



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΙΓΑΙΟΥ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ
ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Ανάπτυξη εξειδικεύμενου GPT chatbot για την ελληνική νομοθεσία

Διπλωματική εργασία
Σπύρου Νικόλαος

Επιβλέπων: Αλεξόπουλος Χαράλαμπος

Μέλη εξεταστικής επιτροπής: Διαμαντοπούλου Βασιλική, Λουκής Ευριπίδης

Σάμος, Οκτώβριος 2024

Περίληψη

Η παρούσα διπλωματική εργασία ασχολείται με την ανάπτυξη ενός εξειδικευμένου chatbot βασισμένου στο μοντέλο GPT για την ελληνική νομοθεσία. Η εργασία επικεντρώνεται στην υλοποίηση ενός συστήματος που μπορεί να κατανοεί και να απαντά σε ερωτήματα σχετικά με νομικά κείμενα και διατάξεις, παρέχοντας άμεση πρόσβαση σε νομικές πληροφορίες.

Για την επίτευξη αυτού του στόχου, χρησιμοποιήθηκαν σύγχρονες τεχνικές επεξεργασίας φυσικής γλώσσας (NLP), με έμφαση στη δημιουργία διανυσματικών αναπαραστάσεων των νομικών κειμένων και στην αποθήκευσή τους σε εξειδικευμένες βάσεις δεδομένων (vectorstores). Το σύστημα αναπτύχθηκε χρησιμοποιώντας τα OpenAI embeddings και το Chroma vectorstore μέσω του Langchain, ενώ η διεπαφή χρήστη υλοποιήθηκε με το Gradio ChatInterface για την παροχή φιλικής και αποτελεσματικής αλληλεπίδρασης.

Τα αποτελέσματα της έρευνας έδειξαν ότι το chatbot μπορεί να παρέχει ακριβείς και χρήσιμες απαντήσεις σε ερωτήματα σχετικά με την ελληνική νομοθεσία, αν και υπήρξαν προκλήσεις στην επεξεργασία της ελληνικής γλώσσας και στην αναγνώριση πολύπλοκων νομικών όρων. Η εργασία επισημαίνει τις δυνατότητες εξέλιξης του συστήματος μέσω της ενσωμάτωσης επιπλέον νομικών δεδομένων και της χρήσης πιο προηγμένων αλγορίθμων NLP.

Συνοψίζοντας, η παρούσα διπλωματική εργασία καταδεικνύει τη δυνατότητα χρήσης τεχνητής νοημοσύνης για τη βελτίωση της πρόσβασης σε νομικές πληροφορίες και θέτει τις βάσεις για μελλοντική έρευνα και ανάπτυξη στον τομέα αυτό.

Λέξεις – Κλειδιά: Chatbot, GPT, ελληνική νομοθεσία, Natural Language Processing (NLP), Μηχανική γνώση, OpenAI, Legal Information Retrieval, Διανυσματικές αναπαραστάσεις, Vectorstore, Gradio, Langchain, Artificial Intelligence (AI), Conversational AI, Retrieval Augmented Generation (RAG)

Abstract

Custom Gpt Chatbot for Greek legislation

This thesis focuses on the development of a specialized chatbot based on the GPT model for Greek legislation. The study aims to implement a system capable of understanding and answering questions related to legal texts and provisions, providing immediate access to legal information.

To achieve this objective, modern techniques in natural language processing (NLP) and machine learning were employed, emphasizing the creation of vector representations of legal texts and their storage in specialized databases (vectorstores). The system was developed using OpenAI embeddings and the Chroma vectorstore through Langchain, while the user interface was implemented with Gradio to provide a user-friendly and efficient interaction.

The results of the research showed that the chatbot can provide accurate and useful answers to questions regarding Greek legislation, although there were challenges in processing the Greek language and recognizing complex legal terms. The thesis highlights the potential for further development of the system by integrating additional legal data and employing more advanced NLP algorithms.

In summary, this thesis demonstrates the feasibility of using artificial intelligence to improve access to legal information and lays the groundwork for future research and development in this field.

Keywords: Chatbot, GPT, Greek Legislation, Natural Language Processing (NLP), Machine Learning, OpenAI, Legal Information Retrieval, Vector Representations, Vectorstore, Gradi, Langchain, Artificial Intelligence (AI), Conversational AI, Retrieval Augmented Generation (RAG)

Contents

1	Εισαγωγή	7
1.1	Το πλαίσιο της Διπλωματικής Εργασίας	7
1.2	Σκοπός της Διπλωματικής Εργασίας	7
1.3	Αντικείμενο της Διπλωματικής Εργασίας.....	7
1.4	Δομή	8
2	Ανάλυση Τρέχοντος Επιπέδου	8
2.1	Τεχνητή Νοημοσύνη και Γλωσσικά Μοντέλα	8
2.2	Τρέχουσα Κατάσταση στην Ελλάδα	9
2.3	Τρέχουσα Κατάσταση στο Εξωτερικό	12
2.4	Συμπεράσματα	17
3	Μεθοδολογία	18
4	Αρχιτεκτονική Συστήματος.....	19
4.1	Απαιτήσεις Συστήματος	19
4.2	Διαγράμματα	21
5	Εργαλεία Ανάπτυξης συστήματος.....	22
5.1	ChatGPT	22
5.2	VsCodium.....	23
5.3	Mozilla Firefox	24
5.4	Pip.....	24
5.5	OpenAI API.....	24
5.6	LangChain	25
5.7	Gradio	26
6	Το Σύστημα.....	27
6.1	Εντοπισμός σχετικών και βέλτιστων ερεθισμάτων (prompts).....	27
6.1.1	ChatGPT 4.0	28
6.1.2	ChatGPT 3.5	30
6.2	Εύρεση Νομικών Κειμένων και Ορολογίας.....	31
6.3	Επιλογή Μοντέλου Γλώσσας.....	33
6.3.1	GPT-3.5	34
6.3.2	GPT-4	35
6.3.3	Παράμετροι για την επιλογή του μοντέλου	35
6.4	Αρχική Ανάπτυξη του Bot.....	37
6.4.1	Εγκατάσταση Απαραίτητων Βιβλιοθηκών και Plugin.....	37
6.4.2	Ενσωμάτωση του OpenAI API & του Κλειδιού του API:.....	38
6.4.3	Πρώτο Στάδιο Υλοποίησης για Αλληλεπίδραση Χρήστη – Bot	41

6.4.4	Ενσωμάτωση Βέλτιστων Ερεθισμάτων (Prompts)	43
6.5	Ανάπτυξη Διεπαφής Χρήστη	46
6.5.1	Υλοποίηση συνάρτησης παραγωγής απάντησης.....	46
6.5.2	Ανάπτυξη γραφικής διεπαφής	47
6.6	Χρήση Langchain και τελική υλοποίηση του Chatbot.....	51
6.6.1	Προετοιμασία αρχείων.....	53
6.6.2	Μέθοδοι διαίρεσης του εξαγόμενου κειμένου	54
6.6.3	Διανυσματικές αναπαραστάσεις και αποθήκευση τους	57
6.6.4	Αλυσίδα ερώτησης-κειμένων-OpenAI	60
6.6.5	Προσαρμογή της διεπαφής χρήστη	61
7	Αξιολόγηση Συστήματος	66
7.1	Παρόμοιες ερωτήσεις στο ChatGPT	73
8	Συμπεράσματα και Προοπτικές	74
8.1	Συμπεράσματα επί των αποτελεσμάτων	74
8.2	Συμπεράσματα επί της μεθοδολογίας.....	75
8.3	Συμπεράσματα επί των εργαλείων	75
8.4	Προοπτικές/ Επόμενα Βήματα	75
9	Αναφορές	76
10	Παράρτημα.....	77
10.1	Κώδικας και αποτελέσματα	77

1 Εισαγωγή

1.1 Το πλαίσιο της Διπλωματικής Εργασίας

Η παρούσα διπλωματική εργασία εντάσσεται στο πλαίσιο της έρευνας σχετικά με την ενσωμάτωση τεχνολογιών τεχνητής νοημοσύνης στην ελληνική νομοθεσία. Στο συγκεκριμένο υποκεφάλαιο θα τεκμηριωθεί η σπουδαιότητα της έρευνας, η σημασία του θέματος και το ερευνητικό πρόβλημα, οδηγώντας τον αναγνώστη στην κατανόηση της ανάγκης και του σκοπού της παρούσας μελέτης.

Η παρούσα έρευνα εξετάζει τη δυνατότητα χρήσης των τεχνολογιών τεχνητής νοημοσύνης στην ελληνική νομοθεσία. Με την αύξηση της πολυπλοκότητας των νομικών κειμένων και την ανάγκη για γρήγορη και ακριβή πρόσβαση σε νομικές πληροφορίες, η εφαρμογή των chatbots μπορεί να συμβάλει κομβικά στην επίλυση αυτού του προβλήματος. Η ικανότητα των chatbots να επεξεργάζονται και να απαντούν σε ερωτήματα σχετικά με τη νομοθεσία θα μπορούσε να βελτιώσει σημαντικά την απόδοση και την ακρίβεια στη νομική πρακτική.

Το θέμα της ανάπτυξης ενός εξειδικευμένου GPT chatbot για την ελληνική νομοθεσία συνδυάζει την τεχνητή νοημοσύνη με τη νομική επιστήμη. Η εξειδίκευση του bot στην ελληνική γλώσσα και νομοθεσία αποτελεί μια πρωτοποριακή προσέγγιση που μπορεί να έχει ευρείες εφαρμογές στη νομική πρακτική και την εκπαίδευση. Η εφαρμογή τέτοιων τεχνολογιών μπορεί να οδηγήσει σε σημαντικές βελτιώσεις στη διαχείριση νομικών πληροφοριών και στην υποστήριξη νομικών επαγγελματιών.

Το κύριο ερευνητικό πρόβλημα που αντιμετωπίζει αυτή η έρευνα είναι η ανάπτυξη ενός GPT chatbot που θα είναι σε θέση να κατανοεί και να ανταποκρίνεται στην ελληνική νομοθεσία. Οι επιμέρους στόχοι περιλαμβάνουν:

Ανάλυση των προκλήσεων της ελληνικής γλώσσας στην επεξεργασία φυσικής γλώσσας: Η ελληνική γλώσσα παρουσιάζει μοναδικές προκλήσεις λόγω της πολυπλοκότητάς της και της πλούσιας μορφολογίας της.

Επιλογή και προετοιμασία των κατάλληλων δεδομένων για την χρήση τους στο σύστημα: Η συλλογή και η προετοιμασία αξιόπιστων και ποικίλων δεδομένων είναι κρίσιμη για την επιτυχή εκπαίδευση του μοντέλου.

Ανάπτυξη του συστήματος και πειραματισμός: Χρήση ορισμένων σημαντικών εργαλείων, τα οποία θα αναφερθούν παρακάτω, που συμβάλουν στον πειραματισμό, στην ανάπτυξη και στην υλοποίηση του συστήματος.

Εξέταση αποτελεσμάτων: Παρουσίαση της λειτουργικότητας του συστήματος και η ανταπόκριση του ή μη στις ερωτήσεις που τίθενται.

Αξιολόγηση αποτελεσμάτων: Χρήση μοντέλου βαθμολογίας για την ακριβέστερη αξιολόγηση των αποτελεσμάτων του συστήματος που εξετάζεται στην παρούσα διπλωματική και εν συνεχεία σύγκριση του με το ChatGPT.

Μέσω της παρούσας έρευνας, αναδεικνύεται η σπουδαιότητα και η σημασία του θέματος μέσω της αναφοράς άλλων προσεγγίσεων και του επιστημονικού πεδίου στο οποίο ανήκει. Τα αποτελέσματα αυτής της έρευνας μπορούν να συμβάλλουν σημαντικά στην κατανόηση και την εφαρμογή της τεχνητής νοημοσύνης στην ελληνική νομοθεσία, προωθώντας την καινοτομία και την αποτελεσματικότητα στον νομικό τομέα.

1.2 Σκοπός της Διπλωματικής Εργασίας

Ο σκοπός της παρούσας διπλωματικής έρευνας είναι η ανάπτυξη ενός εξειδικευμένου GPT chatbot που θα είναι σε θέση να αφοηγκραστεί και να ανταποκρίνεται στην ελληνική νομοθεσία. Ο στόχος αυτός έχει ως σκοπό τη βελτίωση της πρόσβασης και της κατανόησης των νομικών κειμένων από επαγγελματίες και από το ευρύ κοινό. Η επιτυχής υλοποίηση αυτού του στόχου θα συμβάλει στην προώθηση της τεχνολογίας τεχνητής νοημοσύνης στον τομέα της ελληνικής νομοθεσίας και θα προσφέρει νέες δυνατότητες για τη νομική κοινότητα και την κοινωνία γενικότερα.

1.3 Αντικείμενο της Διπλωματικής Εργασίας

Για την επίτευξη του σκοπού της διπλωματικής εργασίας, το αντικείμενο της έρευνας περιλαμβάνει τις ακόλουθες πτυχές:

- **Διερεύνηση των GPT μοντέλων:** Αξιολόγηση των ικανοτήτων των μοντέλων GPT 3.5 & 4.0 στην κατανόηση νομικών ζητημάτων στην ελληνική γλώσσα και παραγωγής απαντήσεων από την υφισταμένη γνώση τους.
- **Επιλογή και προετοιμασία δεδομένων:** Εύρεση των κατάλληλων δεδομένων για την χρήση του chatbot, περιλαμβάνοντας νομικά κείμενα και νομολογία.
- **Ανάπτυξη του GPT chatbot:** Σχεδιασμός και υλοποίηση του μοντέλου, χρησιμοποιώντας τεχνικές επεξεργασίας φυσικής γλώσσας.
- **Ενίσχυση του chatbot με τεχνολογίες RAG:** Εφαρμογή τεχνικών retrieval, embeddings για την προσθήκη επιπλέον υλικού στην γνώση του μοντέλου στην ελληνική γλώσσα και νομοθεσία.
- **Πειραματισμός και αξιολόγηση αποτελεσμάτων:** Διεξαγωγή αξιολόγησης του chatbot με χρήση διαφορετικών μετρικών και παράλληλη δοκιμή με διάφορες ερωτήσεις.

1.4 Δομή

Η συγκεκριμένη διπλωματική εργασία παρουσιάζει στην αφετηρία της το σκοπό και το αντικείμενο που πρόκειται να πραγματευτεί καθώς επίσης και τους βασικούς πυλώνες στους οποίους θα συνταχθεί. Πραγματοποιείται μια πρώτη προσέγγιση με το αναλυόμενο αντικείμενο και παρέχονται οι αναγκαίες πληροφορίες οι οποίες θα συμβάλλουν στην όσο το δυνατόν καλύτερη κατανόηση του περιεχομένου.

Στο δεύτερο κεφάλαιο δίνονται ορισμοί και επεξηγούνται σημαντικοί για την εξέλιξη της έρευνας όροι οι οποίοι θα χρησιμοποιηθούν διαρκώς κατά την διάρκεια της διπλωματικής με στόχο ο αναγνώστης να τους αφογκραστεί και να τους κατανοήσει. Επιπροσθέτως στο ίδιο κεφάλαιο εκτίθεται μια έρευνα με τίτλο «Pythia Project» η οποία διαδραματίζει κομβικό ρόλο στη σύσταση της παρούσας εργασίας καθώς μέσα από αυτήν μπορούμε να «αλιεύσουμε» καίρια συμπεράσματα για την τρέχουσα κατάσταση τόσο στην Ελλάδα όσο και στο εξωτερικό.

Στο κεφάλαιο που ακολουθεί, αναδεικνύεται η μεθοδολογία στην οποία στηρίχθηκε το θέμα της διπλωματικής και εν συνεχεία παρουσιάζονται τα βήματα που ακολουθήθηκαν μέχρι την πραγμάτωση της .

Στο τέταρτο κεφάλαιο θα καταγραφεί η αρχιτεκτονική του συστήματος καθώς και οι απαιτήσεις που δημιουργούνται για την υλοποίησή του ενώ παράλληλα θα παρουσιαστούν και ορισμένα διαγράμματα που συνεπικουρούν στην καλύτερη κατανόηση του κεφαλαίου. Στο ίδιο κεφάλαιο εκθέτονται τα αποτελέσματα της Μελέτης μέσα από συγκριτική ανάλυση και κριτική των αποτελεσμάτων.

Το πέμπτο κεφάλαιο της εργασίας συντελείται από τα θεμελιώδη εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος ενώ ταυτόχρονα πραγματοποιείται και μια ενδελεχής ανάλυση τους.

Η επόμενη θεματική ενότητα είναι ζωτικής σημασίας για την εξέλιξη της διπλωματικής καθώς σε αυτό παρουσιάζεται το σύστημα το οποίο πραγματεύεται η παρούσα έρευνα. Σε αυτό το κεφάλαιο πραγματοποιείται ο εντοπισμός των σχετικών και βέλτιστων ερεθισμάτων (prompts) καθώς και η εύρεση των νομικών κειμένων και της απαραίτητης ορολογίας που πλαισιώνουν το σύστημα με στόχο την σωστή πραγματοποίηση του σκοπού του. Στη συνέχεια του κεφαλαίου ο αναγνώστης έχει τη δυνατότητα να παρατηρήσει ποια επιλογή μοντέλου γλώσσας έγινε από τον γράφοντα καθώς και την αρχική ανάπτυξη του bot και ταυτόχρονα την ανάπτυξη διεκπαφής του χρήστη. Το κεφάλαιο ολοκληρώνεται με την παρουσίαση του υλοποιημένου συστήματος με χρήση στιγμιότυπων (printscreens) των κυριότερων οθονών του.

Στο έβδομο κεφάλαιο η διπλωματική παρέχει τα συμπεράσματα και τις προοπτικές που προέκυψαν μέσα από την σύνθεσή της. Συγκεκριμένα παρουσιάζονται συμπεράσματα τόσο επί των αποτελεσμάτων, επί της μεθοδολογίας όσο και επί των εργαλείων που συνέδραμαν στην υλοποίηση της διπλωματικής εργασίας. Το κεφάλαιο κλείνει με την παρουσίαση των προοπτικών και των επόμενων βημάτων που μπορούν να πραγματοποιηθούν μελλοντικά για το συγκεκριμένο θέμα.

2 Ανάλυση Τρέχοντος Επιπέδου

2.1 Τεχνητή Νοημοσύνη και Γλωσσικά Μοντέλα

Η τεχνητή νοημοσύνη (TN) έχει σημειώσει σημαντική πρόοδο τα τελευταία χρόνια, με τα γλωσσικά μοντέλα να βρίσκονται στην πρωτοπορία της έρευνας. Τα γλωσσικά μοντέλα όπως το GPT έχουν τη

δυνατότητα να κατανοούν και να παράγουν φυσική γλώσσα, επιτρέποντας την ανάπτυξη εφαρμογών που μπορούν να απαντούν σε ερωτήματα, να δημιουργούν κείμενα και να αλληλεπιδρούν με τους χρήστες σε πραγματικό χρόνο.¹ Το GPT, συγκεκριμένα, αποτελεί ένα από τα πιο προηγμένα μοντέλα στον τομέα της TN, βασισμένο σε μια αρχιτεκτονική που επιτρέπει την κατανόηση σύνθετων γλωσσικών δομών και την παραγωγή απαντήσεων υψηλής ποιότητας.²

Το μοντέλο RAG(retrieval augmented generation) συνδυάζει τις δυνατότητες του GPT με τεχνικές ανάκτησης πληροφοριών (retrieval), ενισχύοντας την ακρίβεια και την σχετικότητα των παραγόμενων απαντήσεων. Αντί να βασίζεται αποκλειστικά στην γενική γνώση που έχει αποκτήσει το μοντέλο κατά την εκπαίδευσή του, το RAG ενσωματώνει μηχανισμούς που επιτρέπουν την ανάκτηση συγκεκριμένων πληροφοριών από εξωτερικές βάσεις δεδομένων ή κείμενα κατά την παραγωγή των απαντήσεων. Αυτό το χαρακτηριστικό το καθιστά ιδανικό για την ανάπτυξη εξειδικευμένων chatbots, όπως αυτό που θα εξεταστεί στην παρούσα εργασία.³

Η χρήση της TN στον τομέα της νομικής πρακτικής είναι ένα ταχέως αναπτυσσόμενο πεδίο. Από την αυτοματοποίηση της ανάλυσης νομικών κειμένων έως την ανάπτυξη εργαλείων για τη διευκόλυνση της νομικής έρευνας, οι εφαρμογές της TN έχουν αρχίσει να αλλάζουν τον τρόπο με τον οποίο οι νομικοί επαγγελματίες προσεγγίζουν τη δουλειά τους.⁴ Τα chatbots που βασίζονται σε γλωσσικά μοντέλα, όπως το GPT, μπορούν να αναλάβουν τον ρόλο του εικονικού νομικού συμβούλου, παρέχοντας πληροφορίες και καθοδήγηση σε πελάτες και επαγγελματίες με ακρίβεια και ταχύτητα.

2.2 Τρέχουσα Κατάσταση στην Ελλάδα

Η ενότητα αυτή ανασκοπεί την τρέχουσα κατάσταση των chatbots στην Ελλάδα, εξετάζοντας την υιοθέτηση και την αποτελεσματικότητά τους τόσο στον δημόσιο όσο και στον ιδιωτικό τομέα. Για την ανάλυση αυτή, θα αναλυθούν διάφορες περιπτώσεις χρήσης chatbots στην Ελλάδα, αξιολογώντας τις λειτουργίες και τις επιδόσεις τους.

1. Chatbots στην Ελληνική γλώσσα

Η χρήση των chatbots στην ελληνική γλώσσα έχει αυξηθεί σημαντικά τα τελευταία χρόνια. Οι επιχειρήσεις και οι οργανισμοί αναγνωρίζουν τις δυνατότητές τους να βελτιώσουν την αποτελεσματικότητα και την ικανοποίηση των πελατών. Παρά την αυξημένη χρήση, η ανάπτυξη των chatbots στην ελληνική γλώσσα αντιμετωπίζει προκλήσεις λόγω της πολυπλοκότητας της γλώσσας και της έλλειψης τυποποίησης.

2. Καταγραφή Chatbots στον Ιδιωτικό Τομέα

Ο ιδιωτικός τομέας στην Ελλάδα έχει υιοθετήσει ευρέως τα chatbots για την παροχή υποστήριξης πελατών και την ενίσχυση της αφοσίωσης των πελατών. Ενδεικτικά παραδείγματα περιλαμβάνουν:

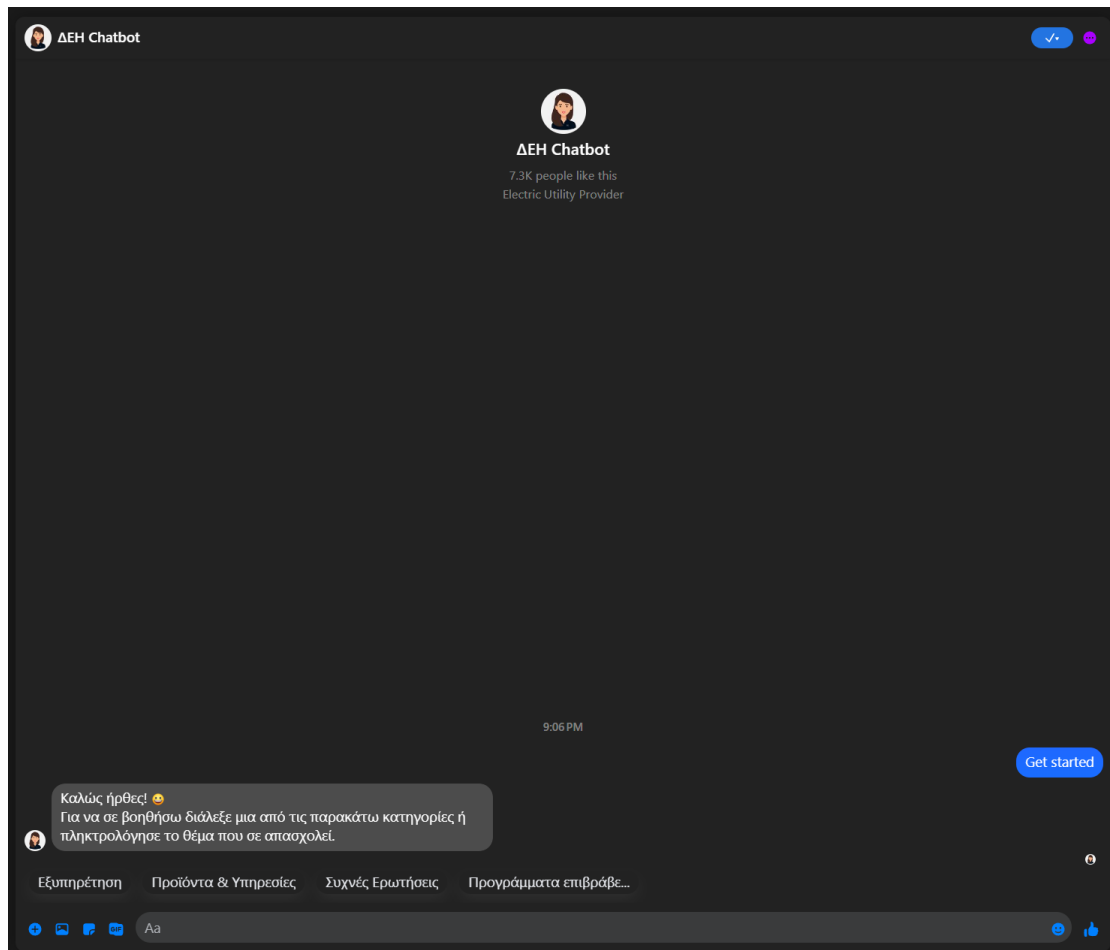
ΔΕΗ: Το chatbot της ΔΕΗ παρέχει πληροφορίες μέσω μενού επιλογών και αναγνώρισης λέξεων-κλειδίων, εξυπηρετώντας τους χρήστες τόσο μέσω Viber όσο και Messenger.

¹ <https://journals.plos.org/plosone/article/file?id=10.1371/journal.pone.0174698&type=printable>

² https://www.lawspot.gr/nomika-blogs/spiros_tassis/dikaio-stin-epohi-tis-tehneitis-noimosynis

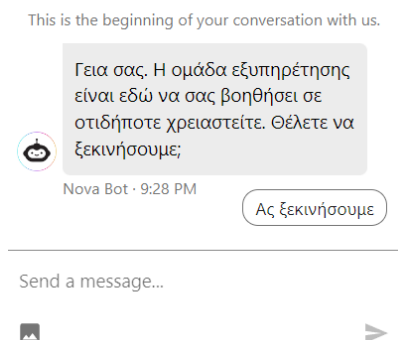
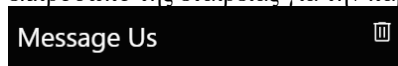
³ P. Lewis et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” May 2020.

⁴ Βλ. P. Lewis, ό.π.



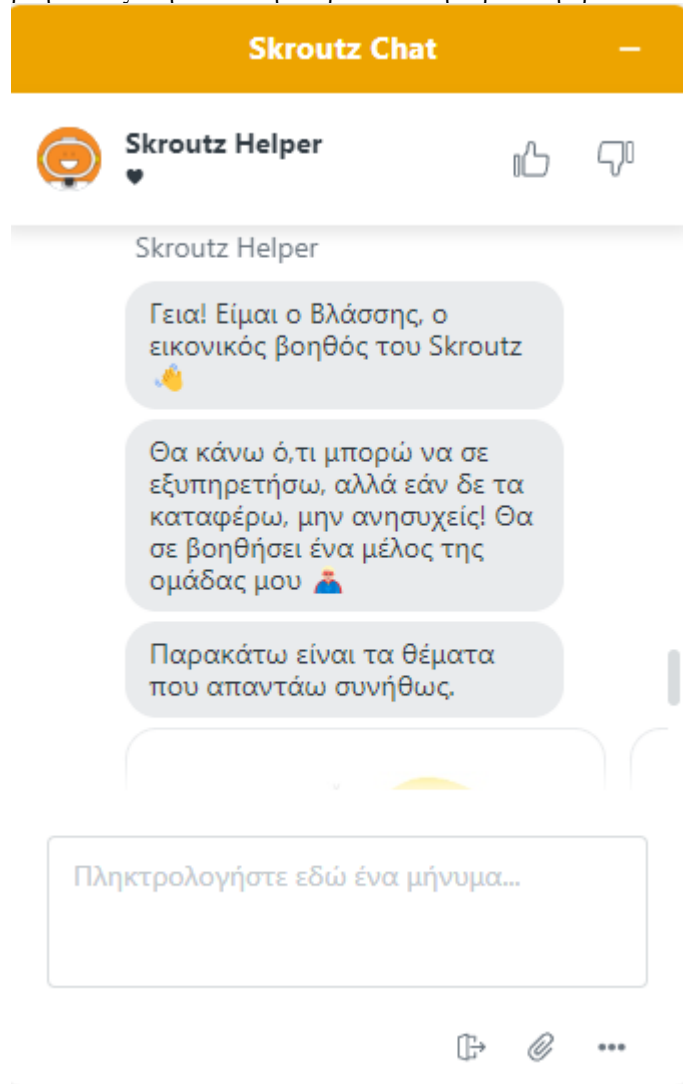
Εικόνα 1 Δεη chatbot

Nova - Wind: Χρησιμοποιεί μενού επιλογών για την αναγνώριση εντολών και συνδέει τον χρήστη με εκπρόσωπο της εταιρείας για την παροχή υπηρεσιών.



Εικόνα 2 Nova chatbot

Skroutz: Παρέχει πληροφορίες και υποστήριξη μέσω συνομιλίας με υπάλληλο της Skroutz, βοηθώντας στην επίλυση αποριών και την πρόταση προϊόντων.

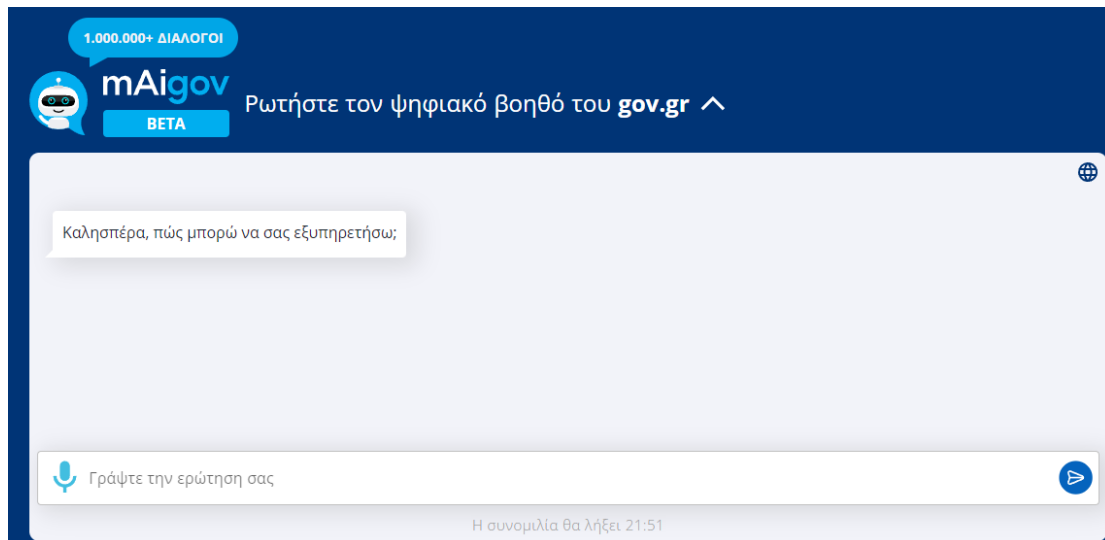


Εικόνα 3 Skroutz chatbot

3. Καταγραφή Chatbots στον Δημόσιο Τομέα

Στον δημόσιο τομέα, τα chatbots χρησιμοποιούνται για την παροχή πληροφοριών και υπηρεσιών στους πολίτες με αποδοτικό τρόπο.

