθεται παράγραφος 18, ως εξής:
1. Τα δικαιώματα χρήσης ή διοίκησης, διαχείρισης και εκμετάλλευσης επί ακινήτων που έχουν περιέλθει με οποιονδήποτε τρόπο στην εταιρεία με την επωνυμία «Ελληνικά Τουριστικά Ακίνητα Α.Ε.», σημειώνονται στο περιθώριο των οικείων βιβλίων μεταγραφών των αρμόδιων υποθηκοφυλακείων ή κτηματολογικών γραφείων. Σημειωτέα πράξη αποτελούν τα σχετικά αποσπάσματα αποφάσεων έγκρισης της σημείωσης του Διοικητικού Συμβουλίου της εταιρείας.

Case addition

http://legislation.di.uoa.gr/eli/law/2012/4052/article/49/paragraph/1

Στο τέλος της παραγράφου 2 του άρθρου 2 του ν. 3996/2011 (Α΄ 170) προστίθεται περίπτωση κβ΄ ως εξής:

κβ. Ελέγχει την τήρηση των νόμιμων προϋποθέσεων για την άσκηση της δραστηριότητας των Επιχειρήσεων Προσωρινής Απασχόλησης (Ε.Π.Α.).

Passage addition

http://legislation.di.uoa.gr/eli/law/2011/3986/article/9/paragraph/3

Στο τέλος της παρ. 4 του άρθρου 5Α του ν. 3049/2002 προστίθεται από τότε που ίσχυσε ο ν. 3965/ 2011 (Α΄ 113) εδάφιο ως εξής:

Μέχρι την πρώτη συγκρότηση της Επιτροπής της παρούσας παραγράφου, η ανάθεση των συμβάσεων του παρόντος άρθρου διενεργείται χωρίς γνώμη της Επιτροπής, με απόφαση της Δ.Ε.Α.Α..

3.1.2. Replacement

Title replacement

http://legislation.di.uoa.gr/eli/law/2015/4324/article/1/paragraph/1

1. *Ο τίτλος του ν. 4173/2013 (Α΄169) αντικαθίσταται ως εξής:*

   Ελληνική Ραδιοφωνία Τηλεόραση Ανώνυμη Εταιρεία (Ε.Ρ.Τ. Α.Ε.).

Article replacement

http://legislation.di.uoa.gr/eli/law/2015/4346/article/2/paragraph/1

Το *άρθρο 61* του *Ν. 4342/2015* (*Α΄ 143*) *αντικαθίσταται* ως εξής:

Άρθρο 61

1. Το άρθρο 26 του Ν. 4174/2013 (Α΄ 170) αντικαθίσταται ως εξής: «Άρθρο 26 ...»

Paragraph replacement

http://legislation.di.uoa.gr/eli/law/2010/3863/article/8/paragraph/1

Οριστικοποίηση σύνταξης αναπηρίας των ασφαλισμένων από 1.1.1993 Η παράγραφος 3 του άρθρου 25 παρ. 3 του ν. 2084/1992(ΦΕΚ 165 Α΄) τροποποιείται ως ακολούθως:

3. Το δικαίωμα συνταξιοδότησης λόγω αναπηρίας, υφίσταται για όσο χρόνο ορίζεται από τις αρμόδιες Υγειονομικές Επιτροπές, παρατείνεται δε με τις ίδιες προϋποθέσεις ενώ δύναται να ελέγχεται αυτεπαγγέλτως οποτεδήποτε, με την υποβολή του συνταξιούχου σε ιατρική εξέταση από τις ανωτέρω επιτροπές. Οι συντάξεις λόγω αναπηρίας είναι οριστικές για τις περιπτώσεις των ασθενειών που προβλέπονται από ρητή διάταξη, μπορεί δε να είναι οριστικές, εφόσον οι υγειονομικές επιτροπές γνωματεύουν ότι η ανικανότητα είναι μόνιμη. Οι προσωρινές συντάξεις λόγω αναπηρίας καθίστανται οριστικές, μετά και την τελική γνωμάτευση των αρμόδιων Υγειονομικών Επιτροπών εφόσον: α) Ο συνταξιούχος έχει συμπληρώσει το 55ο έτος της ηλικίας του και χρόνο συνταξιοδότησης επτά (7) ετών συνεχώς, κατά τη διάρκεια των οποίων υποβλήθηκε σε τρεις τουλάχιστον εξετάσεις από τις οικείες υγειονομικές επιτροπές. β) Ο συνταξιούχος έχει συμπληρώσει το 60ό έτος της ηλικίας του και χρόνο συνταξιοδότησης *πέντε (5) ετών συνεχώς, κατά τη διάρκεια των οποίων υποβλήθηκε σε δύο τουλάχιστον εξετάσεις από τις οικείες υγειονομικές επιτροπές.*

Case replacement

http://legislation.di.uoa.gr/eli/law/2011/3986/article/17/paragraph/3

Η περίπτωση α΄ της παρ. 1 του άρθρου 49 του ν. 3220/2004 (Α΄ 15) αντικαθίσταται ως εξής:

> *α. Η κυριότητα επί περιουσιακών στοιχείων του Ε.Ο.Τ. δύναται να μεταβιβάζεται στην εταιρεία με την επωνυμία «Ελληνικά Τουριστικά Ακίνητα Α.Ε.» ή σε θυγατρικές της εταιρείες, με Πράξη του Υπουργικού Συμβουλίου, ύστερα από αιτιολογημένη εισήγηση του Υπουργού Πολιτισμού και Τουρισμού, η οποία δημοσιεύεται στην Εφημερίδα της Κυβερνήσεως. Η απόφαση αυτή αποτελεί το μεταγραπτέο τίτλο, όταν απαιτείται μεταγραφή για τη μεταβίβαση, η δε μεταγραφή της στα οικεία υποθηκοφυλακεία ή κτηματολογικά γραφεία απαλλάσσεται από κάθε τέλος ή δικαίωμα υπέρ του Δημοσίου, Ο.Τ.Α., Ν.Π.Δ.Δ. και γενικά υπέρ οποιουδήποτε τρίτου φυσικού ή νομικού προσώπου.*

