DEEP LEARNING - OBJECT DETECTION

by

Konstantinos Skepetaris

A master's thesis submitted in partial fulfillment of the requirements for the degree of

Information Systems & Services

Department of Digital Systems

University of Piraeus

2022

Approved by Associate Professor Dimosthenis Kyriazis
Chairperson of Supervisory Committee

Supervised by Professor Michael Filippakis

ABSTRACT

A thesis presented on solving computer vision problems for electronic documents by transforming them to text files and then using deep learning methods to build a natural language processing model for feature and information extraction for the same type of documents. An extensive analysis in the natural language process framework, the deep learning architectures, approach of manually training a model to achieve good accuracy in the results and explaining the asset management business problems regarding manipulating huge number of documents. We will try to go in details and mention every step in the information system we are building in python language, the pros and cons of investing in such machines and what the future holds in real scenarios in similar technological advancements. This thesis is based on scientific publications and articles, books and documentations.

TABLE OF CONTENTS

FIGURES TABLE

GLOSSARY

Annotation. Extra information associated with a particular point in a document.

Entity. A thing with distinct and independent existence.

Machine. The term computer became less appropriate and was replaced by the more generic term machine, which better suits the general-purpose character of the thing rather than computers that were used to perform calculations.

NLP. A pseudoscientific approach to communication, personal development, and psychotherapy.

Parse. Resolve (a sentence) into its component parts and describe their syntactic roles.

Python.

Train. The action of teaching a particular skill or type of behavior.

Transformer. A deep learning model.

Chapter 1

THE IMPORTANCE OF INFORMATION

All organizations and business operations mainly take place through the exchange of documents and information stored in them; information important, some less important and some irrelevant. Information must stay intact from its source and used accordingly to achieve business targets and help communication inside and outside of the organization. But what is information?

The real definition of information is the knowledge obtained from investigation, study, instruction or news and you share it through the act of communicating, whether verbally, nonverbally, visually, or through written word. The communication or reception of knowledge or intelligence. Also is a set of data which is processed in a meaningful way according to the given requirement. It is processed data which includes data that possess context, relevance, and purpose. It also involves manipulation of raw data. It is described through various types of communication, like written, oral, visual and audio communications. Information assigns meaning and improves the reliability of the data. It helps to ensure undesirability and reduces uncertainty. So, when the data is transformed into information, it never has any useless details. There are six types of information. Conceptual information comes from ideas, theories, concepts, hypotheses. Procedural information, or imperative knowledge, is the method of how someone knows to do something and is used by performing a task. Policy information focuses on decision-making and the design, formation and selection of policies. Stimulatory information is information that creates a response or stimulation amongst a person or group of people. Empirical information means information gained through human senses, observation, experimentation and the testing of a hypothesis by establishing documentation of patterns or behavior. Directive and descriptive information is about providing directions to a person or group of people to achieve a particular result and outcome. Other classifications of information can be defined by the attributes factual, analytical, subjective and objective. According to Adami C. (2016), information is defined as the difference between the actual observed entropy [1]of X, minus the actual observed entropy of X given that I know the state of Y (whatever that state may be). So, to get the most amount of information we need to minimize the uncertainty given of something we know. What an organization should know is the number of documents it has and their category classified. The uncertainty of acquiring all important information out of these documents given that we know how many of them are and what type they are. This is easy, right? Well, sounds so. However, the above conditional entropy is based on probabilities. Probabilities human mind and physical body is trying to maximize and be as efficient as possible. Probabilities that decay over the working hours, over weeks, months and year that come in the same position, doing the same job and information extraction, over the same documents. What a human mind induces to this equation is variation. Variation is something that most businesses cannot afford. They cannot also accept the time a person consumes to keep variating with the information extraction. This is the natural way of thinking in the time of information controlled by information systems and services, the era of zettabytes and be exact 79 zettabytes of data created, captured, copied, and consumed globally in 2021 or 79,000,000,000,000 gigabytes according to Statista. Of course, managing the security of information exchange between businesses and organizations provides guidance for planning, establishing, maintaining, and discontinuing information exchange and access between systems that are owned and operated by different organizations (internal or external) or that cross

[1] Entropy, also known as 'uncertainty', is something that is mathematically defined for a 'random variable'.

authorization boundaries. There are four phases of information exchange management; planning the information exchange, establishing the information exchange, maintaining the exchange and associated agreements, discontinuing the information exchange. Information exchange includes access to or the transfer of data outside of authorization boundaries to accomplish a mission or business function.

DOCUMENT STRUCTURE

The size of the document, the number of pages, the complexity, the structure (or lack of it) and the written method (handwritten or typed) varies vastly throughout the exchange channels of the businesses and organizations. The average silent reading rate in English is 238 words per minute (Marc Brysbaert, 2019). Now, take a sample of business documents and count the average words per page without severe loss of text understanding. Count, the times it might take a human to re-read certain parts or whole pages to extract information. Sometimes, the human brain also does calculations while reading to consolidate all necessary information from documents of business kind. Finally, the same person must write down or transfer this information somewhere and store it (either by handwriting or typing). The most convenient way of reducing these issues, because in 2022 these are issues, is to structure data. One way to do this is by introducing graphical methods of separating lines, words, phrases, creating tables, highlight with bold or underwrite, create sectors where each information in a page is shown etc. This really is helpful and thus can lead to creating a template for many uses and purposes; a template which is filled by everyone who uses it with specific information. Such documents can be receipts, government documents of citizens, a form to apply to something and so on. Then, it only takes a few seconds to look at some specific fields and extract what is necessary. Be aware though, these documents are signed and filled by a human and humans make errors in a much higher rate than a sophisticated system for a specific purpose. This means that information validation is an important part while reading a document. Analytical thinking and cross checking may be more important. What someone expects to find might not always be there, hidden in the document.

Chapter 2

NATURAL LANGUAGE PROCESSING

Moving fast forward and jumping on the speeding rail of technological advancements, the abbreviation NLP is heard all around and especially in the business facilities. Natural language processing or NLP was a motivation and an indexation matter. As stated by Braun, S., & Schwind, C. (1976), an information retrieval system should have an indexing device which automatically extracts concepts from the documents to be stored, in order to represent the content of the documents for later retrieval. Until today, the purpose of artificial intelligence and computational linguistics is still devoted to creating computer systems that allow their users to formulate requests in natural language. Machines that understand and respond to text or voice data like a human would. In technical terms, NLP refers to computer science or more specifically to artificial intelligence, giving computers the ability to understand text and spoken words in much the same way human beings can combining statistical, machine learning, and deep learning models. Included processes in this effort require text translation from one language to another, speech-to-text systems, data analysis in very large datasets, part of speech tagging[2], named entity recognition[3], word sense disambiguation[4], co-reference

[2] process of determining the part of speech of a particular word or piece of text based on its use and context

[3] the identification of key information in the text and classification into a set of predefined categories

[4] the problem of determining which "sense" (meaning) of a word is activated by the use of the word in a particular context

resolution[5], sentiment analysis[6], natural language generation[7]. The main processes in NLP are lexical analysis, syntactic analysis, semantic analysis and output transformation process. Lexical analysis brakes sentence into smaller components called tokens. In the step of syntactic analysis, tokens are validated with a specified sentence structure, to be used in the meaning analysis procedure. The semantic analysis process analyzes the meaning of tokens previously created, to produce meaning. Last step stores the results of previous steps in a specific format to be later retrieved.

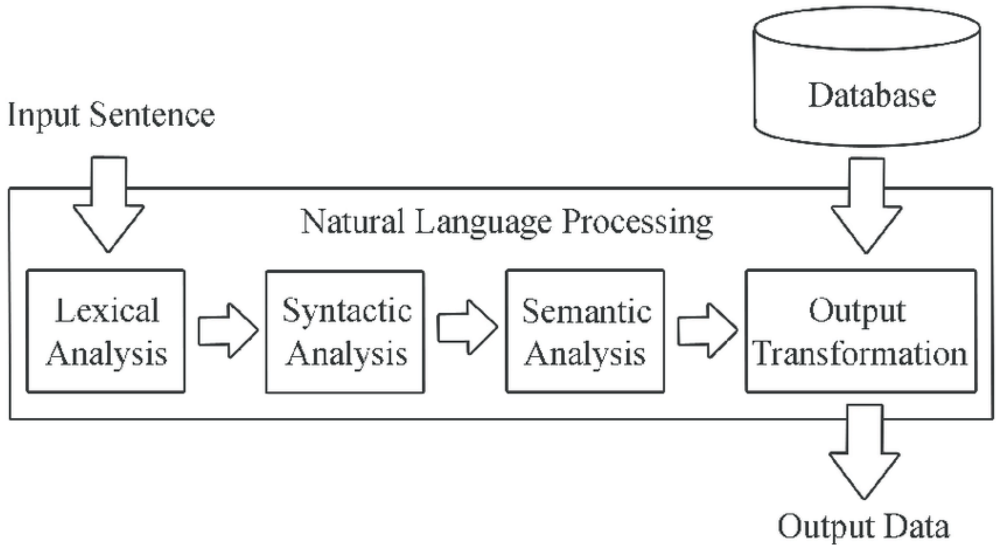


Figure 1: Natural Language Processing steps

Think of NLP as a bridge between natural languages and computers. Newest NLP models, rely on data-driven approaches which help them become more powerful and robust.