Gov.gr: Το chatbot μέσω διαλόγου προσφέρει πληροφορίες και βοήθεια για τις ερωτήσεις του χρήστη.



Εικόνα 4 mAigov chatbot

4. Προκλήσεις και Ευκαιρίες

Παρά την πρόοδο, η ανάπτυξη των chatbots στην ελληνική γλώσσα αντιμετωπίζει προκλήσεις όπως η πολυπλοκότητα της γλώσσας και οι διάφορες διαλέκτους. Ωστόσο, οι προσπάθειες ανάπτυξης μοντέλων NLP ειδικά για την ελληνική γλώσσα και η συνεχής αναγνώριση των πλεονεκτημάτων των chatbots αναμένεται να βελτιώσουν την υιοθέτηση και την αποτελεσματικότητα των chatbots στην Ελλάδα.

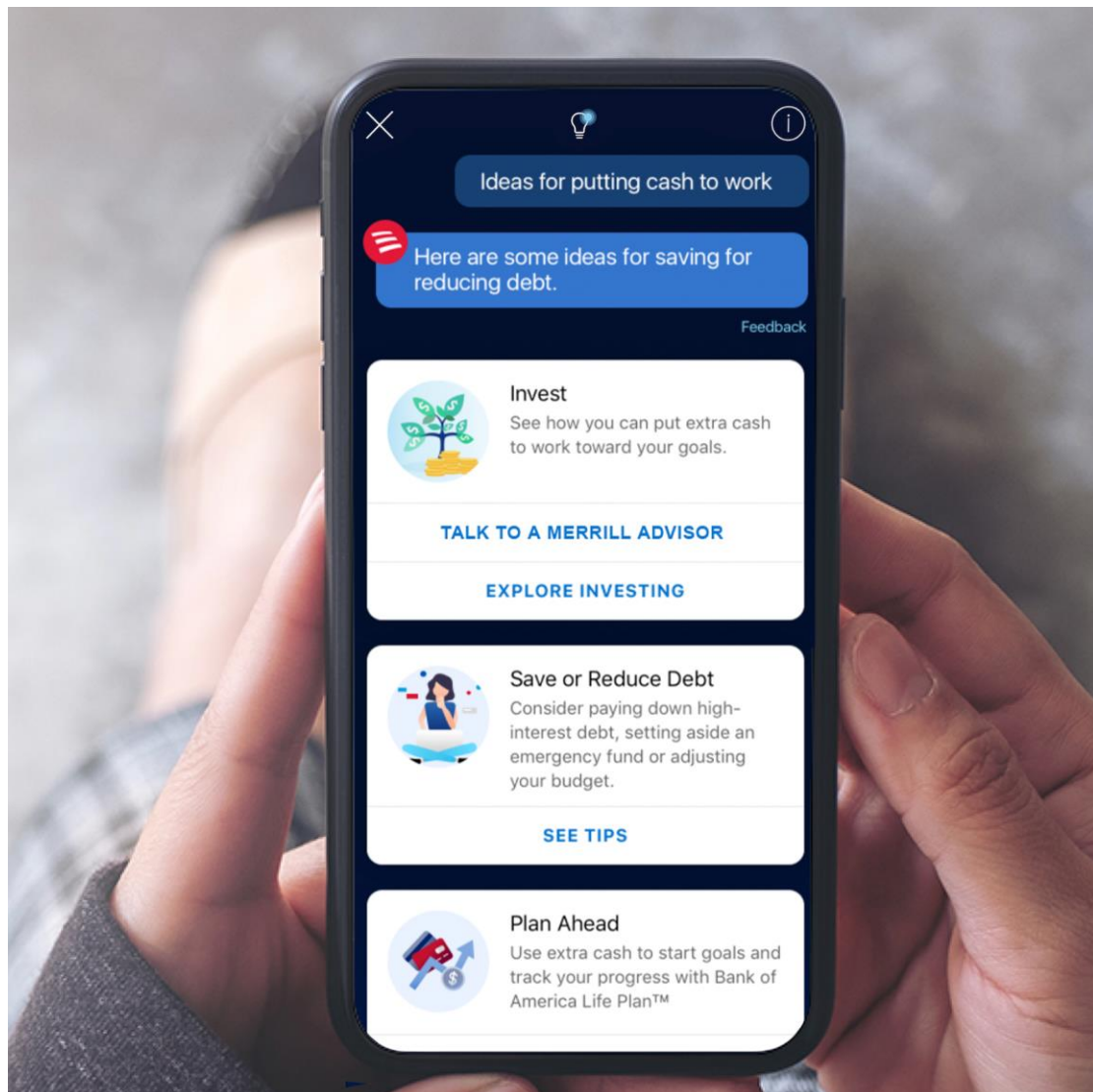
Συμπερασματικά, η χρήση των chatbots στην Ελλάδα αναπτύσσεται με γρήγορο ρυθμό, με πολλές επιχειρήσεις και δημόσιους φορείς να υιοθετούν την τεχνολογία αυτή για τη βελτίωση της επικοινωνίας και της εξυπηρέτησης των χρηστών. Οι προκλήσεις παραμένουν, αλλά οι ευκαιρίες για περαιτέρω ανάπτυξη είναι σημαντικές.

2.3 Τρέχουσα Κατάσταση στο Εξωτερικό

1. Chatbots στη Χρηματοοικονομική Τεχνολογία (Chatbots in Financial Technology)

Στον τομέα της χρηματοοικονομικής τεχνολογίας, τα chatbots έχουν γίνει απαραίτητα εργαλεία για τη βελτίωση της εμπειρίας των πελατών και την παροχή άμεσης υποστήριξης. Παραδείγματα περιλαμβάνουν:

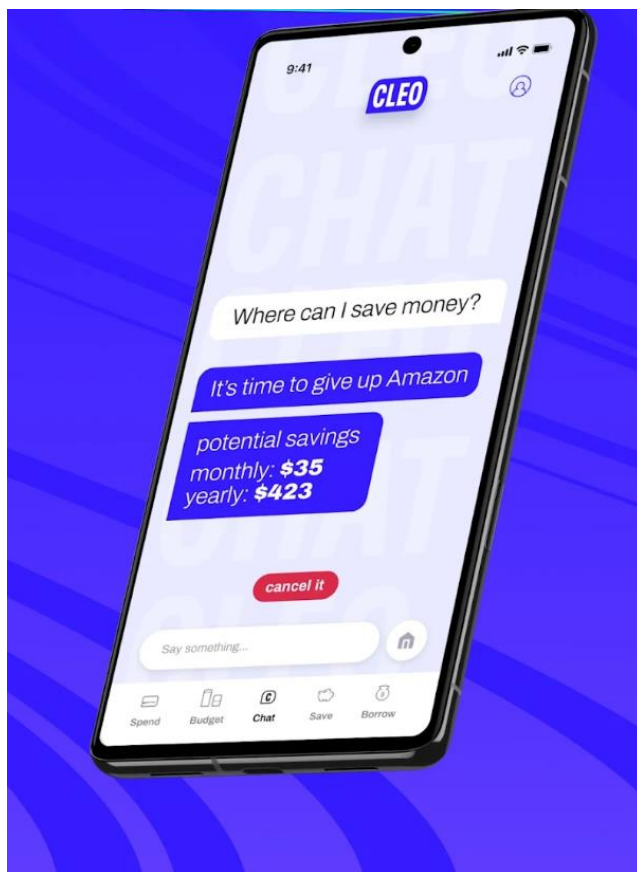
- **Erica από τη Bank of America:** Το chatbot Erica βοηθά τους πελάτες με οικονομικές συναλλαγές, προσφέρει συμβουλές αποταμίευσης και παρέχει ειδοποιήσεις για λογαριασμούς και επενδύσεις. Είναι ενσωματωμένο στην εφαρμογή της τράπεζας και χρησιμοποιεί τεχνολογίες NLP για την κατανόηση και απάντηση σε ερωτήματα πελατών.



Εικόνα 5 Erica chatbot⁵

- **Cleo:** Ένα ανεξάρτητο chatbot που παρέχει συμβουλές διαχείρισης οικονομικών, βοηθώντας τους χρήστες να παρακολουθούν τα έξοδά τους, να δημιουργούν προϋπολογισμούς και να αποταμιεύουν χρήματα. Είναι διαθέσιμο μέσω Facebook Messenger και άλλων πλατφορμών.

⁵ <https://promotions.bankofamerica.com/digitalbanking/mobilebanking/erica>

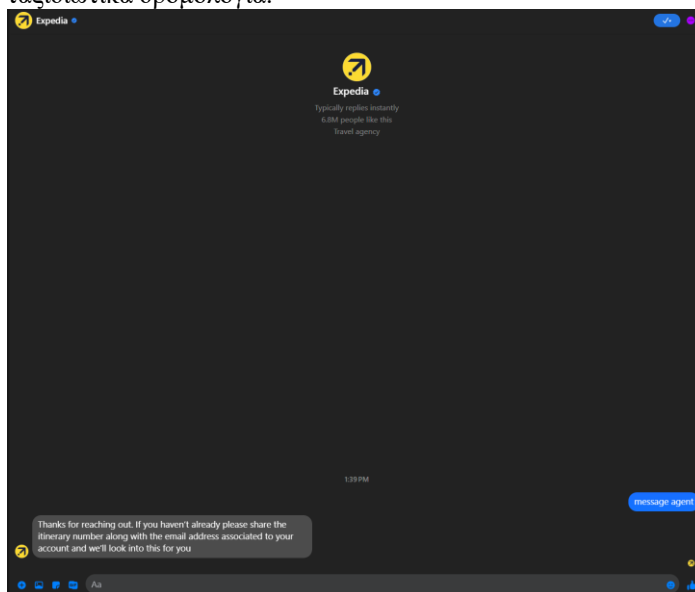


Εικόνα 6 Cleo chatbot⁶

2. Chatbots στον Τουρισμό (Chatbots in Tourism)

Η τουριστική βιομηχανία υιοθετεί τα chatbots για να βελτιώσει την εμπειρία των πελατών, προσφέροντας πληροφορίες και βοήθεια 24/7. Παραδείγματα περιλαμβάνουν:

- **Expedia Virtual Assistant:** Παρέχει υποστήριξη στους χρήστες για την κράτηση πτήσεων και ξενοδοχείων, την αλλαγή ή ακύρωση κρατήσεων, καθώς και πληροφορίες για τα ταξιδιωτικά δρομολόγια.



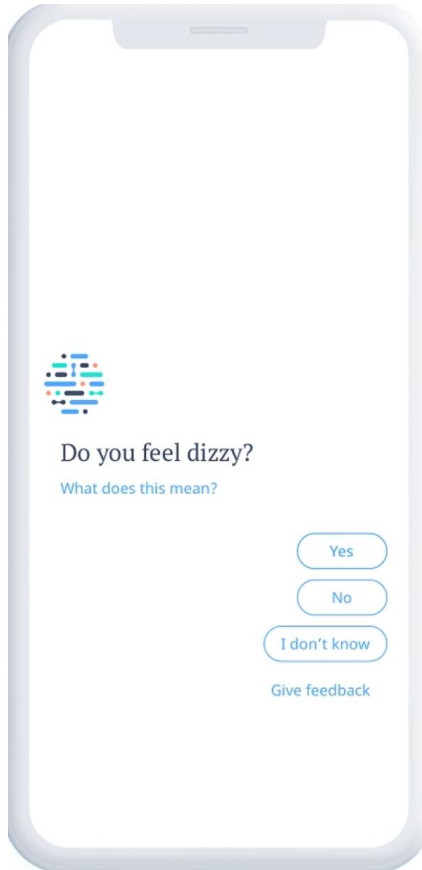
Εικόνα 7 Expedia messenger chatbot

⁶ <https://web.meetcleo.com/>

3. Chatbots στην Υγειονομική Περίθαλψη (Chatbots in Healthcare)

Στον τομέα της υγειονομικής περίθαλψης, τα chatbots χρησιμοποιούνται για την παροχή πληροφοριών στους ασθενείς, τη βοήθεια στον προγραμματισμό ραντεβού και την υποστήριξη διάγνωσης ιατρικών παθήσεων. Παραδείγματα περιλαμβάνουν:

- **Ada Health:** Λειτουργεί παρόμοια με το Babylon, παρέχοντας εξατομικευμένες ιατρικές αξιολογήσεις και συστάσεις με βάση τα συμπτώματα που περιγράφουν οι χρήστες.



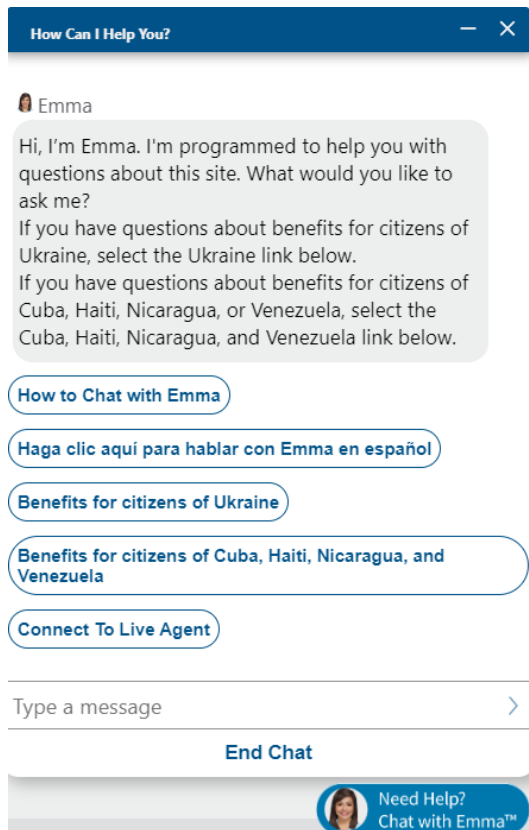
Εικόνα 8 Ada chatbot⁷

4. Chatbots στη Δημόσια Διοίκηση (Chatbots in Public Administration)

Οι κυβερνήσεις και οι δημόσιες υπηρεσίες παγκοσμίως χρησιμοποιούν chatbots για να βελτιώσουν την παροχή υπηρεσιών στους πολίτες. Παραδείγματα περιλαμβάνουν:

- **USCIS Emma:** Ένα chatbot της Υπηρεσίας Ιθαγένειας και Μετανάστευσης των ΗΠΑ, που παρέχει πληροφορίες σχετικά με τις υπηρεσίες μετανάστευσης και βοηθά τους χρήστες να πλοηγηθούν στον ιστότοπο της υπηρεσίας.

⁷ <https://ada.com/app/>

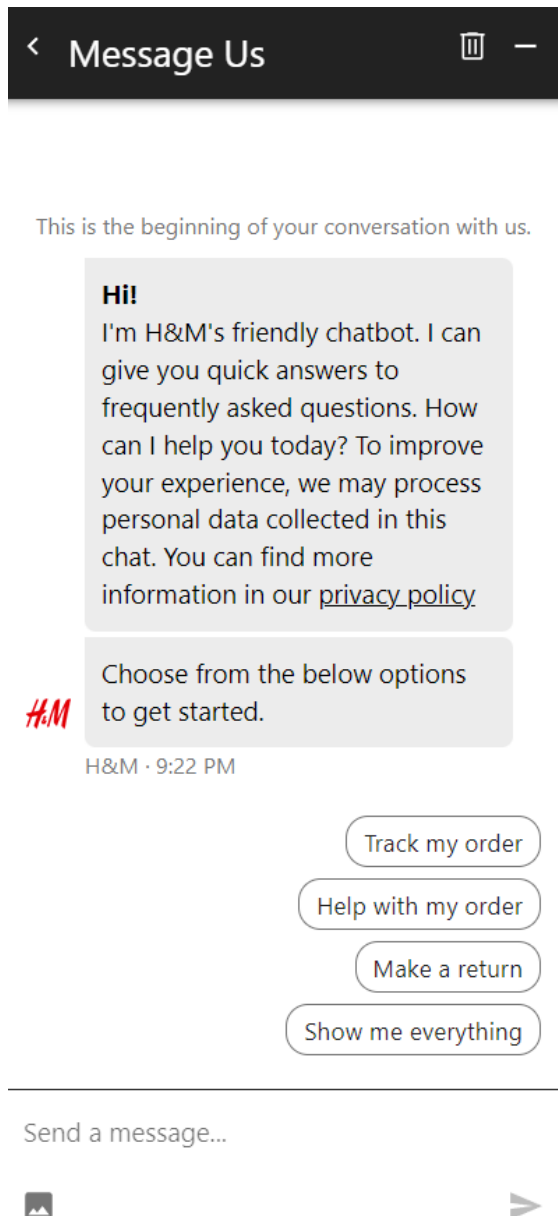


Εικόνα 9 Emma chatbot

5. Chatbots στην Εξυπηρέτηση Πελατών (Chatbots in Customer Service)

Πολλές επιχειρήσεις χρησιμοποιούν chatbots για την εξυπηρέτηση πελατών, προσφέροντας άμεση υποστήριξη και βελτιώνοντας την εμπειρία των χρηστών. Παραδείγματα περιλαμβάνουν:

- **H&M Virtual Assistant:** Βοηθά τους πελάτες να βρουν προϊόντα, να κάνουν παραγγελίες και να παρακολουθούν τις παραγγελίες τους μέσω της ιστοσελίδας και της εφαρμογής της H&M.



Εικόνα 10 H&M chatbot

Τα παραπάνω παραδείγματα δείχνουν τη δυναμική υιοθέτηση των chatbots σε παγκόσμιο επίπεδο και την ποικιλία των εφαρμογών τους σε διαφορετικούς τομείς. Με την πρόοδο της τεχνολογίας και την αύξηση της αποδοχής από τους χρήστες, τα chatbots αναμένεται να παίξουν ακόμη μεγαλύτερο ρόλο στην καθημερινή αλληλεπίδραση μεταξύ επιχειρήσεων και πελατών.

2.4 Συμπεράσματα

Η χρήση των chatbots αυξάνεται σταθερά στην Ελλάδα, με σημαντική υιοθέτηση τόσο στον ιδιωτικό όσο και στον δημόσιο τομέα, παρά τις προκλήσεις που παρουσιάζει η πολυπλοκότητα της ελληνικής γλώσσας. Στο εξωτερικό, η χρήση των chatbots έχει αυξηθεί ραγδαία, με πολλές επιχειρήσεις και οργανισμούς να τα ενσωματώνουν σε διάφορους τομείς, όπως η χρηματοοικονομική τεχνολογία, ο τουρισμός, η υγειονομική περίθαλψη και η δημόσια διοίκηση. Ωστόσο, η επεξεργασία φυσικής γλώσσας (NLP) παραμένει πρόκληση, ιδιαίτερα για λιγότερο διαδεδομένες γλώσσες όπως η ελληνική, και η ανάγκη για μεγάλα σύνολα δεδομένων για την εκπαίδευση των chatbots είναι κοινή πρόκληση, ειδικά για τις μικρότερες επιχειρήσεις. Παρά τις τεχνολογικές προκλήσεις, οι εξελίξεις στην τεχνητή νοημοσύνη και το NLP προσφέρουν σημαντικές ευκαιρίες για τη βελτίωση των chatbots, ενώ η αυξημένη αποδοχή και εμπιστοσύνη των χρηστών ενισχύει τη δυναμική υιοθέτησή τους τόσο στην Ελλάδα όσο και διεθνώς.

Συγκεντρωτικός Πίνακας Ευρημάτων

Τομέας	Παράδειγμα Chatbot	Χώρα	Κύριες Λειτουργίες	Τεχνολογίες Χρησιμοποιούμενες	Προκλήσεις
Χρηματοοικονομική Τεχνολογία	Erica (Bank of America)	ΗΠΑ	Οικονομικές συναλλαγές, συμβουλές αποταμίευσης, ειδοποιήσεις	NLP, AI	Πολυπλοκότητα γλώσσας
Χρηματοοικονομική Τεχνολογία	Cleo	Ηνωμένο Βασίλειο	Διαχείριση οικονομικών, παρακολούθηση εξόδων, προϋπολογισμοί	NLP, AI	Εκπαίδευση μοντέλων
Τουρισμός	Expedia Virtual Assistant	Διεθνές	Κρατήσεις πτήσεων και ξενοδοχείων, αλλαγές κρατήσεων	NLP, AI	Διεπαφή χρήστη
Υγειονομική Περίθαλψη	Ada Health	Διεθνές	Ιατρικές αξιολογήσεις, συστάσεις	NLP, AI	Αξιοπιστία δεδομένων
Δημόσια Διοίκηση	USCIS Emma	ΗΠΑ	Υπηρεσίες μετανάστευσης, πληροφορίες	NLP, AI	Διεπαφή χρήστη
Εξυπηρέτηση Πελατών	H&M Virtual Assistant	Διεθνές	Εύρεση προϊόντων, παραγγελίες, παρακολούθηση παραγγελιών	NLP	Προσωποποίηση
Δημόσιες Υπηρεσίες	mAigov	Ελλάδα	Παροχή πληροφοριών, υποστήριξη μέσω μενού επιλογών	NLP, AI	Εύρος θεμάτων
Εξυπηρέτηση Πελατών	ΔΕΗ Chatbot	Ελλάδα	Πληροφορίες μέσω Viber και Messenger	NLP, AI	Αναγνώριση λέξεων-κλειδιών
Εξυπηρέτηση Πελατών	Nova - Wind Chatbot	Ελλάδα	Παροχή υπηρεσιών, σύνδεση με εκπρόσωπο	NLP, AI	Διεπαφή χρήστη
Εξυπηρέτηση Πελατών	Skrouz Chatbot	Ελλάδα	Υποστήριξη πελατών, προτάσεις προϊόντων	NLP, AI	Διαχείριση δεδομένων πελατών

8

Η έρευνα δείχνει ότι τα chatbots έχουν γίνει ένα απαραίτητο εργαλείο σε πολλούς τομείς, προσφέροντας σημαντικά οφέλη όπως η βελτίωση της εξυπηρέτησης πελατών και η αυτοματοποίηση διαδικασιών. Παρά τις προκλήσεις, οι τεχνολογικές εξελίξεις προσφέρουν νέες ευκαιρίες για περαιτέρω ανάπτυξη και βελτίωση. Οι επιχειρήσεις και οι οργανισμοί πρέπει να συνεχίσουν να επενδύουν στην ανάπτυξη και την εκπαίδευση των chatbots για να διασφαλίσουν ότι θα παραμείνουν στην αιχμή της τεχνολογίας και θα προσφέρουν την καλύτερη δυνατή εμπειρία στους χρήστες.

3 Μεθοδολογία

Για την εκπόνηση της παρούσας εργασίας, ακολουθήθηκε μια διαδικασία που περιλαμβάνει τα ακόλουθα βήματα:

⁸ <https://thepythiaproject.com/>

Αρχικά θα προσδιοριστεί το «πρόβλημα» που στην προκειμένη περίπτωση είναι ο ευρύς όγκος πληροφοριών που βρίσκονται διάσπαρτες, τόσο στο διαδίκτυο όσο και σε φυσική μορφή και στη συνέχεια θα δοθεί η «λύση» η οποία δίνεται με τη δημιουργία του συστήματος που θα αναλυθεί παρακάτω.

Εύρεση τεχνολογιών και εργαλείων: Αρχικά πραγματοποιήθηκε η εύρεση τεχνολογιών και εργαλείων όπως OpenAI API, Gradio και Langchain και η διεξοδική μελέτη του documentation τους.

Ανασκόπηση της Βιβλιογραφίας: Στη συνέχεια, πραγματοποιήθηκε μια εκτενής ανασκόπηση της ελληνικής και διεθνούς βιβλιογραφίας που αφορά το θέμα της εργασίας. Αυτή συντελέστηκε με την αναζήτηση σε ακαδημαϊκές βάσεις δεδομένων, διαδυσκτικά περιοδικά, επιστημονικά άρθρα και τέλος στο διαδίκτυο.

Συλλογή Δεδομένων: Μετά την ανασκόπηση της βιβλιογραφίας, πραγματοποιήθηκε συλλογή νομικών κειμένων και νομοθετικών κωδίκων. Η συλλογή των συγκεκριμένων δεδομένων πραγματοποιήθηκε μέσω μηχανής αναζήτησης (Google) χρησιμοποιώντας συγκεκριμένες λέξεις-κλειδιά όπως "ελληνική νομοθεσία", "νομικά κείμενα" και "νομικοί κώδικες".

Σχεδιασμός συστήματος: Κλείνοντας το θεωρητικό κομμάτι της εργασίας πραγματοποιείται ο σχεδιασμός του συστήματος ο οποίος βασίζεται στις εξειδικευμένες απαιτήσεις, λειτουργικές και μη, που είναι αναγκαίες για τη σωστή υλοποίηση του προγράμματος.

Πειραματισμός: Μετά το θεωρητικό κομμάτι της έρευνας υλοποιήθηκε το πρακτικό μέρος της εργασίας. Το πρώτο στάδιο περιλαμβάνει τον πειραματισμό με το ChatGPT για την εύρεση του κατάλληλου μοντέλου και ποιο από αυτά ανταποκρίνεται καλύτερα στις νομικές ερωτήσεις που του τέθηκαν. Παράλληλα εξετάστηκε η απόκριση του μοντέλου στα βέλτιστα ερεθίσματα (οδηγίες) που του δόθηκαν.

Ανάπτυξη του συστήματος: Εν συνεχεία του πρακτικού μέρους της εργασίας πραγματοποιήθηκε μια πρώτη υλοποίηση του συστήματος. Παράλληλα υφίσταται η ανάπτυξη της γραφικής διεπαφής του χρήστη μέσω του Gradio ενώ ταυτόχρονα χρησιμοποιείται το Langchain και τα επιμέρους εργαλεία του τα οποία διαδραματίζουν κομβικό ρόλο στην επιτυχή λειτουργία του συστήματος καθώς δρουν σαν συνδετικοί κρίκοι που ενώνουν όλα τα παραπάνω.

Αξιολόγηση Αποτελεσμάτων: Βαθμολόγηση των αποτελεσμάτων (με συγκεκριμένο μοντέλο βαθμολόγησης) του συστήματος που αναπτύχθηκε και σύγκριση του με το ChatGPT.

Ερμηνεία Αποτελεσμάτων: Τα αποτελέσματα της ανάλυσης δεδομένων ερμηνεύθηκαν με βάση την ερευνητική ερώτηση και τους στόχους της εργασίας.

Σύνθεση Εργασίας: Τέλος, η εργασία συντάχθηκε με βάση τα αποτελέσματα της ανάλυσης, την ανασκόπηση της βιβλιογραφίας, και την ερμηνεία των αποτελεσμάτων.

Αυτή η μεθοδολογία επέτρεψε την εκπόνηση μιας εργασίας που στηρίζεται σε επιστημονική έρευνα και ανάλυση, τόσο σε θεωρητικό όσο και πρακτικό επίπεδο ενώ παράλληλα παρέχει ένα σαφές πλαίσιο για την κατανόηση των αποτελεσμάτων και των συμπερασμάτων που προέκυψαν.

4 Αρχιτεκτονική Συστήματος

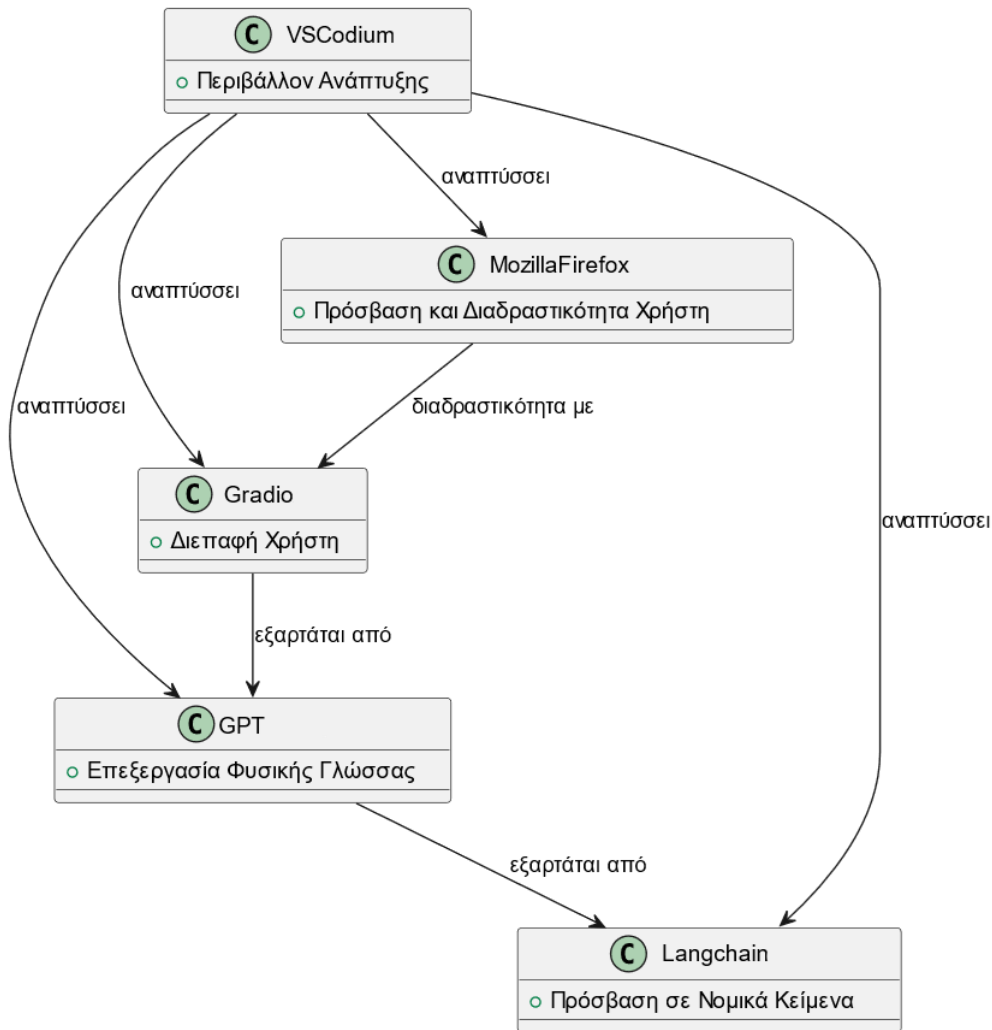
4.1 Απαιτήσεις Συστήματος

Στο πλαίσιο κάθε πληροφοριακού συστήματος, η διαδικασία της ανάλυσης απαιτήσεων αποτελεί έναν ουσιαστικό παράγοντα για την επιτυχή υλοποίηση και λειτουργία του. Οι απαιτήσεις είναι δηλώσεις που καθορίζουν τις λειτουργικές και τεχνικές προδιαγραφές που πρέπει να πληροί το σύστημα. Ειδικότερα, οι λειτουργικές απαιτήσεις αναφέρονται στις υπηρεσίες και τις δυνατότητες που το σύστημα πρέπει να προσφέρει, ενώ οι μη λειτουργικές απαιτήσεις καταγράφουν τους περιορισμούς και τις συνθήκες υπό τις οποίες αυτές οι λειτουργίες πρέπει να εκτελούνται.

Στην περίπτωση της παρούσας διπλωματικής εργασίας, οι απαιτήσεις επικεντρώνονται στον εξειδικευμένο GPT chatbot για την ελληνική νομοθεσία, και όχι στην ιστοσελίδα ή την πλατφόρμα στην οποία ο bot θα ενσωματωθεί. Αυτό εξασφαλίζει ότι οι απαιτήσεις είναι εξατομικευμένες και εστιασμένες στον κύριο στόχο της εργασίας.