Subcase replacement

http://legislation.di.uoa.gr/eli/law/2012/4052/article/93/paragraph/7

Η υποπερίπτωση εε΄ της περίπτωσης α΄ της παραγράφου 7 του άρθρου 77 του ν.3996/2011 (Α΄ 170), αντικαθίσταται ως εξής:

Για τον προσδιορισμό των ανωτέρω ασφαλιστικών εισφορών λαμβάνεται ως βάση υπολογισμού για τον κάθε ασφαλιστικό φορέα - τομέα κύριας και επικουρικής ασφάλισης, καθώς και πρόνοιας, ο ασφαλιστέος μισθός

Passage replacement

http://legislation.di.uoa.gr/eli/law/2010/3863/article/5/paragraph/2

Το εδάφιο β΄ της παραγράφου 4 του άρθρου 1 του ν. 3232/2004 (ΦΕΚ 48 Α΄) αντικαθίσταται ως ακολούθως:

4. Το ανωτέρω τμηματικό ποσό δύναται κατ' επιλογή του ασφαλισμένου να καταβληθεί ταυτόχρονα με αυτό του απονέμοντα, μειωμένο κατά 1/200 για κάθε μήνα που υπολείπεται έως τη συμπλήρωση των προβλεπόμενων από τις διατάξεις του άρθρου 69 του ν. 2084/1992 (ΦΕΚ 165 Α΄) ορίων ηλικίας.

http://legislation.di.uoa.gr/eli/law/2010/3863/article/10/paragraph/1

1. Τα δύο πρώτα εδάφια της παραγράφου 1 του άρθρου 10 του ν. 825/1978 (ΦΕΚ 189 Α), όπως ισχύουν, αντικαθίστανται ως εξής:

«1. Ο ασφαλισμένος του ΙΚΑ−ΕΤΑΜ δικαιούται σύνταξης, αν κατά την υποβολή της αίτησης έχει πραγματοποιήσει 10.500 τουλάχιστον ημέρες εργασίας στην ασφάλιση του ΙΚΑ−ΕΤΑΜ και έχει συμπληρώσει το 58ο έτος της ηλικίας του.

Ο ανωτέρω χρόνος ασφάλισης για τους ασφαλισμένους που συμπληρώνουν αυτόν από 1.1.2011 αυξάνεται κατά 300 ημέρες κάθε χρόνο και μέχρι τη συμπλήρωση 12.000 ημερών ασφάλισης.Το όριο ηλικίας που προβλέπεται από το πρώτο εδάφιο, αυξάνεται σταδιακά από 1.1.2012 κατά ένα (1) έτος κάθε χρόνο και μέχρι τη συμπλήρωση του 60ού έτους της ηλικίας.»

Phrase replacement

http://legislation.di.uoa.gr/eli/law/2015/4346/article/5/paragraph/15

Στην παράγραφο 6 του άρθρου 6 η φράση «το Ταμείο παράσχει» αντικαθίσταται με τη φράση «το Ταμείο παρέχει»

3.1.3. Deletion

Case deletion

http://legislation.di.uoa.gr/eli/law/2014/4305/article/33/paragraph/2

Στο π.δ. 106/2014 (Α΄ 173) «Οργανισμός του Υπουργείου Υγείας» επέρχονται οι ακόλουθες τροποποιήσεις:

Η περίπτωση γ΄ της παραγράφου 3 του άρθρου 2 διαγράφεται.

Subcase deletion

http://legislation.di.uoa.gr/eli/law/2014/4305/article/8/paragraph/32

Η υποπερίπτωση ζζ΄ της περίπτωσης γ΄ της παραγράφου 3 του άρθρου 37 διαγράφεται.

Case and articles deletion

http://legislation.di.uoa.gr/eli/law/2014/4305/article/33/paragraph/3

Στο π.δ. 107/2014 (Α΄ 174) «Οργανισμός του Υπουργείου Αγροτικής Ανάπτυξης και Τροφίμων» επέρχονται οι ακόλουθες τροποποιήσεις:

α) η περίπτωση ε΄ της παραγράφου 3 του άρθρου 2 διαγράφεται.

β) Τα άρθρα 11, 12, 13 και 14 διαγράφονται.

Passage deletion

http://legislation.di.uoa.gr/eli/la/2015/2015-10-08/article/6/paragraph/5

Το πρώτο εδάφιο της περίπτ. β΄ της παρ. 2 του άρθρου 9 του Ν. 4009/2011, όπως ισχύει, καταργείται.

Phrase deletion

http://legislation.di.uoa.gr/eli/law/2015/4346/article/5/paragraph/14

Στην παράγραφο 4 του άρθρου 6 διαγράφονται τα εισαγωγικά « » από τις λέξεις συμφωνίας-πλαίσιο

3.1.4. Renumbering

http://legislation.di.uoa.gr/eli/law/2014/4281/article/194/paragraph/2

Οι παράγραφοι 7, 8, 9, 10, 11 και 12 του άρθρου 28 του ν. 4070/2012 αναριθμούνται σε παραγράφους 8, 9, 10, 11, 12 και 13, αντίστοιχα.