[5] the task of finding all expressions that refer to the same entity in a text

[6] a natural language processing (NLP) technique used to determine whether data is positive, negative or neutral

[7] a process that produces natural language output

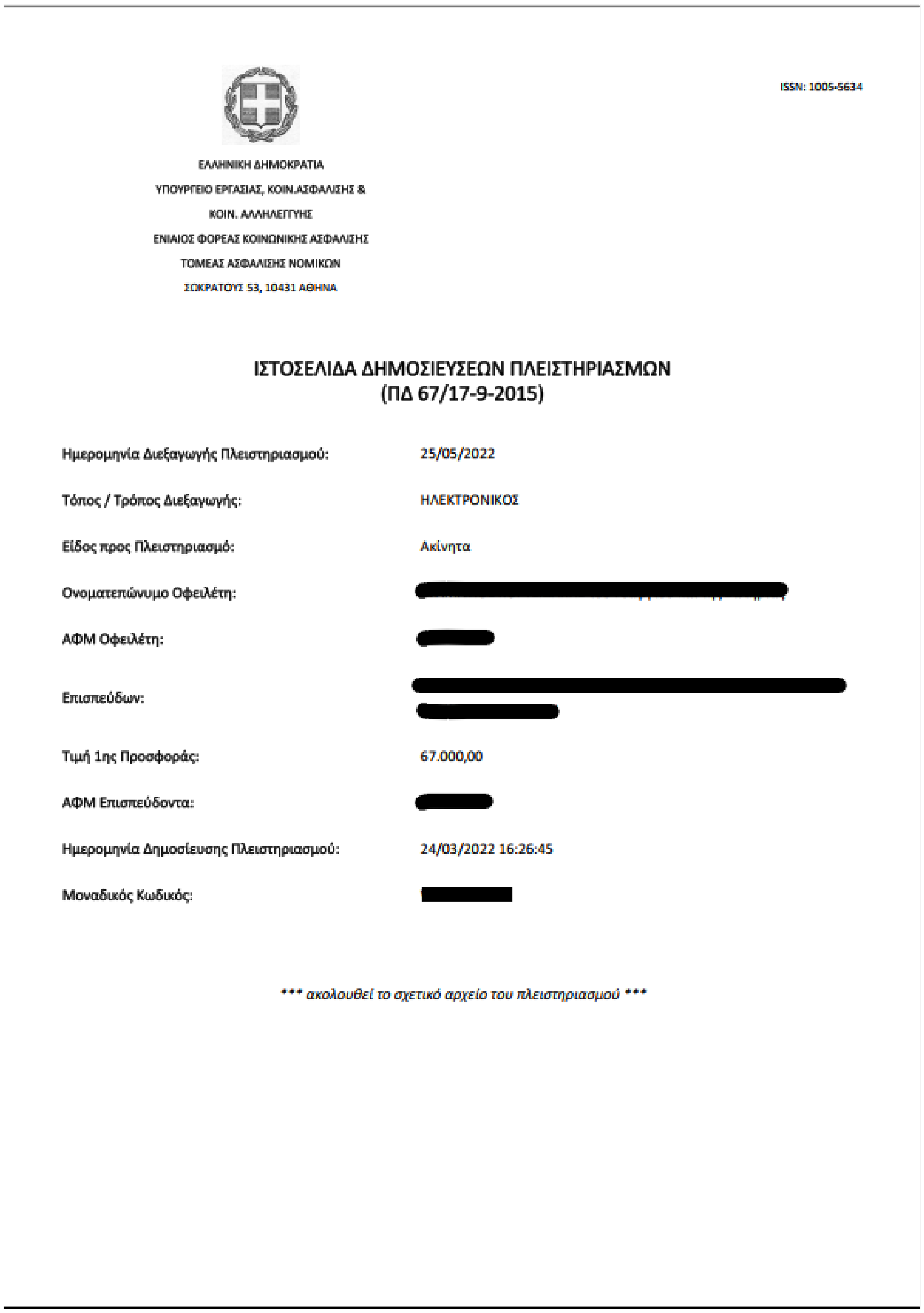
NLP systems need to learn word representations because of problems with the atomic representation of the symbols. Characters, words, sentences or other linguistic components can all be included in the encoded input features but it is preferred to provide a compact representation of the words rather than a sparse one. There are unsupervised algorithms such as word2vec, doc2vec, BoW and the goal is to learn fixed-length representations from variable-length text parts such as sentences and documents. For feature representation we can find encoder-decoder architectures such as One-Hot Representation, Continuous Bag of Words, Word-Level Embedding, Character-Level Embedding. In one-hot encoding representation, each distinct element that must be represented, has its own dimension, resulting in a very high dimensional representation. For instance, less correlated pairs such as "sky" and "car" will be closer to each other in the representation space than higher correlated pairs like "cream" and "cheese". The Continuous Bag-of-Words model (CBOW) has been applied frequently in NLP settings. CBOW tries to predict a word given its surrounding context, which usually consists of a few nearby words. Word embedding is a technique in which words with related semantics become highly correlated in the representation space. Character level embeddings include key signal to improve model performance. Having the character embedding, every single word's vector can be formed. Future directions and challenges deal with training large models like BERT and GPT-2 on large amounts of text data. Due to the high dimensionality of embedding vectors, even small amounts of noise can have a big impact on how quickly models train and how well they perform. A core problem is to find a way to create a private embedding that has good performance for downstream tasks. Thus, we do not want to leak information of training data when releasing a language model.

NLP USE CASES

In practice NLP is used by many machines in our everyday life. Thing we do and interact with also respond back like a human. Machines are there to help us humans achieve more, faster and precisely. Virtual assistants, that you interact with daily use NLP-based machine learning technology to decode and process your voice request automatically. Using NLP algorithms, these assistants can adjust to each individual user's needs and learn exactly what to do by assessing previous interactions, recalling queries, and connecting with other information systems. Chatbots, that use NLP to simulate human conversation by interpreting data based on the text you input. Therefore, chatbots can understand and decode the intention behind each sentence. This includes picking out the overall tone, emotion, and even identifying closely related topics or keywords. Search engines, where NLP machine learning is what helps these engines understand the intent behind each word and suggest the most relevant topics or subjects in context and bring the most in trend results. Predictive text, to predict, auto-complete, and suggest the most grammatically correct word without the hassle of typing each letter one by one. Sentiment analysis, to categorize the gradation of sentences, monitor public's overall sentiment in social media conversations regarding your brand or product., the detection of specific keywords related to your brand or product contain which variation so you can make data-driven decisions accordingly. Spam detection, where text classification teaches the machines to scan emails for language that often indicates spam or phishing.

ASSET MANAGEMENT

Previously mentioned, business documents have overflowed the organizations. One big example is the management of bank portfolios with collateral loans and foreclosured assets, especially in countries where the problem is huge and a lot of them are in need of external help to manage and sell. In the banking sector, agencies and companies who offer asset management and real estate services are trying to catch a share of the problem, invest time and make a deal to help desizure the problem, manage the assets in a way a bank was not able to do it (not a surprise after all the 2008 financial and mortgage crisis), sell or rent the assets and provide advices, solutions and expertise where needed. The legal part of these deals, between a bank and an asset management organization, is very complicated. An even big obstacle is when such organizations finally achieve to manage such portfolios but come up with the piles and piles of documents banks share with them that are attached to the mortgaged (or not) assets of the collateral portfolio. The types of documents vary a lot and not all of them are easy and fast to read and extract information while other have references to other documents and the flow does not end. However, this does not mean there is not a partial solution to this problem. A problem that is timing consuming and nerve recking. What if a machine could automate this process, classify documents per type and a human learns it to read specific type of asset documents and extract fast and accurately information out of them? Information which can later be stored by the employee and used to manage this asset, reach out clients to sell it and deal with the debts of the borrowers. Such documents could be valuation reports, technical reports, title deed documents, allocation reports, records of liens, real property reports, cadaster documents, auction reports and so on. It would not make a sense to write down all the possible type of documents as for the case of the thesis it is more convenient and correct to focus in on type of document and maybe integrate NLP and deep learning to read that document and convince an asset management organization to invest in such machine. As a data analyst who would work for such company interested in investing time and money on such information system, a good approach would be to apply to one type of document. In our case, we get into the shoes of that data analyst and we choose to work with auction documents because they are unstructured and a good test for the machine's NLP capabilities to extract accurately the information.



ISSN: 1005-5634

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΕΡΓΑΣΙΑΣ, ΚΟΙΝ.ΑΣΦΑΛΙΣΗΣ &
ΚΟΙΝ. ΑΛΛΗΛΕΓΓΥΗΣ
ΕΝΙΑΙΟΣ ΦΟΡΕΑΣ ΚΟΙΝΩΝΙΚΗΣ ΑΣΦΑΛΙΣΗΣ
ΤΟΜΕΑΣ ΑΣΦΑΛΙΣΗΣ ΝΟΜΙΚΩΝ
ΣΩΚΡΑΤΟΥΣ 53, 10431 ΑΘΗΝΑ

ΙΣΤΟΣΕΛΙΔΑ ΔΗΜΟΣΙΕΥΣΕΩΝ ΠΛΕΙΣΤΗΡΙΑΣΜΩΝ
(ΠΔ 67/17-9-2015)

Ημερομηνία Διεξαγωγής Πλειστηριασμού:	25/05/2022
Τόπος / Τρόπος Διεξαγωγής:	ΗΛΕΚΤΡΟΝΙΚΟΣ
Είδος προς Πλειστηριασμό:	Ακίνητα
Ονοματεπώνυμο Οφειλέτη:	[redacted]
ΑΦΜ Οφειλέτη:	[redacted]
Επισπεύδων:	[redacted]
Τιμή 1ης Προσφοράς:	67.000,00
ΑΦΜ Επισπεύδοντα:	[redacted]
Ημερομηνία Δημοσίευσης Πλειστηριασμού:	24/03/2022 16:26:45
Μοναδικός Κωδικός:	[redacted]

**** ακολουθεί το σχετικό αρχείο του πλειστηριασμού ****

Figure 2: Auction document example (page1/2)