Τίτλος απαίτησης	Περιγραφή	Σημαντικότητα	Σύνδεση με άλλες απαιτήσεις	Version Υλοποίησης	Εργαλεία
Συμβατότητα με ChatGPT	Το σύστημα πρέπει να είναι συμβατό με το ChatGPT για την επεξεργασία φυσικής γλώσσας	Υψηλή	-		ChatGPT, OpenAI API
Πρόσβαση σε Νομικά Κείμενα	Το σύστημα πρέπει να έχει πρόσβαση σε ελληνικά νομικά κείμενα για εκπαίδευση και αναζήτηση	Υψηλή	Συμβατότητα με ChatGPT		Langchain
Περιβάλλον Ανάπτυξης	Το σύστημα πρέπει να είναι αναπτύξιμο και τρέξιμο στο VSCodium.	Μεσαία	-		VSCodium
Συμβατότητα με Firefox	Το σύστημα πρέπει να είναι συμβατό με τον περιηγητή Mozilla Firefox για την πρόσβαση και την αλληλεπίδραση του χρήστη.	Μεσαία			Mozilla Firefox
Διεπαφή Χρήστη	Το σύστημα πρέπει να διαθέτει μια φιλική προς τον χρήστη διεπαφή μέσω του Gradio API.	Υψηλή	Συμβατότητα με ChatGPT		Gradio

4.2 Διαγράμματα



Διάγραμμα 1 Διάγραμμα εξαρτήσεων

Το διάγραμμα που παρουσιάζεται είναι ένα διάγραμμα εξαρτήσεων που απεικονίζει την αλληλεπίδραση μεταξύ των διαφόρων εργαλείων και APIs που χρησιμοποιούνται στην ανάπτυξη του εξειδικευμένου GPT chatbot για την ελληνική νομοθεσία.

1. VSCodium (Περιβάλλον Ανάπτυξης):

- Το VSCodium είναι ένα περιβάλλον ανάπτυξης κώδικα (IDE) που χρησιμοποιείται για τη συγγραφή, επεξεργασία και διαχείριση του κώδικα του chatbot.
- Από το VSCodium, οι προγραμματιστές έχουν τη δυνατότητα να αναπτύξουν και να ενσωματώσουν τον κώδικα που απαιτείται για τις υπόλοιπες συνιστώσες του συστήματος.

2. Mozilla Firefox (Πρόσβαση και Διαδραστικότητα Χρήστη):

- Ο Mozilla Firefox χρησιμοποιείται ως φυλλομετρητής (browser) για την πρόσβαση και τη διαδραστικότητα του χρήστη με τη διαδικτυακή εφαρμογή.
- Οι χρήστες μπορούν να χρησιμοποιήσουν τον Firefox για να αλληλεπιδρούν με το Gradio, το οποίο παρέχει τη διεπαφή χρήστη.

3. Gradio (Διεπαφή Χρήστη):

- Το Gradio χρησιμοποιείται για την υλοποίηση της διεπαφής χρήστη. Μέσω αυτού, οι χρήστες μπορούν να εισάγουν τα ερωτήματά τους και να λαμβάνουν απαντήσεις.
- Παρέχει ένα φιλικό περιβάλλον αλληλεπίδρασης όπου οι χρήστες μπορούν να επικοινωνούν με το ChatGPT.

4. GPT (Επεξεργασία Φυσικής Γλώσσας):

- Το GPT είναι το κεντρικό μοντέλο τεχνητής νοημοσύνης που είναι υπεύθυνο για την επεξεργασία της φυσικής γλώσσας. Λαμβάνει τις ερωτήσεις από το Gradio και παράγει απαντήσεις.
 - Το μοντέλο GPT κατανοεί και επεξεργάζεται τα ερωτήματα που του υποβάλλονται, παρέχοντας σχετικές απαντήσεις βάσει των δεδομένων που διαθέτει.
5. **Langchain (Πρόσβαση σε Νομικά Κείμενα):**
- Το Langchain χρησιμοποιείται για την πρόσβαση και την ανάκτηση νομικών κειμένων. Παρέχει στο GPT τις αναγκαίες πληροφορίες από νομικά έγγραφα ώστε να μπορεί να δώσει ακριβείς και τεκμηριωμένες απαντήσεις.
 - Μέσω του Langchain, το σύστημα μπορεί να αναζητήσει και να επεξεργαστεί νομικά δεδομένα από τις πηγές που είναι διαθέσιμες.

Ροή Δεδομένων

- **Ανάπτυξη Κώδικα:**
 - Ο κώδικας αναπτύσσεται στο VSCode, όπου οι προγραμματιστές δημιουργούν και διαχειρίζονται τις διάφορες λειτουργίες του συστήματος.
- **Αλληλεπίδραση Χρήστη:**
 - Οι χρήστες χρησιμοποιούν τον Mozilla Firefox για να έχουν πρόσβαση στη διεπαφή χρήστη που παρέχεται από το Gradio.
 - Μέσω του Gradio, οι χρήστες εισάγουν τα ερωτήματά τους, τα οποία μεταφέρονται στο GPT για επεξεργασία.
- **Επεξεργασία Ερωτημάτων:**
 - Το GPT λαμβάνει τα ερωτήματα από το Gradio και τα επεξεργάζεται.
 - Ανάλογα με το ερώτημα, το GPT μπορεί να ζητήσει πληροφορίες από το Langchain για να ανακτήσει σχετικές νομικές πληροφορίες.
- **Παραγωγή Απαντήσεων:**
 - Το GPT παράγει τις απαντήσεις βασιζόμενο στις επεξεργασμένες πληροφορίες και τις δεδομένες νομικές πηγές.
 - Οι απαντήσεις αποστέλλονται πίσω στο Gradio, το οποίο τις παρουσιάζει στον χρήστη μέσω της διεπαφής χρήστη στον Mozilla Firefox.

Αυτό το διάγραμμα παρέχει μια σαφή και συνοπτική εικόνα των διαφόρων εργαλείων που χρησιμοποιούνται στην ανάπτυξη και λειτουργία του GPT chatbot για την ελληνική νομοθεσία, καθώς και τον τρόπο με τον οποίο αλληλεπιδρούν μεταξύ τους για την παροχή ακριβών και χρήσιμων απαντήσεων στους χρήστες.

5 Εργαλεία Ανάπτυξης συστήματος

Στο σημείο αυτό είναι εύλογο να αναφερθεί ότι καταβλήθηκε η αναγκαία προσπάθεια εύρεσης ποικίλων εργαλείων, από τα οποία η πλειοψηφία υπήρξε εργαλεία ανοικτού κώδικα, μέσα από μία ευρεία γκάμα επιλογών. Με αυτό τον τρόπο επιλέχθηκαν τα καταλληλότερα εργαλεία με σκοπό να επιτευχθεί το ζητούμενο της διπλωματικής εργασίας.

5.1 ChatGPT

Το ChatGPT είναι ένα μοντέλο γλωσσικής παραγωγής που αναπτύχθηκε από την OpenAI. Το συγκεκριμένο μοντέλο χρησιμοποιεί μια αρχιτεκτονική που βασίζεται στους μετασχηματιστές (Transformers), η οποία επιτρέπει την επεξεργασία και την παραγωγή φυσικής γλώσσας σε πολύπλοκες δομές και μορφές.

Το ChatGPT έχει εκπαιδευτεί σε μεγάλους όγκους κειμένου και είναι ικανό να παράγει συνεκτικό και συνεχή λόγο, απαντώντας σε ερωτήσεις, δημιουργώντας κείμενα και εκτελώντας άλλες γλωσσικές εργασίες. Η ευελιξία και η ικανότητά του να κατανοεί και να παράγει φυσική γλώσσα καθιστούν το ChatGPT ένα πολύτιμο εργαλείο σε πολλούς τομείς, όπως η επιστημονική έρευνα, η εκπαίδευση, η δημιουργική γραφή και άλλα.

Η αρχιτεκτονική του ChatGPT περιλαμβάνει πολλαπλά επίπεδα μετασχηματιστών, τα οποία επιτρέπουν την παράλληλη επεξεργασία πληροφοριών και την αποτελεσματική εκμάθηση σχέσεων μεταξύ διαφορετικών μερών ενός κειμένου. Αυτό επιτρέπει στο μοντέλο να κατανοεί και να

αναπαράγει πολύπλοκες γλωσσικές δομές και να ανταποκρίνεται σε ερωτήσεις και εργασίες με υψηλή ακρίβεια.

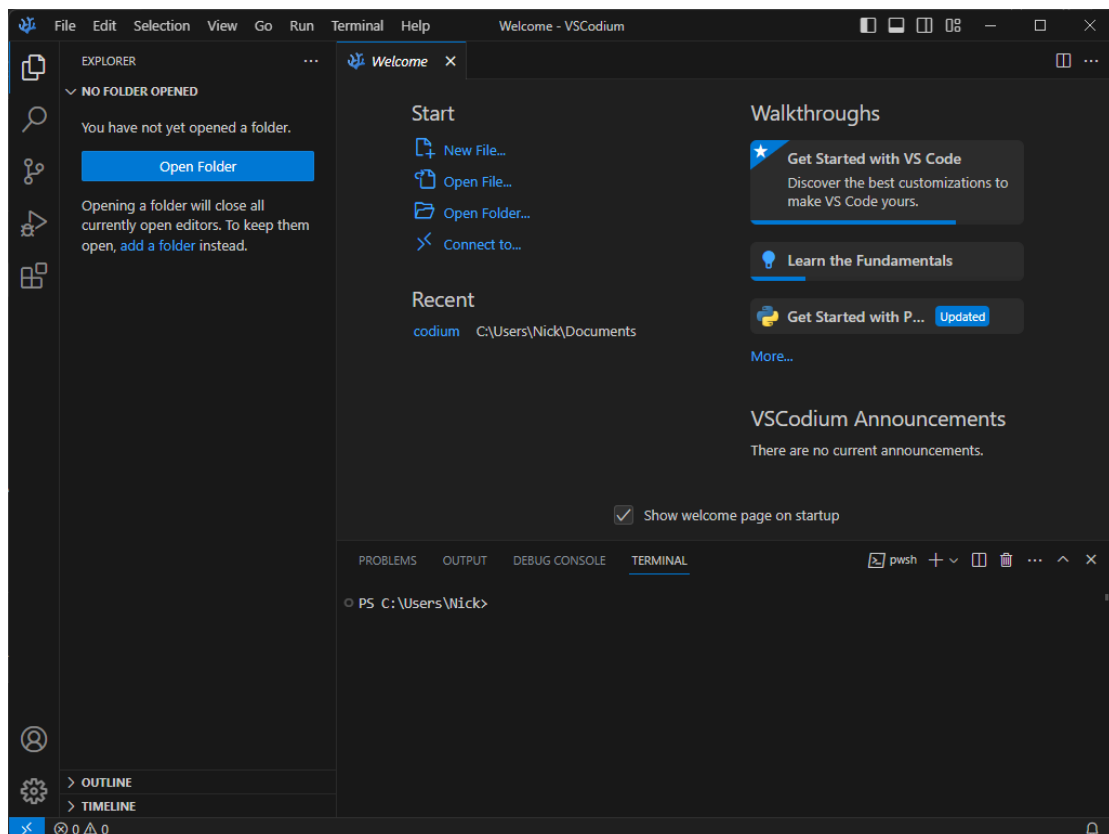
Είναι σημαντικό να σημειωθεί ότι τα μοντέλα της σειράς GPT, όπως το ChatGPT, μπορούν να είναι πολύπλοκα και να απαιτούν σημαντικούς πόρους υπολογιστικής ισχύος για την εκπαίδευση και τη λειτουργία τους.

5.2 VsCodium

Το VSCodium είναι ένα ελεύθερο και ανοιχτού κώδικα πρόγραμμα επεξεργασίας κειμένου και περιβάλλον ανάπτυξης (Integrated Development Environment - IDE). Πρόκειται για μια κοινοτικά διαχειριζόμενη έκδοση του Visual Studio Code, το οποίο είναι ένα δημοφιλές IDE που αναπτύχθηκε από την Microsoft.⁹

Η κύρια διαφορά μεταξύ του VSCodium και του Visual Studio Code είναι ότι το VSCodium δεν περιλαμβάνει κάποιες εμπορικές εξαρτήσεις και τηλεμετρικά δεδομένα που υπάρχουν στην επίσημη έκδοση του Visual Studio Code. Αυτό σημαίνει ότι το VSCodium προσφέρει μια εμπειρία που σέβεται περισσότερο την ιδιωτικότητα του χρήστη και συμμορφώνεται πλήρως με τις αρχές του ανοιχτού κώδικα.

Το VSCodium παρέχει πολλές από τις ίδιες λειτουργίες που κάνουν το Visual Studio Code ένα ισχυρό εργαλείο για προγραμματιστές, όπως η υποστήριξη πολλαπλών γλωσσών προγραμματισμού, η ενσωμάτωση με συστήματα ελέγχου εκδόσεων, η δυνατότητα επέκτασης μέσω πρόσθετων, και άλλα. Το VSCodium είναι διαθέσιμο για διάφορα λειτουργικά συστήματα, συμπεριλαμβανομένων των Windows, macOS και Linux, και μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών σε πολλές διαφορετικές γλώσσες και πλατφόρμες. Είναι ιδιαίτερα δημοφιλές μεταξύ των προγραμματιστών που προτιμούν ελεύθερο και ανοιχτό λογισμικό και θέλουν ένα περιβάλλον ανάπτυξης που σέβεται τις αρχές της κοινότητας του ανοιχτού κώδικα.



Εικόνα 11. VSCodium

⁹ <https://vscodium.com/#why>

5.3 Mozilla Firefox

Το Mozilla Firefox είναι ένας ελεύθερος και ανοιχτού κώδικα περιηγητής ιστού, που αναπτύσσεται από το Mozilla Foundation και το Mozilla Corporation. Πρόκειται για έναν από τους πιο δημοφιλείς περιηγητές ιστού στον κόσμο και είναι διαθέσιμος για πολλαπλές πλατφόρμες, συμπεριλαμβανομένων των Windows, macOS, Linux, καθώς και για κινητές συσκευές.¹⁰

Οι κύριοι πυλώνες πάνω στους οποίους έχει χτιστεί ο Mozilla Firefox είναι:

1. **Ανοιχτός Κώδικας:** Ο κώδικας του Firefox είναι ανοιχτός, πράγμα που σημαίνει ότι οποιοσδήποτε μπορεί να τον εξετάσει, να τον τροποποιήσει και να συμβάλει στην ανάπτυξή του.
2. **Προστασία της Ιδιωτικότητας:** Ο Firefox δίνει ιδιαίτερη έμφαση στην προστασία της ιδιωτικότητας των χρηστών, προσφέροντας διάφορες λειτουργίες όπως τον περιορισμό της παρακολούθησης από τρίτους και την ασφαλή αποθήκευση κωδικών πρόσβασης.
3. **Επεκτασιμότητα:** Ο Firefox υποστηρίζει μια πληθώρα επεκτάσεων και πρόσθετων, τα οποία επιτρέπουν στους χρήστες να προσαρμόσουν την εμπειρία περιήγησής τους σύμφωνα με τις ανάγκες τους.
4. **Ταχύτητα και Απόδοση:** Ο Firefox είναι γνωστός για την υψηλή του απόδοση και ταχύτητα, χάρη σε σύγχρονες τεχνολογίες όπως ο JavaScript engine και η υποστήριξη για WebAssembly.
5. **Κοινωνική Συμμετοχή:** Λόγω του ανοιχτού του χαρακτήρα, ο Firefox έχει μια ενεργή κοινότητα που συμβάλλει στην ανάπτυξη και τη βελτίωση του προγράμματος.

5.4 Pip

Το *pip* (που αντιπροσωπεύει το "Pip Installs Packages" ή "Pip Installs Python") είναι ένα σύστημα διαχείρισης πακέτων που χρησιμοποιείται για την εγκατάσταση και διαχείριση πακέτων λογισμικού γραμμένων σε Python. Αναπτύχθηκε αρχικά για να ενισχύσει την εγκατάσταση βιβλιοθηκών και εξαρτήσεων για τη γλώσσα προγραμματισμού Python και είναι ένα από τα πιο δημοφιλή εργαλεία για τη διαχείριση πακέτων Python.¹¹

Ευκολία Χρήσης: Το *pip* είναι σχεδιασμένο για να είναι εύκολο στη χρήση, με απλές εντολές που επιτρέπουν την εγκατάσταση, αναβάθμιση, και απεγκατάσταση πακέτων.

Ευελιξία: Το *pip* υποστηρίζει την εγκατάσταση πακέτων από διάφορες πηγές, συμπεριλαμβανομένου του PyPI (Python Package Index), τοπικών αρχείων, και αποθετηρίων VCS (Version Control Systems).

Συμβατότητα: Το *pip* είναι συμβατό με όλες τις μείζονες εκδόσεις της Python και λειτουργεί σε διάφορα λειτουργικά συστήματα.

Διαχείριση Εξαρτήσεων: Το *pip* επιτρέπει την αυτόματη εγκατάσταση όλων των εξαρτήσεων ενός πακέτου, διευκολύνοντας έτσι τη διαχείριση περίπλοκων έργων.

Κοινότητα και Υποστήριξη: Λόγω της ευρείας χρήσης του, υπάρχει μια μεγάλη κοινότητα που παρέχει υποστήριξη και συμβουλές για την αποτελεσματική χρήση του *pip*.

5.5 OpenAI API

OpenAI API είναι μια διεπαφή προγραμματισμού εφαρμογών (API) που παρέχεται από την OpenAI. Αυτή η διεπαφή επιτρέπει στους προγραμματιστές να αλληλεπιδρούν με τα μοντέλα μηχανικής μάθησης της OpenAI, όπως τα μοντέλα της σειράς GPT, μέσω ενός προγραμματιστικού περιβάλλοντος.¹²

Ο ορισμός του OpenAI API μπορεί να κατανοηθεί μέσω των εξής κύριων στοιχείων:

Πρόσβαση σε μοντέλα: Το API παρέχει πρόσβαση σε διάφορα προηγμένα μοντέλα μηχανικής μάθησης που έχουν αναπτυχθεί από την OpenAI. Αυτό περιλαμβάνει τόσο γενικούς σκοπούς όσο και εξειδικευμένα μοντέλα.

¹⁰ <https://www.mozilla.org/en-US/firefox/>

¹¹ <https://pip.pypa.io/en/stable/>

¹² <https://openai.com/index/openai-api/>

Ευελιξία και Προσαρμοστικότητα: Το API επιτρέπει την εκτέλεση διαφόρων εργασιών, όπως η παραγωγή κειμένου, η ανάλυση συναισθημάτων, η μετάφραση γλώσσας και άλλα, μέσω μιας ενιαίας διεπαφής.

Ασφάλεια και προστασία: Το OpenAI API χρησιμοποιεί πρωτόκολλα ασφαλείας για να διασφαλίσει ότι οι αλληλεπιδράσεις με τα μοντέλα γίνονται με ασφαλή τρόπο.

Εμπορική χρήση: Πολλές επιχειρήσεις και οργανισμοί χρησιμοποιούν το OpenAI API για να ενσωματώσουν τις δυνατότητες της τεχνητής νοημοσύνης στα προϊόντα και τις υπηρεσίες τους.

Ευκολία χρήσης: Το API είναι σχεδιασμένο για να είναι προσβάσιμο και εύκολο στη χρήση, ακόμη και για όσους δεν έχουν εξειδικευμένες γνώσεις στη μηχανική μάθηση.

Συνολικά, το OpenAI API αποτελεί ένα ισχυρό εργαλείο που διευκολύνει την πρόσβαση και τη χρήση των τεχνολογιών τεχνητής νοημοσύνης της OpenAI, επιτρέποντας την ανάπτυξη καινοτόμων εφαρμογών και υπηρεσιών σε διάφορους τομείς και βιομηχανίες.

5.6 LangChain

Το LangChain είναι μια διεπαφή προγραμματισμού εφαρμογών που επιτρέπει στους προγραμματιστές να ενσωματώσουν τεχνητή νοημοσύνη συνομιλίας στις εφαρμογές τους.¹³

Χαρακτηριστικά του LangChain:

- Υποστηρίζει συνομιλίες σε φυσική γλώσσα σε πολλές γλώσσες όπως αγγλικά, ελληνικά, ισπανικά, γαλλικά κτλ.
- Επιτρέπει στους προγραμματιστές να κάνουν ερωτήσεις και να λαμβάνουν απαντήσεις από το μοντέλο τεχνητής νοημοσύνης.
- Υποστηρίζει εξειδικευμένα μοντέλα τεχνητής νοημοσύνης για συγκεκριμένους τομείς όπως ιατρική, νομικά κτλ.
- Επιτρέπει την προσαρμογή των απαντήσεων με βάση το πλαίσιο της συνομιλίας.
- Έχει χαμηλό κόστος χρήσης ανάλογα με τον αριθμό των αιτημάτων.

Συνοπτικά, το LangChain καθιστά εύκολη την ενσωμάτωση ικανοτήτων φυσικής γλώσσας σε εφαρμογές λογισμικού.

¹³ <https://www.langchain.com/>



Εικόνα 12. LangChain¹⁴

5.7 Gradio

Το Gradio είναι ένα framework που επιτρέπει τη δημιουργία διεπαφών χρήστη για μοντέλα μηχανικής μάθησης και τεχνητής νοημοσύνης.¹⁵

Χαρακτηριστικά του Gradio:

- Δημιουργεί γραφικές διεπαφές χρήστη χωρίς κώδικα για μοντέλα Python.
- Υποστηρίζει διάφορα είδη εισόδων (κείμενο, εικόνες, ήχος) και εξόδων.
- Επιτρέπει τον έλεγχο και τον έλεγχο μοντέλων τεχνητής νοημοσύνης μέσω διαδραστικών διεπαφών.
- Διευκολύνει την ανάπτυξη και τον έλεγχο πρωτοτύπων μοντέλων τεχνητής νοημοσύνης.
- Μπορεί να χρησιμοποιηθεί για τη δημιουργία εφαρμογών τελικού χρήστη με δυνατότητα προβλέψεων τεχνητής νοημοσύνης.
- Διαθέτει ενσωματωμένες λειτουργίες αποσφαλμάτωσης και καταγραφής.
- Είναι ανοιχτού κώδικα και δωρεάν για χρήση.

Συνολικά, το Gradio απλοποιεί τη δημιουργία διεπαφών χρήστη για μοντέλα τεχνητής νοημοσύνης σε Python.

¹⁴ <https://www.godaddy.com/resources/news/everything-ive-learned-building-an-ai-app>

¹⁵ <https://www.gradio.app/>

6 Το Σύστημα

Στο παρόν κεφάλαιο πραγματοποιείται η υλοποίηση του προτεινόμενου συστήματος βάσει ενός συγκεκριμένου πλαισίου, το οποίο αποτελείται από τα ακόλουθα βήματα:

- **Εντοπισμός Σχετικών Ερεθισμάτων (Prompts):** Αρχικά, επιδιώκεται η εντοπισμός των κατάλληλων ερεθισμάτων (prompts) που θα συμβάλλουν στην εξατομίκευση και βελτιστοποίηση των αποτελεσμάτων του GPT, ειδικά για τις ερωτήσεις που συνδέονται με την ελληνική νομοθεσία.
- **Εύρεση Νομικών Κειμένων και Ορολογίας:** Στη συνέχεια, εντοπίζονται νομικά κείμενα, ορολογία και κοινές νομικές πρακτικές που εφαρμόζονται στην ελληνική νομοθεσία, προκειμένου να εμπλουτιστεί το σύστημα με τις απαραίτητες γνώσεις.
- **Επιλογή Μοντέλου Γλώσσας:** Επιλέγεται το κατάλληλο μοντέλο γλώσσας που ταιριάζει στην κλίμακα των δεδομένων και στις απαιτήσεις της υλοποίησης του chatbot.
- **Αρχική Ανάπτυξη του Bot:** Υλοποιείται το chatbot σε αρχικό στάδιο, χρησιμοποιώντας το OpenAI API για την παραγωγή των απαντήσεων.
- **Ανάπτυξη Διεπαφής Χρήστη:** Δημιουργείται μια διεπαφή χρήστη για την αλληλεπίδραση με το chatbot, χρησιμοποιώντας το Gradio API για την παροχή μιας φιλικής προς τον χρήστη εμπειρίας.
- **Επιλογή και Καθαρισμός Νομικών Κειμένων:** Επιλέγονται και καθαρίζονται τα κείμενα της ελληνικής νομοθεσίας, προκειμένου να είναι κατάλληλα για χρήση και ενσωμάτωση στο σύστημα.
- **Ενσωμάτωση με το Langchain:** Τέλος, γίνεται η ενσωμάτωση και η αλληλεπίδραση όλων των παραπάνω στοιχείων με το Langchain, προκειμένου να υλοποιηθεί μια συνεκτική και λειτουργική λύση για την παροχή νομικών συμβουλών και πληροφοριών.

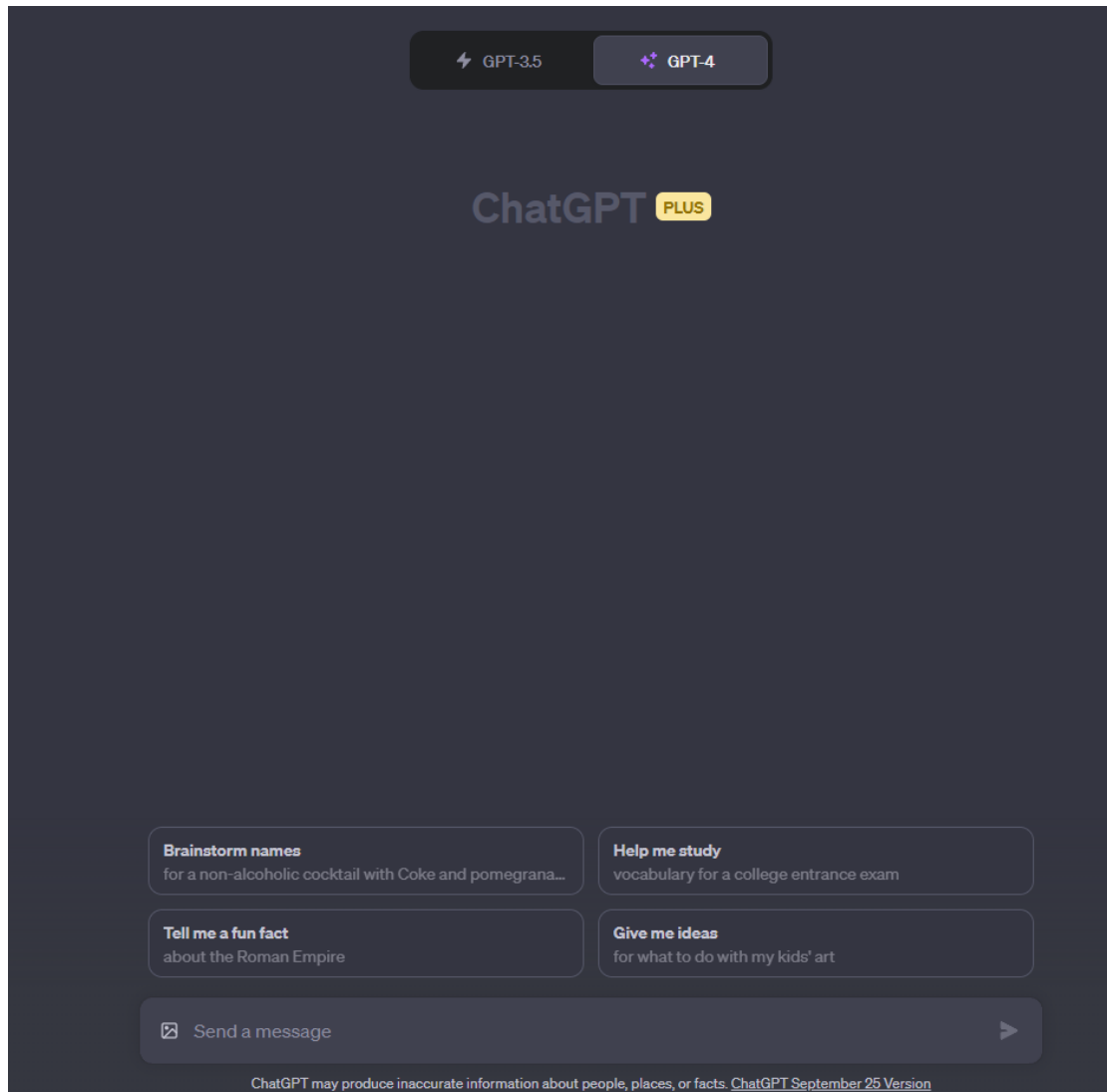
Καθένα από τα παραπάνω βήματα αποτελεί κρίσιμο στοιχείο της υλοποίησης, και προσφέρει το θεμέλιο για την ανάπτυξη ενός αποτελεσματικού και προσαρμοσμένου συστήματος chatbot για την ελληνική νομοθεσία.

6.1 Εντοπισμός σχετικών και βέλτιστων ερεθισμάτων (prompts)

Στο παρόν υποκεφάλαιο, η προσπάθειά επικεντρώνεται στην ανάπτυξη των κατάλληλων ερεθισμάτων και στην αρχική αξιολόγηση της εγκυρότητας των αποκρίσεων βασισμένες σε έναν επιλεγμένο νόμο. Στόχος είναι να κρίνουμε τα αποτελέσματα για ένα συγκεκριμένο μοντέλο γλώσσας, αρχικά επικεντρωνόμενοι στο ChatGPT-4, και στη συνέχεια στο ChatGPT-3.5, προκειμένου να εξακριβώσουμε την επίδοση και την ακρίβεια των αποκρίσεών τους. Αυτή η συγκριτική αξιολόγηση θα συμβάλει σημαντικά στην επιλογή του κατάλληλου μοντέλου γλώσσας για την επόμενη φάση της υλοποίησης.

Το κριτήριο αξιολόγησης της εγκυρότητας των αποκρίσεων είναι κεντρικό για την κατανόηση της ικανότητας των μοντέλων γλώσσας να παράγουν νομικά συνεπείς και ενημερωμένες απαντήσεις, που αντικατοπτρίζουν την τρέχουσα ελληνική νομοθεσία. Η διαδικασία αυτή αποτελεί βασικό βήμα για την περαιτέρω βελτίωση και προσαρμογή του συστήματος, ενισχύοντας την ικανότητα του chatbot να παρέχει ακριβείς και χρήσιμες νομικές πληροφορίες.

6.1.1 ChatGPT 4.0



Εικόνα 13. ChatGPT 4.0 September 25 version

Παραπάνω βλέπουμε την αρχική σελίδα του default ChatGPT 4.0 χωρίς κανένα επιπρόσθετο.