3.1.5. Combinations

Finally, we cite some examples containing several modification categories:

http://legislation.di.uoa.gr/eli/pd/2014/166/article/2/paragraph/1
η περίπτωση α της παραγράφου 1 τροποποιείται ως εξής:αα) η υποπερίπτωση ζζ΄) διαγράφεται και η υποπερίπτωση ηη΄ αναριθμείται ως ζζ΄.

http://legislation.di.uoa.gr/eli/law/2014/4305/article/9/paragraph/15
Η υποπερίπτωση γγ΄ της περίπτωσης γ΄ της παραγράφου 3 του άρθρου 34 διαγράφεται. Η υποπερίπτωση δδ΄ της ίδιας περίπτωσης αναριθμείται ως γγ΄.

http://legislation.di.uoa.gr/eli/law/2015/4346/article/1/paragraph/2
Μετά την παρ. 1 του άρθρου 17 του Ν. 4174/2013 προστίθεται παράγραφος 2, αναριθμουμένων των επόμενων παραγράφων, ως εξής:
2. Τα πρόσωπα της προηγούμενης παραγράφου χορηγούν σε εξουσιοδοτημένο προσωπικό της Ελληνικής Στατιστικής Αρχής (ΕΛ.ΣΤΑΤ.) ή σε εξουσιοδοτημένα πρόσωπα από την Ελληνική Στατιστική Αρχή (ΕΛ.ΣΤΑΤ.), προσωποιημένα στοιχεία ανά ΑΦΜ, καθώς και συγκεντρωτικά στοιχεία, τα οποία τηρούνται στη Φορολογική Διοίκηση, με την υποχρέωση χρήσης αυτών, αποκλειστικά για το σκοπό για τον οποίο ζητούνται στο πλαίσιο των αρμοδιοτήτων της ΕΛ.ΣΤΑΤ. και σύμφωνα με τις διατάξεις της παρ. 3 του άρθρου 8 του Ν. 3832/2010, όπως ισχύει.

In this chapter we have defined our conceptual model and the entities we want our neural network to recognize. In the next chapter we will describe the methodology we followed in its implementation.

4. METHODOLOGY

The process of preparing our neural network comes in two parts, the first being the preprocessing of the data in order for them to be in the appropriate format and the second the development of the neural network model.

4.1. Pre-processing

The dataset we began with was a list of documents, each one consisting of a passage from a law containing one or more modifications.

4.1.1. Labeling

The first step was to collect examples of modifications from the corpus that would be necessary in the next phase of the training. As a background for our work we used the Government Gazettes, available as pdf files in the National Printing House's website [9], which have been converted for the Νομοθεσι@ platform through optical character recognition (OCR) into text file format. In an effort to filter this huge amount of data, we also used a script made by the ΝΟΜΟΘΕΣΙ@ team to recognize modifications by keywords found through regular expressions. Thus we singled out a set of laws that were quite likely to contain at least one modification.

To mark the annotations we used doccano, an open source tool [10].

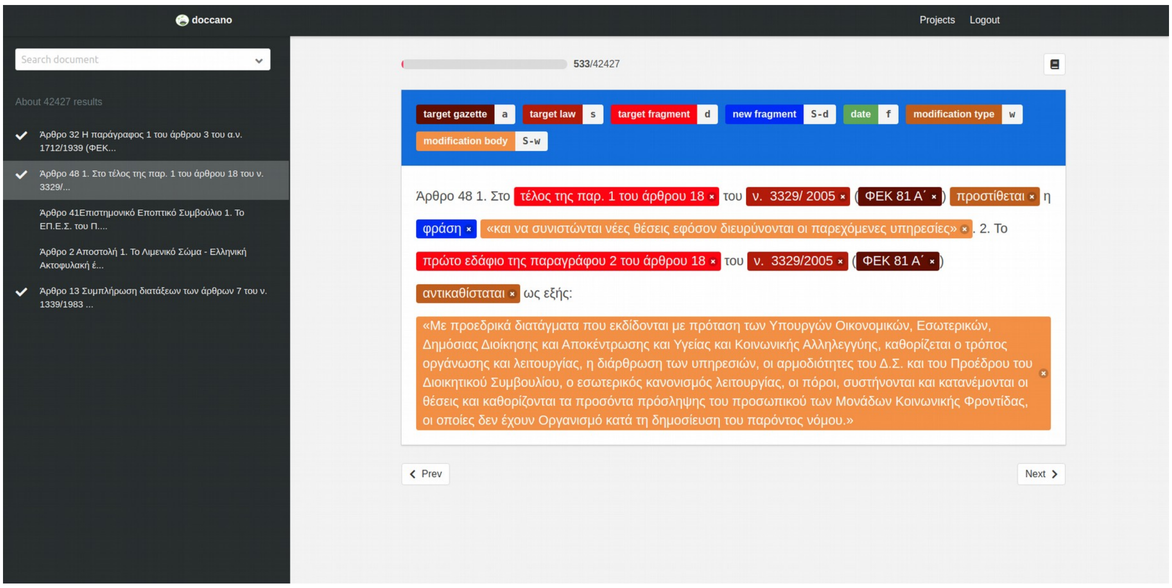

Illustration 12: Screenshot of the Doccano workspace

9 http://www.et.gr/index.php/f-e-k/diathesi-fek

10 https://doccano.herokuapp.com/

As our goal was to annotate a large number of modification examples while working independently of each other, and these examples had to be consistent in their method of labeling for a smooth training of the neural network, we first set some labeling rules and annotated a few example documents. We repeated this process, adjusting the rules, until the identification of the labels we noted reached a satisfactory percentage (less than one discrepancy in twenty documents). Some of these rules are listed in Chapter 2.

When we had collected a sufficient volume of recorded data (533 laws, containing 1548 modifications and 6929 labels in total) we proceeded to the next stage.