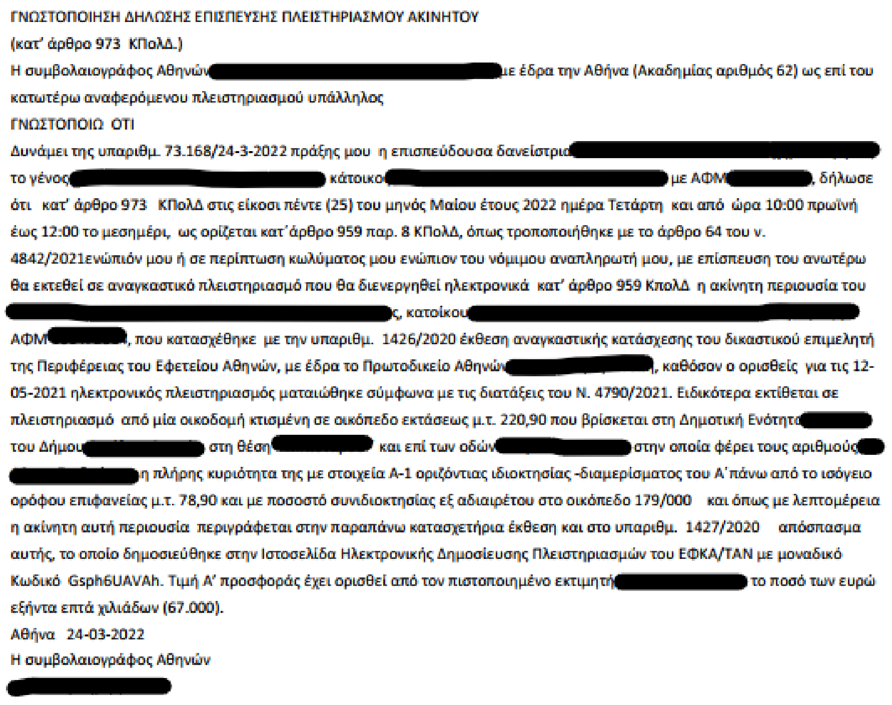

ΓΝΩΣΤΟΠΟΙΗΣΗ ΔΗΛΩΣΗΣ ΕΠΙΣΠΕΥΣΗΣ ΠΛΕΙΣΤΗΡΙΑΣΜΟΥ ΑΚΙΝΗΤΟΥ
(κατ' άρθρο 973 ΚΠολΔ.)
Η συμβολαιογράφος Αθηνών [redacted] με έδρα την Αθήνα (Ακαδημίας αριθμός 62) ως επί του κατωτέρω αναφερόμενου πλειστηριασμού υπάλληλος
ΓΝΩΣΤΟΠΟΙΩ ΟΤΙ
Δυνάμει της υπαριθμ. 73.168/24-3-2022 πράξης μου η επισπεύδουσα δανείστρια [redacted] το γένος [redacted] κάτοικο [redacted] με ΑΦΜ [redacted], δήλωσε ότι κατ' άρθρο 973 ΚΠολΔ στις είκοσι πέντε (25) του μηνός Μαίου έτους 2022 ημέρα Τετάρτη και από ώρα 10:00 πρωϊνή έως 12:00 το μεσημέρι, ως ορίζεται κατ΄άρθρο 959 παρ. 8 ΚΠολΔ, όπως τροποποιήθηκε με το άρθρο 64 του ν. 4842/2021ενώπιόν μου ή σε περίπτωση κωλύματος μου ενώπιον του νόμιμου αναπληρωτή μου, με επίσπευση του ανωτέρω θα εκτεθεί σε αναγκαστικό πλειστηριασμό που θα διενεργηθεί ηλεκτρονικά κατ' άρθρο 959 ΚπολΔ η ακίνητη περιουσία του [redacted]ς, κατοίκου [redacted] ΑΦΜ [redacted], που κατασχέθηκε με την υπαριθμ. 1426/2020 έκθεση αναγκαστικής κατάσχεσης του δικαστικού επιμελητή της Περιφέρειας του Εφετείου Αθηνών, με έδρα το Πρωτοδικείο Αθηνών [redacted], καθόσον ο ορισθείς για τις 12-05-2021 ηλεκτρονικός πλειστηριασμός ματαιώθηκε σύμφωνα με τις διατάξεις του Ν. 4790/2021. Ειδικότερα εκτίθεται σε πλειστηριασμό από μία οικοδομή κτισμένη σε οικόπεδο εκτάσεως μ.τ. 220,90 που βρίσκεται στη Δημοτική Ενότητα [redacted] του Δήμου [redacted] στη θέση [redacted] και επί των οδών [redacted] στην οποία φέρει τους αριθμούς [redacted] η πλήρης κυριότητα της με στοιχεία Α-1 οριζόντιας ιδιοκτησίας -διαμερίσματος του Α΄πάνω από το ισόγειο ορόφου επιφανείας μ.τ. 78,90 και με ποσοστό συνιδιοκτησίας εξ αδιαιρέτου στο οικόπεδο 179/000 και όπως με λεπτομέρεια η ακίνητη αυτή περιουσία περιγράφεται στην παραπάνω κατασχετήρια έκθεση και στο υπαριθμ. 1427/2020 απόσπασμα αυτής, το οποίο δημοσιεύθηκε στην Ιστοσελίδα Ηλεκτρονικής Δημοσίευσης Πλειστηριασμών του ΕΦΚΑ/ΤΑΝ με μοναδικό Κωδικό Gsph6UAVAh. Τιμή Α' προσφοράς έχει ορισθεί από τον πιστοποιημένο εκτιμητή [redacted] το ποσό των ευρώ εξήντα επτά χιλιάδων (67.000).
Αθήνα 24-03-2022
Η συμβολαιογράφος Αθηνών
[redacted]

Figure 3: Auction document example (page 2/2)

REQUIREMENTS AND FIRST APPROACH

The first and most important step is to write down the requirements of what we try to implement and build. Simplicity is the base but intention to detail is perfection and accuracy. Requirements are basically the needs and the problem. The needs are information extraction of specific fields, and the problem is we need that fast and as accurate as possible to justify the costs. We will define some labels that contain important information about assets and debtors of mortgaged assets that are now going through an auction due to foreclosure. Auction date; the date that the auction is taking place. Tax id of debtor; the unique tax id number of the debtor as was created by the state in which he lives and has citizenship. Auction id; a unique reference code for each auction which takes place through auction sites like https://deltio.tnomik.gr/. Foreclosure id; a unique reference code created by the bailiff responsible for the auction. Accelerator; the one who initiates the auction process. Tax id accelerator; the unique tax id number of the accelerator as was created by the state in which it pays taxes to. Property type; the property's type such as house, apartment, maisonette, storage, hotel etc. Size of land; the size in square meters of the land in which the property is built. Percentage of ownership; is the percentage of the direct, universal and absolute authority of the person over the thing (or asset in this case). Notary public; of the common law is a public officer constituted by law to serve the public in non-contentious matters usually concerned with general financial transactions, estates, deeds, powers-of-attorney, and foreign and international business. Ten labels or characteristics of the auction reports (document type) that we think are important and want to be extracted because this saves a lot of time from a person to do it. Previously, we talked about NLP and how machines are trained like humans. It is true, we must train the machine to read each document line by line, show it where each of the above labels is present in a big sample of documents and let it show the results of its apprenticeship. Then, it will easily and fast find what we want in documents of the same type that it has not seen before. Voila! We answered one of the difficulties we will come up with. We need a set of training documents and a set of test documents to see the accuracy. Like a student who learns mathematics and does his exercises and the same type of exercises repeatedly, only to be tested once, in the exams. He is then declared aware and knowledgeable of mathematics. Labels are another thing that needs attention on how to approach. Such a process is called named entity recognition or NER. Wait, will the machine be able to read the documents? Not all of them are in the form we want. Text files around hard to find around the web. Very few businesses use them to trade information to say the least. Nor that I know any other organizations or companies that do. And if the files are in the most common extension of PDF? Yes, Portable Document Format (PDF), standardized as ISO 32000, is a file format developed by Adobe in 1992 to present documents, including text formatting and images, in a manner independent of application software, hardware, and operating systems. However not all PDFs have the stored information in text lines. PDFs can contain images or even worse image that have text. Now that is one more difficulty a machine faces and struggles with sources and time to fight.

SOURCE OF DATA

Yet, the important piece of the puzzle is the source of documents. Organizations today understand the value of data. Most use their internal data to drive important decision-making processes. Businesses have private or public access to such auction reports. Not being able to access such data sources (because information is data) is a problem. They have made great strides in collecting and utilizing data from their own activities. It is like the student who is missing the books to learn. As organizations scrambled to understand the world's changing patterns around them, they discovered little of use in their internal data. A source of external data could—and still can—help organizations plan and respond at a granular level. To predict outcomes and generate investment returns, analysts and data scientists in investment firms have gathered "alternative data" from a variety of licensed and public data sources. The rule of three steps applies over again; it is summarizing the necessary steps. So, three steps are to create value out of the external source that provide data. Establish a dedicated team for external data sourcing. A key role to find a dedicated data scout to identify operational, cost, and growth improvements that could be powered by external data. Someone who would be responsible to use that data, plan the use cases to focus on, identifying and prioritizing data sources for investigation, and measuring the value generated through use of external data. Second step; develop relationships with data marketplaces and aggregators. These relationships can give organizations ready access to broader data. Finally, prepare the data architecture for new external-data streams. Ensure that the data architecture is flexible enough to support the integration of incoming data. For our purpose, the type of documents was not chosen by luck as the source we are looking for is open to the public. We speak of auctions; people are the asking factor and the demand is https://deltio.tnomik.gr/ or other similar auctions sites in Greece. Auction is the legal process whereby assets (e.g. a property) of the debtor are publicly sold off in order to satisfy the claims of creditors. The process of managing and conducting electronic auctions is undertaken by certified notaries who act as Auction Clerks. Therefore, by using the electronic auction service, anyone interested can participate in electronic auctions from wherever they are without their physical presence, but simply by using a computer. A property may be sold at a public auction, only when there is an enforceable title (e.g. payment order or final court decision), which is notified to the debtor. Immediately, the next stage of the process is the publication of the auction by the bailiff, who prepares and posts the excerpt of the confiscation report, on the Auction Publications Website of the Judicial Publications Bulletin of the Legal Fund (https://deltio.tnomik.gr/). After the publication in the Legal Fund, the competent Auction Officer must post it on the website https://www.eauction.gr. The latter is the platform for conducting electronic auctions. An auction report in PDF format is provided. Working every day in a company with such documents gives us the opportunity to responsibly have a big amount of them. Take also into consideration data preparation, which is an important step before analysis; reformat and correct data; identify and assign two different values as the same across multiple data sources and enrich the one bussiness' dataset with the external.

C h a p t e r 3

PYTHON PROGRAMMING LANGUAGE

To build such an information system to handle NLP and implement object detection techniques and deep learning methods, Python programming language will be used. Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed. Python was originally conceived by Guido van Rossum as a hobby project in December 1989. As Van Rossum's office remained closed during Christmas, he was looking for a hobby project that will keep him occupied during the holidays. He planned to create an interpreter for a new scripting language, and named the project Python. Thus, Python was originally designed as a successor to ABC programming language. The first version of Python was released in January 1994 and included a number of new features and functional programming tools including lambda, filter, map and reduce. Later, several new features like keyword arguments, built-in support for complex numbers, and a basic form of data hiding. The second version introduced a new list comprehension feature and a garbage collection system; gave preference to alphabetic keywords over punctuation characters. Also, the garbage collection system effectuated collection of reference cycles. There was also support for nested scopes, and unification of Python's classes and types into a single hierarchy. The third and current version of Python added deprecated features and backward incompatibility but foremost now uses text and data instead of Unicode and 8-bit strings.