Άρθρο: 2

Ημ/νία: 23.02.1946

Ημ/νία Ισχύος: 23.02.1946

Τίτλος Άρθρου

Αναδρομική δύναμη του νόμου

Λήμματα

ΚΑΝΟΝΕΣ ΔΙΚΑΙΟΥ, ΝΟΜΟΙ, ΑΝΑΔΡΟΜΙΚΟΤΗΤΑ ΝΟΜΩΝ

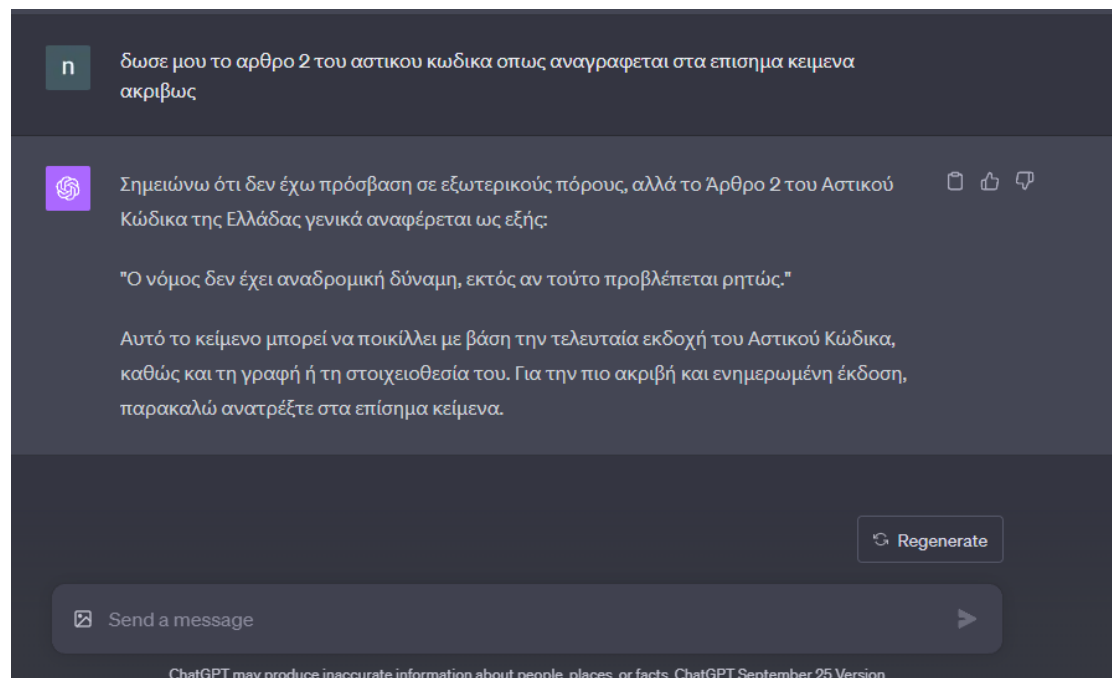
Κείμενο Άρθρου

Ο νόμος ορίζει για το μέλλον, δεν έχει αναδρομική δύναμη και διατηρεί την ισχύ του εφόσον άλλος κανόνας δικαίου δεν τον καταργήσει ρητά ή σιωπηρά.

Εικόνα 14. Άρθρο 2 του αστικού δικαίου όπως εμφανίζεται στα επίσημα κείμενα του υπουργείου δικαιοσύνης

Επιλέγοντας το άρθρο 2 του Αστικού Κώδικα ως σημείο αναφοράς, θα ξεκινήσουμε με μια απλοποιημένη ερώτηση, δίχως οποιαδήποτε παράμετρο σχετικά με το επιθυμητό αποτέλεσμα, προκειμένου να παρατηρήσουμε την απάντηση που θα λάβουμε. Αυτό θα μας επιτρέψει να αξιολογήσουμε την αρχική ανταπόκριση του μοντέλου γλώσσας σε μια βασική νομική ερώτηση, καθώς και να κατανοήσουμε τον τρόπο με τον οποίο το μοντέλο διαχειρίζεται τη νομική ορολογία και το περιεχόμενο.

Ερώτηση: «δώσε μου το άρθρο 2 του αστικού κώδικα όπως αναγράφεται στα επίσημα κείμενα ακριβώς»



Εικόνα 15. Αρχική ερώτηση στο ChatGPT 4.0 και πρώτο αποτέλεσμα

Από την ανάλυση της απάντησης που παρείχε το ChatGPT, αναγνωρίσαμε την πρώτη αδυναμία του μοντέλου, η οποία είναι η έλλειψη πρόσβασης σε εξωτερικούς πόρους. Αυτό το ζήτημα επισημαίνει την ανάγκη για ενσωμάτωση επιπρόσθετων νομικών κειμένων που θα ενισχύσουν την ικανότητα του συστήματος να παράγει έγκυρες και κατανοητές απαντήσεις. Στην επόμενη φάση της υλοποίησης, θα επιδιώξουμε να αντιμετωπίσουμε αυτήν την αδυναμία, προσθέτοντας τα απαραίτητα νομικά κείμενα

που θα επιτρέψουν την επίτευξη του επιθυμητού αποτελέσματος. Αλλά για αυτή την ενότητα θα προσπαθήσουμε πρώτα να πάρουμε το καλύτερο επιθυμητό αποτέλεσμα με ποιο στοχευμένες ερωτήσεις.

Ας δοκιμάσουμε να κάνουμε την ίδια ερώτηση θέτοντας ένα ρόλο στο ChatGPT και κάποιες οδηγίες. Ερώτηση: «Ως δικηγόρος εξειδικευμένος στον αστικό δίκαιο θέλω να μου εξηγήσεις το άρθρο 2 του αστικού κώδικα»



Εικόνα 16. Αποτέλεσμα μετά από ερώτηση στο ChatGPT 4.0 με οδηγίες

Από την παρατήρηση της ανωτέρω εικόνας, διαπιστώνουμε ότι η απάντηση είναι πιο αναλυτική και πλουσιότερη σε πληροφορίες. Επιπλέον, ακολουθώντας αυτήν την τεχνική, μπορούμε να δημιουργήσουμε ένα μοτίβο ερώτησης που θα μας φανεί χρήσιμο εν συνεχεία για οποιοδήποτε δίκαιο επιθυμούμε να εξετάσουμε, για παράδειγμα: «Ως δικηγόρος εξειδικευμένος στο "επιθυμητό δίκαιο", "ερώτηση"». Αυτό το μοτίβο ερώτησης θα επιτρέψει την εξατομίκευση των ερωτήσεων μας ανάλογα με το ενδιαφέρον του νομικού πεδίου.

6.1.2 ChatGPT 3.5

Σε αυτό το κεφάλαιο, θα ακολουθηθεί η ίδια μεθοδολογία, προκειμένου να αποκτήσουμε τη δυνατότητα σύγκρισης των αποτελεσμάτων, η οποία θα συμβάλει στην επιλογή του καταλλήλου μοντέλου



Εικόνα 17. Αποτέλεσμα μετά απο ερώτηση στο ChatGPT3.5 με οδηγίες

Αναλύοντας το αποτέλεσμα που λάβαμε από και τις δύο ερωτήσεις, το ChatGPT3.5 φαίνεται να είναι ακατάλληλο για τις ανάγκες μας, καθώς ίσως είναι πιο κατάλληλο για πιο απλά και καθημερινά ζητήματα, δεδομένου ότι οι απαντήσεις παρέχονται σε μια πιο ταχεία βάση.

6.2 Εύρεση Νομικών Κειμένων και Ορολογίας

Όλες οι πληροφορίες και νόμοι που θα χρησιμοποιηθούν για την υλοποίηση του bot μας θα αντληθούν από τα επίσημα αρχεία του Ελληνικού Υπουργείου Δικαιοσύνης, όπως διατίθενται στην επίσημη ιστοσελίδα. Οι αντίστοιχοι κώδικες και τα αντίστοιχα pdf αρχεία παρουσιάζονται στην παρακάτω εικόνα

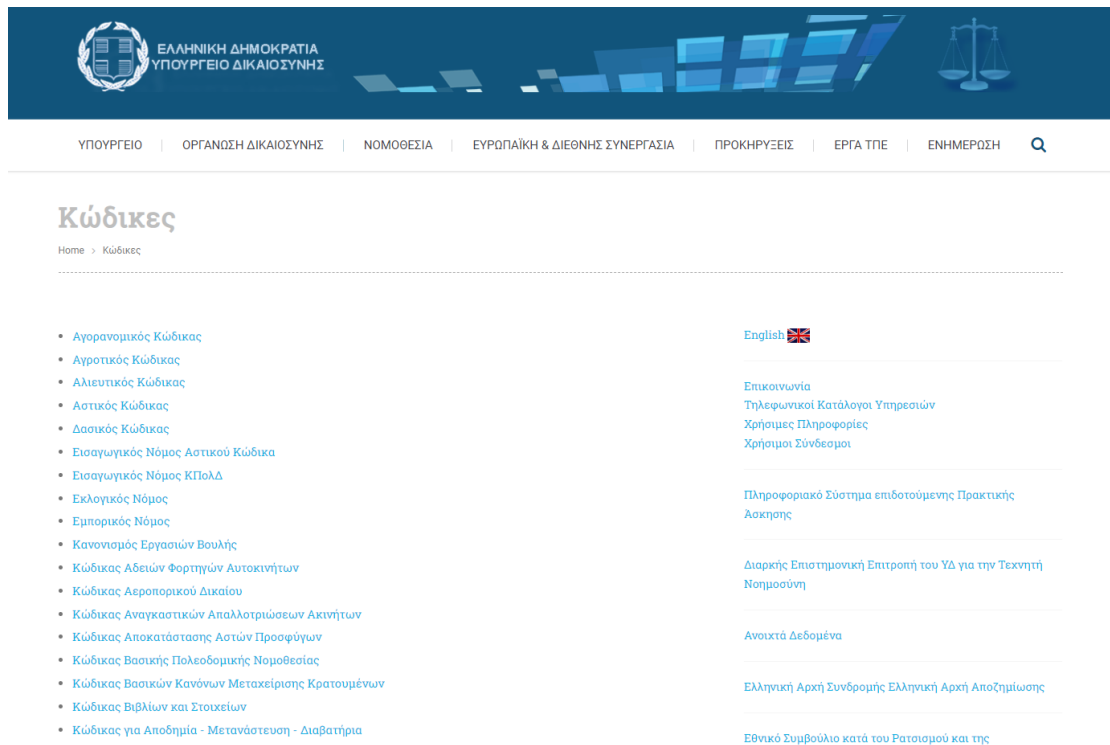


Figure 1. Ιστότοπος ελληνικού υπουργείου δικαιοσύνης

Όλοι οι κώδικες με τους συμπεριλαμβανόμενους συνδέσμους είναι οι ακόλουθοι:

- [Αγορανομικός Κώδικας](#)
- [Αγροτικός Κώδικας](#)
- [Αλιευτικός Κώδικας](#)
- [Αστικός Κώδικας](#)
- [Δασικός Κώδικας](#)
- [Εισαγωγικός Νόμος Αστικού Κώδικα](#)
- [Εισαγωγικός Νόμος ΚΠολΔ](#)
- [Εκλογικός Νόμος](#)
- [Εμπορικός Νόμος](#)
- [Κανονισμός Εργασιών Βουλής](#)
- [Κώδικας Αδειών Φορτηγών Αυτοκινήτων](#)
- [Κώδικας Αεροπορικού Δικαίου](#)
- [Κώδικας Αναγκαστικών Απαλλοτριώσεων Ακινήτων](#)
- [Κώδικας Αποκατάστασης Αστών Προσφύγων](#)
- [Κώδικας Βασικής Πολεοδομικής Νομοθεσίας](#)
- [Κώδικας Βασικών Κανόνων Μεταχείρισης Κρατουμένων](#)
- [Κώδικας Βιβλίων και Στοιχείων](#)
- [Κώδικας για Αποδημία - Μετανάστευση - Διαβατήρια](#)
- [Κώδικας για την Αποκατάσταση Κτηνοτρόφων](#)
- [Κώδικας Δεοντολογίας Δικηγορικού Λειτουργήματος](#)
- [Κώδικας Δεοντολογίας Επαγγέλματος Συντηρητή Αρχαιοτήτων](#)
- [Κώδικας Δημόσιου Λογιστικού](#)
- [Κώδικας Δημόσιου Ναυτικού Δικαίου](#)
- [Κώδικας Δήμων και Κοινοτήτων](#)
- [Κώδικας Διαταγμάτων για τα Δημοτολόγια](#)
- [Κώδικας Διαχείρισης Συστήματος](#)
- [Κώδικας Δικαστικού Σώματος Ενόπλων Δυνάμεων](#)
- [Κώδικας Δικαστικών Επιμελητών](#)
- [Κώδικας Δικαστικών Λειτουργών](#)
- [Κώδικας Δικαστικών Υπαλλήλων](#)
- [Κώδικας Δικηγόρων](#)

- [Κώδικας Δικών Δημοσίου](#)
- [Κώδικας Διοικητικής Διαδικασίας](#)
- [Κώδικας Διοικητικής Δικονομίας](#)
- [Κώδικας Εγγραφών κλπ Μαθητών στα ξένα σχολεία](#)
- [Κώδικας Εισπράξεως Δημοσίων Εσόδων](#)
- [Κώδικας Ελληνικής Ιθαγένειας](#)
- [Κώδικας Ελληνικής Φαρμακευτικής Δεοντολογίας](#)
- [Κώδικας Εσόδων Δήμων και Κοινοτήτων](#)
- [Κώδικας Ιδιωτικού Ναυτικού Δικαίου](#)
- [Κώδικας Κανονιστικών Διατάξεων για τα Ναρκωτικά](#)
- [Κώδικας Κατασκευής Δημοσίων Έργων](#)
- [Κώδικας Καταστάσεως Προσωπικού ΟΤΑ](#)
- [Κώδικας Κεντρικής Ενώσεως Δήμων και Κοινοτήτων](#)
- [Κώδικας Μετοχικού Ταμείου Πολιτικών Υπαλλήλων](#)
- [Κώδικας Μετοχικού Ταμείου Στρατού](#)
- [Κώδικας Ναξίας Σμύριδας](#)
- [Κώδικας Νομοθεσίας Κυβέρνησης _ Κυβερνητικών Οργάνων](#)
- [Κώδικας Νομοργιακής Αυτοδιοίκησης](#)
- [Κώδικας Νόμων για τα Ναρκωτικά](#)
- [Κώδικας Οδικής Κυκλοφορίας](#)
- [Κώδικας Οργανισμού Δικαστηρίων](#)
- [Κώδικας Οργανισμού Φορολογικών Δικαστηρίων](#)
- [Κώδικας Ορθής Γεωργικής Πρακτικής](#)
- [Κώδικας Παροχής Επενδυτικών Κινήτρων](#)
- [Κώδικας Περί Αργαιοτήτων](#)
- [Κώδικας Περί Μητρώων Αρρένων](#)
- [Κώδικας Ποινικής Δικονομίας](#)
- [Κώδικας Πολεμικών Συντάξεων](#)
- [Κώδικας Πολιτικής Δικονομίας](#)
- [Κώδικας Πολιτικών και Στρατιωτικών Συντάξεων](#)
- [Κώδικας Προσωπικού Λιμενικού Σώματος](#)
- [Κώδικας Συμβολαιογράφων](#)
- [Κώδικας Συντάξεων Προσωπικού Σιδηροδρόμων](#)
- [Κώδικας Ταμείου Νομικών](#)
- [Κώδικας Τελών Χαρτοσήμου](#)
- [Κώδικας του Ανωτάτου Ειδικού Δικαστηρίου](#)
- [Κώδικας Τροφίμων και Ποτών](#)
- [Κώδικας Φορολογίας Εισοδήματος](#)
- [Κώδικας Φορολογίας Καπνού](#)
- [Κώδικας Φορολογίας Κληρονομιών](#)
- [Κώδικας Φορολογίας Οινοπνεύματος](#)
- [Κώδικας Φορολογικής Δικονομίας](#)
- [Κώδικας Φορολογικών Στοιχείων](#)
- [Κώδικας Φόρου Προστιθέμενης Αξίας](#)
- [Νομοθεσία για την εκλογή βουλευτών](#)
- [Ποινικός Κώδικας](#)
- [Πτωχευτικός Κώδικας](#)
- [Στρατιωτικός Ποινικός Κώδικας](#)
- [Τελωνειακός Κώδικας](#)
- [Υπαλληλικός Κώδικας](#)

6.3 Επιλογή Μοντέλου Γλώσσας

Ανατρέχοντας στον ιστότοπο του OpenAI API, παρατηρούμε τα διαθέσιμα μοντέλα και τις ικανότητές τους, τα οποία παρουσιάζονται ως εξής:

Όνομα	Περιγραφή
<u>GPT-4</u>	Ένα σύνολο μοντέλων που βελτιώνουν το GPT-3.5 και μπορούν να κατανοήσουν καθώς και να δημιουργήσουν φυσική γλώσσα ή κώδικα
<u>GPT-3.5</u>	Ένα σύνολο μοντέλων που βελτιώνουν το GPT-3 και μπορούν να κατανοήσουν καθώς και να δημιουργήσουν φυσική γλώσσα ή κώδικα
<u>GPT base</u>	Ένα σύνολο μοντέλων χωρίς οδηγίες που μπορούν να κατανοήσουν καθώς και να δημιουργήσουν φυσική γλώσσα ή κώδικα
<u>DALL-E</u>	Ένα μοντέλο που μπορεί να παράγει και να επεξεργάζεται εικόνες με προτροπή φυσικής γλώσσας
<u>Whisper</u>	Ένα μοντέλο που μπορεί να μετατρέψει τον ήχο σε κείμενο
<u>Embeddings</u>	Ένα σύνολο μοντέλων που μπορούν να μετατρέψουν κείμενο σε αριθμητική μορφή
<u>Moderation</u>	Ένα προσαρμοσμένο μοντέλο που μπορεί να ανιχνεύσει εάν το κείμενο μπορεί να είναι ευαίσθητο ή μη ασφαλές
<u>GPT-3</u>	Ένα σύνολο μοντέλων που μπορούν να κατανοήσουν και να δημιουργήσουν φυσική γλώσσα

Για την υλοποίηση του εξειδικευμένου bot μας θα επικεντρωθούμε στις κατηγορίες του ChatGPT-3.5 ChatGPT-4. Συνεχίζοντας στην ιστοσελίδα του API του OpenAI οι κατηγορίες εμφανίζονται ως εξής:

6.3.1 GPT-3.5

GPT-3.5	Περιγραφή	Μέγιστα Tokens	Δεδομένα Εκπαίδευσης
gpt-3.5-turbo	Το πιο ικανό μοντέλο GPT-3.5 που έχει βελτιστοποιηθεί για συνομιλία με κόστος 1/10 του text-davinci-003. Θα ενημερωθεί με την τελευταία έκδοση του μοντέλου 2 εβδομάδες μετά την κυκλοφορία του.	4,097 tokens	Έως Σεπτέμβριος 2021
gpt-3.5-turbo-16k	Ίδιες δυνατότητες με το στάνταρ gpt-3.5-turbo αλλά με 4 φορές το περιεχόμενο.	16,385 tokens	Έως Σεπτέμβριος 2021
gpt-3.5-turbo-instruct	Δυνατότητες παρόμοιες με το text-davinci-003 αλλά συμβατές με το παλαιότερο endpoint Completions και όχι Chat Completions.	4,097 tokens	Έως Σεπτέμβριος 2021
gpt-3.5-turbo-0613	Στιγμιότυπο του gpt-3.5-turbo από 13 Ιουνίου 2023 με δεδομένα κλήσεων λειτουργίας. Σε αντίθεση με το gpt-3.5-turbo, αυτό το μοντέλο δεν θα λάβει ενημερώσεις και θα καταργηθεί 3 μήνες μετά την κυκλοφορία της νέας έκδοσης.	4,097 tokens	Έως Σεπτέμβριος 2021

gpt-3.5-turbo-16k-0613	Στιγμιότυπο του gpt-3.5-turbo-16k από 13 Ιουνίου 2023. Σε αντίθεση με το gpt-3.5-turbo-16k, αυτό το μοντέλο δεν θα λάβει ενημερώσεις και θα καταργηθεί 3 μήνες μετά την κυκλοφορία της νέας έκδοσης.	16,385 tokens	Έως Σεπτέμβριος 2021
-------------------------------	--	---------------	----------------------

6.3.2 GPT-4

ΤΕΛΕΥΤΑΙΟ ΜΟΝΤΕΛΟ	ΠΕΡΙΓΡΑΦΗ	ΜΕΓΙΣΤΑ TOKENS	ΔΕΔΟΜΕΝΑ ΕΚΠΑΙΔΕΥΣΗΣ
gpt-4	Πιο ικανό από κάθε μοντέλο GPT-3.5, μπορεί να εκτελέσει πιο σύνθετες εργασίες, και έχει βελτιστοποιηθεί για συνομιλία. Θα ενημερωθεί με την τελευταία έκδοσή του μοντέλου 2 εβδομάδες μετά την κυκλοφορία του.	8,192 tokens	Έως Σεπτέμβριος 2021
gpt-4-0613	Στιγμιότυπο του gpt-4 από 13 Ιουνίου 2023 με δεδομένα κλήσεων λειτουργιών. Σε αντίθεση με το gpt-4, αυτό το μοντέλο δεν θα λαμβάνει ενημερώσεις και θα καταργηθεί 3 μήνες μετά την κυκλοφορία μιας νέας έκδοσης.	8,192 tokens	Έως Σεπτέμβριος 2021
gpt-4-32k	Ίδιες δυνατότητες με το στάνταρ μοντέλο gpt-4 αλλά με τετραπλάσιο μήκος περιεχομένου. Θα ενημερωθεί με την τελευταία έκδοσή του μοντέλου.	32,768 tokens	Έως Σεπτέμβριος 2021
gpt-4-32k-0613	Στιγμιότυπο του gpt-4-32k από 13 Ιουνίου 2023. Σε αντίθεση με το gpt-4-32k, αυτό το μοντέλο δεν θα λαμβάνει ενημερώσεις και θα καταργηθεί 3 μήνες μετά την κυκλοφορία μιας νέας έκδοσης.	32,768 tokens	Έως Σεπτέμβριος 2021

6.3.3 Παράμετροι για την επιλογή του μοντέλου

Ο κύριος παράγοντας που πρέπει να συμπεριληφθεί στις σκέψεις μας είναι τα tokens, γι' αυτό το λόγο θα αναλύσουμε τα παρακάτω ερωτήματα στο παρόν υποκεφάλαιο:

1. **Τι αντιπροσωπεύουν τα tokens;**
 - Τα tokens αποτελούν τις βασικές μονάδες εισόδου σε ένα μοντέλο γλώσσας, και μπορεί να αντιπροσωπεύουν χαρακτήρες, λέξεις, ή ακόμη και μεγαλύτερες ακολουθίες κειμένου, ανάλογα με την κωδικοποίηση της γλώσσας.¹⁶
2. **Τι επιπτώσεις υπάρχουν με την αυξημένη χρήση tokens;**
 - Η αυξημένη χρήση tokens μπορεί να οδηγήσει σε πιο ακριβή αποτελέσματα, αλλά επίσης αυξάνει τον υπολογιστικό χρόνο και τους απαιτούμενους πόρους.
3. **Ποιος είναι ο κατάλληλος αριθμός token για το bot που θέλουμε να υλοποιήσουμε;**
 - Ο κατάλληλος αριθμός tokens εξαρτάται από τις απαιτήσεις της εφαρμογής, τον διαθέσιμο υπολογιστικό πόρο, και το επίπεδο λεπτομέρειας και ακρίβειας που απαιτείται.
4. **Πώς η κατανόηση των tokens επηρεάζει την απόδοση και την ακρίβεια του bot;**
 - Η κατανόηση των tokens είναι κρίσιμη για την αποτελεσματική επεξεργασία των ερωτημάτων και την παραγωγή ακριβών και συνεκτικών απαντήσεων, καθώς οι λανθασμένες αναλύσεις μπορούν να οδηγήσουν σε ανακριβείς ή ακατανόητες απαντήσεις.
5. **Πώς μπορεί η διαχείριση των tokens να βελτιώσει την εμπειρία του χρήστη και την αποτελεσματικότητα του bot;**
 - Οι σωστές τακτικές διαχείρισης tokens μπορούν να μειώσουν τους χρόνους απόκρισης, να βελτιώσουν την ακρίβεια των απαντήσεων, και να διευκολύνουν την κατανόηση των ερωτημάτων και απαντήσεων.
6. **Ποιες είναι οι προκλήσεις που συνδέονται με τη διαχείριση των tokens σε έναν νομικό bot;**
 - Οι προκλήσεις περιλαμβάνουν τη διαχείριση μεγάλου όγκου νομικού κειμένου, την αντιμετώπιση πολύπλοκων ερωτημάτων, και τη διατήρηση της ακρίβειας και συνεκτικότητας στις απαντήσεις.
7. **Πώς η αλλαγή του αριθμού των tokens επηρεάζει το κόστος και τους πόρους που απαιτούνται για τη λειτουργία του bot;**
 - Η αύξηση των tokens μπορεί να αυξήσει το κόστος λειτουργίας λόγω της αυξημένης υπολογιστικής απαίτησης, ενώ η μείωση των tokens μπορεί να μειώσει την ακρίβεια ή την ποιότητα των απαντήσεων.
8. **Πώς η επιλογή του μοντέλου γλώσσας επηρεάζει τον αριθμό των διαθέσιμων tokens και την απόδοση του bot;**
 - Διαφορετικά μοντέλα γλώσσας έχουν διαφορετικούς περιορισμούς και δυνατότητες σε σχέση με τα tokens, το οποίο μπορεί να επηρεάσει την απόδοση, την ακρίβεια, και την εμπειρία του χρήστη.
9. **Υπάρχουν τεχνικές ή πρακτικές για τη βελτιστοποίηση της χρήσης των tokens χωρίς να επηρεάζεται η ποιότητα των απαντήσεων του bot;**
 - Τεχνικές όπως η προ-επεξεργασία του κειμένου, η επιλογή του κατάλληλου μοντέλου γλώσσας, και η βελτιστοποίηση της διαχείρισης των ερωτήσεων και απαντήσεων μπορούν να βοηθήσουν στην βελτιστοποίηση της χρήσης των tokens.
10. **Πώς μπορεί η σωστή κατανομή των tokens να συμβάλει στην καλύτερη κατανόηση των νομικών ερωτημάτων και απαντήσεων;**
 - Μια ορθή κατανομή των tokens μπορεί να βοηθήσει στην αποφυγή των ασαφειών, να διευκολύνει την αναγνώριση των νομικών εννοιών και να βελτιώσει την ακρίβεια και την ποιότητα των απαντήσεων.

¹⁶ <https://pub.towardsai.net/langchain-101-part-2ab-all-you-need-to-know-about-large-language-models-3512ae41dfc3>

Εκτός από τα tokens, υπάρχουν πολλοί άλλοι παράγοντες που πρέπει να ληφθούν υπόψη κατά την επιλογή ενός μοντέλου για την υλοποίηση ενός νομικού bot. Ενδεικτικά, ορισμένοι από αυτούς τους παράγοντες περιλαμβάνουν:

1. **Ακρίβεια και Συνεκτικότητα:** Είναι σημαντικό να επιλεγεί ένα μοντέλο που παρέχει ακριβείς και συνεκτικές απαντήσεις, ειδικά στο πεδίο του νομικού.
2. **Πρόσβαση σε Εξωτερικούς Πόρους:** Η πρόσβαση σε εξωτερικές νομικές βάσεις δεδομένων ή άλλους πόρους μπορεί να είναι κρίσιμη για την παροχή ενημερωμένων και ακριβών απαντήσεων.
3. **Εκπαίδευση και Προσαρμογή:** Η δυνατότητα εκπαίδευσης ή προσαρμογής του μοντέλου στις ειδικές ανάγκες της νομικής εφαρμογής σας.
4. **Υπολογιστική Απαιτήση:** Οι απαιτήσεις σε υπολογιστικούς πόρους και η απόδοση του μοντέλου σε σχέση με τον χρόνο απόκρισης.
5. **Κόστος:** Το κόστος λειτουργίας, συμπεριλαμβανομένων των τελών για την πρόσβαση στο μοντέλο, των υπολογιστικών πόρων, και της αποθήκευσης και διαχείρισης των δεδομένων.
6. **Ασφάλεια και Προστασία Δεδομένων:** Ειδικά στο νομικό πεδίο, η ασφάλεια και η προστασία των δεδομένων είναι κρίσιμης σημασίας.
7. **Υποστήριξη Πολλαπλών Γλωσσών:** Εάν το bot πρέπει να υποστηρίξει πολλαπλές γλώσσες, η υποστήριξη γλώσσας είναι ένας σημαντικός παράγοντας.
8. **Ευκολία Χρήσης και Συντηρητικότητα:** Η ευκολία χρήσης, η δυνατότητα διαχείρισης, και η συντηρητικότητα του μοντέλου.

6.4 Αρχική Ανάπτυξη του Bot

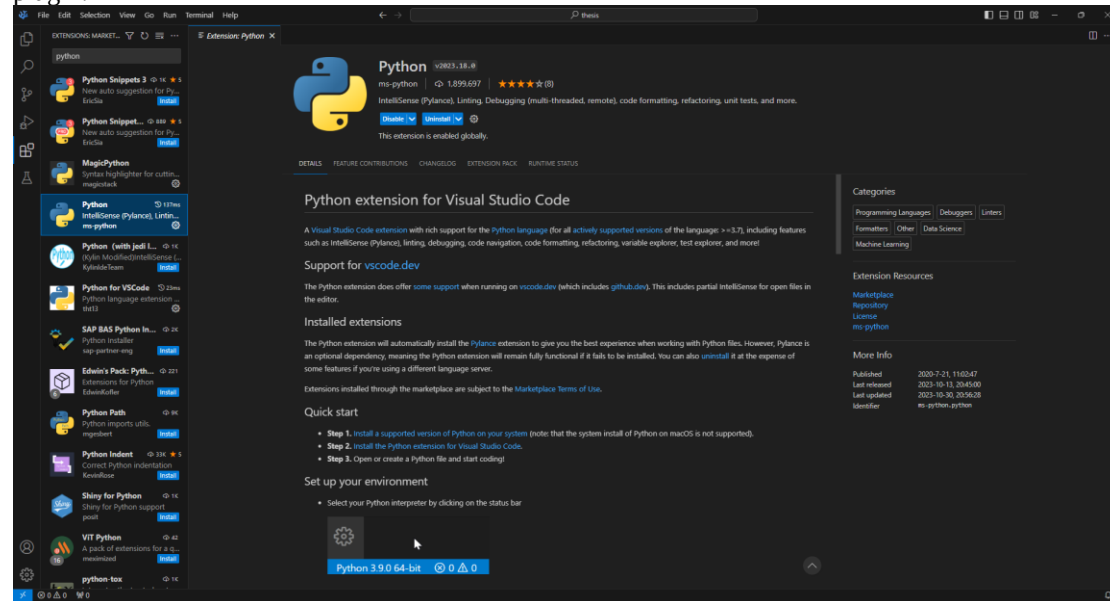
Σε αυτή την ενότητα, πρόκειται να ξεκινήσουμε την υλοποίηση του bot, χρησιμοποιώντας τα εργαλεία VSCodium, Python, pip, καθώς και το OpenAI API, με αφετηρία τους παρακάτω στόχους:

- **Εγκατάσταση Απαραίτητων Βιβλιοθηκών και Plugin:** Είναι απαραίτητο να εγκαταστήσουμε τις αναγκαίες βιβλιοθήκες και τα plugin που θα συμβάλλουν στην υλοποίηση του bot, εξασφαλίζοντας την ομαλή λειτουργία και την επικοινωνία με το OpenAI API.
- **Ενσωμάτωση του OpenAI API:** Η ενσωμάτωση του OpenAI API είναι κρίσιμης σημασίας για την ανάπτυξη του bot, καθώς αυτό θα παρέχει την ικανότητα προώθησης ερωτημάτων και λήψης απαντήσεων βάσει των νομικών κειμένων και παραμέτρων που θα παρασχεθούν.
- **Ενσωμάτωση του Κλειδιού του API:** Η ενσωμάτωση του κλειδιού του API είναι ζωτικής σημασίας για την εξασφάλιση της ασφαλούς και εξουσιοδοτημένης πρόσβασης στο OpenAI API, επιτρέποντας την ομαλή αλληλεπίδραση μεταξύ της εφαρμογής μας και του API.
- **Πρώτο Στάδιο Υλοποίησης για Αλληλεπίδραση Χρήστη – Bot:** Στο πρώτο στάδιο της υλοποίησης, πρόκειται να αναπτύξουμε τις βασικές λειτουργίες για την αλληλεπίδραση μεταξύ του χρήστη και του bot, επιτρέποντας την υποβολή ερωτημάτων και την λήψη απαντήσεων με βάση τα εισαγόμενα νομικά κείμενα.
- **Ενσωμάτωση Βέλτιστων Ερεθισμάτων (Prompts):** Θα πραγματοποιήσουμε την ενσωμάτωση των οδηγιών που δόθηκαν στο ChatGPT κατά τη διάρκεια της ανάλυσης που πραγματοποιήθηκε στο κεφάλαιο 7.1. Η ενσωμάτωση αυτών των βέλτιστων ερεθισμάτων είναι σημαντική για την επίτευξη της επιθυμητής απόδοσης του bot και τη βελτίωση της ακρίβειας των απαντήσεων που παρέχει. Μέσω αυτής της διαδικασίας, στοχεύουμε στην καλύτερη κατανόηση και επεξεργασία των νομικών ερωτημάτων που τίθενται από τους χρήστες, διασφαλίζοντας μια αποτελεσματική και ακριβής αλληλεπίδραση μεταξύ του bot και των χρηστών.