4.1.2. Tokenization

The next step was to convert the dataset from string sequences to word sequences, translating accordingly all the label information we added during the annotation. This process is called tokenization. We used the tokenizer which was developed for the ΝΟΜΟΘΕΣΙ@ project [19], including lists of word abbreviations corresponding to the same token. In addition, we used the open source library spacy[11]. For better homogenization, all letters were converted to uppercase without intonation and all digits were replaced with a special placeholder symbol. Words with non-Greek characters are also represented by a common token. Each punctuation mark corresponds to a different token, unless they are part of an abbreviation.

4.1.3. Vectorization

As the last step of the pre-processing stage we converted the documents into sequences of vectors, using the word2vec algorithm. Here again we relied on the valuable work of ΝΟΜΟΘΕΣΙ@ team, using a pre-trained model in the same corpus, that is Greek legislative text from Government Gazettes.

11 https://spacy.io/

4.2. Model Development

We decided to develop our model through the popular Keras API, using the Tensorflow framework [20] as a backend. The development environment was Ubuntu 18.04, with all code written and run on a Jupyter notebook running on a Docker container. The version of Tensorflow used throughout the project was 2.1.0.

As explained in previous chapters, the main architecture we wanted to use was BiLSTM, a kind of RNN which is considered state of the art for sequence labeling problems. The exact structure we eventually used is as follows:

- First, an **Input layer**,
- a **Masking layer** which is used to mark the padding put at the end of shorter documents so that it can be ignored by following layers,
- two **BiLSTM layers**,
- a **Dropout layer** and, finally,
- a **Dense layer** with Softmax as its activation function, meant to format the output into probability values for each of our ten classes.

The loss function the network would be trained with is categorical cross-entropy. The hyperparameters we would keep as variables to fine-tune in order to create the most suitable model are:

- The **number of epochs** the model will train for,
- the **batch size**,
- the **initial weight values**,
- the **weight constraint** function,
- the **number of recurrent neurons** per layer,
- the **dropout rate** value, and
- the **optimization algorithm**.

In this chapter, we have explained the development of our neural network model. Next we will present the results of our experiments using different hyperparameter values.

5. EXPERIMENTS

In order to reduce the dimensionality of the combination space we split the hyperparameters in smaller groups [21]. We ran a k-fold cross validation with a fold parameter of five for each group. Following are the tables with the metrics for each experiment, with the values having been the averaged values over the five runs of the cross validation. Highlighted are the parameter value combinations we chose to use in the final model.

5.1. Batch size and number of epochs.

Table 1: Results for different number of epochs and batch sizes

epochs	batch_size	loss	accuracy	precision
10		0.1987394532	0.943038824	0.9479698
10	16	0.1724772134	0.957107678	0.9588619
10	32	0.1749375622	0.950657132	0.95379968
10	64	0.1720855257	0.948168126	0.952315588
10	128	0.251519454	0.914983488	0.923575112
50		0.243524125	0.953504466	0.954498992
50	16	0.2778220906	0.9517598	0.952588928
50	32	0.1990034085	0.961398032	0.962673718
50	64	0.2148139398	0.957540398	0.958530606
50	128	0.1870346848	0.955092786	0.95709626
100		0.2475726526	0.958659274	0.959218004
100	16	0.2775270908	0.959608218	0.96000182
100	32	0.2197551827	0.964737088	0.965227258
100	64	0.218872849	0.95967064	0.960209684
100	128	0.2305038312	0.952585268	0.953699648

We chose a batch size of 32 as it has the better metrics. We elected to set the number of epochs equal to 50, as it gave only marginally worse results than a value of 100 while running in half the time.

5.2. Optimizer

Table 2: Results for different optimization algorithms

optimizer	accuracy	precision
adadelta	0.494156864	0
sgd	0.742685762	0.802297688
adam	0.95656944	0.958524184
nadam	0.959250688	0.9607998
adagrad	0.674079952	0.76296502
adamax	0.94622862	0.950959378
rmsprop	0.961767394	0.96244512

While the Adam optimizer and its derivatives are largely considered to be the best in related tasks, the RMSProp algorithm proved to deliver the best results with Nadam coming in as a close second.

5.3. Initial weight values, weight constraints

Table 3: Results for different neuron initialization values and weight constraint functions

init mode	weight constraint	accuracy	precision	recall
glorot normal		0.95880122	0.959382032	0.958348536
glorot normal	min max norm	0.961783186	0.96275384	0.961153058
glorot normal	non neg	0.9537656	0.955271936	0.95268564
glorot normal	unit norm	0.955074632	0.956146308	0.954244148
glorot uniform		0.95691177	0.9583567	0.95653057
glorot uniform	min max norm	0.96129592	0.96236995	0.96053341
glorot uniform	non neg	0.9483329	0.95076452	0.947506875
glorot uniform	unit norm	0.956402185	0.957153675	0.955449025
ones		0.78200577	0.81429215	0.738280875
ones	min max norm	0.949856	0.952658815	0.948775735
ones	non neg	0.784618775	0.81714163	0.73848058
ones	unit norm	0.93842015	0.94201845	0.935814915
zeros		0.6395238	0.70121545	0.55118862
zeros	min max norm	0.644610945	0.71175505	0.5328278
zeros	non neg	0.60680965	0.64215928	0.536129
zeros	unit norm	0.64175068	0.712149075	0.5420998

A min-max normalising weight constraint function seems to have the best results, with glorot normal and glorot uniform being equivalent.

5.4. Number of neurons and dropout rate

Table 4: Results for different number of input neurons and dropout rate

neurons	dropout	accuracy	precision	recall
50	0.001	0.95212414	0.953508866	0.95135214
50	0.01	0.94949617	0.951421902	0.948164888
50	0.2	0.959749912	0.961502078	0.958120596
50	0.5	0.948093366	0.953285278	0.94390384
100	0.001	0.944646852	0.94559018	0.944218086
100	0.01	0.962315488	0.963295404	0.961684152
100	0.2	0.961676144	0.962650604	0.961223864
100	0.5	0.960138338	0.962061272	0.958663954
150	0.001	0.960264442	0.960730022	0.959887386
150	0.01	0.960632272	0.961246994	0.960360164
150	0.2	0.960756108	0.96145671	0.960156526
150	0.5	0.961348662	0.962831152	0.960492698

We see that 100 neurons in each layer are enough, with the performance even falling when the neurons reach 150. A dropout rate of 0.01 also gives the best results.