FROM DOCUMENT TO TEXT FILES: OCR AND NEURAL NETWORKS

The goal to achieve here is to read pdf documents, auction reports to be more specific, that contain only text. We treat it as an image to test the machine's object detection capabilities, meaning that these pdfs could easily be read by a Python library directly and copy its contents to a text file to work later on with NLP. However, there may be a few cases where these auction reports may have the text as an image and such python libraries could not parse the text. There are libraries however that would fit the purpose but let us not rush there yet. We talked about python but how do we use it? Python is free and available free of charge, even for commercial purposes. It is open source and anyone can contribute to Python development (by fixing code bugs, writing unit tests for functions in the standard library, writing documentation for functions in the standard library etc.). It is accessible and people of all ages, from school children to retirees, have learned Python. Can be versatile, meaning Python can help you solve problems in many fields, including scripting, data science, web development, GUI development, and more and it is also powerful enough for anyone to code small scripts to automate repetitive tasks, and create complex and large-scale enterprise solutions. Python works on Linux, Mac, Windows, and several other platforms. It comes preinstalled on macOS and on most Linux distributions. However, if you want to be up to date, then you probably need to download and install the latest version. You also have the choice of using different Python versions in different projects if you want to.

To start programming someone will need a text editor / integrated development environment (IDE), such as VS Code and PyCharm. An integrated development environment (IDE) is software for building applications that combines common developer tools into a single graphical user interface (GUI). An IDE typically consists of the source code editor, local build automation and debugger. The source code editor is a text editor that can assist in writing software code with features such as syntax highlighting with visual cues, providing language specific auto-completion, and checking for bugs as code is being written. The local build automation creates a local build of the software for use by the developer, like compiling computer source code into binary code, packaging binary code, and running automated tests while the debugger on the other hand is a program for testing other programs that can graphically display the location of a bug in the original code. On the other side are the libraries. Generally, a library is a collection of books or is a room or place where many books are stored to be used later. Similarly, in the programming world, a library is a collection of precompiled codes that can be used later in a program for some specific well-defined operations. Libraries in Python are a collection of related modules. It contains bundles of code that can be used repeatedly in different programs. It makes Python Programming simpler and convenient for the programmer. As we don't need to write the same code again and again for different programs. Python libraries play a very vital role in fields of Machine Learning, Data Science, Data Visualization, etc. We use libraries so that we don't need to write the code again in our program that is already available. Python comes with a standard library which is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below. The library contains built-in modules (written in C) that provide access to system functionality such as file I/O that would otherwise be inaccessible to Python programmers, as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming. Some of these modules are explicitly designed to encourage and enhance the portability of Python programs by abstracting away platform-specifics into platform-neutral APIs. The Python Standard Library plays a very important role. Without it, the programmers cannot have access to the functionalities of Python. Libraries are imported in each program's code (generally in the beginning). This enables a programmer to utilize all of their functions and capabilities. From a first time programmer to a more experienced one we move forward our time learning machine a few months and now we start approaching the problem of which library to use to help in building a program that will extract text from auction report documents in pdf format. OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV like all libraries have some methods that a programmer can use to achieve different results. OpenCV started in the early 2000s from Gray Bradsky, a computer vision engineer at Intel back then. Bradsky and a team of engineers, mostly from Russia, developed the first versions of OpenCV internally at Intel before making v0.9 of it open source software (OSS) in 2002. Bradsky then transitioned to Willow Garage, with the former founding members of OpenCV. Among them were Viktor Eurkhimov, Sergey Molinov, Alexander Shishkov, and Vadim Pisarevsky (who eventually started the company ItSeez, which was acquired in 2016 by Intel), who began supporting the young library as an open-source project. OpenCV is written in C++ and its primary interface is in C++. There are bindings in Python, Java and MATLAB/OCTAVE. The API for these interfaces can be found in the online documentation. It supports hardware acceleration and runs on desktop operating systems of Windows, Linux, macOS, FreeBSD, NetBSD, OpenBSD and mobile operating systems of Android, iOS, Maemo, BlackBerry 10. Importing OpenCV library in python is easy and we can start by loading the auction report, the pdf pages need to be split and turn into images. PyMuPDF library is a Python binding for MuPDF. It supports not only PDF, but also XPS, OpenXPS, CBZ, CBR, FB2 and EPUB formats and is maintained and developed by Artifex Software, Inc. MuPDF stands out among all similar products for its top rendering capability and unsurpassed processing speed. At the same time, its "light weight" makes it an excellent choice for platforms where resources are typically limited, like smartphones; so does PyMuPDF. It runs and has been tested on Mac, Linux and Windows for Python versions 3.6 and up. Other platforms should work too, as long as MuPDF and Python support them. It fully supports standard metadata since not all entries may always contain data. To access the auction report pdf document, it must be opened with the a statement and stored as a doc object (doc = fitz.open(filename)). Filename must be a Python string (or a pathlib.Path) specifying the name of an existing file. It is also possible to open a document from memory data, or to create a new, empty PDF. Now we can render a page into a raster or vector (SVG) image, optionally zooming, rotating, shifting or shearing it. Raster (or bitmap) images are the types of images that are produced when scanning or photographing an object. Raster images are compiled using pixels, or tiny dots, containing unique color and tonal information that come together to create the image. Since raster images are pixel based, they are resolution dependent. The number of pixels that make up an image as well as how many of those pixels are displayed per inch, both determine the quality of an image. As you may have guessed, the more pixels in the image and the higher the resolution is, the higher quality the image will be. On the other hand, instead of trying to keep track of the millions of tiny pixels in a raster image, vector images, or line art, keep track of points and the equations for the lines that connect them. Vector images are made up of paths or line art that can infinitely scalable because they work based on algorithms rather than pixels. One of the greatest things about vector images is that you can re-size them infinitely larger or smaller, and they will still print out just as clearly, with no increase (or decrease) in file size. After iterating over the pages of the doc (for page in doc:), we can create an object which (in this case) contains an RGB image of the page, ready to be used for many purposes (pix = page.get_pixmap()). Tweaking resolution / DPI, colorspace (e.g., to produce a grayscale image or an image with a subtractive color scheme), transparency, rotation, mirroring, shifting, shearing, etc. is available. The consistency of this object mathematically speaking is a number of methods and attributes. Among them are the integers width, height (each in pixels) and stride (number of bytes of one horizontal image line). Attribute samples represent a rectangular area of bytes representing the image data (a Python bytes object). Furthermore, there are methods (techniques actually) that will improve the preprocess of the image. Using gray image and applying Otsu's automatic thresholding algorithm and inverting the binary threshold (cv2.THRESH_BINARY_INV), the text is now white (foreground) and background is cleansed. We still have many ways to precleans images that have text. Application of a distance transform to thresh image using a maskSize (cv2.distanceTransform(thresh, cv2.DIST_L2, x)), where x is the magnification times. This is basically a calculation that determines the distance from each pixel to the nearest 0 pixel (black) in the input image. Next, can follow a normalization and scale distance to a specific range (cv2.normalize(dist, dist, 0, 1.0, cv2.NORM_MINMAX)). Distance transformation applies a larger distance from the foreground pixels to the background helping to clean the noise in the image's background. To summarize, opencv offers many image filtering option, geometric and miscellaneous image transformations, drawing function, color space conversions, colormaps, planar subdivision, feature detection, object detection and so on, all found in the official documentation to fine tune accordingly in every application (https://docs.opencv.org/4.4.0/d7/dbd/group__imgproc.html). Yet we are not finished. On the now processed images, Optical Character Recognition or OCR is applied, to transform a two-dimensional image of text, that could contain machine printed or handwritten text from its image representation into machine-readable text. OCR as a process generally consists of several sub-processes to perform as accurately as possible, such as preprocessing of the image, text localization, character segmentation, character recognition and post processing. Optical Character Recognition remains a challenging problem when text occurs in unconstrained environments, like natural scenes, due to geometrical distortions, complex backgrounds, and diverse fonts. Handwritten text recognition and handwritten document recognition is the ability to recognize all the text from an input image. Ability not only to recognize all the text from the input document in a humanly correct order, but also to tag sub-sequences of the predicted sequence of characters given some layout classes. Images have specific dimensions; height, width and channels (1 for gray-scaled images and 3for RGB images). Document understanding includes a set of tasks whose purpose is to extract, classify, interpret, contextualize, and search information from documents; it implies Optical Character Recognition (OCR). The authors of "Chargrid: Towards Understanding 2D Documents" (2018) suggested representing textual documents as a 2D representation of one-hot encoded characters, which are produced by an OCR called chargrid. The idea is to keep the spatial information between the textual entities. To implement chargrid we need to extract information such as- each character coordinates (positional information), textual information, etcetera from the image of the document and this will be done with the help of one open-source OCR tool that I have also implemented and user friendly is pytesseract. The idea behind chargrid representation of document is to take each character and replace it's bounding box pixels with specific integer index. The character's encoding is superseded by word embeddings through the use of the BERT model.

Pytesseract's engine is popular, even though it can be painful to implement and modify sometimes. Tesseract was originally developed at Hewlett-Packard Laboratories Bristol UK and at Hewlett-Packard Co, Greeley Colorado USA between 1985 and 1994, with some more changes made in 1996 to port to Windows, and some C++ in 1998. In 2005 Tesseract was open sourced by HP. From 2006 until November 2018 it was developed by Google.. It supports a wide variety of languages, is compatible with many programming languages and frameworks and it can be used with the existing layout analysis to recognize text within a large document, or it can be used in conjunction with an external text detector to recognize text from an image of a single text line. The basic OCR flow is an API request [8], input image, pre-process, ingest that to tesseract OCR engine combined with Leptonica and trained data set and out of it we have an export of the text. Leptonica is a pedagogically oriented open source library containing software that is broadly useful for image processing and image analysis applications while trained data set can be provided per each language. Tesseract uses neural network subsystem which is focused on line recognition, but also still supports character pattern recognition. Tesseract has Unicode (UTF-8) support and can recognize more than 100 languages "out of the box". It supports various image formats including PNG, JPEG and TIFF and can also be trained to recognize other languages. To recognize an image containing a single character, Convolutional Neural Networks (CNN) are used. Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.

[8] Application programming interfaces (APIs) requests, are a message sent to a server asking an API to provide a service or information.