Με την εκτέλεση αυτών των βημάτων, θα έχουμε τεθεί τα θεμέλια για την περαιτέρω ανάπτυξη και βελτίωση του bot, στοχεύοντας στην παροχή ακριβών και συνεπών απαντήσεων σε νομικά ερωτήματα με βάση την ελληνική νομοθεσία.

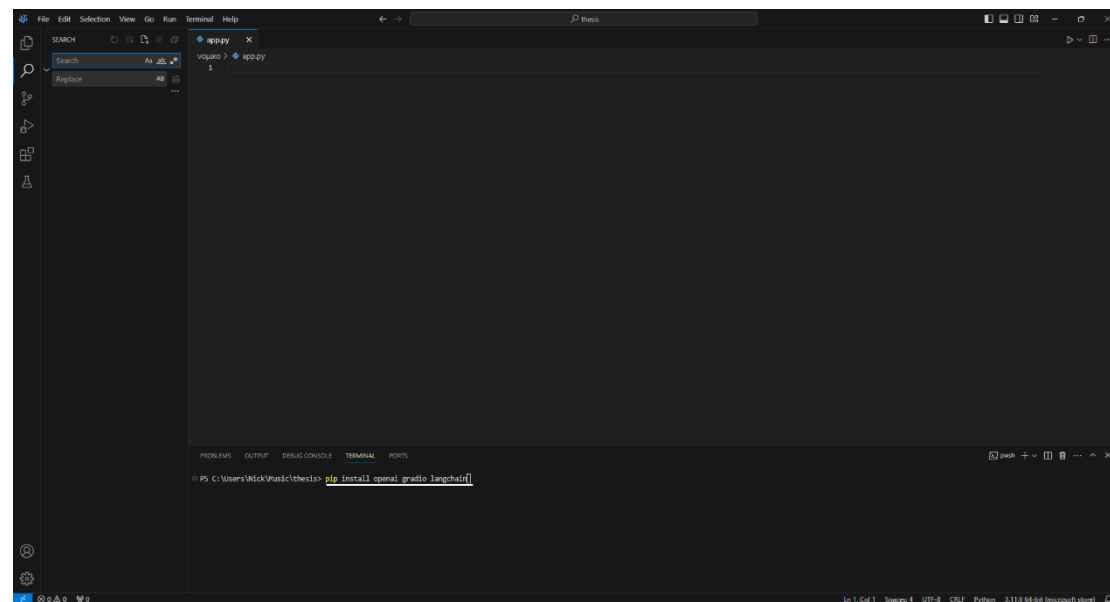
6.4.1 Εγκατάσταση Απαραίτητων Βιβλιοθηκών και Plugin

Σε αυτή την ενότητα, θα αρχίσουμε προετοιμάζοντας το περιβάλλον για την ανάπτυξη του bot, καταρχάς με την προσθήκη του plugin της Python. Για να το πράξουμε αυτό, θα επιλέξουμε την επιλογή εύρεσης plugin από την αριστερή πλευρά της οθόνης και θα εγκαταστήσουμε το Python plugin.



Εικόνα 18. Εγκατάσταση python plugin στο VSCodium

Στη συνέχεια, θα προχωρήσουμε στην εγκατάσταση των απαραίτητων βιβλιοθηκών που θα χρειαστούν κατά τη διάρκεια της ανάπτυξης του bot, με τη βοήθεια του εργαλείου `pip`. Η εντολή που θα χρησιμοποιηθεί στον τερματικό στο κάτω μέρος του VSCodium για αυτό τον σκοπό είναι: `pip install openai gradio langchain`. Αυτό το βήμα είναι κρίσιμο για την επιτυχή υλοποίηση του bot, καθώς οι βιβλιοθήκες αυτές θα παρέχουν τα απαραίτητα εργαλεία και τις λειτουργίες για την ανάπτυξη και την αλληλεπίδραση με το bot.



Εικόνα 19. Εγκατάσταση απαραίτητων βιβλιοθηκών

6.4.2 Ενσωμάτωση του OpenAI API & του Κλειδιού του API:

Στην υποενότητα 7.4.2, πραγματοποιείται η ενσωμάτωση του OpenAI API, ενός κρίσιμου σταδίου για την υλοποίηση του bot. Η διαδικασία περιλαμβάνει τα εξής βήματα:

1. Δημιουργία Λογαριασμού στο OpenAI:

Σε περίπτωση που δεν υπάρχει προηγούμενος λογαριασμός, απαιτείται η δημιουργία ενός νέου λογαριασμού στον ιστότοπο του OpenAI. Παρακάτω εμφανίζεται εικόνα με την οθόνη εγγραφής.



Create your account

Note that phone verification may be required for signup. Your number will only be used to verify your identity for security purposes.

Continue

Already have an account? [Log in](#)

OR



Continue with Google



Continue with Microsoft Account



Continue with Apple

[Terms of use](#) | [Privacy policy](#)

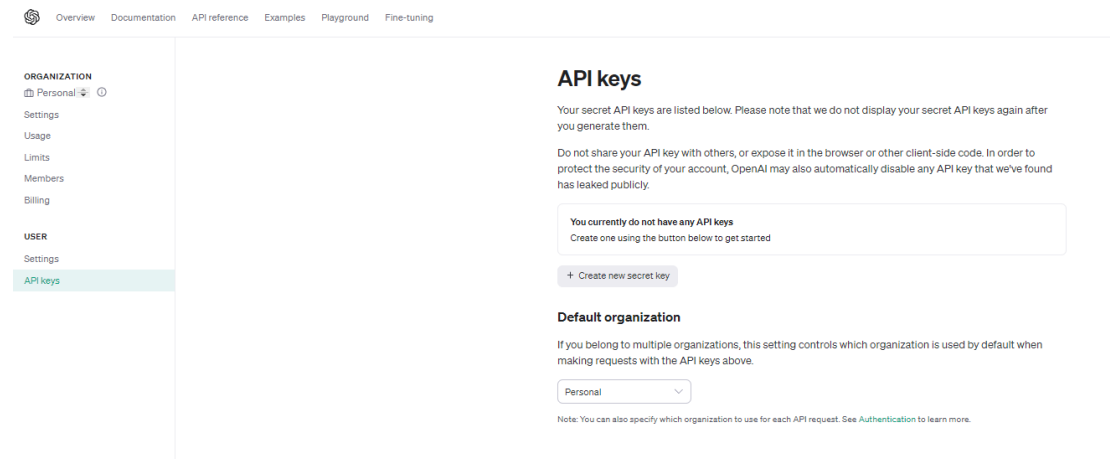
Εικόνα 20. Δημιουργία λογαριασμού OpenAI

2. Πρόσβαση στο OpenAI API:

- Με την ολοκλήρωση της δημιουργίας λογαριασμού, ακολουθείται η διαδικασία πρόσβασης στο OpenAI API. Εισάγονται οι λεπτομέρειες του λογαριασμού και ακολουθούνται οι οδηγίες πρόσβασης στο API.

3. Λήψη του API Κλειδιού:

- Επιβεβαιωμένη η πρόσβαση, λαμβάνεται το API κλειδί που θα χρησιμοποιηθεί για την αυθεντικοποίηση των αιτήσεων προς το OpenAI API. Παρακάτω φαίνεται η οθόνη δημιουργίας κλειδιού.



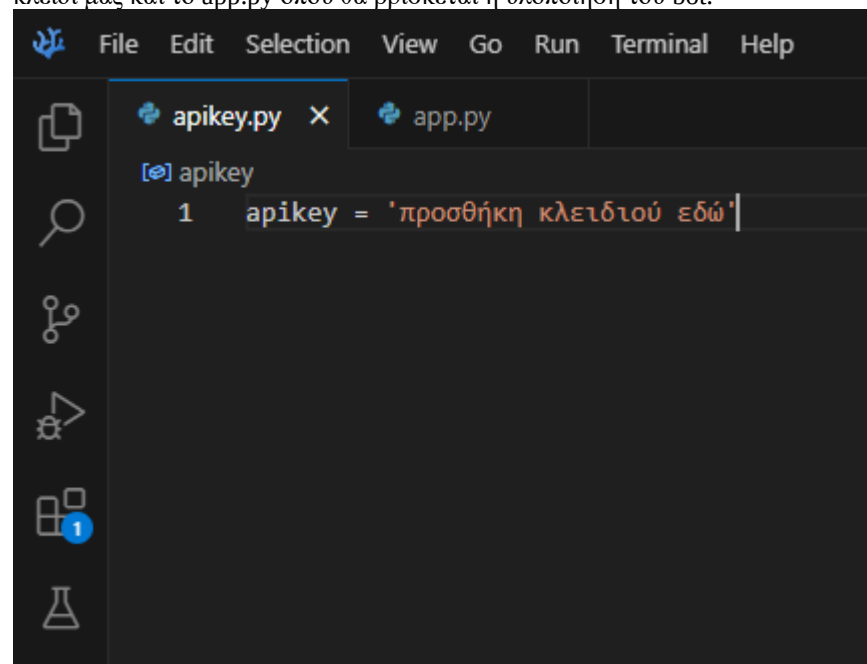
Εικόνα 21. Δημιουργία κλειδιού OpenAI

4. Εγκατάσταση της Βιβλιοθήκης OpenAI:

Εγκαθίσταται η βιβλιοθήκη OpenAI στο περιβάλλον ανάπτυξης του Python μέσω της εντολής `pip install openai`, όπου έχει γίνει στο κεφάλαιο 7.4.1.

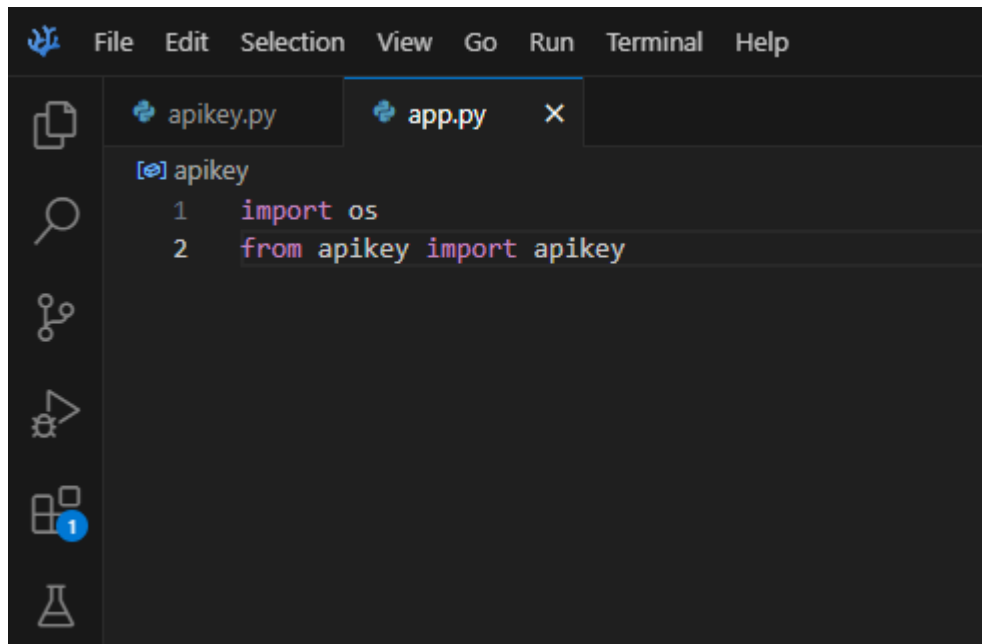
5. Ενσωμάτωση του API Κλειδιού στον Κώδικα:

Ενσωματώνεται το API κλειδί στον κώδικα. Για την εγκατάσταση του OpenAI κλειδιού θα δημιουργήσουμε 2 αρχεία στο περιβάλλον ανάπτυξης μας, το `apikey.py` όπου θα συμπεριλαμβάνει το κλειδί μας και το `app.py` όπου θα βρίσκεται η υλοποίηση του bot.



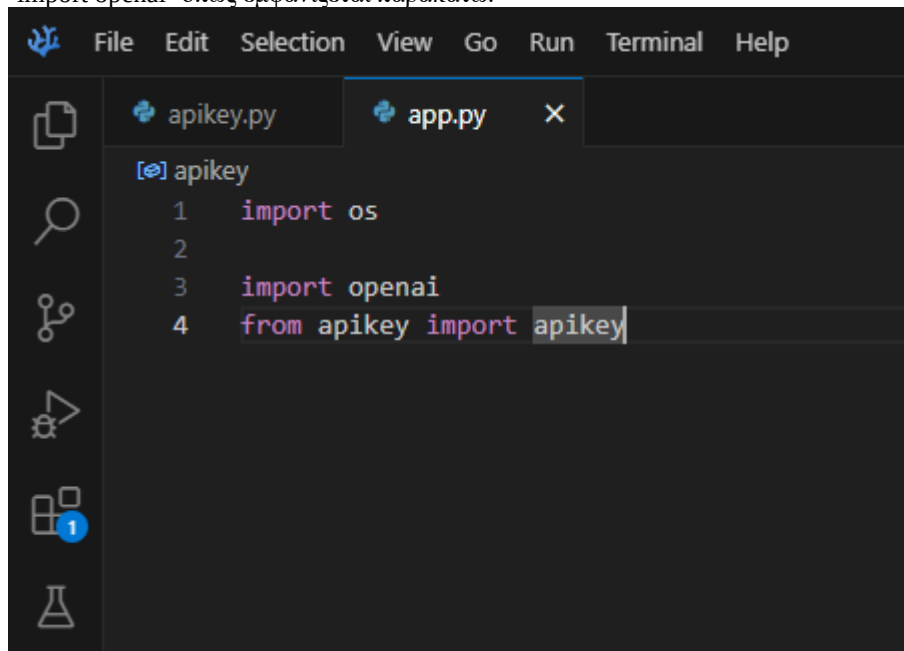
Εικόνα 22. `apikey.py` αρχείο με το κλειδί

Για την προσθήκη του κλειδιού στο app.py θα γράψουμε στην αρχή του προγράμματος ‘from apikey import apikey’ όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 23. Προσθήκη κλειδιού στο πρόγραμμα

Και το τελικό στάδιο είναι να προσθέσουμε την βιβλιοθήκη της python για το OpenAI γράφοντας ‘import openai’ όπως εμφανίζεται παρακάτω.



6.4.3 Πρώτο Στάδιο Υλοποίησης για Αλληλεπίδραση Χρήστη – Bot

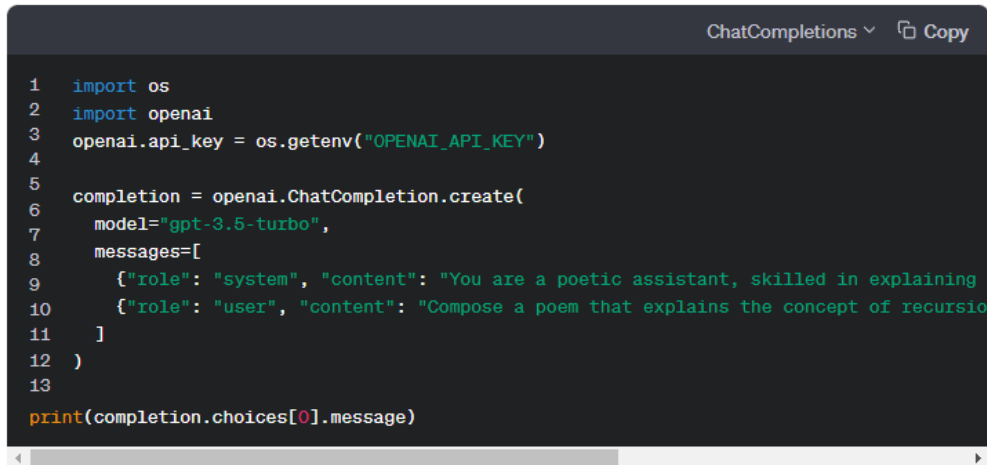
Ακολουθώντας τον quickstart οδηγό στην ιστοσελίδα του OpenAI μπορούμε να κάνουμε την πρώτη μας ερώτηση αλλά και να δώσουμε τις οδηγίες μας στο bot που αναπτύσσουμε. Ο οδηγός εμφανίζεται ως εξής:

Step 3: Sending your first API request

✓ Making an API request

After you have Python configured and an API key setup, the final step is to send a request to the OpenAI API using the Python library. To do this, create a file named `openai-test.py` using the terminal or an IDE.

Inside the file, copy and paste one of the examples below:



```

1  import os
2  import openai
3  openai.api_key = os.getenv("OPENAI_API_KEY")
4
5  completion = openai.ChatCompletion.create(
6      model="gpt-3.5-turbo",
7      messages=[
8          {"role": "system", "content": "You are a poetic assistant, skilled in explaining"},
9          {"role": "user", "content": "Compose a poem that explains the concept of recursion"}
10     ]
11 )
12
13 print(completion.choices[0].message)

```

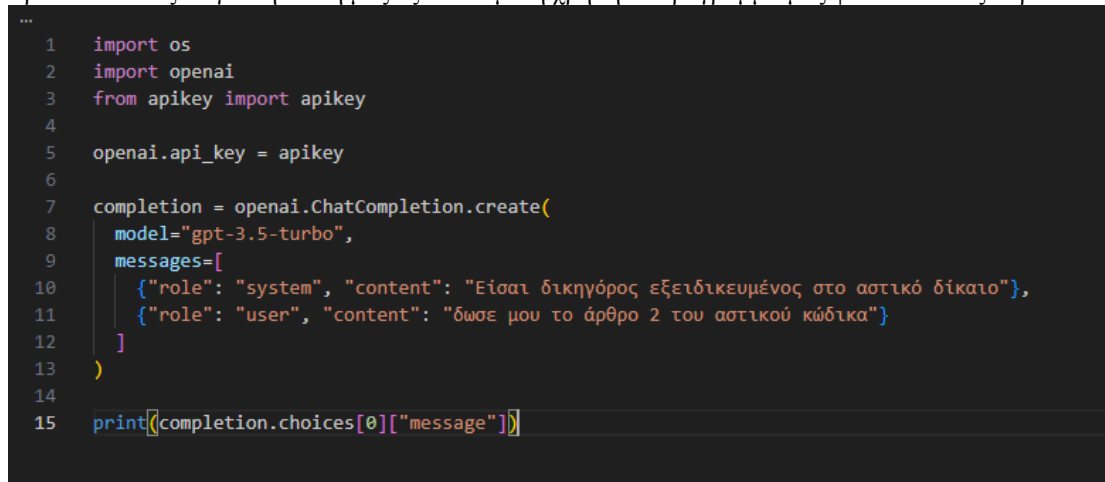
To run the code, enter `python openai-test.py` into the terminal / command line.

The **Chat Completions** example highlights just one area of strength for our models: creative ability. Explaining recursion (the programming topic) in a well formatted poem is something both the best developers and best poets would struggle with. In this case, `gpt-3.5-turbo` does it effortlessly.

Εικόνα 24. Quickstart guide του OpenAI

Έπειτα τροποποιείται το παραπάνω πρόγραμμα για να εξυπηρετήσει τον δικό μας σκοπό αλλάζοντας την ερώτηση και δίνοντας ένα σταθερό ρόλο στο bot όπου στην συνέχεια θα αλλάξουμε στη συνέχεια για να είναι δυναμική η

Τροποποιώντας το για την δική μας εξειδικευμένη χρήση το πρόγραμμα μας φαίνεται όπως παρακάτω:



```

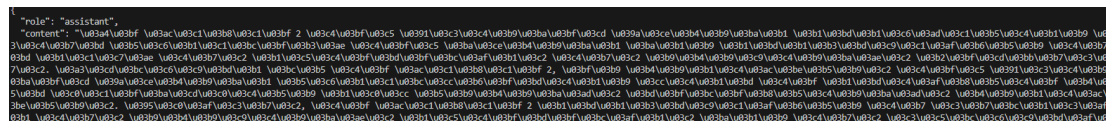
...
1  import os
2  import openai
3  from apikey import apikey
4
5  openai.api_key = apikey
6
7  completion = openai.ChatCompletion.create(
8      model="gpt-3.5-turbo",
9      messages=[
10         {"role": "system", "content": "Είσαι δικηγόρος εξειδικευμένος στο αστικό δίκαιο"},
11         {"role": "user", "content": "δωσε μου το άρθρο 2 του αστικού κώδικα"}
12     ]
13 )
14
15 print(completion.choices[0]["message"])

```

Εικόνα 25. Αρχική υλοποίηση ακολουθώντας τον οδηγό

Τρέχοντας το πρόγραμμα χρησιμοποιώντας το τερματικό έχουμε την πρώτη μας απάντηση μέσω του API. Δυστυχώς η απάντηση είναι σε διαφορετικό encoding που δεν υποστηρίζεται άμεσα από το

τερματικό. Επίσης στο παρόν υποκεφάλαιο χρησιμοποιείται το μοντέλο *gpt-3.5-turbo* λόγω μικρότερου κόστους και ταχύτητας, εφόσον η υλοποίηση βρίσκεται σε δοκιμαστικό στάδιο.

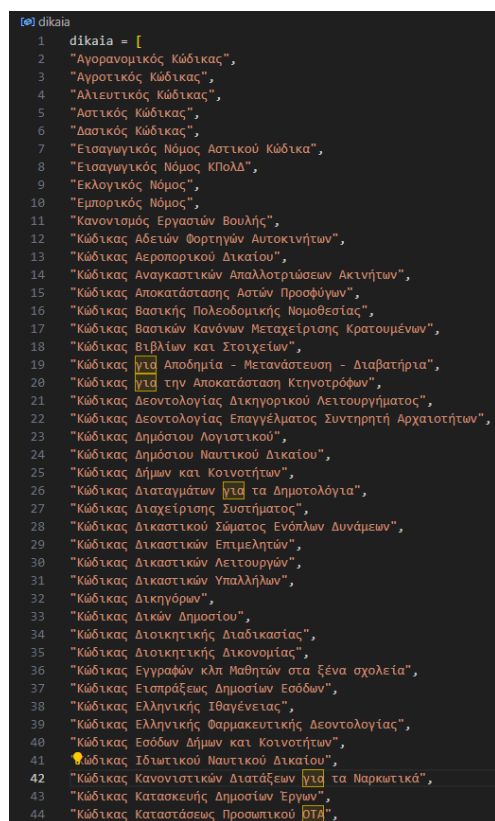


Εικόνα 26. Απάντηση του bot μέσω API

Για την λύση του προηγούμενου προβλήματος θα αλλάξουμε την γραμμή 15 από `'print(completion["choices"][0]["message"])` σε `'print(completion["choices"][0]["message"]["content"])` ώστε να επιλεχθεί μόνο η απάντηση του bot και να εμφανιστεί κανονικά.

6.4.4 Ενσωμάτωση Βέλτιστων Ερεθισμάτων (Prompts)

Σε αυτό το υποκεφάλαιο θα εξελίξουμε την υλοποίηση προσθέτοντας επιλογή στον χρήστη για το νομικό κώδικα που ενδιαφέρεται. Ξεκινώντας με την δημιουργία ενός καινούργιου αρχείου με το όνομα `'dikaia.py'` δημιουργείται μια λίστα με όλους τους κώδικες όπως εμφανίζεται παρακάτω:



Εικόνα 27. Λίστα δικαίων

Στη συνέχεια προστίθεται στο πρόγραμμα μας και ξεκινάει η δημιουργία της επιλογής του χρήστη. Καταρχάς εμφανίζουμε την λίστα με την εκκίνηση του προγράμματος με τον παρακάτω τρόπο:

```
1. #Εμφάνιση λίστας
2. for kwdikas in dikaia:
3.     print(enumerator, "." + kwdikas + "\n")
4.     enumerator = enumerator + 1
5.
```

Εικόνα 28. Εμφάνιση της αριθμημένης λίστας

Τρέχοντας το πρόγραμμα εμφανίζεται το παρακάτω αποτέλεσμα:

```

1 .Αγορανομικός Κώδικας
2 .Αγροτικός Κώδικας
3 .Αλιευτικός Κώδικας
4 .Αστικός Κώδικας
5 .Δασικός Κώδικας
6 .Εισαγωγικός Νόμος Αστικού Κώδικα
7 .Εισαγωγικός Νόμος ΚΠολΔ
8 .Εκλογικός Νόμος
9 .Εμπορικός Νόμος
10 .Κανονισμός Εργασιών Βουλής
    
```

Εικόνα 29. Εμφάνιση λίστας κατά την εκκίνηση

Έπειτα με αυτό το παρακάτω κομμάτι του προγράμματος επιλέγει ο χρήστης τον κώδικα που τον ενδιαφέρει.

```

1.     user_input = 100
2.
3.     #επιλογή κώδικα από τον χρήστη
4.     while(user_input < 1 or user_input > 80):
5.         print("\n")
6.         user_input = int(input("Επέλεξε κώδικα γράφοντας το αντίστοιχο νούμερο:
7.         "))
    
```

Τέλος θα προσθέσουμε την δυνατότητα να ρωτάει ο χρήστης με είσοδο από το πληκτρολόγιο και να στέλνεται η ερώτηση μαζί με της οδηγίες.

```

1.     #ερώτηση χρήστη
2.     user_question = input("Κάνε την ερώτηση σου: ")
3.
4.     completion = openai.ChatCompletion.create(
5.         model="gpt-3.5-turbo",
6.         messages=[
7.             {"role": "system", "content": "Είσαι δικηγόρος εξειδικευμένος στο " +
8.             dikaia[user_input - 1]},
9.             {"role": "user", "content": "δωσε μου το άρθρο 2 του αστικού κώδικα"}
10.        ]
11.    )
12.    print(completion["choices"][0]["message"]["content"].strip())
    
```

Επίσης προσθέτοντας στην γραμμή 12 το ["content"] λύνουμε το πρόβλημα που είχαμε στην αρχή και η απάντηση εμφανίζεται κανονικά. Το πρόγραμμα ολοκληρωμένο και οθόνη εκτέλεσης φαίνονται παρακάτω:

```

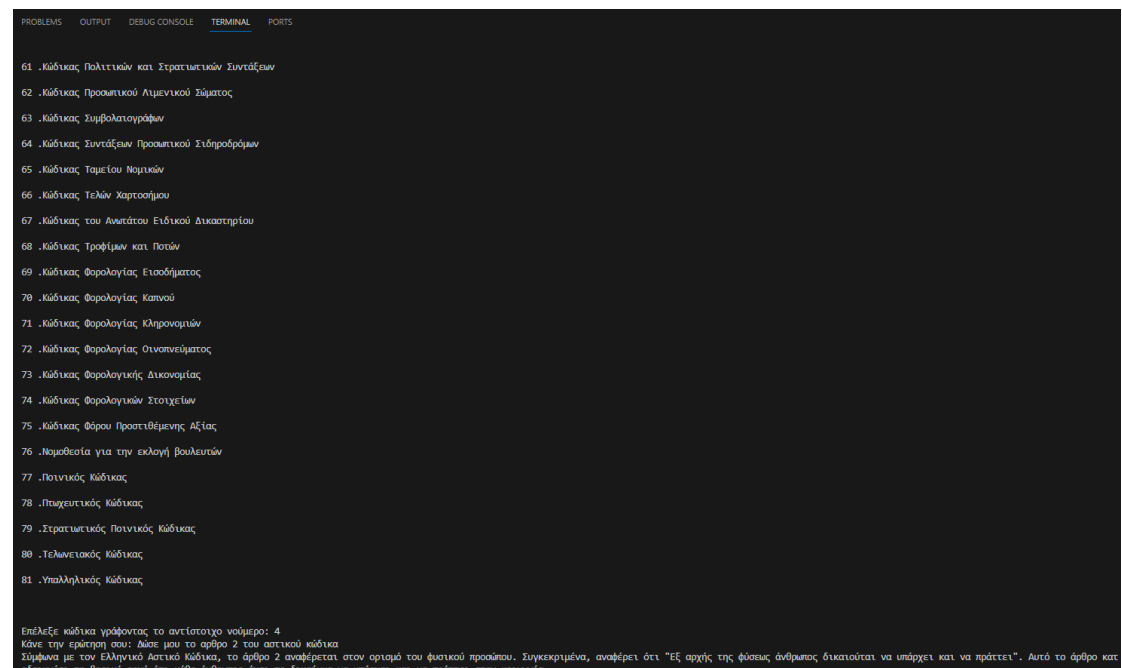
1. import os
2. import openai
3. from apikey import apikey
4. from dikaia import dikaia
5.
6. #αρχικοποίηση κλειδιού
7. openai.api_key = apikey
8.
    
```



```

9. def main():
10.     enumerator = 1
11.
12.     #Εμφάνιση λίστας
13.     for kwdikas in dikaia:
14.         print(enumerator, "." + kwdikas + "\n")
15.         enumerator = enumerator + 1
16.
17.     user_input = 100
18.
19.     #επιλογή κώδικα από τον χρήστη
20.     while(user_input < 1 or user_input > 80):
21.         print("\n")
22.         user_input = int(input("Επέλεξε κώδικα γράφοντας το αντίστοιχο νούμερο:
23.     "))
24.
25.     #ερώτηση χρήστη
26.     user_question = input("Κάνε την ερώτηση σου: ")
27.
28.     completion = openai.ChatCompletion.create(
29.         model="gpt-3.5-turbo",
30.         messages=[
31.             {"role": "system", "content": "Είσαι δικηγόρος εξειδικευμένος στο " +
32.             dikaia[user_input - 1]},
33.             {"role": "user", "content": user_question}
34.         ]
35.     )
36.
37.     print(completion["choices"][0]["message"]["content"].strip())
38.
39. if __name__ == "__main__":
40.     main()

```



Εικόνα 30. Εκτέλεση προγράμματος στην αρχική του υλοποίηση

Εν κατακλείδι σε αυτή την ενότητα υλοποιήθηκε:

- Εισαγωγή κλειδιού του API
- Αλληλεπίδραση προγράμματος με OpenAI API
- Εισαγωγή βέλτιστων ερεθισμάτων (prompts) με δυναμική επιλογή κώδικα
- Αλληλεπίδραση χρήστη – bot

6.5 Ανάπτυξη Διεπαφής Χρήστη

Κατασκευάζοντας τη γραφική διεπαφή του bot, το επόμενο βήμα αφορά την ανάπτυξη ενός φιλικού προς τον χρήστη περιβάλλοντος, όπου οι λειτουργίες που υλοποιήθηκαν στις προηγούμενες ενότητες θα είναι προσβάσιμες μέσω μιας οπτικής διεπαφής. Με την υποστήριξη του Gradio API, η διεπαφή θα σχεδιαστεί για να διευκολύνει την αλληλεπίδραση με το bot, επιτρέποντας στους χρήστες να εκτελούν ερωτήματα και να λαμβάνουν απαντήσεις με έναν απλοποιημένο τρόπο. Αρχικά για να χρησιμοποιήσουμε το Gradio API πρέπει να γίνουν κάποιες αλλαγές στο πρόγραμμα μας.