5.5. Final results

In the end, we used the hyperparameters we arrived at through the tuning process to create one final model, trained on the full training and validation dataset, and used it to predict the labels of the documents in the test set. The following table is a classification report produced by the scikit-learn module, giving precision, recall and f1-score per label, as well as overall averages.

Table 5: Test results with evaluation metrics per class

	precision	recall	f1-score	support
none	0.98	0.99	0.98	12731
gazette	0.99	0.83	0.9	242
law	0.98	0.9	0.94	461
target fragment	0.95	0.92	0.94	1430
new fragment	0.96	0.6	0.74	149
modification body	1	1	1	14760
addition	0.94	0.85	0.89	54
deletion	1	0.9	0.95	39
replacement	0.97	0.85	0.91	121
renumbering	1	0.38	0.56	13
accuracy			0.99	30000
macro avg	0.98	0.82	0.88	30000
weighted avg	0.99	0.99	0.99	30000

5.6. Examples

And here are a few examples of the output the neural network produced when tasked with identifying that same test dataset:

Example 1:

ΑΡΘΡΟ d d. ΣΤΗΝ ΠΑΡΑΓΡΑΦΟ d ΤΟΥ ΑΡΘΡΟΥ d ΤΟΥ Ν . dddd/dddd , ΤΟ ΤΕΛΕΥΤΑΙΟ ΕΔΑΦΙΟ ΤΗΣ ΠΕΡΙΠΤΩΣΗΣ Γ΄ ΥΠΟΠΕΡΙΠΤΩΣΗ ΙΙ ΔΙΑΓΡΑΦΕΤΑΙ . d. Η ΠΑΡ . d ΤΟΥ ΑΡΘΡΟΥ d ΤΟΥ Ν . dddd/dddd ΑΝΤΙΚΑΘΙΣΤΑΤΑΙ ΩΣ ΕΞΗΣ : « d . ΟΙ ΔΗΛΩΣΕΙΣ ΠΕΡΙΟΥΣΙΑΚΗΣ ΚΑΤΑΣΤΑΣΗΣ ΤΩΝ ΠΡΟΣΩΠΩΝ ΠΟΥ ΑΝΑΦΕΡΟΝΤΑΙ ΣΤΙΣ ΠΕΡΙΠΤΩΣΕΙΣ ΣΤ΄ ΕΩΣ ΚΑΙ ΙΕ΄ ΤΗΣ ΠΑΡ . d ΤΟΥ ΑΡΘΡΟΥ d ΤΟΥ ΠΑΡΟΝΤΟΣ ΝΟΜΟΥ ΥΠΟΒΑΛΛΟΝΤΑΙ ΣΤΗ Γ΄ ΜΟΝΑΔΑ ΤΗΣ ΑΡΧΗΣ ΤΟΥ ΑΡΘΡΟΥ d ΤΟΥ Ν . dddd/dddd . » d. ΣΤΗΝ ΠΑΡ . d ΤΟΥ ΑΡΘΡΟΥ d ΤΟΥ Ν . dddd/dddd , Η ΦΡΑΣΗ « ΑΠΟ ΤΙΣ ΕΠΙΤΡΟΠΕΣ ΤΩΝ ΠΑΡΑΓΡΑΦΩΝ d ΚΑΙ d ΤΟΥ ΑΡΘΡΟΥ ΑΥΤΟΥ , ΤΟΣΟ ΟΙ ΙΔΙΕΣ » ΑΝΤΙΚΑΘΙΣΤΑΤΑΙ ΑΠΟ ΤΗ ΦΡΑΣΗ « ΑΠΟ ΤΗΝ ΕΠΙΤΡΟΠΗ ΤΗΣ ΠΑΡΑΓΡΑΦΟΥ d , ΤΟΣΟ Η ΙΔΙΑ » . d. Η ΠΑΡ . d ΤΟΥ ΑΡΘΡΟΥ d ΤΟΥ Ν . dddd/dddd ΑΝΤΙΚΑΘΙΣΤΑΤΑΙ ΩΣ ΕΞΗΣ : « d . ΜΕΤΑ ΤΟ ΠΕΡΑΣ ΤΟΥ ΕΛΕΓΧΟΥ ΑΠΟ ΤΗΝ ΕΠΙΤΡΟΠΗ ΤΗΣ ΠΑΡΑΓΡΑΦΟΥ d , ΑΝ ΔΕΝ ΔΙΑΠΙΣΤΩΘΕΙ ΠΑΡΑΒΑΣΗ ΚΑΙ Η ΔΗΛΩΣΗ ΚΡΙΘΕΙ ΕΙΛΙΚΡΙΝΗΣ , ΣΥΝΤΑΣΣΕΤΑΙ ΣΤΟ ΣΩΜΑ ΤΗΣ ΠΡΑΞΗ ΤΟΥ ΔΙΕΝΕΡΓΗΣΑΝΤΟΣ ΤΟΝ ΕΛΕΓΧΟ ΚΑΙ ΤΙΘΕΤΑΙ ΣΤΟ ΑΡΧΕΙΟ . ΕΦΟΣΟΝ ΔΙΑΠΙΣΤΩΝΟΝΤΑΙ ΠΑΡΑΒΑΣΕΙΣ ΤΟΥ ΝΟΜΟΥ ΚΑΙ ΣΥΝΤΡΕΧΕΙ ΠΕΡΙΠΤΩΣΗ ΚΑΤΑΛΟΓΙΣΜΟΥ ΚΑΤΑ ΤΟ ΑΡΘΡΟ dd ΤΟΥ ΠΑΡΟΝΤΟΣ ΝΟΜΟΥ , ΣΥΝΤΑΣΣΕΤΑΙ ΣΧΕΤΙΚΗ ΕΚΘΕΣΗ , Η ΟΠΟΙΑ ΑΠΟΣΤΕΛΛΕΤΑΙ ΣΤΟΝ ΓΕΝΙΚΟ ΕΠΙΤΡΟΠΟ ΤΗΣ ΕΠΙΚΡΑΤΕΙΑΣ ΣΤΟ ΕΛΕΓΚΤΙΚΟ ΣΥΝΕΔΡΙΟ . ΑΝ ΑΝΑΚΥΠΤΕΙ ΠΕΡΙΠΤΩΣΗ ΠΟΙΝΙΚΗΣ ΕΥΘΥΝΗΣ , Η ΕΚΘΕΣΗ ΑΠΟΣΤΕΛΛΕΤΑΙ ΣΤΟ ΑΡΜΟΔΙΟ ΓΙΑ ΤΗΝ ΑΣΚΗΣΗ ΠΟΙΝΙΚΗΣ ΔΙΩΞΗΣ ΟΡΓΑΝΟ . ΕΦΟΣΟΝ ΔΙΑΠΙΣΤΩΘΕΙ ΑΝΑΓΚΗ ΔΙΕΡΕΥΝΗΣΗΣ ΘΕΜΑΤΩΝ ΠΟΥ ΕΜΠΙΠΤΟΥΝ ΣΤΗΝ ΑΡΜΟΔΙΟΤΗΤΑ ΦΟΡΟΛΟΓΙΚΗΣ Η ΑΛΛΗΣ ΑΡΧΗΣ , Η ΕΚΘΕΣΗ ΑΠΟΣΤΕΛΛΕΤΑΙ ΣΤΗΝ ΑΡΧΗ ΑΥΤΗ . » d. Η ΠΑΡ . d ΤΟΥ ΑΡΘΡΟΥ d ΤΟΥ Ν . dddd/dddd ΑΝΤΙΚΑΘΙΣΤΑΤΑΙ ΩΣ ΕΞΗΣ : « d . ΚΑΤΑ ΤΗ ΔΙΑΡΚΕΙΑ ΤΟΥ ΕΛΕΓΧΟΥ , Η ΕΠΙΤΡΟΠΗ ΤΗΣ ΠΑΡΑΓΡΑΦΟΥ d , ΔΙΑ ΤΟΥ