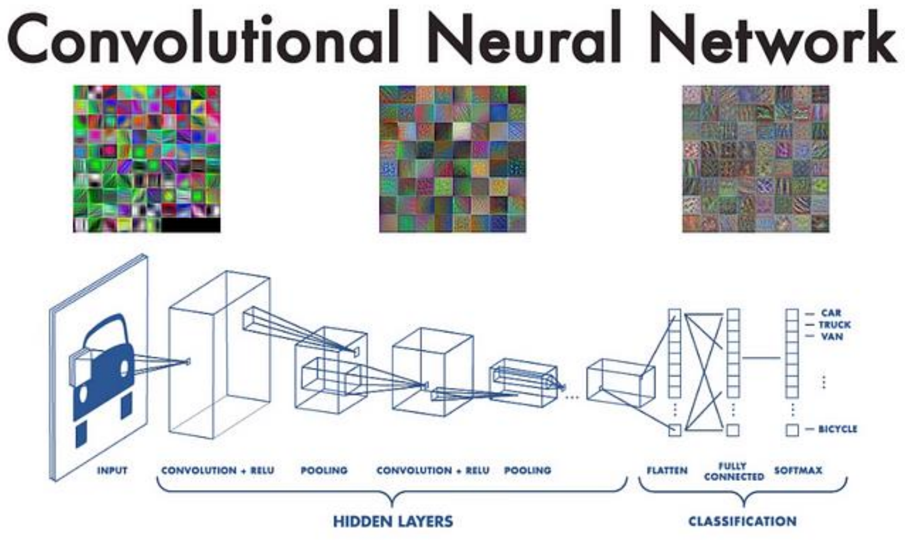


Figure 4: The architecture of Convolutional Neural Network

However, there are recurrent neural networks (RNN) and Long Short-Term Memory networks (LSTM) which are commonly used for natural language processing and speech recognition whereas convolutional neural networks (ConvNets or CNNs) are more often utilized for classification and computer vision tasks. How do RNNs work? Actually, they are networks with loops in them, allowing information to persist; can be thought of as multiple copies of the same network, each passing a message to a successor and being able to connect previous information to the present task. For example, in a sentence where there is an obvious (for a human) missing word, where the gap between the relevant information and the place that it's needed is small, RNNs can learn to use the past information. However, RNNs become unable to learn to connect the information, a gap between the relevant information and the point where it is needed to become very large. In theory, RNNs are capable of handling such "long-term dependencies." Fortunately, LSTMs are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber

(1997). They can store information for long periods of time. LSTMs also have the chain structure of repeating modules like RNNs.

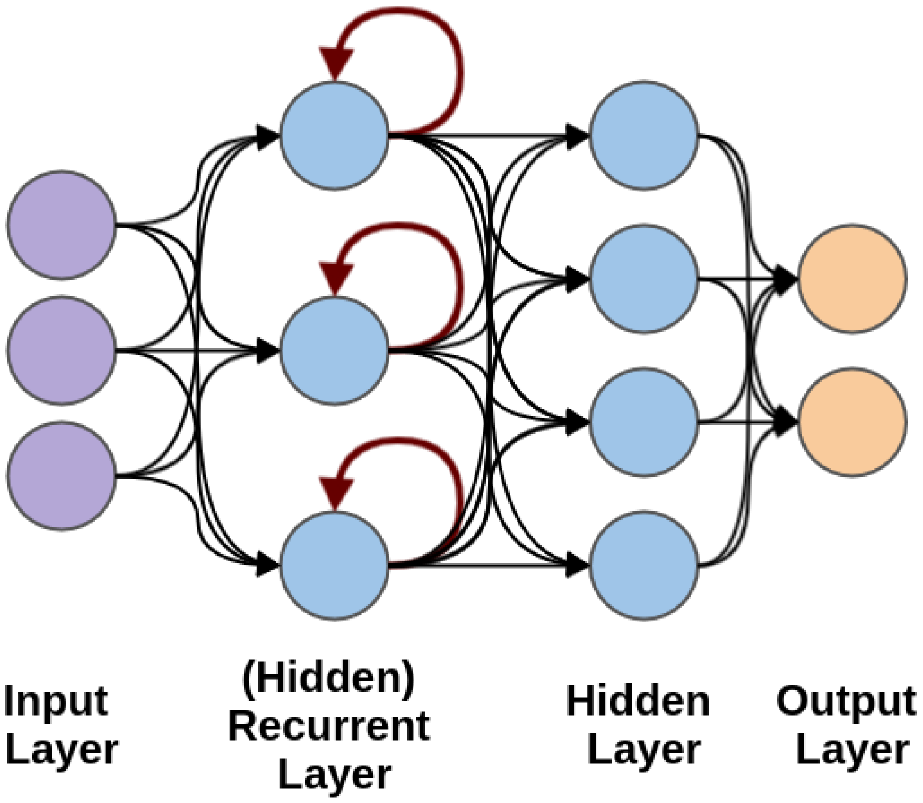


Figure 5: Recurrent Neural Network (RNN), with additional feed-forward layer

Back to our problem, all of what we learn packs up in pytesseract method image_to_string. Takes as an input the processed image, languages English and Greek and the installation path of tesseract. We need to specify the language because auction reports contain both English and Greek characters. This will make things complicated and surely when someone builds a machine like this it requires a lot of fine tuning of the opencv's parameters and methods of processing an image to be as “readable” as possible by tesseract. Supposedly everything worked right for us, meaning the output text was extracted almost intact from the pdf, we are happy and good to move with further analysis.

NAMED ENTITY RECOGNITION AND HUMAN ANNOTATION

A programmer will face many difficulties and obstacles on his way to completing tasks. A decision must be taken that may impact the performance, the expectations but may favor time and budget. The Python community is very big in scale with many libraries, tools and whole information systems available to the users for free; meaning free of charge. For anyone to build and program everything from scratch it is time consuming and sometimes requires skills that need development, require sources (hardware, money etc.) and may not work as good as something that is already implemented. When we talked about NER or Named Entity Recognition step, the goal was to do text analysis that allows classifying the named entities under various predefined classes. Named-entity recognition is a subtask of information extraction that seeks to locate and classify named entities in text into predefined categories such as the names of persons, organizations, locations, expressions of time, quantities. It detects the sentence boundaries in each document based on capitalization rules. is a fundamental task in the field of information extraction, which involves determining entity boundaries from free text and classifying them into pre-defined categories, such as person (PER), location (LOC), and organization (ORG) (Zhao et al., 2021b). Use cases of NER, besides the one described here, can be customer support; like when companies receive a large amount of customer feedback and assists in classifying customer complaints that are automatically routed to the appropriate department or branch. Could also be applied in human resources, speeding up the hiring process by automatically filtering out resumes to find the appropriate candidates with the required skills. Another filed could publishing; a house, a media organization or a research institution, managing the online content generated by news outlets and publishing houses correctly; analyze entire articles to identify the prominent people, organizations, and places mentioned. To enable smooth content discovery, knowing the relevant tags for each article. An entity is an individual person, place, or thing in the world, while a mention is a phrase of text that refers to an entity using a proper name. There are three methods to integrate visual information into NER. First, to encode the whole image into a global feature vector to increase each word representation Secondly, to divide the feature map extracted from the whole image into multiple regions averagely. It uses a transformer framework based on the combination of self-modal and cross-modal attention to interact textual and visual information (Yu et al., 2020). Finally, to use noun phrases to detect the image bounding boxes and intermix words and visual objects by graph neural networks (GNN). The two most common algorithms that can be used for Named Entity Recognition are Maximum Entropy Markov Model (MEMM) and Common Random Fields (CRF). MEMM is a directed discriminative model from the Markov random field[9] and the way it works is based on a probabilistic model that takes into account the previous words in a sentence in order to predict the next word. The algorithm is trained on a corpus of text that has already been annotated with named entities and can be used to predict in new text. Benefits are the easy adaptation to languages and domains and its high speed. On the other side, CRFs are a type of latent semantic indexing model that can be used to automatically extract information from text. CRFs are essentially a way of combining the advantages of discriminative classification and graphical modeling, combining the ability to compactly model multivariate outputs y with the ability to leverage many input

[9] A set of random variables having a Markov property described by an undirected graph.

features x for prediction. Keeping that in mind and heading back to the thoughts of the programmer now it is the time to choose a tool that is already implemented to help annotation and do NER or build a new from scratch. Since that would be time consuming and it does not reach the depth I wanted to reach in this information system, I used Doccano tool. It is an open-source text annotation tool for humans. It provides annotation features for text classification, sequence labeling and sequence to sequence tasks. So, everyone can create labeled data for sentiment analysis, named entity recognition, text summarization and so on. Official documentation is easy to read and start working with the annotation tool. The basic idea is to upload a size x of the same document type (auction report in our case), define the labels and manually annotate one document by one. Sounds really time consuming but the later benefits are bigger than the effort. Think of an organization using it to extract information from thousands of documents. Doccano will read previously created text files by uploading to the platform and save all the results in a key-value pair dictionary to be used by our NLP framework. We will come to that later. The amount of auction reports to annotate or so-called training dataset for the predictive modeling problem, depends on the complexity of the problem and the learning algorithm. My suggestion is that a few hundred annotations that compile a training dataset are enough to give good accuracy on a test data set, meaning on something over 70%. Generally, the bigger the training dataset the better but if the goal is not to achieve 99% accuracy for example and as a company does not waste time from employees to annotate documents over months (because for a decent 10,000 training dataset it would surely take months), then a few hundred are enough to start with. I chose to stick with 272 annotations and split 80% of them for training data set and 20% for test dataset by the machine to test itself the accuracy of the model. Also, time was not in my favor but here we do not strike for highest accuracy rather than learning in the process.