1. Η παραγωγή της απάντησης πρέπει να γίνεται σε ξεχωριστή συνάρτηση
2. Υλοποίηση της Gradio διεπαφής με λίστα για επιλογή κώδικα και κουτί για τεκμηρίωση της ερώτησης
3. Προσθήκη των παραμέτρων που παράγονται από τις επιλογές του χρήστη στην συνάρτηση του grt για την παραγωγή της απάντησης
4. Δοκιμή της γραφικής διεπαφής Gradio

6.5.1 Υλοποίηση συνάρτησης παραγωγής απάντησης

Εστιάζοντας στη βελτίωση της αναγνωσιμότητας του κώδικα και προετοιμάζοντας τη βάση για την ενσωμάτωση της γραφικής διεπαφής, η δομή του προγράμματος υφίσταται αναδιάρθρωση. Αυτό συνεπάγεται την ανασχεδίαση της αρχιτεκτονικής των συναρτήσεων, ώστε να εξυπηρετούνται οι απαιτήσεις του εργαλείου δημιουργίας της διεπαφής, το οποίο απαιτεί συγκεκριμένες συναρτήσεις ως παραμέτρους για τη λειτουργικότητά του. Στην ουσία θα μετατραπεί το κομμάτι που χρησιμοποιήθηκε στην ενότητα 7.4.3 όπως ακολουθεί:

```

19.     #επιλογή κώδικα από τον χρήστη
20.     while(user_input < 1 or user_input > 80):
21.         print("\n")
22.         user_input = int(input("Επέλεξε κώδικα γράφοντας το αντίστοιχο νούμερο:
23.     "))
24.     #ερώτηση χρήστη
25.     user_question = input("Κάνε την ερώτηση σου: ")
26.
27.     completion = openai.ChatCompletion.create(
28.         model="gpt-3.5-turbo",
29.         messages=[
30.             {"role": "system", "content": "Είσαι δικηγόρος εξειδικευμένος στο " +
31. dikaia[user_input - 1]},
32.             {"role": "user", "content": user_question}
33.         ]
34.     )

```

Παραπάνω βλέπουμε την προηγούμενη υλοποίηση και ακολουθούν οι αλλαγές:

```

1. def ask_question_with_instruction(instruction, question):
2.     instruction = "Είσαι δικηγόρος εξειδικευμένος στο " + instruction
3.     # Πρόσθεση ερώτησης με το δίκαιο
4.     prompt = f"{instruction}\n{question}"
5.     # Παραγωγή απάντησης με το GPT
6.     response = openai.ChatCompletion.create(
7.         model = "gpt-3.5-turbo",
8.         messages = [{"role": "user", "content": prompt}],
9.         max_tokens = 1024
10.    )
11.
12.    return response["choices"][0]["message"]["content"].strip()
13.
14.

```

Παρατηρείται πως το κομμάτι παραγωγής της απάντησης έχει μεταφερθεί σε ξεχωριστή συνάρτηση. Επίσης στην γραμμή 4 βλέπουμε τις παραμέτρους {instruction} και {question} να βρίσκονται μέσα σε

εισαγωγικά, ο λόγος είναι πως για να επιλεγθούν τα αποτελέσματα από την επιλογή του χρήστη από την γραφική διεπαφή οι οδηγίες (Gradio, 2023)¹⁷ μας παρουσιάζουν τον παραπάνω τρόπο.

6.5.2 Ανάπτυξη γραφικής διεπαφής

Σε αυτό το υποκεφάλαιο θα υλοποιηθεί η γραφική διεπαφή με τη βοήθεια του Gradio¹⁸ με τα παρακάτω βήματα:

1. Ανάπτυξη διεπαφής καλώντας συνάρτηση του Gradio
2. Ανάπτυξη του dropdown menu για την επιλογή του νομικού κώδικα
3. Ανάπτυξη της φόρμας εισαγωγής της ερώτησης
4. Τελικό αποτέλεσμα και δοκιμή της διεπαφής

Για την υλοποίηση της διεπαφής θα διατρέξουμε στο documentation του Gradio για την σωστή χρήση της συνάρτησης της διεπαφής. Ακολουθεί κομμάτι από την ιστοσελίδα¹⁹:

Παράμετρος	Περιγραφή
fn	Απαιτείται. Η συνάρτηση για την οποία θα δημιουργηθεί η διεπαφή. Κάθε παράμετρος της συνάρτησης αντιστοιχίζεται σε ένα είσοδο, και η συνάρτηση πρέπει να επιστρέφει μία τιμή ή μία πλειάδα τιμών, κάθε μία αντιστοιχίζεται σε ένα στοιχείο εξόδου.
inputs	Απαιτείται. Ένα στοιχείο εισόδου Gradio, ή λίστα στοιχείων. Μπορεί να δοθούν ως αντικείμενα ή μέσω των συντομεύσεων τους ως συμβολοσειρές. Ο αριθμός των στοιχείων εισόδου πρέπει να ταιριάζει με τον αριθμό των παραμέτρων στη συνάρτηση fn.
outputs	Απαιτείται. Ένα στοιχείο εξόδου Gradio, ή λίστα στοιχείων. Μπορεί να δοθούν ως αντικείμενα ή μέσω των συντομεύσεων τους ως συμβολοσειρές. Ο αριθμός των στοιχείων εξόδου πρέπει να ταιριάζει με τον αριθμό των τιμών που επιστρέφει η συνάρτηση fn.
examples	Προεπιλογή: Καμία. Δείγματα εισόδου για τη συνάρτηση· αν παρέχονται, εμφανίζονται κάτω από τα στοιχεία της διεπαφής και μπορούν να κλικαριστούν για να πληρωθούν οι εισοδοί. Πρέπει να είναι σε μορφή εμφωλευμένης λίστας, όπου η εξωτερική λίστα αποτελείται από δείγματα και κάθε εσωτερική λίστα από μία είσοδο ανά στοιχείο εισόδου. Μπορεί επίσης να δοθεί ένας συνδετήρας σε κατάλογο με δείγματα, αλλά πρέπει να βρίσκεται στον κατάλογο με το Python αρχείο που τρέχει την εφαρμογή gradio. Αν υπάρχουν πολλαπλές εισοδοί και δίνεται ένας κατάλογος, πρέπει να υπάρχει ένα αρχείο log.csv στον κατάλογο για να συνδέσει τις αντίστοιχες εισόδους.
Cache_examples	Προεπιλογή: Καμία. Αν ισχύει η τιμή True, τα δείγματα αποθηκεύονται στον διακομιστή για γρήγορη εκτέλεση στα δείγματα. Αν fn είναι μια συνάρτηση γεννήτορας, τότε η τελευταία παραγόμενη τιμή θα χρησιμοποιηθεί ως έξοδος.

¹⁷ Gradio documentation, <https://www.gradio.app/docs/interface>

¹⁸ Ιστοσελίδα gradio, <https://www.gradio.app/>

¹⁹ Gradio documentation, <https://www.gradio.app/docs/interface>

	Η προεπιλογή στους Χώρους της HuggingFace είναι True. Η προεπιλογή αλλού είναι False.
examples_per_page	Προεπιλογή: 10. Αν παρέχονται δείγματα, πόσα να εμφανίζονται ανά σελίδα.
live	Προεπιλογή: Ψευδές. Εάν η διεπαφή πρέπει να επανατρέχει αυτόματα αν κάποια από τις εισόδους αλλάξει.
title	Προεπιλογή: Καμία. Ένας τίτλος για τη διεπαφή· αν παρέχεται, εμφανίζεται πάνω από τα στοιχεία εισόδου και εξόδου με μεγάλα γράμματα. Χρησιμοποιείται επίσης ως τίτλος της καρτέλας όταν ανοίγει σε παράθυρο περιηγητή.
description	Προεπιλογή: Καμία. Μια περιγραφή για τη διεπαφή· αν παρέχεται, εμφανίζεται πάνω από τα στοιχεία εισόδου και εξόδου και κάτω από τον τίτλο με κανονικά γράμματα.

(Gradio, 2023)

Ο παραπάνω πίνακας έχει αντιγραφεί από την ιστοσελίδα του Gradio με την δεύτερη στήλη μεταφρασμένη στα ελληνικά. Περιέχει μόνο τα στοιχεία που τέθηκαν απαραίτητα για την δική μας υλοποίηση. Παρακάτω ακολουθεί ο πίνακας για την συνάρτηση Dropdown menu:

Παράμετρος	Περιγραφή
choices <i>list[str int float tuple[str, str int float]] None</i> default: None	Μια λίστα με επιλογές συμβολοσειρών για να επιλέξετε. Μία επιλογή μπορεί επίσης να είναι ένα ζεύγος της μορφής (όνομα, αξία), όπου το όνομα είναι το εμφανιζόμενο όνομα της επιλογής στο αναπτυσσόμενο μενού και η αξία είναι η τιμή που θα περαστεί στη συνάρτηση ή θα επιστραφεί από τη συνάρτηση.
value <i>str int float list[str int float] Callable None</i> default: None	Η προεπιλεγμένη τιμή(ές) που επιλέγεται στο αναπτυσσόμενο μενού. Εάν καμία, καμία τιμή δεν ορίζεται από προεπιλογή. Εάν είναι εφικτό, η συνάρτηση θα κληθεί όποτε η εφαρμογή φορτώνει για να ορίσει την αρχική τιμή του συστατικού.
type <i>Literal[('value', 'index')]</i> default: "value"	Ο τύπος της τιμής που θα επιστρέφεται από το συστατικό. "value" επιστρέφει τη συμβολοσειρά της επιλεγμένης επιλογής, "index" επιστρέφει τον δείκτη της επιλεγμένης επιλογής.
multiselect <i>bool None</i> default: None	Εάν είναι αληθές, μπορούν να επιλεγούν πολλαπλές επιλογές.
allow_custom_value <i>bool</i> default: False	Εάν είναι αληθές, επιτρέπει στο χρήστη να εισάγει μια προσαρμοσμένη τιμή που δεν βρίσκεται στη λίστα με τις επιλογές.
max_choices <i>int None</i> default: None	Ο μέγιστος αριθμός επιλογών που μπορούν να επιλεγούν. Αν δεν υπάρχει, δεν επιβάλλεται κανένα όριο.
filterable <i>bool</i> default: True	Αν είναι αληθές, ο χρήστης θα είναι σε θέση να πληκτρολογήσει μέσα στο αναπτυσσόμενο μενού και να φιλτράρει τις επιλογές πληκτρολογώντας.
label <i>str None</i> default: None	Η ετικέτα για αυτό το συστατικό. Εμφανίζεται πάνω από το συστατικό και χρησιμοποιείται επίσης ως η επικεφαλίδα εάν υπάρχουν πίνακες

	παραδειγμάτων για αυτό το συστατικό. Αν δεν υπάρχει και χρησιμοποιείται σε ένα gr.Interface , η ετικέτα θα είναι το όνομα της παραμέτρου που αντιστοιχεί σε αυτό το συστατικό.
--	---

20

Όπως προηγουμένως αυτή η λίστα δεν είναι ολοκληρωμένη αλλά μόνο τα στοιχεία που μας ενδιαφέρουν. Τελικά ακολουθεί ο πίνακας για το textbox.

Παράμετρος	Περιγραφή
value str Callable None default: ""	Προεπιλεγμένο κείμενο για παροχή στο textarea. Αν είναι callable, η συνάρτηση θα κληθεί κάθε φορά που φορτώνει η εφαρμογή για να ορίσει την αρχική τιμή του συστατικού.
lines int default: 1	Ελάχιστος αριθμός γραμμών που πρέπει να παρέχεται στο textarea.
max_lines int default: 20	Μέγιστος αριθμός γραμμών που πρέπει να παρέχεται στο textarea.
placeholder str None default: None	Υπόδειξη placeholder για παροχή πίσω από το textarea.
label str None default: None	Η ετικέτα για αυτό το συστατικό. Εμφανίζεται πάνω από το συστατικό και χρησιμοποιείται επίσης ως η κεφαλίδα αν υπάρχουν πίνακες παραδειγμάτων για αυτό το συστατικό. Αν δεν υπάρχει και χρησιμοποιείται σε μια gradio.Interface , η ετικέτα θα είναι το όνομα της παραμέτρου που ανατίθεται σε αυτό το συστατικό.
info str None default: None	Πρόσθετη περιγραφή του συστατικού.

21

Θα ακολουθήσει το τμήμα κώδικα για την υλοποίηση της γραφικής διεπαφής:

```

1. def main():
2.     iface = gradio.Interface(
3.         ask_question_with_instruction,
4.         [
5.             #οι παράμετροι που θα δωθούν
6.             gradio.Dropdown(choices = dikaia, label = "Διάλεξε ένα δίκαιο"),
7.             gradio.Textbox(lines = 2, label = "Κάνε την ερώτηση σου εδώ:"),
8.             #τι επιστρέφει
9.             "text",
10.            title = "GPT-based Chatbot για την ελληνική νομοθεσία",
11.            description = "Διάλεξε ένα δίκαιο από την λίστα και κάνε την ερώτηση σου",
12.        )
13.     iface.launch()
14.

```

Αναλύοντας το παραπάνω πρόγραμμα παρατηρείται πως:

- Στην γραμμή 2 υπάρχει η συνάρτηση(fn)

²⁰ <https://www.gradio.app/docs/dropdown#initialization>

²¹ <https://www.gradio.app/docs/textbox#initialization>

- Στις γραμμές 4 – 7 υπάρχουν οι παράμετροι εισόδου
 - Στις γραμμές 10 – 11 ο τίτλος και η περιγραφή
- Εκτελώντας το πρόγραμμα εμφανίζεται στον τερματικό το εξής μήνυμα:

```
Running on local URL:  http://127.0.0.1:7860

To create a public link, set `share=True` in `launch()`.
█
```

Εικόνα 31. Εκτέλεση προγράμματος με την γραφική διεπαφή

Ακολουθώντας το σύνδεσμο στο αντίστοιχο web browser, στην δική μας περίπτωση Mozilla Firefox, υπάρχει η υλοποίηση της διεπαφής όπως εμφανίζεται παρακάτω:

Εικόνα 32.Γραφική διεπαφή Gradio

Εικόνα 33. Dropdown menu

Επιλέγοντας ένα δίκαιο και γράφοντας μία ερώτηση το αποτέλεσμα είναι το εξής:

Εικόνα 34. Παραγωγή Απάντησης

Εικόνα 35. Απάντηση από το μοντέλο gpt-3.5-turbo

Στην εν λόγω ενότητα πραγματοποιήθηκε η ενσωμάτωση της διεπαφής χρήστη για την απρόσκοπτη επικοινωνία με το bot, καθώς και η διαχείριση της εισαγωγής και παραγωγής δεδομένων. Η διαδικασία αυτή διευκολύνει την άμεση παροχή οδηγιών και την καταγραφή των ερωταποκρίσεων. Ακολουθεί το πρόγραμμα ως είναι:

```
1. import os
2. import openai
3. import gradio
4.
5. from apikey import apikey
6. from dikaia import dikaia
7.
8. #αρχικοποίηση κλειδιού
9. openai.api_key = apikey
10.
11. def ask_question_with_instruction(instruction, question):
12.     instruction = "Είσαι δικηγόρος εξειδικευμένος στο " + instruction
13.     # Πρόσθεση ερώτησης με το δίκαιο
14.     prompt = f"{instruction}\n{question}"
15.     # Παραγωγή απάντησης με το GPT
16.     response = openai.ChatCompletion.create(
17.         model = "gpt-3.5-turbo",
18.         messages = [{"role": "user", "content": prompt}],
19.         max_tokens = 1024
20.     )
21.
22.     return response["choices"][0]["message"]["content"].strip()
23.
24. def main():
25.     iface = gradio.Interface(
26.         ask_question_with_instruction,
27.         [
28.             #οι παράμετροι που θα δωθούνε
29.             gradio.Dropdown(choices = dikaia, label = "Διάλεξε ένα δίκαιο"),
30.             gradio.Textbox(lines = 2, label = "Κάνε την ερώτησή σου εδώ:)],
31.             #τι επιστρέφει
32.             "text",
33.             title = "GPT-based Chatbot για την ελληνική νομοθεσία",
34.             description = "Διάλεξε ένα δίκαιο από την λίστα και κάνε την ερώτησή σου",
35.         )
36.     iface.launch()
37.
38. if __name__ == "__main__":
39.     main()
40.
```

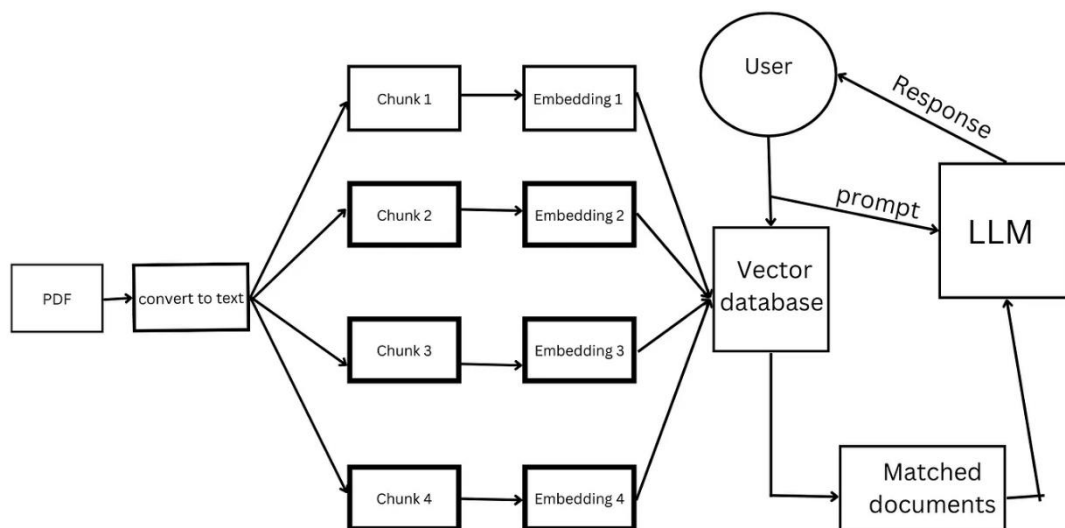
6.6 Χρήση Langchain και τελική υλοποίηση του Chatbot

Στην ενότητα αυτή, θα αναλύσουμε την ενσωμάτωση του εργαλείου Langchain στο σύστημα του ChatBot. Η διαδικασία αυτή περιλαμβάνει την προετοιμασία των αρχείων PDF για τη δημιουργία της γνωστικής βάσης που θα ενσωματωθεί στο ChatBot, καθώς και την ανάπτυξη του μηχανισμού που θα διαχειρίζεται το ιστορικό της συνομιλίας.

Αρχικά, τα PDF θα μετατραπούν σε κείμενο, το οποίο θα χωριστεί σε τμήματα (chunks) για να διευκολυνθεί η επεξεργασία τους. Στη συνέχεια, κάθε τμήμα θα μετατραπεί σε μια διανυσματική αναπαράσταση (embeddings), η οποία θα αποθηκευτεί σε μια βάση δεδομένων διανυσμάτων (vector database).

Όταν ο χρήστης καταθέσει μια ερώτηση, το σύστημα θα χρησιμοποιήσει το Langchain για να συσχετίσει την ερώτηση με την διανυσματική αναπαράσταση των τμημάτων κειμένου στη βάση δεδομένων. Η διαδικασία αυτή θα επιτρέψει στο μοντέλο γλωσσικής μάθησης (LLM) να εντοπίσει τα ταιριαστά έγγραφα που θα παρέχουν το πλαίσιο για την κατάλληλη απάντηση στον χρήστη. Το κεφάλαιο αυτό θα εξετάσει την ενσωμάτωση και την αλληλεπίδραση μεταξύ των διαφορετικών συστημάτων και θα περιγράψει τις αλλαγές που έχουν γίνει για την τελική υλοποίηση του ChatBot. Θα παρουσιαστούν ακόμη οι τεχνικές λεπτομέρειες, η μεθοδολογία που ακολουθήθηκε, καθώς και τα αποτελέσματα από την πρακτική εφαρμογή του συστήματος. Η προσέγγιση αυτή στοχεύει στην ανάδειξη του πόσο κρίσιμη είναι η ενσωμάτωση των σύγχρονων τεχνολογιών τεχνητής νοημοσύνης στην αυτοματοποίηση των νομικών διαδικασιών και στην παροχή αξιόπιστης και ταχύτερης πρόσβασης στη νομική πληροφορία. Συνοπτικά οι στόχοι του κεφαλαίου είναι:

- **Προετοιμασία των Αρχείων PDF**
 - Μετατροπή σε κείμενο για χρήση ως γνωστική βάση του ChatBot.
- **Δημιουργία Διανυσμάτων (Embeddings)**
 - Τμηματοποίηση του κειμένου και μετατροπή σε διανυσματικές αναπαραστάσεις.
 - Αποθήκευση των embeddings σε βάση δεδομένων διανυσμάτων.
- **Διαχείριση Ερωτήσεων Χρήστη**
 - Συσχέτιση ερωτήσεων με τα αντίστοιχα διανύσματα της βάσης δεδομένων.
- **Χρήση του Μοντέλου Γλωσσικής Μάθησης (LLM)**
 - Αναζήτηση ταιριαστών εγγράφων για τη δημιουργία κατάλληλων απαντήσεων.
 - Παραγωγή της απάντησης από το LLM βασισμένη στα εντοπισμένα έγγραφα.
- **Αλληλεπίδραση Συστημάτων και Τελική Υλοποίηση**
 - Περιγραφή της ενσωμάτωσης των συστημάτων Langchain, vector database, και LLM.
 - Ανάλυση των τεχνικών διαδικασιών και εκτίμηση της απόδοσης του ChatBot.



Εικόνα 36 Αναπαράσταση συστήματος²²

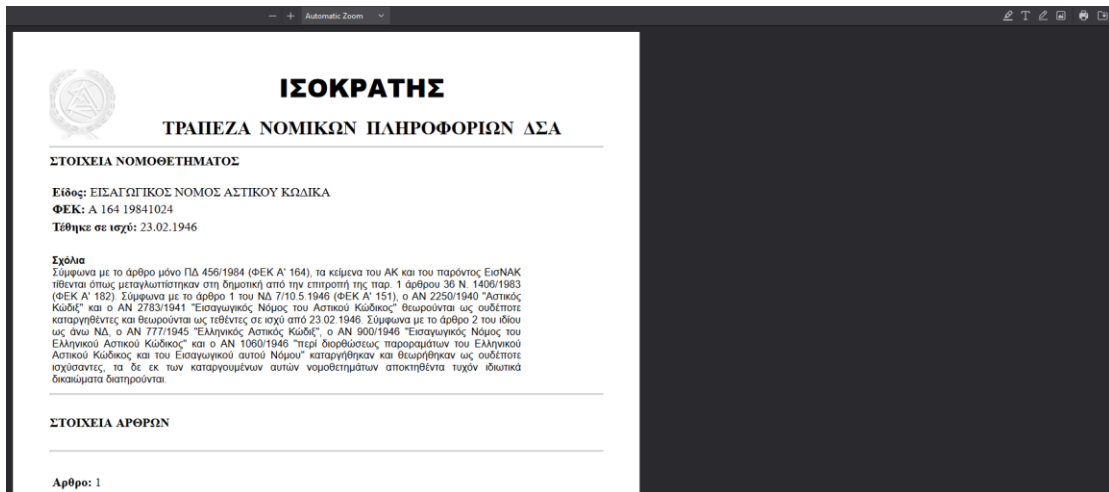
²² <https://medium.com/getpieces/how-to-build-a-langchain-pdf-chatbot-b407fcd750b9>

6.6.1 Προετοιμασία αρχείων

Σε αυτό το κεφάλαιο θα πραγματοποιηθεί η συλλογή των νομικών κειμένων και θα υλοποιηθούν οι απαραίτητες συναρτήσεις για την επεξεργασία τους.

1. Συλλογή νομικών κειμένων

Η διαδικασία συλλογής νομικών κειμένων ξεκινά με την πρόσβαση στις πηγές που αναφέρονται στο κεφάλαιο 7.2, όπου υπάρχουν σύνδεσμοι προς τα απαιτούμενα νομικά έγγραφα. Αφού εντοπιστούν τα σχετικά έγγραφα, πραγματοποιείται η λήψη τους και αποθηκεύονται σε έναν καθορισμένο φάκελο με την ονομασία "pdfs". Παρακάτω ακολουθεί εικόνα με την λήψη του Εισαγωγικού νόμου του αστικού κώδικα. Πατώντας το τελευταίο κουμπί στην πάνω δεξιά θέση γίνεται η λήψη και αποθήκευση του αρχείου.



Εικόνα 37 Λήψη Εισαγωγικού νόμου του αστικού κώδικα

2. Εγκατάσταση Langchain

Για να χρησιμοποιηθούν όλα τα εργαλεία που προσφέρει το Langchain για την υλοποίηση των απαραίτητων συναρτήσεων, θα εγκατασταθεί το Langchain τρέχοντας το πρόγραμμα **pip** για την εγκατάσταση του. Τρέχοντας την παρακάτω εντολή στον τερματικό ξεκινάει η εγκατάσταση.

Για την εγκατάσταση του Langchain, θα χρησιμοποιηθεί το **pip**, ένα εργαλείο διαχείρισης πακέτων Python που διευκολύνει την εγκατάσταση και διαχείριση βιβλιοθηκών. Η εγκατάσταση πραγματοποιείται μέσω της παρακάτω εντολής σε ένα τερματικό:

```
pip install langchain
```

Εικόνα 38 Εγκατάσταση Langchain από τον τερματικό

Αυτή η εντολή θα κατεβάσει και θα εγκαταστήσει το Langchain καθώς και όλες τις απαραίτητες εξαρτήσεις του. Η εγκατάσταση αυτή εξασφαλίζει ότι τα εργαλεία της βιβλιοθήκης θα είναι διαθέσιμα για χρήση στην επόμενη φάση της διπλωματικής, η οποία αφορά την επεξεργασία και ανάλυση των νομικών κειμένων.

3. Συνάρτηση Εξαγωγής των κειμένων από τα pdf αρχεία

```
1. def get_text(pdf_path):
2.     pdf_files = glob.glob("%s/*.pdf" % pdf_path)
3.     text = ""
4.     for pdf in pdf_files:
5.         pdf_reader = PyPDFLoader(pdf)
6.         for page in pdf_reader.pages:
7.             text += page.extract_text()
8.     return text
```

Παραπάνω παρουσιάζεται μια απλή συνάρτηση εξαγωγής κειμένου από τα pdf αρχεία. Στην γραμμή 2 συλλέγονται όλα τα ονόματα των αρχείων που τελειώνουν με ".pdf" με την βοήθεια του glob. Στην συνέχεια για κάθε αρχείο προστίθενται στην μεταβλητή "pdf_reader", και εξάγεται το κείμενο και αποθηκεύεται στη μεταβλητή "text" και επιστρέφεται. Στην επόμενη υποενότητα θα εξεταστούν

διαφορετικές μέθοδοι διαίρεσης και μετατροπής του κειμένου σε διανυσματικές αναπαραστάσεις και αποθήκευσή τους.

6.6.2 Μέθοδοι διαίρεσης του εξαγόμενου κειμένου

Στην ενότητα αυτή θα εξετάσουμε τις διάφορες τεχνικές διαίρεσης του εξαγόμενου κειμένου. Οι μέθοδοι αυτές είναι κρίσιμες για την ανάλυση και την επεξεργασία μεγάλων όγκων κειμένου, διότι επιτρέπουν την αποτελεσματικότερη διαχείριση και κατανόηση των δεδομένων. Ειδικότερα, θα αναλυθούν οι εξής τεχνικές: Διαίρεση με χαρακτήρες (Character Split), Αναδρομική διαίρεση με χαρακτήρες (Recursive Character Split) και Σημασιολογική διαίρεση (Semantic Splitting).

Διαίρεση με χαρακτήρες (Character Split)

Η μέθοδος Διαίρεσης με χαρακτήρες (Character Split) είναι μια από τις πιο απλές τεχνικές διαίρεσης κειμένου.²³ Σε αυτή τη μέθοδο, το κείμενο διασπάται σε μικρότερα τμήματα βάσει συγκεκριμένων χαρακτήρων. Οι χαρακτήρες αυτοί μπορεί να είναι κενά διαστήματα, σημεία στίξης ή άλλοι ειδικοί χαρακτήρες. Για παράδειγμα, ένα κείμενο μπορεί να διαχωριστεί σε λέξεις χρησιμοποιώντας το κενό διάστημα ως σημείο διαίρεσης. Αυτή η μέθοδος είναι ιδιαίτερα χρήσιμη όταν θέλουμε να επεξεργαστούμε κείμενο σε επίπεδο λέξης ή πρότασης. Παρακάτω ακολουθεί η διαδικασία όπως παρουσιάζεται στην ιστοσελίδα του Langchain. Αρχικά γίνεται η εγκατάσταση των Text-splitter με την βοήθεια του pip.