In this example we can see that the model succesfully identifies all the labels despite the text containing many different modifications in the same article.

Example 2:

ΑΡΘΡΟ dd ΤΟ ΑΡΘΡΟ ddd ΤΟΥ Π.Δ. ddd/dddd , ΟΠΩΣ ΕΧΕΙ ΤΡΟΠΟΠΟΙΗΘΕΙ ΜΕ ΤΙΣ ΔΙΑΤΑΞΕΙΣ ΤΟΥ ΑΡΘΡΟΥ dd ΤΟΥ Π.Δ. ddd/dddd (Α΄ ddd) , ΑΝΤΙΚΑΘΙΣΤΑΤΑΙ ΩΣ ΕΞΗΣ : « ΑΡΘΡΟ dddΤΕΛΩΝΕΙΑ d. ΣΤΟ ΝΟΜΟ ΠΡΕΒΕΖΑΣ ΟΙ ΑΡΜΟΔΙΟΤΗΤΕΣ ΤΟΥ ΑΡΘΡΟΥ d ΑΣΚΟΥΝΤΑΙ ΑΠΟ ΤΟ ΤΕΛΩΝΕΙΟ ΠΡΕΒΕΖΑΣ ΚΑΙ ΤΟ ΤΟΠΙΚΟ ΤΕΛΩΝΕΙΑΚΟ ΓΡΑΦΕΙΟ ΠΑΡΓΑΣ . d. ΤΟ ΤΕΛΩΝΕΙΟ ΠΡΕΒΕΖΑΣ ΕΙΝΑΙ Α΄ ΤΑΞΗΣ , ΕΧΕΙ ΕΔΡΑ ΤΗΝ ΠΡΕΒΕΖΑ ΚΑΙ ΔΙΑΡΘΡΩΝΕΤΑΙ ΣΤΑ ΠΑΡΑΚΑΤΩ ΤΜΗΜΑΤΑ , ΠΟΥ ΕΧΟΥΝ ΩΣ ΕΞΗΣ : Α) ΤΜΗΜΑ ΔΙΟΙΚΗΤΙΚΗΣ ΥΠΟΣΤΗΡΙΞΗΣ ΚΑΙ ΔΙΚΑΣΤΙΚΟΥ Β) ΤΜΗΜΑ ΔΙΑΔΙΚΑΣΙΩΝ ΚΑΙ Ε.Φ.Κ . Γ) ΤΜΗΜΑ ΕΛΕΓΧΟΥ ΤΑΞΙΔΙΩΤΩΝ Δ) ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΩΝ , ΠΑΡΑΠΟΙΗΜΕΝΩΝ ΚΑΙ ΔΙΩΞΗΣ Ε) ΤΜΗΜΑ ΤΕΛΩΝΙΣΜΟΥ . d. ΣΤΗΝ ΚΑΤΑ ΤΟΠΟΝ ΑΡΜΟΔΙΟΤΗΤΑ ΤΟΥ ΤΕΛΩΝΕΙΟΥ ΠΡΕΒΕΖΑΣ ΠΕΡΙΛΑΜΒΑΝΕΤΑΙ : Ι) Η ΠΕΡΙΟΧΗ ΟΛΟΚΛΗΡΟΥ ΤΟΥ ΟΜΩΝΥΜΟΥ ΝΟΜΟΥ ΚΑΙ ΙΙ) ΟΙ ΧΩΡΟΙ ΤΟΥ ΑΕΡΟΔΡΟΜΙΟΥ ΑΚΤΙΟΥ (ΤΟΥ ΔΗΜΟΥ ΑΝΑΚΤΟΡΙΟΥ ΠΟΥ ΑΝΗΚΕΙ ΣΤΟ ΝΟΜΟ ΑΙΤΩΛΙΑΣ ΚΑΙ ΑΚΑΡΝΑΝΙΑΣ) . d. ΣΤΗΝ ΠΑΡΓΑ ΤΟΥ ΝΟΜΟΥ ΠΡΕΒΕΖΑΣ ΕΔΡΕΥΕΙ ΟΜΩΝΥΜΟ ΤΟΠΙΚΟ ΤΕΛΩΝΕΙΑΚΟ ΓΡΑΦΕΙΟ (ΑΡΘΡΟΥ dd ΠΑΡ . dd ΕΔ . d Ν . dddd/dddd – ΦΕΚ ddd Α΄) ΥΠΑΓΟΜΕΝΟ ΣΤΟ ΤΕΛΩΝΕΙΟ ΠΡΕΒΕΖΑΣ . »