SPACY FRAMEWORK FOR ADVANCED NATURAL LANGUAGE PROCESSING

Coming up to the 2022 technological era and all the support from community, we now have tools, end-to-end[10] applications, service and frameworks and libraries to utilize deep learning methods like never before. For every programmer simplicity of use is the key in combination with the most precise fine tuning of parameters to achieve great results. Such a framework that handles advanced natural language processes is SpaCy; designed specifically for production use and helps you build applications that process and "understand" large volumes of text. It can be used to build information extraction or natural language understanding systems, or to pre-process text for deep learning. Loaded with trained pipelines (like the one we will build with our annotations exported as json format) enables spaCy to predict linguistic annotations such as, whether a word is a verb or a noun. A trained pipeline can consist of multiple components that use a statistical model trained on labeled data and spaCy offers trained pipelines for a variety of languages, which can be installed as individual Python modules. Pipeline packages can differ in size, speed, memory usage, accuracy and the data they include. The chosen package always depends on the use case and the texts someone wants to work with. For a general-purpose use case, the small, default packages are always a good start that include components like binary weights, lexical entries, data files, word vectors, configuration option. The above mean accordingly that included someone will find part-of-speech tagger, dependency parser and named entity recognizer to predict those annotations in context; words and their context-independent attributes like the shape or spelling; lemmatization rules and lookup tables; multi-dimensional meaning representations of words that let someone determine how similar they are to each other; options, like the language and processing pipeline settings and model

[10] A full process from start to finish.

implementations to use, to put spaCy in the correct state when loading the pipeline. The full features of SpaCy are tokenization, part-of-speech (POS) tagging, dependency parsing, lemmatization, sentence boundary detection (SBD), named entity recognition, entity linking, similarity, text classification rule-based matching, training and serialization. To explain further as provided by the official documentation, tokenization means segmenting text into words, punctuations mark etc. Part-of-speech (POS) tagging is to assign word types to tokens, like verb or noun. Dependency parsing is to assign syntactic dependency labels, describing the relations between individual tokens, like subject or object. Lemmatization, assigning the base forms of words. For example, the lemma of "was" is "be", and the lemma of "rats" is "rat". Sentence Boundary Detection (SBD), to find and segment individual sentences. Named Entity Recognition (NER), to label named "real-world" objects, like persons, companies or locations. Entity Linking (EL), means disambiguating textual entities to unique identifiers in a knowledge base. Similarty, is comparing words, text spans and documents and how similar they are to each other. Text classification, to assign categories or labels to a whole document, or parts of a document. Rule-based Matching, by sequences of tokens based on their texts and linguistic annotations, like regular expressions. Training, to update and improve a statistical model's predictions. Serialization, to save objects to files or byte strings. The whole SpaCy architecture is simplified to a programmer offering 3 central data structures, the Language class, the Vocab and the Doc object. The Language class is used to process a text and turn it into a Doc object. It's typically stored as a variable called nlp. The Doc object owns the sequence of tokens and all their annotations. By centralizing strings, word vectors and lexical attributes in the Vocab, we avoid storing multiple copies of this data. The Doc object owns the data, and Span and Token are views that point into it. The Doc object is constructed by the Tokenizer, and then modified in place by the components of the pipeline. The Language object coordinates these components. It takes raw text and sends it through the pipeline, returning an annotated document. It also orchestrates training and serialization. In this particular framework, a certain sequence of actions taking place behind the scenes; from the input text file to the output Doc object and is a processing pipeline that has components that are called in specific order. They can contain a statistical model and trained weights, or only make rule-based modifications to the Doc. spaCy provides a range of built-in components for different language processing tasks and also allows adding custom components. Such SpaCy's components are powered by statistical models. Every "decision" these components make is a prediction based on the model's current weight values. The weight values are estimated based on examples the model has seen during training. We now can train a model, by first training data such the text and labels we would like the model to predict from auction reports. In other words, train our own NER component based on the type of document we want. Is not that freedom and power or what? Training is an iterative process in which the model's predictions are compared against the reference annotations in order to estimate the gradient of the loss. The gradient of the loss is then used to calculate the gradient of the weights through backpropagation[11]. The gradients indicate how the weight values should be changed so that the model's predictions become more like the reference labels over time. The purpose of the training as mentioned previously is the information we want to extract over never seen before auction reports. SpaCy team also suggests my later discovery, that to create a training and evaluation dataset you usually need at least a few hundred examples for both training and evaluation.

[11] A widely used algorithm for training feedforward neural networks.

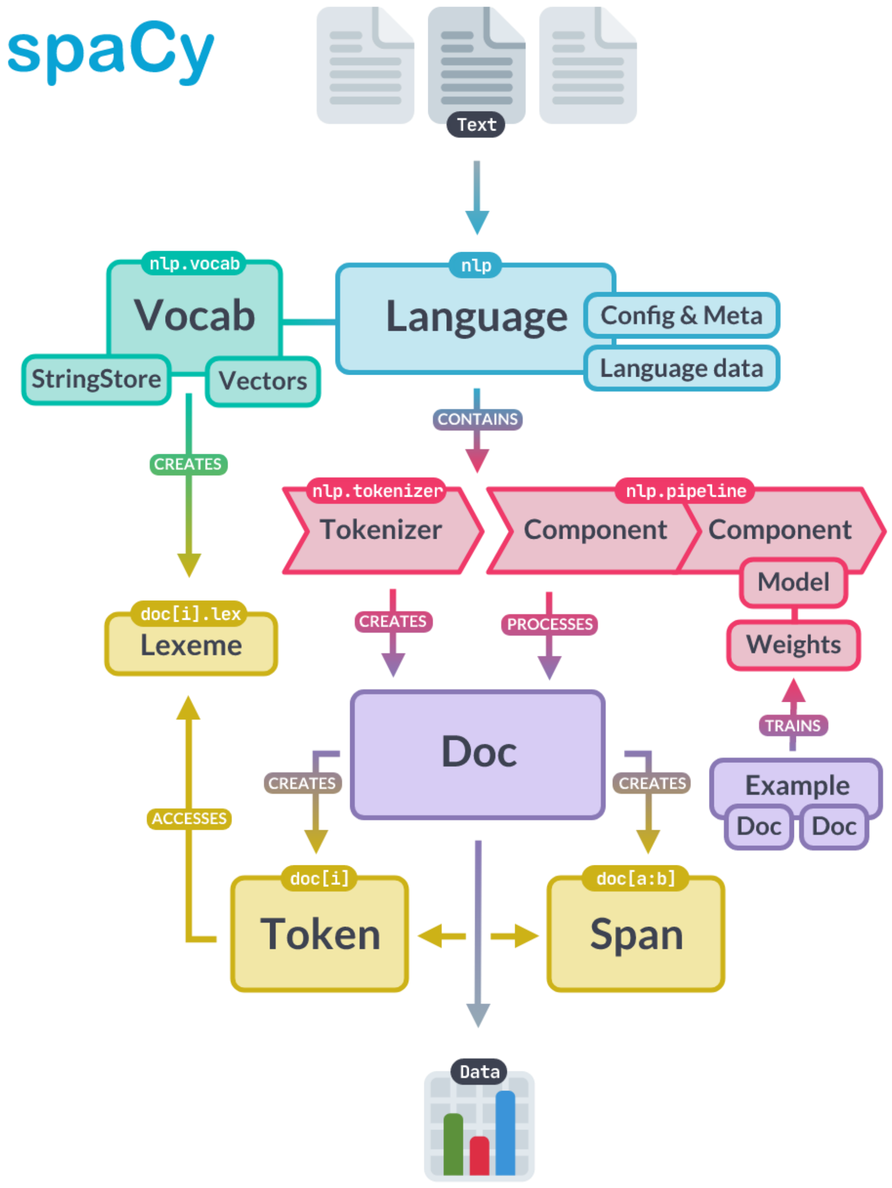


Figure 6: spaCy architecture

TRANSFORMERS

Rising to the surface, I would like to focus a little more on SpaCy's transformers; newly introduced deep neural networks that will face NLP knowledge acquisition bottleneck. A human brain can be described as a biological neural network, a web of neurons transmitting elaborate patterns of electrical signals. Receives input signals and, based on those inputs, sends an output signal. Computer scientists have been inspired by the human brain. Back in 1994 and written by Warren S. McCulloch, artificial neural networks were a success in performing relatively low-level functions like perception, segmentation and classification of patterns. Artificial neural networks were founded in the early 1940s. The artificial neurons of McCulloch and Pitts, which could be designed to reproduce an arbitrary logical function, introduced the concepts of activity, linear threshold logic, and inhibitory reset. Components that could be used in the design of logic circuits and memories. The birth of artificial intelligence (AI) was marked by Dartmouth Conference of 1956. Invented in 1957 by Frank Rosenblatt at the Cornell Aeronautical Laboratory, perceptron is the simplest neural network possible: a computational model of a single neuron. A perceptron consists of one or more inputs, a processor, and a single output. It receives inputs, weights the inputs with a number often between -1 and 1, sums the inputs and generates outputs. The basic idea of a neural network is its ability to learn and change its internal structure based on the information flowing through it. For this learning there are strategies. Supervised learning, which involves a teacher that is smarter than the network itself. Unsupervised learning, is when there is not a pattern in a dataset. Reinforcement learning; is observation and then decision making. Seems that a neural network is a supervised machine learning algorithm. Nowadays, neural networks have evolved so much that they are used for pattern recognition, time series prediction, signal processing, control, soft sensors, anomaly detection and many more. Applications can be found with an increasing rate in medical science, insurance companies, car manufacturers, safety and control zones like in airports and everywhere you can imagine. Neural networks operate with epochs which means training the neural network with all the training data for one cycle. In an epoch, we use all the data exactly once. A forward pass and a backward pass together are counted as one pass. An epoch is made up of one or more batches of the dataset to train the neural network. During the training phase, we aim to minimize the error rate as well as to make sure that the model generalizes well on new data. The problems we are facing are overfitting (high variance) when the model fits perfectly to the training examples but has limited capability generalization. On the other hand, if the model is said to be underfitting (high bias) if it didn't learn the data well enough. Transformers are a type of artificial neural network architecture that is used to solve the problem of transduction or transformation of input sequences into output sequences in deep learning applications (Dr J Rogel-Salazar, May 25, 2022). As written in their Transformers book, by open source developers at Hugging Face—including the creator of the Transformers library, a miracle is taking place, a revolution goes far beyond text generation; the whole realm of natural language processing. Wherever there's language, speech or text, there is an application for NLP. The transformer is a neural network architecture proposed in 2017 in a groundbreaking paper called "Attention Is All You Need", published by a team of Google researchers. Prior to transformers, recurrent architectures such as LSTMs were top notch in NLP.