Εικόνα 39 Αναπαράσταση της διαίρεσης²⁴

²³ https://github.com/FullStackRetrieval-com/RetrievalTutorials/blob/main/tutorials/LevelsOfTextSplitting/5_Levels_Of_Text_Splitting.ipynb

²⁴ <https://chunkviz.up.railway.app/>

```
1. pip install -qU langchain-text-splitters
```

Και παρακάτω φαίνεται ένα παράδειγμα χρήσης του.

```
1. from langchain_text_splitters import CharacterTextSplitter
2.
3. text_splitter = CharacterTextSplitter(
4.     separator="\n\n",
5.     chunk_size=1000,
6.     chunk_overlap=200,
7.     length_function=len,
8.     is_separator_regex=False,
9. )
```

Χρησιμοποιώντας το pdf του αστικού κώδικα και τις δύο συναρτήσεις, εμφανίζοντας το πρώτο κομμάτι που έχει χωριστεί το κείμενο, έχουμε το εξής αποτέλεσμα:

```
#συλλογή κειμένου από τα pdf
def get_text(pdf_path):
    pdf_files = glob.glob("%s/*.pdf" % pdf_path)

    text = ""
    for pdf in pdf_files:
        pdf_reader = PyPDFLoader(pdf)

        for page in pdf_reader.load():
            text += page.page_content

    return text

#χωρισμός του κειμένου σε chunks
def get_chunks(text):
    text_splitter = CharacterTextSplitter(
        separator = "\n",
        chunk_size = 300,
        chunk_overlap = 50,
        length_function = len
    )

    chunks = text_splitter.split_text(text)

    return chunks
```

Εικόνα 40 Συναρτήσεις εξαγωγής και διαίρεσης κειμένου

```
text = get_text("pdfs")
chunks = get_chunks(text)

print(chunks[0])
```

Εικόνα 41 Main συνάρτηση

```
ΙΣΟΚΡΑΤΗΣ
ΤΡΑΠΕΖΑ ΝΟΜΙΚΩΝ ΠΛΗΡΟΦΟΡΙΩΝ ΔΣΑ
ΣΤΟΙΧΕΙΑ ΝΟΜΟΘΕΤΗΜΑ ΤΟΣ

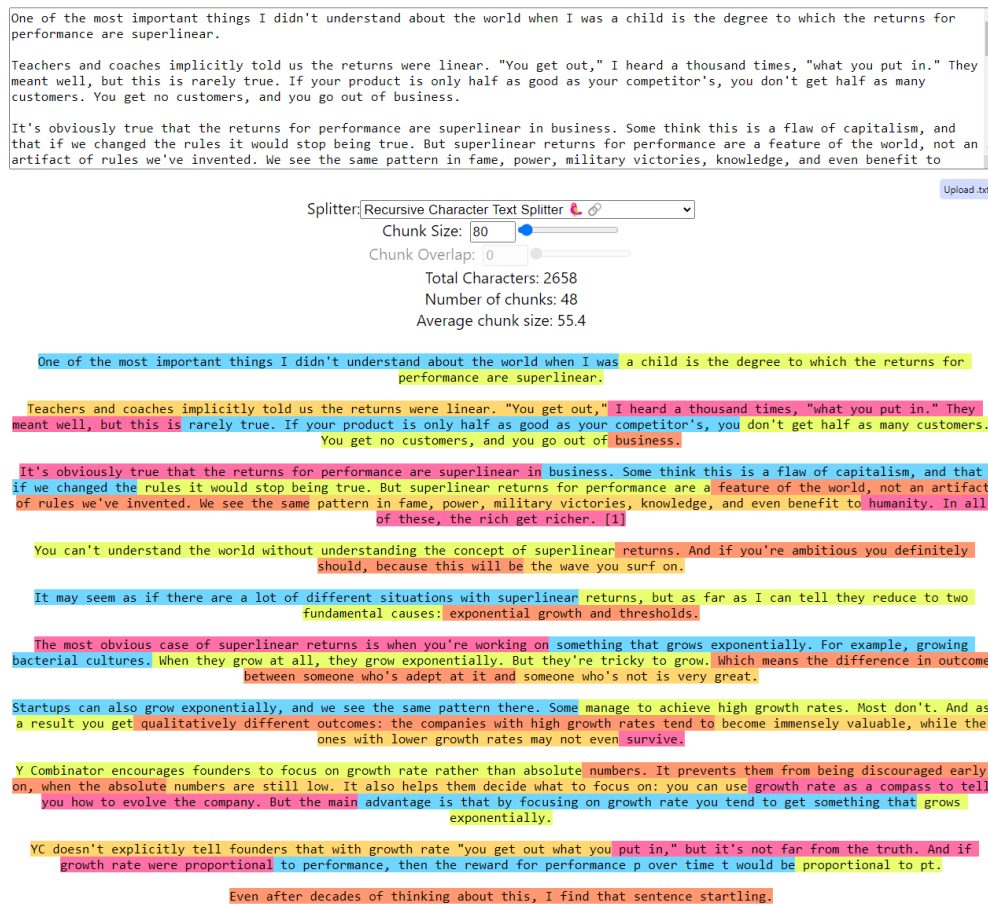
Είδος: ΑΣΤΙΚ ΟΣ ΚΩΔΙΚΑΣ
ΦΕΚ: Α 164 19841024
Τέθηκε σε ισχύ: 23.02.1946

Θέματ α
Γενικές Αρχές, Ενο χικό Δίκ ατο, Εμπράγματ ο Δίκ ατο, Οικ ογενειακ ό Δίκ ατο, Κληρονομικ ό Δίκ ατο
Σχόλια
```

Εικόνα 42 Αποτέλεσμα

Αναδρομική διαίρεση με χαρακτήρες (Recursive Character Split)

Η Αναδρομική διαίρεση με χαρακτήρες (Recursive Character Split) είναι μια πιο εξελιγμένη εκδοχή της Διαίρεσης με χαρακτήρες. Σε αυτή την τεχνική, η διαδικασία διαίρεσης εφαρμόζεται επαναληπτικά σε κάθε τμήμα του κειμένου που προκύπτει από την προηγούμενη διαίρεση. Για παράδειγμα, μετά τη διαίρεση ενός κειμένου σε προτάσεις, μπορούμε να εφαρμόσουμε περαιτέρω διαίρεση για να διαχωρίσουμε τις προτάσεις σε λέξεις. Αυτή η μέθοδος επιτρέπει μια πιο λεπτομερή ανάλυση του κειμένου και μπορεί να προσαρμοστεί ανάλογα με τις ανάγκες της ανάλυσης. Ακολουθώντας την ίδια μέθοδο με το προηγούμενο κομμάτι και κάνοντας τις απαραίτητες αλλαγές έχουμε τα παρακάτω:



Εικόνα 43 Αναπαράσταση αναδρομικής διαίρεσης²⁵

```
1. from langchain_text_splitters import RecursiveCharacterTextSplitter
2.     text_splitter = RecursiveCharacterTextSplitter(
3.         chunk_size = 300,
4.         chunk_overlap = 50,
5.         length_function = len
6.     )
```

ΙΣΟΚΡΑΤΗΣ
ΤΡΑΠΕΖΑ ΝΟΜΙΚΩΝ ΠΛΗΡΟΦΟΡΙΩΝ ΔΣΑ
ΣΤΟΙΧΕΙΑ ΝΟΜΟΘΕΤΗΜΑ ΤΟΣ

Είδος: ΑΣΤΙΚ ΟΣ ΚΩΔΙΚΑΣ
ΦΕΚ: Α 164 19841024
Τέθηκε σε ισχύ: 23.02.1946

Θέματ α
Γενικές Αρχές, Ενο χικό Δίκ αιο, Εμπράγματ ο Δίκ αιο, Οικ ογενειακ ό Δίκ αιο, Κληρονομικ ό Δίκ αιο
Σχόλια

Εικόνα 44Εκτύπωση πρώτου κομματιού

Παρατηρείτε πως δεν υπάρχει διαφορά για το πρώτο κομμάτι.

Σημασιολογική διαίρεση (Semantic Splitting)

Η Σημασιολογική διαίρεση (Semantic Splitting) επικεντρώνεται στη διαίρεση του κειμένου με βάση το νόημα και τη σημασία των λέξεων και των προτάσεων.²⁶ Αντί να βασίζεται σε απλούς χαρακτήρες ή

²⁵ <https://chunkviz.up.railway.app/>

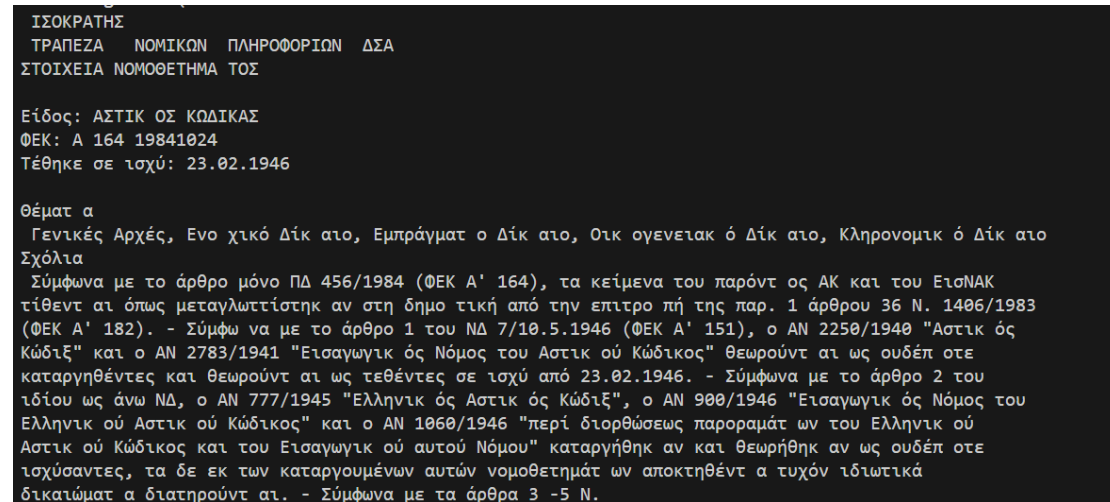
²⁶ https://github.com/FullStackRetrieval-com/RetrievalTutorials/blob/main/tutorials/LevelsOfTextSplitting/5_Levels_Of_Text_Splitting.ipynb

μορφολογικά στοιχεία, αυτή η μέθοδος χρησιμοποιεί τεχνικές επεξεργασίας φυσικής γλώσσας (NLP) για να αναγνωρίσει νοηματικές μονάδες μέσα στο κείμενο. Για παράδειγμα, μπορεί να διαχωρίσει ένα κείμενο σε θεματικές ενότητες ή σε παραγράφους που καλύπτουν συγκεκριμένα υποθέματα. Με την χρήση του OpenAI Embeddings (θα αναλυθεί στην επόμενη υποενότητα) και του SemanticChunker²⁷ ο συγκεκριμένος διαιρέτης κειμένου καταφέρνει να χωρίσει το κείμενο σε κομμάτια που είναι κατανοητά. Θα μπορούσε να επιτευχθεί το ίδιο και με τις προηγούμενες μεθόδους αν η κάθε παράγραφος/σελίδα ακολουθούσε ένα μοτίβο αλλά δεν είναι δυνατό να υλοποιηθεί για τον όγκο των αρχείων που θα επεξεργάζεται το chatbot. Μειονεκτήματα αυτής της προσέγγισης είναι το επιπλέον κόστος με την χρήση του OpenAI Embeddings και η ταχύτητα διότι χρειάζεται παραπάνω επεξεργασία το κείμενο ώστε να πραγματοποιηθεί η διαίρεση του κειμένου. Παρακάτω ακολουθούν οι αλλαγές στην συνάρτηση `get_chunks` και εικόνα με την εμφάνιση του πρώτου κομματιού.

```
1. pip install langchain_experimental langchain_openai
```

```
1. from langchain_experimental.text_splitter import SemanticChunker
2. from langchain_openai.embeddings import OpenAIEmbeddings
```

```
1. def get_chunks(text):
2.     embeddings = OpenAIEmbeddings()
3.     text_splitter = SemanticChunker(embeddings, breakpoint_threshold_type =
4.         "percentile")
5.     chunks = text_splitter.split_text(text)
6.
7.     return chunks
8.
```



ΕΙΣΑΓΩΓΙΚΟΣ ΚΩΔΙΚΑΣ
ΕΠΕΡΕΥΧΗ ΝΟΜΙΚΩΝ ΠΛΗΡΟΦΟΡΙΩΝ ΔΣΑ
ΣΤΟΙΧΕΙΑ ΝΟΜΟΘΕΤΗΜΑ ΤΟΣ

Είδος: ΑΣΤΙΚ ΟΣ ΚΩΔΙΚΑΣ
ΦΕΚ: Α 164 19841024
Τέθηκε σε ισχύ: 23.02.1946

Θέμα α
Γενικές Αρχές, Ενο χικό Δίκ αιο, Εμπράγματ ο Δίκ αιο, Οικ ογενειακ ό Δίκ αιο, Κληρονομικ ό Δίκ αιο

Σχόλια
Σύμφωνα με το άρθρο μόνο ΠΔ 456/1984 (ΦΕΚ Α' 164), τα κείμενα του παρόντος ΑΚ και του ΕισΝΑΚ τίθενται ως προς μεταγωγική αν στη δημοτική από την επιτροπή της παρ. 1 άρθρου 36 Ν. 1406/1983 (ΦΕΚ Α' 182). - Σύμφωνα με το άρθρο 1 του ΝΔ 7/10.5.1946 (ΦΕΚ Α' 151), ο ΑΝ 2250/1940 "Αστικ ός Κώδικας" και ο ΑΝ 2783/1941 "Εισαγωγικ ός Νόμος του Αστικ ού Κώδικος" θεωρούνται ως ουδέποτε καταργηθέντες και θεωρούνται ως τεθέντες σε ισχύ από 23.02.1946. - Σύμφωνα με το άρθρο 2 του ιδίου ως άνω ΝΔ, ο ΑΝ 777/1945 "Ελληνικ ός Αστικ ός Κώδικας", ο ΑΝ 900/1946 "Εισαγωγικ ός Νόμος του Ελληνικ ού Αστικ ού Κώδικος" και ο ΑΝ 1060/1946 "περί διορθώσεως παραραμάτων του Ελληνικ ού Αστικ ού Κώδικος και του Εισαγωγικ ού αυτού Νόμου" καταργήθηκαν και θεωρήθηκαν ως ουδέποτε ισχύσαντες, τα δε εκ των καταργουμένων αυτών νομοθετημάτων αποκτηθέντα τυχόν ιδιωτικά δικαιώματα διατηρούνται. - Σύμφωνα με τα άρθρα 3 -5 Ν.

Εικόνα 45 Εκτύπωση πρώτου κομματιού

6.6.3 Διανυσματικές αναπαραστάσεις και αποθήκευσή τους

Στο παρόν κεφάλαιο θα αναλυθούν οι τεχνικές μετατροπής κειμένου σε διανυσματικές αναπαραστάσεις αλλά και η αποθήκευσή τους σε vectorstore. Για την επίτευξη αυτού του στόχου θα χρησιμοποιηθούν τα OpenAI embeddings και το chroma²⁸ vectorstore μέσω του API του Langchain.

²⁷ "Semantic Search — Sentence Transformers documentation."
<https://sbert.net/examples/applications/semantic-search/README.html>

²⁸ <https://python.langchain.com/v0.1/docs/integrations/vectorstores/chroma/>

Σε αυτό το κεφάλαιο θα αναλυθούν οι τεχνικές μετατροπής κειμένου σε διανυσματικές αναπαραστάσεις (vector representations), καθώς και η αποθήκευσή τους σε μια βάση δεδομένων διανυσμάτων (vectorstore). Για την επίτευξη αυτού του στόχου θα χρησιμοποιηθούν τα embeddings του OpenAI και το vectorstore Chroma μέσω του API του Langchain.

Τι είναι τα Embeddings

Τα embeddings είναι διανυσματικές αναπαραστάσεις (vector representations) λέξεων ή φράσεων που χρησιμοποιούνται στην επεξεργασία φυσικής γλώσσας (Natural Language Processing - NLP). Τα embeddings μετατρέπουν τις λέξεις σε αριθμητικούς διανυσματικούς χώρους (vector spaces), επιτρέποντας στους αλγόριθμους να κατανοήσουν τις σημασιολογικές σχέσεις μεταξύ των λέξεων. Οι λέξεις με παρόμοια σημασία βρίσκονται κοντά στον διανυσματικό χώρο, ενώ οι λέξεις με διαφορετική σημασία βρίσκονται μακριά ή μία από την άλλη.²⁹

Τι είναι το Vectorstore

Το vectorstore είναι μια ειδική βάση δεδομένων που έχει σχεδιαστεί για την αποθήκευση και την αναζήτηση διανυσματικών αναπαραστάσεων. Σε αντίθεση με τις παραδοσιακές βάσεις δεδομένων που αποθηκεύουν δεδομένα σε μορφή πίνακα, το vectorstore αποθηκεύει δεδομένα σε διανυσματική μορφή, επιτρέποντας γρήγορες και αποδοτικές αναζητήσεις βασισμένες στην ομοιότητα μεταξύ των διανυσμάτων.³⁰

Πώς Χρησιμοποιούνται τα Embeddings και το Vectorstore

1. Δημιουργία Embeddings:

- Τα embeddings δημιουργούνται χρησιμοποιώντας μοντέλα NLP που εκπαιδεύονται σε μεγάλα σύνολα δεδομένων. Αυτά τα μοντέλα μετατρέπουν τις λέξεις σε διανύσματα με καθορισμένες διαστάσεις. Ένα παράδειγμα τέτοιου μοντέλου είναι τα OpenAI embeddings.

2. Αποθήκευση σε Vectorstore:

- Μετά τη δημιουργία των embeddings, αποθηκεύονται σε ένα vectorstore όπως το Chroma για μελλοντική χρήση.
- Το vectorstore επιτρέπει την αποθήκευση μεγάλων όγκων διανυσματικών δεδομένων και την ταχεία ανάκτηση τους μέσω αναζητήσεων βασισμένων στην ομοιότητα των διανυσμάτων.

3. Χρήση Embeddings:

- Αναζήτηση και Ανάκτηση Πληροφορίας (Search and Information Retrieval):** Οι εφαρμογές μπορούν να αναζητήσουν κείμενα που είναι σημασιολογικά παρόμοια με ένα δεδομένο ερώτημα, χρησιμοποιώντας τα embeddings.
- Συστάσεις (Recommendations):** Τα embeddings χρησιμοποιούνται για να προσφέρουν προσωποποιημένες συστάσεις βάσει της σημασιολογικής ομοιότητας μεταξύ των προτιμήσεων των χρηστών.
- Κατηγοριοποίηση και Ταξινόμηση (Categorization and Classification):** Τα embeddings βοηθούν στη βελτίωση της ακρίβειας των μοντέλων κατηγοριοποίησης και ταξινόμησης κειμένου, επιτρέποντας την κατανόηση του νοήματος του κειμένου.

Παράδειγμα Χρήσης με OpenAI Embeddings και Chroma Vectorstore

1. Εγκατάσταση Απαιτούμενων Πακέτων:

```
1. pip install langchain_chroma
```

2. Δημιουργία και Αποθήκευση Embeddings:

```
1. from langchain.embeddings import OpenAIEmbeddings
2. from langchain.vectorstores import Chroma
3.
4. def gt_vectorstore(text_chunks):
5.     embeddings = OpenAIEmbeddings()
6.
7.     vectorstore = Chroma.from_documents(text_chunks, embeddings)
8.
9.     return vectorstore
```

²⁹ <https://pub.towardsai.net/langchain-101-part-2ab-all-you-need-to-know-about-large-language-models-3512ae41dfc3#12b9>

³⁰ <https://pub.towardsai.net/langchain-101-part-2ab-all-you-need-to-know-about-large-language-models-3512ae41dfc3#12b9>

3. Αναζήτηση στο Vectorstore:

- Μόλις τα embeddings έχουν αποθηκευτεί, μπορούμε να εκτελέσουμε αναζητήσεις στο vectorstore για να βρούμε παρόμοια κείμενα.
- Παράδειγμα κώδικα για αναζήτηση:

```
1. #διάβασμα κειμένου από τα pdf
2. raw_text = get_text("pdfs")
3.
4. #χωρισμός του κειμένου σε chunks
5. text_chunks = get_chunks(raw_text)
6.     vectorstore = get_vectorstore(documents)
7.
8. result = vectorstore.similarity_search("Δωσε μου το αρθρο 2 του αστικού κωδικα")
9.
10. for page in result:
11.     print(page.page_content)
```

Και παρακάτω βλέπουμε κείμενα που μας έχουνε επιστραφεί από την μέθοδο similarity_search().

Περιγραφή όρου θησαυρού: ΙΚΑΝΟΠΟΙΗΣΗ ΗΘΙΚΗΣ ΒΛΑΒΗΣ

Τίτλος Αρθρου
 Ικανοποίηση της ηθικής βλ άβης. Λήμματ α
 ΕΝΟΧΕΣ, ΑΔΙΚ ΟΠΡ ΑΞΙΑ, ΙΚΑΝΟΠΟΙΗΣΗ ΗΘΙΚΗΣ ΒΛΑΒΗΣ, ΙΚΑΝΟΠΟΙΗΣΗ ΨΥΧΙΚΗΣ Ο ΔΥΝΗΣ

Σχόλια
 Σύμφωνα με την παρ. 2 του άρθρου 14 του ν. 3471/2006 (Α' 133/28.6.2006), η -κατά το πα ρ ό ν
 άρθρο- χρηματική ικανοποίηση λόγω ηθικής βλάβης για παράβαση του ανωτέρω νόμου ορίζετ αι, κατ'
 ελάχιστ ο, στο ποσό των δέκα χιλιάδων ευρώ (10.000), εκτός αν ζητηθεί από τον ενάγο ντα μικρό τερο
 ποσό. Η χρηματική ικανοποίηση επιδικ άζεται ανεξάρτητα από την αιτούμενη αποζημίωση για
 περιουσιακή βλάβη. Σύμφωνα με το άρθρο 5 του ν. 3500/2006 για την αντιμετ ώπιση της
 ενδοοικ ογενειακής βίας (Α' 232/24.10.2006), η κατά το πα ρ ό ν άρθρο χρηματική ικανοποίηση, λόγω
 ηθικής βλάβης του παθόντ ος για μία από τις πράξεις του ανωτέρω νόμου, δ ε ν μπορεί να είναι
 κατώτερη τ ων χιλίων (1.000) ευρώ, ε κ τ ό ς αν ο ίδιος ο π αθών ζ ήτησε μ ι κ ρ ό τ ε ρ ο π οσό. Κείμενο Αρθρου

Σε περίπτωση αδικοπραξίας, ανεξάρτητα από την αποζημίωση για την περιουσιακή ζημία, το
 δικαστήριο μπορεί να επιδικ άσει εύλογη κατά την κρίση του χρηματική ικανοποίηση λόγω ηθικής

Εικόνα 46Επιστροφή κειμένων από similarity_search()

Κύρια Μοντέλα για Δημιουργία Embeddings

1. Word2Vec:

- Αναπτύχθηκε από την Google και χρησιμοποιεί δύο κύριες προσεγγίσεις: Continuous Bag of Words (CBOW) και Skip-Gram.
- Εκπαιδεύεται σε μεγάλα σύνολα δεδομένων για να μάθει τις σχέσεις μεταξύ των λέξεων.

2. GloVe (Global Vectors for Word Representation):

- Αναπτύχθηκε από το Stanford και βασίζεται στην ανάλυση της συνύπαρξης των λέξεων σε μεγάλες ομάδες κειμένων.
- Δημιουργεί στατικά embeddings που είναι χρήσιμα για πολλές εφαρμογές.

3. FastText:

- Αναπτύχθηκε από το Facebook και επεκτείνει το Word2Vec λαμβάνοντας υπόψη τα υπολέξια (subwords).
- Παρέχει καλύτερη αναπαράσταση για σπάνιες ή σύνθετες λέξεις.

4. BERT (Bidirectional Encoder Representations from Transformers):

- Αναπτύχθηκε από την Google και χρησιμοποιεί την αρχιτεκτονική των Transformers για να δημιουργήσει δυναμικά embeddings.
- Είναι ικανό να κατανοεί το συμφραζόμενο και τις πολλαπλές σημασίες μιας λέξης.

5. GPT (Generative Pre-trained Transformer):

- Αναπτύχθηκε από την OpenAI και χρησιμοποιεί επίσης την αρχιτεκτονική των Transformers.
- Είναι γνωστό για τις ικανότητές του στην παραγωγή κειμένου και τη δημιουργία υψηλής ποιότητας embeddings.

6.6.4 Αλυσίδα ερώτησης-κειμένων-OpenAI

Σε αυτό το υπό κεφάλαιο θα υλοποιηθεί η ροή που θα ακολουθεί η ερώτηση του χρήστη, ερώτηση χρήστη -> αναζήτηση κειμένων στο vectorstore -> αποστολή στο OpenAI για παραγωγή απάντησης -> ιστορικό συνομιλίας. Για την επίτευξη αυτού του στόχου θα χρησιμοποιηθούν από το Langchain τα εξής: ChatOpenAI, ConversationBufferMemory, ConversationalRetrievalChain.

Ροή Διαδικασίας

1. **Ερώτηση Χρήστη**
 - Ο χρήστης υποβάλλει μια ερώτηση μέσω της διεπαφής του chatbot. Αυτή η ερώτηση μπορεί να είναι οποιοδήποτε κείμενο ή φράση που αναζητά πληροφορίες ή βοήθεια.
2. **Αναζήτηση Κειμένων στο Vectorstore**
 - Το ερώτημα του χρήστη μετατρέπεται σε διανυσματική αναπαράσταση (embedding) χρησιμοποιώντας το μοντέλο OpenAI Embeddings.
 - Η διανυσματική αναπαράσταση του ερωτήματος χρησιμοποιείται για την αναζήτηση σχετικών κειμένων στο vectorstore. Το vectorstore περιέχει αποθηκευμένα embeddings κειμένων που έχουν ήδη μετατραπεί σε διανυσματικές μορφές.
 - Η αναζήτηση στο vectorstore επιστρέφει τα πιο σχετικά κείμενα βάσει της ομοιότητας των διανυσμάτων.
3. **Αποστολή στο OpenAI για Παραγωγή Απάντησης**
 - Τα κείμενα που ανακτήθηκαν από το vectorstore αποστέλλονται στο μοντέλο OpenAI για την παραγωγή μιας απάντησης.
 - Το μοντέλο OpenAI χρησιμοποιεί τις πληροφορίες από τα ανακτηθέντα κείμενα για να δημιουργήσει μια απάντηση που είναι ακριβής και συναφής με το ερώτημα του χρήστη.
4. **Ιστορικό Συνομιλίας**
 - Η απάντηση του OpenAI καταγράφεται στο ιστορικό συνομιλίας του χρήστη μέσω του ConversationBufferMemory.³¹
 - Η διατήρηση του ιστορικού συνομιλιών επιτρέπει στο σύστημα να παρέχει συνεπείς και σχετικές απαντήσεις σε επόμενες ερωτήσεις, λαμβάνοντας υπόψη το προηγούμενο πλαίσιο της συνομιλίας.

Παράδειγμα Υλοποίησης με Langchain

- 1) **Ορισμός της Αλυσίδας:** Ο κώδικας παρακάτω δείχνει πώς να ορίσουμε την αλυσίδα χρησιμοποιώντας το Langchain:

```
1. def conversation_chain(vectorstore):
2.     llm = ChatOpenAI(model = "gpt-4")
3.     memory = ConversationBufferMemory(memory_key = 'chat_history', return_messages =
True, output_key='answer')
4.     conversationchain = ConversationalRetrievalChain.from_llm(
5.         llm = llm,
6.         retriever = vectorstore.as_retriever(),
7.         memory = memory,
8.         return_source_documents=True
9.     )
10.
11.     return conversationchain
```

- 2) **Χρήση της Αλυσίδας για Απάντηση σε Ερώτημα Χρήστη:** Παρακάτω φαίνεται πώς να χρησιμοποιήσουμε την αλυσίδα για να απαντήσουμε σε ένα ερώτημα χρήστη:

```
1. conversation = conversation_chain(vectorstore)
2.
3. user_query = "Δώσε μου το άρθρο 2 του αστικού κώδικα"
4. response = conversation.invoke(user_input)
5.
6. print(response)
```

- 3) **Διατήρηση Ιστορικού Συνομιλίας (Maintaining Conversation History):**

³¹ <https://python.langchain.com/v0.1/docs/modules/memory/types/buffer/>

- Η χρήση του **ConversationBufferMemory** διασφαλίζει ότι το ιστορικό της συνομιλίας καταγράφεται και χρησιμοποιείται για την παροχή συνεπών απαντήσεων σε μελλοντικές αλληλεπιδράσεις με τον χρήστη.
- Αυτό επιτρέπει στο σύστημα να αναγνωρίζει το πλαίσιο της συνομιλίας και να προσαρμόζει τις απαντήσεις του αναλόγως.

ConversationBufferMemory

Το ConversationBufferMemory είναι μια κλάση που αποθηκεύει το ιστορικό συνομιλιών. Αυτό είναι σημαντικό για την διατήρηση του πλαισίου σε συνομιλίες μεγάλης διάρκειας, επιτρέποντας στο σύστημα να ανακαλεί προηγούμενες συνομιλίες και να παρέχει πιο συναφείς απαντήσεις.

ConversationalRetrievalChain

Η ConversationalRetrievalChain είναι μια κλάση που συνδυάζει τις δυνατότητες ανάκτησης πληροφοριών από ένα vectorstore και την παραγωγή απαντήσεων χρησιμοποιώντας ένα μοντέλο γλώσσας. Αυτή η αλυσίδα εξασφαλίζει ότι οι απαντήσεις που παρέχονται στον χρήστη βασίζονται σε ανακτηθείσες πληροφορίες που είναι σχετικές με το ερώτημά του.

6.6.5 Προσαρμογή της διεπαφής χρήστη

Σε αυτό το κεφάλαιο θα πραγματοποιηθεί η προσαρμογή της διεπαφής για την ενσωμάτωση της παραπάνω αλυσίδας. Θα χρησιμοποιηθεί το ChatInterface ³² για την γρήγορη υλοποίηση του, η μόνη απαραίτητη προϋπόθεση για την υλοποίηση του είναι η δημιουργία μίας συνάρτησης η οποία παίρνει 2 παραμέτρους, την ερώτηση του χρήστη και το ιστορικό της συνομιλίας. Άλλος ένας στόχος είναι η επιστροφή των σελίδων και το όνομα του αρχείου που χρησιμοποιήθηκαν ως πηγές για την ανάπτυξη της απάντησης από το gpt μοντέλο. Ακολουθεί η συνάρτηση chat function:

```
1. def chat_function(user_input, history):
2.
3.     response = conversation.invoke(user_input)
4.     resources = ''
5.
6.     for page in response["source_documents"]:
7.         for metadata in page:
8.             if(metadata[0] == 'metadata'):
9.                 resources += "page: " + str(metadata[1]['page'] + 1)
10.                resources += " document name: " + metadata[1]['source'] + "\n"
11.
12.     print(resources)
13.
14.     return response["answer"] + "\n" + resources
15.
```

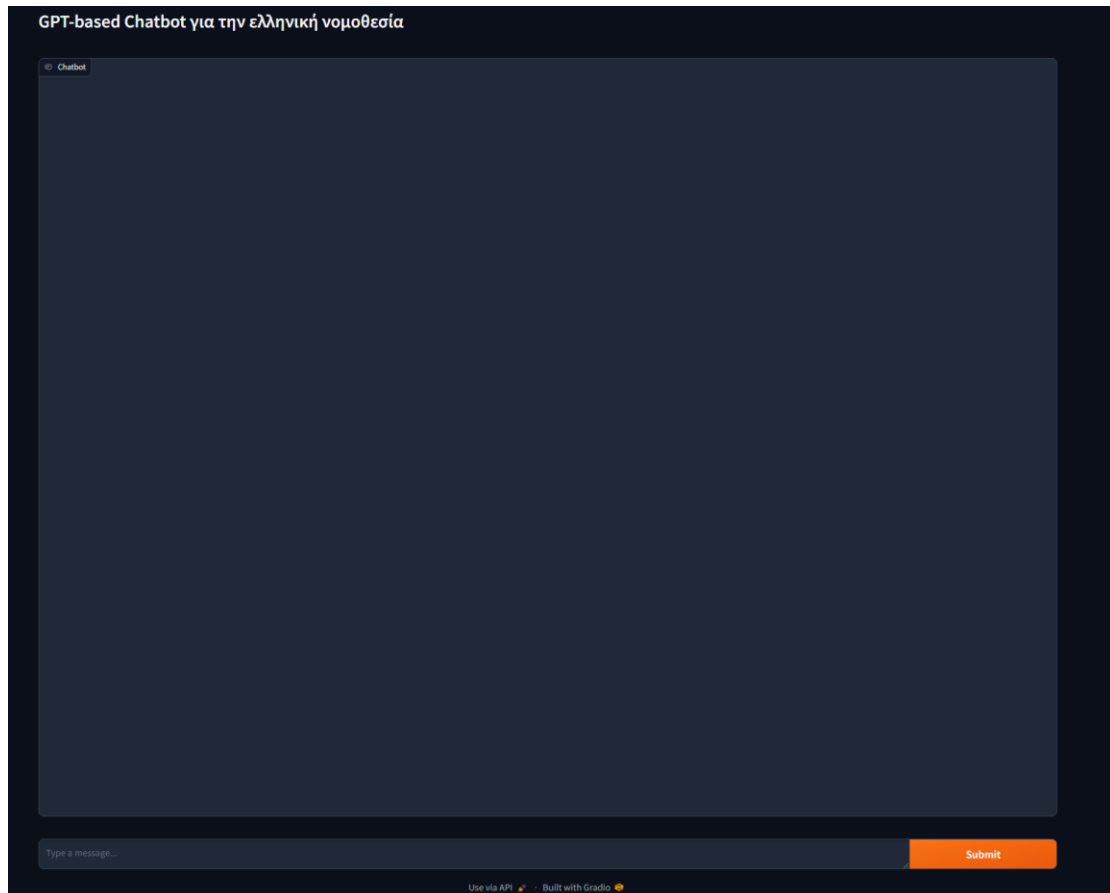
Και η διεπαφή χρήστη χρησιμοποιώντας το ChatInterface(). Ακολουθεί πίνακας για τις παραμέτρους, το κομμάτι κώδικα και αντίστοιχη εικόνα με την διεπαφή.

Παράμετρος	Περιγραφή
fn (Callables)	Η συνάρτηση για την αποθήκευση του chat interface. Πρέπει να δέχεται δύο παραμέτρους: το user_input και το history, και να επιστρέφει την απάντηση, τους πόρους, και τυχόν άλλα δεδομένα.
initialText (str or None)	Εάν καθοριστεί, το chat interface θα εμφανίσει αυτό το μήνυμα αρχικά. Εάν είναι None, το chat interface θα ξεκινήσει χωρίς προεπιλεγμένο μήνυμα
chatbot (Callable[[Any, Any], Component])	Η συνάρτηση που δημιουργεί το component για το chat interface. Εάν δεν καθοριστεί, θα χρησιμοποιηθεί το προεπιλεγμένο component.