Here we can see that the model correctly identifies the law under modification and correctly does not label the other article mentioned.

Example 3:

ΑΡΘΡΟ d d. Η ΠΑΡΑΓΡΑΦΟΣ ΤΟΥ ΑΡΘΡΟΥ d ΤΟΥ Ν . dddd/dddd ΑΡΙΘΜΕΙΤΑΙ ΩΣ ΠΑΡΑΓΡΑΦΟΣ d ΚΑΙ ΣΤΟ ΤΕΛΟΣ ΑΥΤΗΣ ΠΡΟΣΤΙΘΕΤΑΙ ΠΕΡΙΠΤΩΣΗ Ζ ' ΩΣ ΑΚΟΛΟΥΘΩΣ : « Ζ) ΤΗ ΣΥΜΠΡΑΞΗ ΑΠΟ ΚΟΙΝΟΥ ΜΕ ΤΑ ΠΑΝΕΠΙΣΤΗΜΙΑ , ΓΙΑ ΤΗ ΔΙΟΡΓΑΝΩΣΗ ΠΡΟΓΡΑΜΜΑΤΩΝ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ (Π.Μ.Σ.) ΚΑΙ ΤΗ ΔΙΕΞΑΓΩΓΗ ΕΡΕΥΝΑΣ ΣΕ ΤΟΜΕΙΣ ΕΝΔΙΑΦΕΡΟΝΤΟΣ ΤΗΣ ΕΛΛΗΝΙΚΗΣ ΕΜΠΟΡΙΚΗΣ ΝΑΥΤΙΛΙΑΣ . » . d. ΣΤΟ ΑΡΘΡΟ d ΤΟΥ Ν . dddd/dddd ΠΡΟΣΤΙΘΕΤΑΙ ΠΑΡΑΓΡΑΦΟΣ d ΠΟΥ ΕΧΕΙ ΩΣ ΑΚΟΛΟΥΘΩΣ : « d . ΜΕ ΠΡΟΕΔΡΙΚΟ ΔΙΑΤΑΓΜΑ , ΠΟΥ ΕΚΔΙΔΕΤΑΙ ΜΕΤΑ ΑΠΟ ΠΡΟΤΑΣΗ ΤΩΝ ΥΠΟΥΡΓΩΝ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ ΚΑΙ ΕΜΠΟΡΙΚΗΣ ΝΑΥΤΙΛΙΑΣ ΚΑΙ ΜΕ ΤΗ ΣΥΜΦΩΝΗ ΓΝΩΜΗ ΤΩΝ ΕΝΔΙΑΦΕΡΟΜΕΝΩΝ ΠΑΝΕΠΙΣΤΗΜΙΩΝ , ΜΠΟΡΕΙ ΝΑ ΟΡΓΑΝΩΝΟΝΤΑΙ ΚΑΙ ΝΑ ΛΕΙΤΟΥΡΓΟΥΝ , ΝΑ ΣΥΓΧΩΝΕΥΟΝΤΑΙ , ΝΑ ΜΕΤΟΝΟΜΑΖΟΝΤΑΙ Η ΝΑ ΚΑΤΑΡΓΟΥΝΤΑΙ Π.Μ.Σ. ΣΕ ΓΝΩΣΤΙΚΑ ΑΝΤΙΚΕΙΜΕΝΑ ΣΧΕΤΙΚΑ ΜΕ ΤΙΣ ΕΚΠΑΙΔΕΥΤΙΚΕΣ , ΕΡΕΥΝΗΤΙΚΕΣ ΚΑΙ ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΝΑΓΚΕΣ ΤΗΣ ΕΜΠΟΡΙΚΗΣ ΝΑΥΤΙΛΙΑΣ . ΤΑ ΠΡΟΓΡΑΜΜΑΤΑ ΑΥΤΑ ΟΡΓΑΝΩΝΟΝΤΑΙ ΑΠΟ ΤΑ ΠΑΝΕΠΙΣΤΗΜΙΑ ΣΕ ΣΥΝΕΡΓΑΣΙΑ ΜΕ ΤΙΣ Α.Ε.Ν . ΤΗΝ ΑΡΜΟΔΙΟΤΗΤΑ ΧΟΡΗΓΗΣΗΣ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΤΙΤΛΩΝ ΣΠΟΥΔΩΝ ΕΧΟΥΝ ΤΑ ΠΑΝΕΠΙΣΤΗΜΙΑ . » .