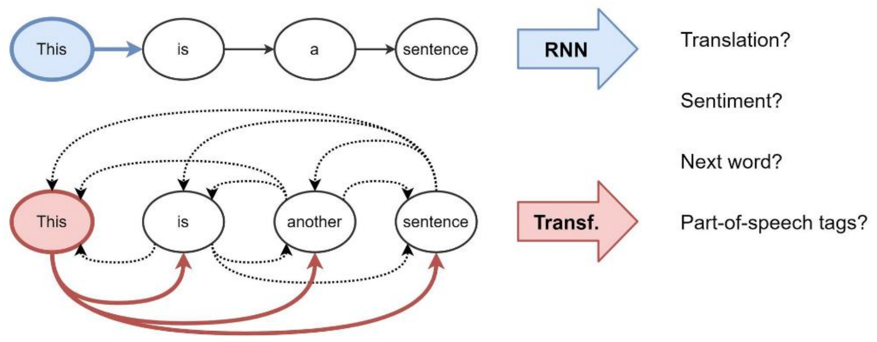


Figure 7: Transformer vs RNN

While recurrent architectures work with feedback loops that allow information to propagate from one step to another, while transformers are based on the encoder-decoder architecture that is widely used for tasks like machine translation, where a sequence of words is translated from one language to another. The two components convert an input sequence of tokens into a sequence of embedding vectors (Encoder) and use the encoder's hidden state to iteratively generate an output sequence of tokens, one token at a time (Decoder). The flow of the architecture is firstly for the text to be input, tokenized and converted to token embeddings using the techniques. Then, the encoder's and decoder's layers are analogous stacking convolutional layers in computer vision. The encoder's output is fed to each decoder layer, and the decoder then generates a prediction for the most probable next token in the sequence. The output of this step is then fed back into the decoder to generate the next token. Transformer architecture consists of over 50 architectures under the hood, some of which predict masked tokens in texts and determining if one text passage is likely to follow another, knowledge distillation for performance efficiency, several pretraining objectives for building multilingual models and more. For this reason, NER is often called a token classification task.

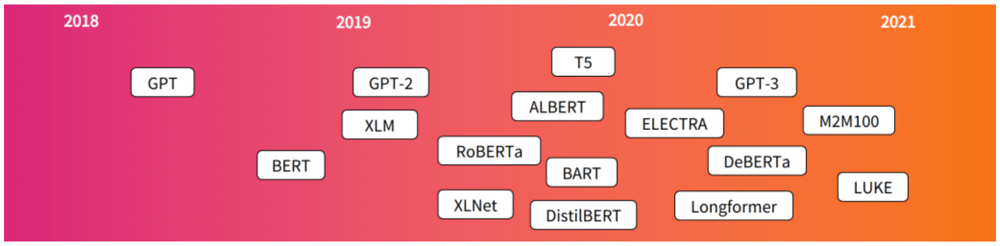


Figure 8: Timeline of popular Transformer model releases

For text classification, one of the above architectures, BERT[12], uses the special token to represent an entire sequence of text, something like each individual input token is fed into the same fully connected layer to output the entity of the token. It is a new technique for NLP pre-training called Bidirectional Encoder Representations from Transformers. To train a model using BERT and transformers in general, a GPU or cloud TPU is required. Training large transformer models is expensive and time-consuming, so if you are not successful the first or second time, projects might be canceled (Mostofa Patwary). Transformers are among the biggest neural-network models in terms of the number of parameters involved. And they are growing much faster than other models. Building accelerators for BERT contain matrix-vector engines for layers which perform matrix-matrix operations as MxN, a GEMM[13] as MxNxK, and a product of two variables as a*b. A GPU has a strong processing capability and high memory bandwidth that can achieve great data throughput because most of the chip's area is the execution unit. GPUs implement GEMMs by partitioning the output matrix into tiles, which are then assigned to thread blocks. Pre-training a

[12]Pretrained with the two objectives of predicting masked tokens in texts and determining if one text passage is likely to follow another. The former task is called masked language modeling (MLM) and the latter next sentence prediction (NSP).

[13] General Matrix Multiply (GEMM) is a common algorithm in linear algebra, machine learning, statistics, and many other domains.

BERT model is necessary only once, for all fine-tuning tasks, which may need hundreds of millions of training samples. To fine-tune BERT using spaCy, we need to provide training and dev data in the spaCy JSON format, to predict entities. This requires a powerful GPU with parallel processing and for this reason, in our case, the BERT model was fine-tuned using a lilliputian Nvidia GTX 1660 Super OC. Fine such pre-trained model, it increases performance with less labelled data, thus saving a lot of human and money effort to label task-specific data. In our case, we needed approximately 60 minutes.

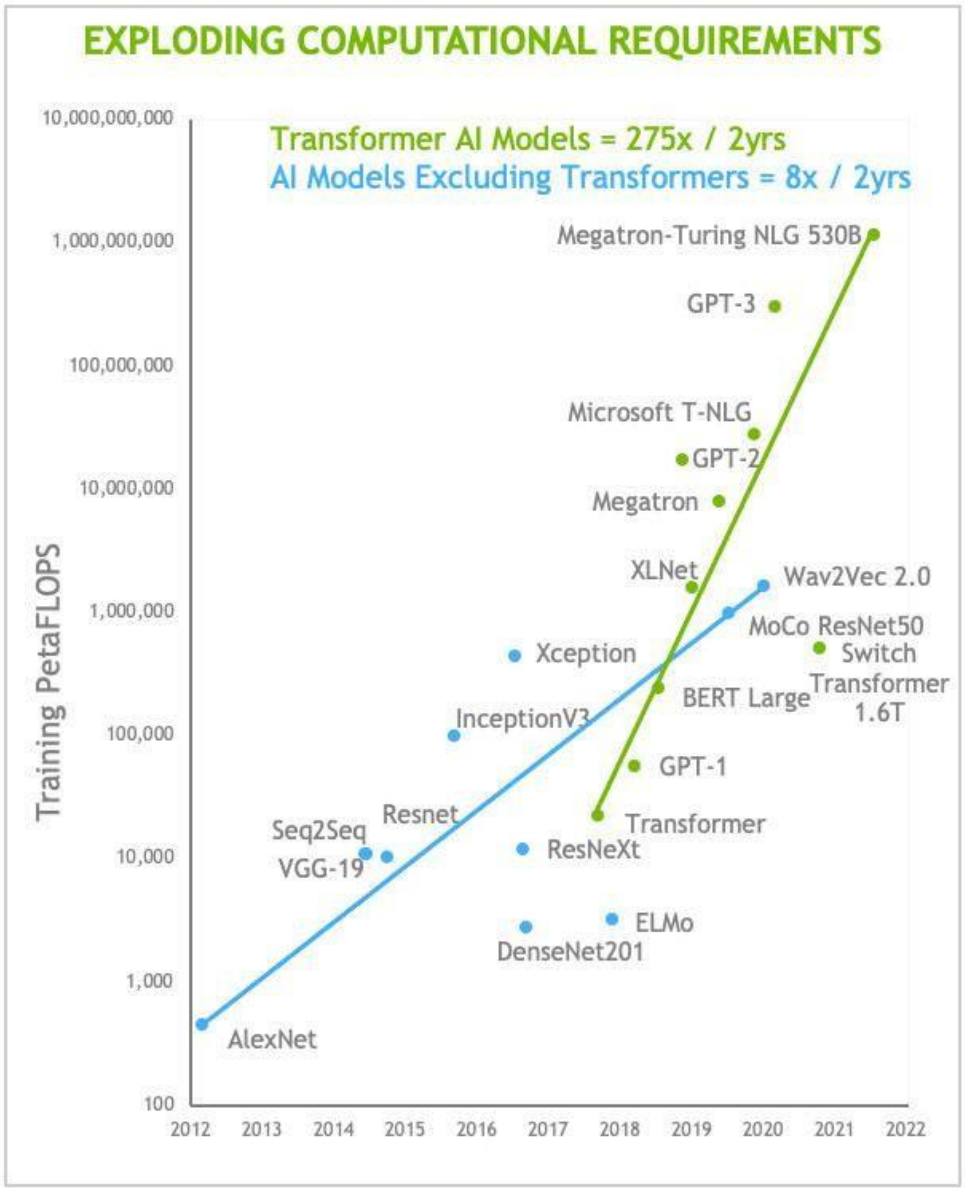


Figure 9: The computational needs of transformers are growing more rapidly than those of other forms of AI. Those are growing really fast, too, of course. NVIDIA

Transformers' architecture basically splits into a body and head. The last layer of the pretraining task moving to the downstream task, is called the model head and it is task specific (BertForMaskedLM[14]) while the rest of the model is called the body and includes the token embeddings and transformer layers that are task-agnostic (BertModel). A Document Attention Network (DAN) is an end-to-end encoder-decoder architecture that jointly recognizes both text and layout, from whole documents. Its architecture uses Fully Convolutional Network (FCN) encoder for feature extraction and a stack of transformer decoder layers for an autoregressive token-by-token prediction process plus a transformer decoder. Coming back to the 2D input document image, DAN extracts 2D feature maps from an input document image. 2D positional encoding is added to these features to keep the spatial information, before being flattened into a 1D sequence of features. This representation is computed only once and serves as input to the transformer decoder. The decoder follows an autoregressive prediction process at character-level given the previously predicted tokens and based on the computed features, it outputs the next token probabilities for each token of D. The final predicted token is the one with the highest probability. In OCR, the original transformer architecture, which is defined for 1D sequences, it is replaced the 1D positional encoding by 2D positional encoding, since the inputs are 2D images.

MULTILINGUAL NER

So far, we talked about auction reports and the contents of them written mainly in Greek language but also containing English corpora. There is a class of multilingual transformers which uses masked language modeling as a pretraining objective on huge corpora across many languages. This means that a model that is fine-tuned

[14] A Language Modeling(LM) head; LM head is a linear layer having input dimension of hidden state and output dimension of vocabulary size and maps to hidden state output of BERT model to a specific token in the vocabulary.

on one language can be applied to others without any further training, saving huge amount to time since that kind of task requires a lot of human effort from community, not even talking about employees doing it. This makes such models versatile when switching through languages. In our case we used SpaCy's sentence transformers (through the relevant configuration file in the code) the pretrained RoBERTa, which is also set by default when choosing two or more languages for the model. The RoBERTa model was proposed in RoBERTa: A Robustly Optimized BERT Pretraining Approach by Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. It is based on Google's BERT model released in 2018. It builds on BERT and modifies key hyperparameters, removing the next-sentence pretraining objective and training with much larger mini-batches and learning rates that replicates a study of BERT pretraining (Devlin et al., 2019) that carefully measures the impact of many key hyperparameters and training data size. It is presented a robustly optimized BERT pretraining approach. Findings showed that BERT was significantly undertrained and the above researchers proposed an improved recipe for training BERT models, which they call RoBERTa, and can match or exceed the performance of all of the post-BERT methods. The way BERT works is that it takes as input a concatenation of two sequences of tokens, which are then presented as a single input sequence to BERT with special tokens delimiting them and constrained by a parameter that controls the maximum sequence length during training. Later the model is first pretrained on a large unlabeled text corpus and subsequently finetuned using end-task labeled data. MLM objective is implied as a cross-entropy loss on predicting the masked tokens. Masked tokens are a random sample of the tokens in the input sequence that is selected and replaced with the special token. BERT selects a specific amount of tokens. Next sentence prediction also is part of the role, predicting whether two segments follow each other in the original text. Positive examples are created by taking consecutive sentences from the text corpus. Negative examples are created by pairing segments from different documents. Positive and negative examples are sampled with equal probability. The Next Sentence Prediction (NSP) objective was designed to improve performance on downstream tasks, such as Natural Language Inference (Bowman et al., 2015), which require reasoning about the relationships between pairs of sentences. BERT is optimized with Adam and is trained on a combination of BOOKCORPUS (Zhu et al., 2015) plus English WIKIPEDIA, which totals 16GB of uncompressed text. Finally, Byte-Pair Encoding (BPE), On the other hand, RoBERTa is trained with dynamic masking FULL-SENTENCES without NSP loss, large mini-batches and a larger byte-level BPE handles the large vocabularies common in natural language corpora. BPE vocabulary sizes typically range from 10K-100K sub word units. Their approach suggests that performance can be substantially improved by training the model longer, with bigger batches over more data; removing the next sentence prediction objective; training on longer sequences; and dynamically changing the masking pattern applied to the training data.

Chapter 4

PERFORMANCE MEASURES AND RESULTS

When training a model in SpaCy framework, an output in console reports results for precision, recall, and F1-score. Focusing on F1 score we a first picture of the NER model performance that we are training. To confirm that our model works as expected, we test one auction report and check the extracted labels to see if meet our expectations. However, F1 score is nothing more than a number and as I mentioned we set the threshold of the accuracy and what we want the machine to do for us. Though, there can be some error analysis to investigate the errors of our model. A thorough error analysis of your model is one of the most important aspects when training and debugging transformers. If bad performance is spotted, error analysis might give insights and reveal things we missed; also, for good performance error analysis is still powerful to stress the weaknesses of the model. For that error analysis we can look at the validation dataset with the highest loss, as provided by SpaCy. There is a supported function called scorer.scores which returns back multiple scores. This step helps establish how confident we can be in the model's ability to get all the tags right. A table is printed in the console revealing tags (labels we set before) their precision score[15], recall score[16], F-score [17]and sample size. Below is the output of the training pipeline.

[15] Precision shows how many entities were correctly labelled for all the tagged entities.

[16] Recall shows based on the actual tags, how many did we correctly.

[17] A balancer of the two other scores.

E	#	LOSS TRANS...	LOSS NER	ENTS_F	ENTS_P	ENTS_R	SCORE
---	------	-------------	--------	------	------	------	------
0	0	719.13	832.39	0.00	0.00	0.00	0.00
1	200	445642.91	55529.02	15.60	9.06	55.96	0.26
3	400	50080.23	28336.46	20.31	11.79	73.33	0.35
4	600	4400.43	26536.53	22.77	13.18	83.43	0.39
6	800	914.31	24959.15	26.37	15.79	80.00	0.40
7	1000	1995.57	24631.79	23.31	13.60	81.41	0.39
9	1200	977.94	24075.59	22.25	12.92	80.00	0.38
10	1400	1772.00	23196.50	23.04	13.37	83.23	0.39
12	1600	929.98	23125.43	24.22	14.07	87.07	0.41
14	1800	370.88	22178.07	23.02	13.34	83.84	0.39
15	2000	625.41	22187.31	23.46	13.60	85.25	0.40
17	2200	730.95	21622.84	83.28	84.58	82.02	0.83
18	2400	333.34	20494.17	75.02	68.15	83.43	0.76
20	2600	234.45	19183.59	80.56	74.66	87.47	0.81
21	2800	28503.55	18414.26	81.65	77.84	85.86	0.82
23	3000	146.81	16720.72	84.92	83.99	85.86	0.85
24	3200	358.25	15184.76	83.69	81.45	86.06	0.84
26	3400	6449.03	13233.76	86.30	88.03	84.65	0.86
28	3600	139.49	10882.02	83.37	79.13	88.08	0.84
29	3800	639.44	8740.30	85.07	82.26	88.08	0.85
31	4000	129.49	6143.54	86.78	86.69	86.87	0.87
32	4200	106.80	4189.76	86.32	86.59	86.06	0.86
34	4400	167.57	2595.30	84.12	81.71	86.67	0.84
35	4600	192.96	1502.75	82.79	79.41	86.46	0.83

Figure 10: Table containing spaCy's NER model training results

Column "E" refers to epochs. In terms of artificial neural networks, an epoch refers to one cycle through the full training dataset. Column "#" refers to iteration. In iterative decoding, training examples are broken down into a sequence of intermediate steps that the transformer learns iteratively. Column "LOSS TRANS..." and "LOSS NER" refers to the Loss function; a method of evaluating how well the algorithm is modeling the dataset. The function we want to minimize or maximize is called the objective function or criterion. When we are minimizing it, we may also call it the cost function, loss function, or error function. Columns ”ENTS_F”, ”ENTS_P”, ”ENTS_R” are the F-score, precision and recall for the NER task respectively. Precision and recall are performance metrics that apply to data retrieved from a collection, corpus or sample space. A measure that combines precision and recall is the harmonic mean of precision and recall, the traditional F-measure or balanced F-score. The “SCORE” column shows the overall score of the pipeline; the higher and closer to 1.00, the better the model output.

To correctly evaluate the performance and accuracy of the model, semantic analysis must be done by us. We created unit tests, where we tested 20 pdfs and output the results. For every pdf, we compared only the found entities by our model and compared them to the actual document. Let us say A= number of wrong entities found by model per pdf, B= number of entities found by model per pdf and N=total number of pdfs tested. The accuracy x then is calculated as $x = \frac{\sum_{k=1}^{N} (1 - \frac{A_k}{B_k})}{N}$. In our case, accuracy was 88.35%, which is very close to spaCy’s score. Even more interesting is the fact that our manually calculated accuracy comes very close to spaCy’s recall score (ENTS_R). The recall is calculated as the ratio between the number of Positive samples correctly classified as Positive to the total number of Positive samples. The recall measures the model's ability to detect Positive samples. The higher the recall, the more positive samples detected. This proves the ability of the model to correctly detect positive samples like we do, humans.

FUTURE WORK AND IMPROVEMENTS

Huge space there is for improvements and I mean it. Think about all the variables that can be improved for every step followed to create a model for natural language processing. Starting with the most important factor, defining the usage of the machine. Is it a scientific purpose or market purpose? A market-oriented machine for NLP can be as complex as the requirements define. What is expected to be achieved and how the results are going to be used. Next, is the collection of quality and good variety of document samples. We could spend more time carefully collecting a good variety of the documents we want to train a model for. Since we are working on documents with unstructured data it is very important to cover all possible cases of where to find the desired labels-information we want to extract. Tagged entities that have similar and various words close to them can be more easily trained. Do not always expect good training by looking for information around the same place on every document. This also wraps up the importance of annotating the training dataset correctly without any mistakes that could "confuse" the model later. Speaking about time, a bigger sample of annotated dataset is desirable since the documents are unstructured. A few hundred annotated documents could easily come close to a few thousand. This is necessary as we are chasing the importance of information and we do not want to overfit or underfit the training dataset. Test a few hundred documents, evaluate scores and if necessary, train more annotated documents, until desired accuracy is hit. A good notice here is that always choose GPU usage to train with transformers neural network for better results. SpaCy gives the freedom with lot of options within its config file to increase training pipeline's scores. Computer hardware will come in handy here to save more time with back-and-forth while testing and training. It can also allow better optimization while setting parameters in the config file. If we would like to take this step of fine tuning to a whole other level, a crazy but very dangerous idea (dangerous by the means of cost and running out of time) would be to implement our transformer neural network from scratch. However, a very excellent and good solution instead would be to auto retrain the model by itself. If deployed, we can me it receive inputs every so often by users. What I mean is that by giving the ability to a user to correct found information in a document by our model, using an annotation tool, we could save the updated correct annotation by user and use them for retraining the model.

BIBLIOGRAPHY

Adami C. (2016), “What is information? Phil. Trans. R. Soc. A 374:20150230”.

Marc Brysbaert,
How many words do we read per minute? A review and meta-analysis of reading rate,
Journal of Memory and Language,
Volume 109,
2019,
104047,
ISSN 0749-596X.

https://opencv.org/about/.

https://guides.lib.umich.edu.

https://docs.opencv.org.

https://pymupdf.readthedocs.io/en/latest/tutorial.html.

http://www.leptonica.org.

http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

https://www.expressanalytics.com/blog/what-is-named-entity-recognition-ner-benefits-use-cases-algorithms/.

https://homepages.inf.ed.ac.uk/csutton/publications/crftut-fnt.pdf.

https://thinc.ai/docs/backprop101.

https://spacy.io/usage/spacy-101.

Asahi Ushio and Jose Camacho-Collados, T-NER: An All-Round Python Library
for Transformer-based Named Entity Recognition, School of Computer Science and Informatics
Cardiff University, United Kingdom, 9 Sep 2022.

Lewis Tunstall, Leandro von Werra, Thomas Wolf, Natural Language Processing with Transformers. Building Language Applications with Hugging Face, O'Reilly Media, 2022.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi,
Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov, RoBERTa: A Robustly Optimized BERT Pretraining Approach, Paul G. Allen School of Computer Science & Engineering,
University of Washington, Seattle, WA, 26 Jul 2019.

Anoop R. Katti, Christian Reisswig, Cordula Guder, Sebastian Brarda, Steffen Bickel,
Johannes Höhne, and Jean Baptiste Faddoul. “Chargrid: Towards Understanding 2D
Documents”. In: Proceedings of the 2018 Conference on Empirical Methods in Natural
Language Processing. 2018, pp. 4459–4469.