This final example showcases one of the mistakes in the results. Despite the overall high success rate, here the model fails to identify one of the two modifications that take place. Specifically, while it identifies the addition (“ΠΡΟΣΤΙΘΕΤΑΙ”) it misses the renumbering (“ΑΡΙΘΜΕΙΤΑΙ”).

In this chapter we have presented the results of our experiments both in tuning our different hyperparameters as well as those produced by our optimal model in a test dataset. Finally, we will comment on those results and present our conclusions.

6. CONCLUSION

With a medium-sized dataset of approximately 7000 annotations our model had an accuracy of 99%, a precision of 98%, a recall of 82% and an f1-score of 88% over all labels.

As we can see from the metrics and from the examples, it is possible to create a neural network with a very high degree of success in labeling the text of Greek legislative modifications. We believe that the results are sufficient to find practical application in automating the bulk of that task.

Should our model be integrated into the ΝΟΜΟΘΕΣΙ@ platform, a possible next step would be the parsing of the law label and automatic linking of the modification to the law it modifies. With some more effort, we believe it might be viable to develop a neural network-based solution to apply the changes to the original law.

As a final note, even though the results are good, the annotation still takes many hours of work and, as is obvious, no automated system could reach an accuracy of 100% with current technology. Given that there are already widely used formalised frameworks of version control (e.g. git) seeing great success in a variety of applications, we believe the benefits of using such a protocol make it the preferable infrastructure for legislative systems as well. We hope that the Greek government will soon recognise the advantages of semantic technologies in the organisation of our societies. With the necessary changes to its current, long outdated procedures it would be possible to have useful metainformation annotated during the inception of a law, rather than left as an afterthought requiring third-party development.

ABBREVIATIONS-ACRONYMS

ΦΕΚ	Φύλλο Εφημερίδας της Κυβέρνυσης
API	Application Programming Interface
ELI	European Legislation Identifier
NLP	Natural Language Processing
DNN	Deep Neural Network
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
BiLSTM	Bidirectional Long Short-Term Memory
BGD	Batch Gradient Descent
SGD	Stochastic Gradient Descent
MGD	Mini-batch Gradient Descent
PPV	Positive Predicted Value
TPR	True Positive Rate
OCR	Optical Character Recognition

REFERENCES

[1] I. Angelidis, I. Chalkidis, C. Nikolaou, P. Soursos, and M. Koubarakis, "Nomothesia: A Linked Data Platform for Greek Legislation," p. 13.

[2] T. Beris *et al.*, "Towards a Decentralized, Trusted, Intelligent and Linked Public Sector: A Report from the Greek Trenches," in *Companion Proceedings of The 2019 World Wide Web Conference*, San Francisco, USA, May 2019, pp. 840–849, doi: 10.1145/3308560.3317077.

[3] I. Angelidis, I. Chalkidis, and M. Koubarakis, "Named Entity Recognition, Linking and Generation for Greek Legislation," p. 10, 2018.

[4] I. Chalkidis, C. Nikolaou, P. Soursos, and M. Koubarakis, "Modeling and Querying Greek Legislation Using Semantic Web Technologies," in *The Semantic Web*, vol. 10249, E. Blomqvist, D. Maynard, A. Gangemi, R. Hoekstra, P. Hitzler, and O. Hartig, Eds. Cham: Springer International Publishing, 2017, pp. 591–606.

[5] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[6] I. Sutskever, *Training recurrent neural networks*. University of Toronto Toronto, Canada, 2013.

[7] A. Graves and J. Schmidhuber, "Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. Curran Associates, Inc., 2009, pp. 545–552.

[8] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," *ArXiv Prepr. ArXiv160301360*, 2016.

[9] T. Wang and K. Cho, "Larger-context language modelling," *ArXiv Prepr. ArXiv151103729*, 2015.

[10] J. Kiefer and J. Wolfowitz, "Stochastic Estimation of the Maximum of a Regression Function," *Ann Math Stat.*, vol. 23, no. 3, pp. 462–466, Sep. 1952, doi: 10.1214/aoms/1177729392.

[11] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *ArXiv14126980 Cs*, Jan. 2017, Accessed: Jun. 21, 2020. [Online]. Available: http://arxiv.org/abs/1412.6980.

[12] "lecture_slides_lec6.pdf." Accessed: Jun. 21, 2020. [Online]. Available: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

[13] S. Ruder, "An overview of gradient descent optimization algorithms," *ArXiv160904747 Cs*, Jun. 2017, Accessed: Jun. 21, 2020. [Online]. Available: http://arxiv.org/abs/1609.04747.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[15] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient BackProp," in *Neural Networks: Tricks of the Trade: Second Edition*, G. Montavon, G. B. Orr, and K.-R. Müller, Eds. Berlin, Heidelberg: Springer, 2012, pp. 9–48.

[16] C. D. Manning, P. Raghavan, and H. Schütze, "Introduction to Information Retrieval," p. 569, 2009.

[17] T. Mikolov, "Computing numeric representations of words in a high-dimensional space," US9740680B1, Aug. 22, 2017.

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *ArXiv13013781 Cs*, Sep. 2013, Accessed: Jun. 21, 2020. [Online]. Available: http://arxiv.org/abs/1301.3781.

[19] I. Angelidis, "Named Entity Recognition and Linking in Greek Legislation," Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών, 2018.

[20] M. Abadi *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *ArXiv160304467 Cs*, Mar. 2016, Accessed: Jun. 21, 2020. [Online]. Available: http://arxiv.org/abs/1603.04467.

[21] N. Reimers and I. Gurevych, "Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks," *ArXiv170706799 Cs*, Aug. 2017, Accessed: Jun. 21, 2020. [Online]. Available: http://arxiv.org/abs/1707.06799.