³² <https://www.gradio.app/docs/gradio/chatinterface>

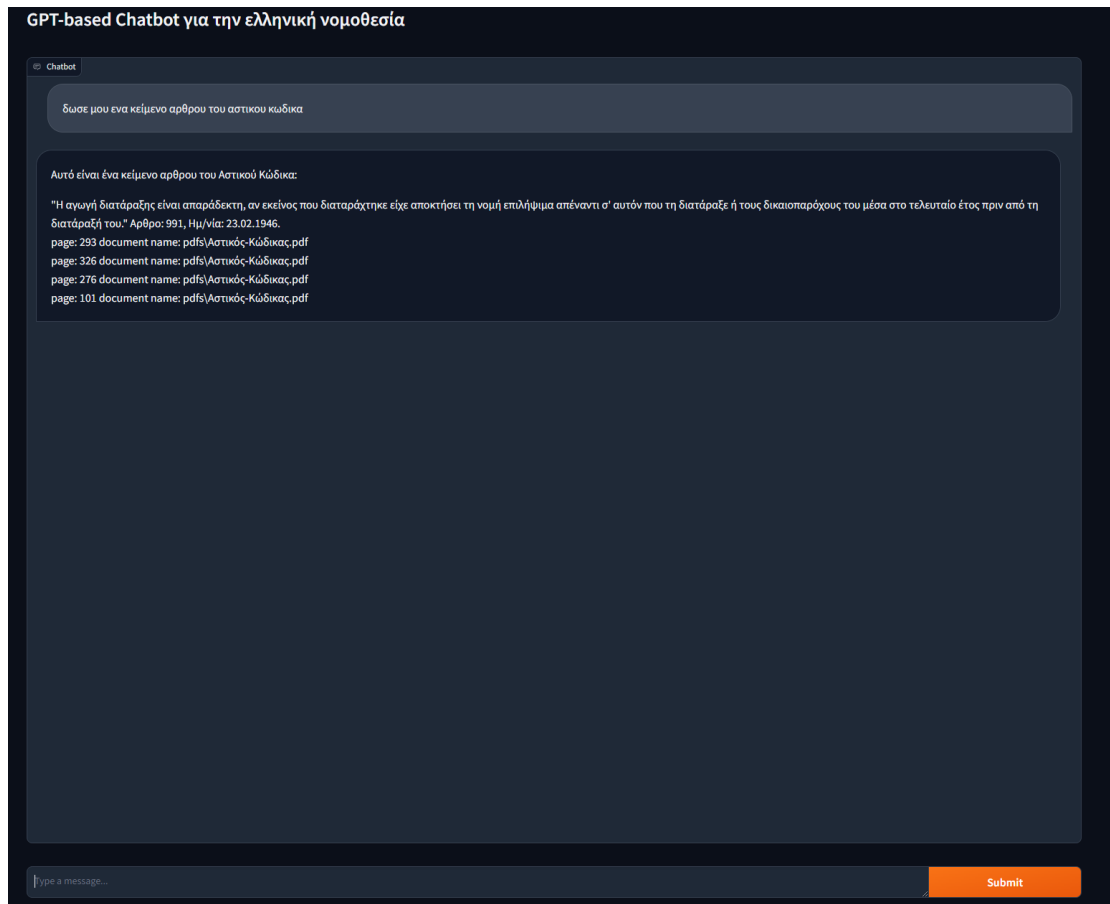
additional_inputs (List[Component])	Ένας κατάλογος πρόσθετων component που θα εμφανίζονται με το chat interface. Εάν δεν καθοριστεί, δεν θα προστεθούν πρόσθετα component.
additional_inputs_accordion (Union[bool, str], optional)	Εάν καθοριστεί, τα πρόσθετα inputs θα εμφανίζονται σε ένα accordion. Χρησιμοποιήστε το additional_inputs_accordion για να καθορίσετε τον τίτλο του accordion.
examples (Union[List[str], List[List[str]], Dict[str, str], Dict[str, List[str]], Callable[[], Union[List[str], List[List[str]], Dict[str, str], Dict[str, List[str]]], optional)	Παρέχει παραδείγματα που εμφανίζονται κάτω από το πεδίο κειμένου για να βοηθήσουν τον χρήστη να ξεκινήσει τη συνομιλία. Εάν καθοριστεί ένα URL ή callable, θα εμφανίζονται τα αποτελέσματα του.
cache_examples (bool)	Εάν είναι True, τα παραδείγματα θα φορτώνονται μόνο μία φορά για την τρέχουσα συνεδρία.
examples_per_page (int)	Εάν παραδείγματα παρέχονται, καθορίζει πόσα παραδείγματα θα εμφανίζονται ανά σελίδα.
title (str)	Τίτλος για τη διεπαφή, εμφανίζεται πάνω από το chatbox. Επίσης χρησιμοποιείται ως τίτλος της σελίδας όταν ανοίγει σε παράθυρο περιηγητή.
description (str)	Μια σύντομη περιγραφή της εφαρμογής που εμφανίζεται πάνω από το chatbox. Μπορεί να χρησιμοποιηθεί Markdown.
theme (str or None)	Θέμα για τη διεπαφή, μπορεί να είναι "default", "monochrome", ή "compact".
css (str)	Προσαρμοσμένα CSS στυλ που εφαρμόζονται στη διεπαφή. Αυτά θα περιληφθούν στην κεφαλίδα της σελίδας.
container (Union[bool, str])	Καθορίζει αν θα χρησιμοποιηθεί κοντέινερ γύρω από το chatbox. Το κοντέινερ βελτιώνει την εμφάνιση σε πολλές διατάξεις, αλλά μπορεί να παραληφθεί σε ορισμένες εφαρμογές.
height (Union[str, int])	Καθορίζει το ύψος της διεπαφής. Το default είναι 500 pixels.
width (Union[str, int])	Καθορίζει το πλάτος της διεπαφής. Το default είναι "full".
scrollbar (bool)	Εάν είναι True, θα εμφανίζεται μια γραμμή κύλισης στο ιστορικό της συνομιλίας.
log_time (bool)	Εάν είναι True, θα καταγράφεται η ώρα για κάθε μήνυμα που αποστέλλεται και λαμβάνεται.
clear_on_submit (bool)	Εάν είναι True, το πεδίο κειμένου θα αδειάζει όταν ο χρήστης υποβάλλει ένα νέο μήνυμα.
input_placeholder (str)	Το κείμενο που θα εμφανίζεται ως placeholder στο πεδίο εισαγωγής κειμένου.
timeout (float)	Καθορίζει το χρονικό όριο για κάθε αίτημα.
retry_on_timeout (bool)	Εάν είναι True, θα γίνεται επαναπροσπάθεια σε περίπτωση χρονικού ορίου.
auto_submit (bool)	Εάν είναι True, το chat interface θα υποβάλλει αυτόματα το πρώτο μήνυμα.
concurrency_limit (Optional[int])	Καθορίζει το μέγιστο αριθμό ταυτόχρονων αιτημάτων που μπορεί να διαχειριστεί το chat interface.
fill_height (bool)	Εάν είναι True, η διεπαφή θα προσαρμόζει το ύψος της ανάλογα με το περιεχόμενο.
disable_cache (bool)	Εάν είναι True, η αποθήκευση προσωρινών δεδομένων θα απενεργοποιηθεί.

```
1. gr.ChatInterface(fn = chat_function, retry_btn = None, undo_btn = None, clear_btn =  
None, title = "GPT-based Chatbot για την ελληνική νομοθεσία").launch()
```



Εικόνα 47 Γραφική διεπαφή Gradio

Στην παραπάνω εικόνα φαίνεται η διεπαφή και θα ακολουθήσουν φωτογραφίες με δοκιμή του συστήματος όπου θα παρουσιαστεί η ικανότητα του να απαντάει σε ερωτήσεις βάση των αρχείων μας, να επιστρέφει τις πηγές που χρησιμοποιήθηκαν για την παραγωγή της απάντησης αλλά και την ικανότητα του να θυμάται και να απαντάει σε ερωτήσεις βάση στα προ αναφερόμενα.



Εικόνα 48 Ερώτηση και απάντηση στο chatbot

293 / 630 | — 100% + | [icon] [icon] παράξης είναι απαράδεκτη 1/1 ^ v X

Αρθρο: 989
Ημ/νία: 23.02.1946
Ημ/νία Ισχύος: 23.02.1946
Περιγραφή όρου θησαυρού: ΠΡΟΣΒΟΛΗ - ΔΙΑΤΑΡΑΞΗ ΝΟΜΗΣ - ΑΠΟΒΟΛΗ ΝΟΜΕΑ - ΠΑΡΑΓΡΑΦΗ ΑΞΙΩΣΕΩΝ

Τίτλος Αρθρου
Προστασία σε περίπτωση διατάραξης.

Λήμματα
ΕΜΠΡΑΓΜΑΤΟ ΔΙΚΑΙΟ, ΝΟΜΗ, ΠΡΟΣΒΟΛΗ ΝΟΜΗΣ, ΔΙΑΤΑΡΑΞΗ ΝΟΜΗΣ, ΠΡΟΣΤΑΣΙΑ

Κείμενο Αρθρου

Ο νομέας που διαταράχτηκε παράνομα έχει δικαίωμα να αξιώσει την παύση της διατάραξης καθώς και την παράλειψή της στο μέλλον. Αξίωση αποζημίωσης κατά τις διατάξεις για τις αδικοπράξεις δεν αποκλείεται.

Αρθρο: 990
Ημ/νία: 23.02.1946
Ημ/νία Ισχύος: 23.02.1946
Περιγραφή όρου θησαυρού: ΠΡΟΣΒΟΛΗ - ΔΙΑΤΑΡΑΞΗ ΝΟΜΗΣ - ΑΠΟΒΟΛΗ ΝΟΜΕΑ - ΠΑΡΑΓΡΑΦΗ ΑΞΙΩΣΕΩΝ

Λήμματα
ΕΜΠΡΑΓΜΑΤΟ ΔΙΚΑΙΟ, ΝΟΜΗ, ΝΟΜΗ ΕΠΙΛΗΨΙΜΗ

Κείμενο Αρθρου

Η αγωγή διατάραξης είναι απαράδεκτη, αν εκείνος που διαταράχτηκε είχε αποκτήσει τη νομή επιλήψιμα απέναντι σ' αυτόν που τη διατάραξε ή τους δικαιιοπαρόχους του μέσα στο τελευταίο έτος πριν από τη διατάραξή του.

Αρθρο: 991
Ημ/νία: 23.02.1946
Ημ/νία Ισχύος: 23.02.1946
Περιγραφή όρου θησαυρού: ΠΡΟΣΒΟΛΗ - ΔΙΑΤΑΡΑΞΗ ΝΟΜΗΣ - ΑΠΟΒΟΛΗ ΝΟΜΕΑ - ΠΑΡΑΓΡΑΦΗ ΑΞΙΩΣΕΩΝ

Λήμματα
ΕΜΠΡΑΓΜΑΤΟ ΔΙΚΑΙΟ, ΝΟΜΗ, ΠΡΟΣΒΟΛΗ ΝΟΜΗΣ, ΔΙΑΤΑΡΑΞΗ ΝΟΜΗΣ, ΑΠΟΒΟΛΗ ΤΟΥ ΝΟΜΕΑ

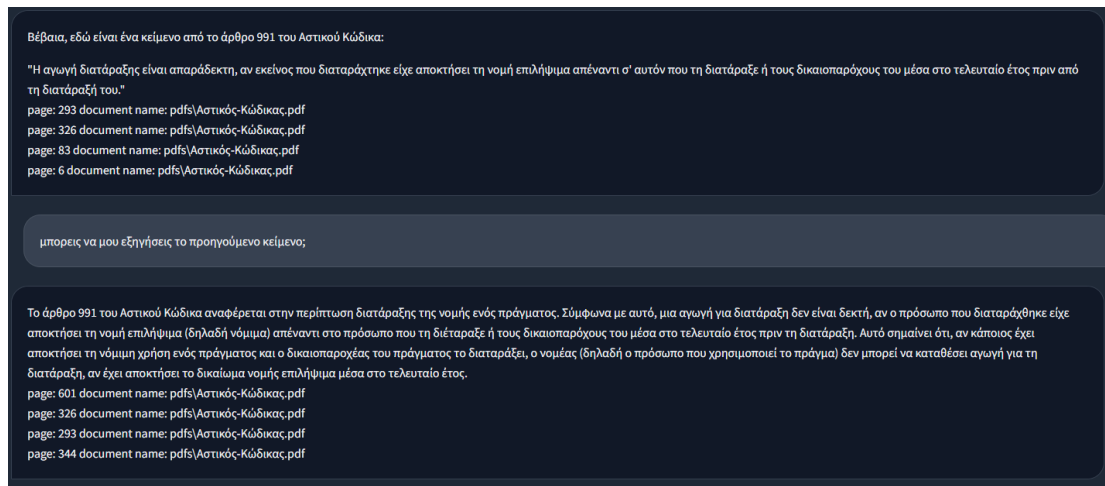
Κείμενο Αρθρου

Ο εναγόμενος για διατάραξη ή αποβολή δεν μπορεί να επικαλεστεί δικαίωμα που του παρέχει εξουσία πάνω στο πράγμα παρά μόνο αν το δικαίωμα έχει αναγνωριστεί τελεσίδικα σε δίκη ανάμεσα σ' αυτόν και τον ενάγοντα.

Αρθρο: 992
Ημ/νία: 23.02.1946
Ημ/νία Ισχύος: 23.02.1946
Περιγραφή όρου θησαυρού: ΠΡΟΣΒΟΛΗ - ΔΙΑΤΑΡΑΞΗ ΝΟΜΗΣ - ΑΠΟΒΟΛΗ ΝΟΜΕΑ - ΠΑΡΑΓΡΑΦΗ ΑΞΙΩΣΕΩΝ

Τίτλος Αρθρου
Προστασία.

Εικόνα 49 σελίδα 293 του αστικού κώδικα



Εικόνα 50 Απάντηση επόμενης ερώτησης με χρήση της μνήμης της συνομιλίας

Εδώ φαίνεται η ικανότητα του chatbot να απαντάει στην ερώτηση “μπορείς να μου εξηγήσεις το προηγούμενο κείμενο;” χωρίς την αναφορά του, αποδεικνύοντας της χρήσης της μνήμης του.

7 Αξιολόγηση Συστήματος

Για να αξιολογηθεί η αποτελεσματικότητα του συστήματος και το κατά πόσο αυτό λειτουργεί με τον επιθυμητό τρόπο πραγματοποιήθηκε μια σειρά ερωτήσεων προς το σύστημα. Οι απαντήσεις αυτών των ερωτήσεων είναι γνωστές σε εμάς. Μία απάντηση για να θεωρηθεί πλήρως σωστή πρέπει να καλύπτει τρεις βασικούς παράγοντες οι οποίοι είναι:

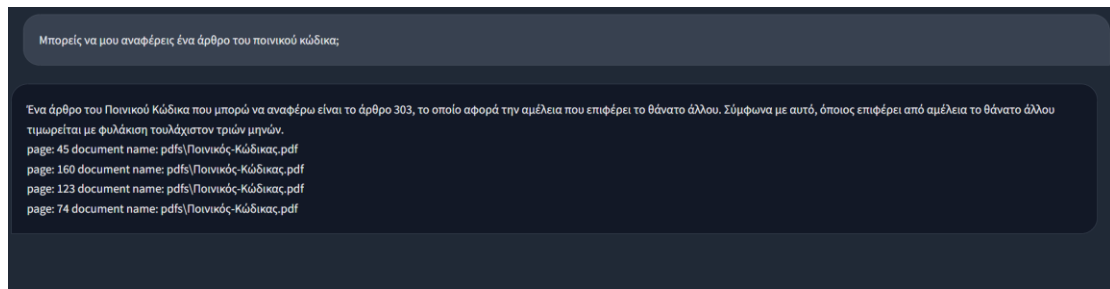
- 1) Το άρθρο ή τις επιμέρους πληροφορίες του (στη συγκεκριμένη περίπτωση του ποινικού και του αγροτικού κώδικα).
- 2) Την επακριβή αναφορά μέσα στο αντίστοιχο αρχείο.
- 3) Την σελίδα στην οποία βρίσκεται.

Η απάντηση που θα καλύπτει και τους τρεις παραπάνω παράγοντες θα επιβραβεύεται με 3 μονάδες (ανώτερη δυνατή βαθμολόγηση) αξιολόγησης, η απάντηση που θα καλύπτει δύο από τους τρεις παράγοντες θα βαθμολογείται με 2 μονάδες, η απάντηση που θα καλύπτει μόνο ένα από τους τρεις παράγοντες θα συγκεντρώνει 1 μονάδα και τέλος η απάντηση που δεν θα καλύπτει κανένα παράγοντα θα αξιολογείται με 0 μονάδες.

Η συγκεκριμένη αλληλεπίδραση ξεκίνησε με μία ερώτηση πιο γενικού περιεχομένου, και στη συνέχεια οι ερωτήσεις που πραγματοποιήθηκαν αναφέρονταν σε πιο ειδικό περιεχόμενο. Τέλος θα παρουσιαστούν όλες οι ερωτήσεις και απαντήσεις που ελήφθησαν ανεξαρτήτου βαθμολογίας.

Η συγκεκριμένη μέθοδος αξιολόγησης είναι εμπνευσμένη από ένα επιστημονικό άρθρο το οποίο μελέτηθηκε κατά την διεκπεραίωση της συγκεκριμένης έρευνας³³.

³³ R. Caspi and P. D. Karp, “An evaluation of ChatGPT and Bard (Gemini) in the context of biological knowledge retrieval,” *Access Microbiology*, vol. 6, no. 6, Jun. 2024, doi: [10.1099/acmi.0.000790.v3](https://doi.org/10.1099/acmi.0.000790.v3).



Εικόνα 51 Πρώτη ερώτηση και απάντηση

Άρθρο: 302
Ημ/νία: 01.01.1951
Ημ/νία Ισχύος: 01.01.1951
Περιγραφή όρου θησαυρού: ΑΝΘΡΩΠΟΚΤΟΝΙΑ ΑΠΟ ΑΜΕΛΕΙΑ

Τίτλος Άρθρου
Ανθρωποκτονία από αμέλεια
Λήμματα
ΠΟΙΝΙΚΟΣ ΚΩΔΙΚΑΣ (Π.Κ.), ΕΓΚΛΗΜΑΤΑ ΚΑΤΑ ΤΗΣ ΖΩΗΣ, ΑΝΘΡΩΠΟΚΤΟΝΙΑ ΑΠΟ ΑΜΕΛΕΙΑ, ΑΠΑΛΛΑΓΗ
Σχόλια
- Το παρόν άρθρο τίθεται όπως αντικαταστάθηκε με το άρθρο 15 του ν. 1419/1984 (Α' 28/14.3.1984).
Κείμενο Άρθρου

- Οποιος επιφέρει από αμέλεια το θάνατο άλλου, τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών.
- Αν το θύμα της πράξης η οποία αναφέρεται στην προηγούμενη παράγραφο είναι οικείος του υπαιτίου, το δικαστήριο μπορεί να απαλλάξει τον υπαίτιο από κάθε ποινή, αν πεισθεί ότι λόγω της ψυχικής οδύνης που υπέστη από τις συνέπειες της πράξης του δε χρειάζεται να υποβληθεί σε ποινή.

Άρθρο: 303
Ημ/νία: 01.01.1951
Ημ/νία Ισχύος: 01.01.1951
Περιγραφή όρου θησαυρού: ΠΑΙΔΟΚΤΟΝΙΑ

Τίτλος Άρθρου
Παιδοκτονία
Λήμματα
ΠΟΙΝΙΚΟΣ ΚΩΔΙΚΑΣ (Π.Κ.), ΕΓΚΛΗΜΑΤΑ ΚΑΤΑ ΤΗΣ ΖΩΗΣ, ΠΑΙΔΟΚΤΟΝΙΑ
Σχόλια
* "ή μετά τον τοκετόν μεν, εφ'όσον όμως", κατά το κείμενο της καθαρεύουσας.

www.dsanet.gr:8380/Webtop/print_law_record.jsp?doc_id=41519&setno=1

160/247

27/8/2019

ΟΘΟΝΗ ΕΚΤΥΠΩΣΗΣ

Κείμενο Άρθρου

Μητέρα που με πρόθεση σκότωσε το παιδί της κατά τον τοκετό [ή μετά τον τοκετό, αλλά ενώ]* εξακολουθούσε ακόμη διατάραξη του οργανισμού της από τον τοκετό, τιμωρείται με κάθειρξη μέχρι δέκα ετών.

Εικόνα 52 Άρθρο 302 & 303 του Ποινικού Κώδικα

Βαθμολογία: 2

Μπορείς να μου αναφέρεις οποιοδήποτε άρθρο του ποινικού κώδικα που να αναφέρεται σε ανθρωποκτονία;

Λυπάμαι, αλλά δεν έχω διαθέσιμες πληροφορίες για άρθρα του ποινικού κώδικα που να αναφέρονται σε ανθρωποκτονία.

page: 29 document name: pdfs\Ποινικός-Κώδικας.pdf

page: 30 document name: pdfs\Ποινικός-Κώδικας.pdf

page: 61 document name: pdfs\Ποινικός-Κώδικας.pdf

page: 160 document name: pdfs\Ποινικός-Κώδικας.pdf

Εικόνα 53 Δεύτερη ερώτηση και απάντηση

Άρθρο: 302

Ημ/νία: 01.01.1951

Ημ/νία Ισχύος: 01.01.1951

Περιγραφή όρου θησαυρού: ΑΝΘΡΩΠΟΚΤΟΝΙΑ ΑΠΟ ΑΜΕΛΕΙΑ

Τίτλος Άρθρου

Ανθρωποκτονία από αμέλεια

Λήμματα

ΠΟΙΝΙΚΟΣ ΚΩΔΙΚΑΣ (Π.Κ.), ΕΓΚΛΗΜΑΤΑ ΚΑΤΑ ΤΗΣ ΖΩΗΣ, ΑΝΘΡΩΠΟΚΤΟΝΙΑ ΑΠΟ ΑΜΕΛΕΙΑ, ΑΠΑΛΛΑΓΗ

Σχόλια

- Το παρόν άρθρο τίθεται όπως αντικαταστάθηκε με το άρθρο 15 του ν. 1419/1984 (Α' 28/14.3.1984).

Κείμενο Άρθρου

1. Οποιος επιφέρει από αμέλεια το θάνατο άλλου, τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών.

2. Αν το θύμα της πράξης η οποία αναφέρεται στην προηγούμενη παράγραφο είναι οικείος του υπαίτιου, το δικαστήριο μπορεί να απαλλάξει τον υπαίτιο από κάθε ποινή, αν πεισθεί ότι λόγω της ψυχικής οδύνης που υπέστη από τις συνέπειες της πράξης του δε χρειάζεται να υποβληθεί σε ποινή.

Άρθρο: 303

Ημ/νία: 01.01.1951

Ημ/νία Ισχύος: 01.01.1951

Περιγραφή όρου θησαυρού: ΠΑΙΔΟΚΤΟΝΙΑ

Τίτλος Άρθρου

Παιδοκτονία

Λήμματα

ΠΟΙΝΙΚΟΣ ΚΩΔΙΚΑΣ (Π.Κ.), ΕΓΚΛΗΜΑΤΑ ΚΑΤΑ ΤΗΣ ΖΩΗΣ, ΠΑΙΔΟΚΤΟΝΙΑ

Σχόλια

* "ή μετά τον τοκετόν μεν, εφ'όσον όμως", κατά το κείμενο της καθαρεύουσας.

www.dsanet.gr:8380/Webtop/print_law_record.jsp?doc_id=41519&setno=1

160/247

Εικόνα 54 Άρθρο 302 του ποινικού κώδικα

Βαθμολογία: 2

Μπορείς να μου αναφέρεις την προβλεπόμενη τιμωρία για την παράβαση του άρθρου 303 που αφορά την παιδοκτονία;

Το άρθρο 303, το οποίο αφορά την παιδοκτονία, προβλέπει ότι όποιος επιφέρει από αμέλεια τον θάνατο άλλου, τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Αν το θύμα είναι οικείος του υπαίτιου, το δικαστήριο μπορεί να απαλλάξει τον υπαίτιο από κάθε ποινή, αν πεισθεί ότι, λόγω της ψυχικής οδύνης που υπέστη από τις συνέπειες της πράξης του, δε χρειάζεται να υποβληθεί σε ποινή.

page: 160 document name: pdfs\Ποινικός-Κώδικας.pdf

page: 171 document name: pdfs\Ποινικός-Κώδικας.pdf

page: 29 document name: pdfs\Ποινικός-Κώδικας.pdf

page: 123 document name: pdfs\Ποινικός-Κώδικας.pdf

Εικόνα 55 Τρίτη ερώτηση & απάντηση

Άρθρο: 302

Ημ/νία: 01.01.1951

Ημ/νία Ισχύος: 01.01.1951

Περιγραφή όρου θησαυρού: ΑΝΘΡΩΠΟΚΤΟΝΙΑ ΑΠΟ ΑΜΕΛΕΙΑ

Τίτλος Άρθρου

Ανθρωποκτονία από αμέλεια

Λήμματα

ΠΟΙΝΙΚΟΣ ΚΩΔΙΚΑΣ (Π.Κ.), ΕΓΚΛΗΜΑΤΑ ΚΑΤΑ ΤΗΣ ΖΩΗΣ, ΑΝΘΡΩΠΟΚΤΟΝΙΑ ΑΠΟ ΑΜΕΛΕΙΑ, ΑΠΑΛΛΑΓΗ

Σχόλια

- Το παρόν άρθρο τίθεται όπως αντικαταστάθηκε με το άρθρο 15 του ν. 1419/1984 (Α' 28/14.3.1984).

Κείμενο Άρθρου

1. Οποιος επιφέρει από αμέλεια το θάνατο άλλου, τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών.
2. Αν το θύμα της πράξης η οποία αναφέρεται στην προηγούμενη παράγραφο είναι οικείος του υπαιτίου, το δικαστήριο μπορεί να απαλλάξει τον υπαίτιο από κάθε ποινή, αν πεισθεί ότι λόγω της ψυχικής οδύνης που υπέστη από τις συνέπειες της πράξης του δε χρειάζεται να υποβληθεί σε ποινή.

Άρθρο: 303

Ημ/νία: 01.01.1951

Ημ/νία Ισχύος: 01.01.1951

Περιγραφή όρου θησαυρού: ΠΑΙΔΟΚΤΟΝΙΑ

Τίτλος Άρθρου

Παιδοκτονία

Λήμματα

ΠΟΙΝΙΚΟΣ ΚΩΔΙΚΑΣ (Π.Κ.), ΕΓΚΛΗΜΑΤΑ ΚΑΤΑ ΤΗΣ ΖΩΗΣ, ΠΑΙΔΟΚΤΟΝΙΑ

Σχόλια

* "ή μετά τον τοκετόν μεν, εφ'όσον όμως", κατά το κείμενο της καθαρεύουσας.

www.dsanet.gr:8380/Webtop/print_law_record.jsp?doc_id=41519&setno=1

160/247

Εικόνα 56 Άρθρο 302 & 303 του Ποινικού Κώδικα

Βαθμολογία: 2

Μπορείς να μου αναφέρεις ένα οποιοδήποτε άρθρο του αγροτικού κώδικα;

Το άρθρο που αναφέρεται στην "Αφάνεια κληρούχου" του Αγροτικού Κώδικα ορίζει ότι η τριετής απουσία στρατευθέντων κληρούχων από την έκδοση της απόφασης της επιτροπής απαλλοτριώσεων, εφόσον δεν έχουν δείξει σημεία ύπαρξης, αποτελεί πλήρη απόδειξη του θανάτου τους, χωρίς να απαιτείται η κήρυξή τους σε αφάνεια σύμφωνα με τις υπάρχουσες διατάξεις.

page: 24 document name: pdfs\Ποινικός-Κώδικας.pdf
page: 65 document name: pdfs\Ποινικός-Κώδικας.pdf
page: 72 document name: pdfs\Αγροτικός-Κώδικας.pdf
page: 43 document name: pdfs\Ποινικός-Κώδικας.pdf

Εικόνα 57 Τέταρτη ερώτηση & απάντηση

72 / 91 | — 189% + | 📄 🔍

Άρθρο: 269
Ημ/νία: 06.12.1949
Ημ/νία Ισχύος: 06.12.1949

Περιγραφή όρου θησαυρού: ΑΓΡΟΤΙΚΟΣ ΚΛΗΡΟΣ

Τίτλος Άρθρου
Αφάνεια κληρούχου
Λήμματα
Αγροτικός Κώδικας, Αφάνεια κληρούχου
Κείμενο Άρθρου
Η από της εκδόσεως της αποφάσεως της επιτροπής απαλλοτριώσεων τριετής απουσία των στρατευθέντων κληρούχων, εάν οι απουσιάζοντες δεν έχωσιν έκτοτε δείξει σημεία υπάρξεως, αποτελεί πλήρη απόδειξιν του θανάτου των, μη απαιτούμενης της κηρύξεως αυτών κατά τας υπάρχουσας διατάξεις εις αφάνειαν.

Βαθμολογία: 3

Μπορείς να μου αναφέρεις οποιοδήποτε άρθρο του αγροτικού κώδικα που να αναφέρεται στον όρο απαλλοτρίωση;

Ναι, το άρθρο 3194/1955 αναφέρεται στην απαλλοτρίωση καλλιεργήσιμων γαιών. Σύμφωνα με αυτό, σε περίπτωση αναγκαστικής απαλλοτρίωσης γαιών ανήκουσων σε κληρούχους ή ιδιοκτήτες γεωργούς για δημόσια έργα, μπορεί να παραχωρηθούν γαίες ίσης αξίας αντί αποζημίωσης.

page: 68 document name: pdfs\Αγροτικός-Κώδικας.pdf
page: 80 document name: pdfs\Αγροτικός-Κώδικας.pdf
page: 72 document name: pdfs\Αγροτικός-Κώδικας.pdf
page: 64 document name: pdfs\Αγροτικός-Κώδικας.pdf

Εικόνα 58 Πέμπτη ερώτηση & απάντηση

68 / 91 | 189% + | [] []

Άρθρο: 263
Ημ/νία: 06.12.1949
Ημ/νία Ισχύος: 06.12.1949
Περιγραφή όρου θησαυρού: ΑΠΑΛΛΟΤΡΙΩΣΗ-ΔΙΑΘΕΣΗ-ΔΙΑΝΟΜΗ ΓΑΙΩΝ-ΚΤΗΜΑΤΩΝ

Λήμματα
Αγροτικός Κώδικας, Απαλλοτρίωση κλήρων, Εγγειοβελτιωτικά έργα

Σχόλια
- Το εντός " " εδάφιο της παρ. 2 προστέθηκε με το άρθρο 61 του Ν. 3194/1955

Κείμενο Άρθρου

1. Εν περιπτώσει αναγκαστικής απαλλοτριώσεως καλλιεργησίμων γαιών ανηκουσών εις κληρούχους ή ιδιοκτήτας κατά κύριον επάγγελμα γεωργούς προς εκτέλεσιν εγγειοβελτιωτικών και εν γένει δημοσίων και κοινής ωφελείας έργων, αντί της οφειλομένης εις τους δικαιούχους κληρούχους ή ιδιοκτήτας αποζημιώσεως παραχωρούνται αυτοίς, εφ' όσον υπάρχουν, τη αιτήσει των, γαίαι ίσης αξίας εξ εκτάσεων ανηκουσών κατά το άρθ. 31 εδάφ. δ' 164 και 197 του παρόντος Κώδικος εις Κοινότητας, Δήμους ή Συνεταιρισμούς αποκαταστάσεως ακτημόνων καλλιεργητών (Σ.Α.Α.Κ.) ή εις το Δημόσιον κατά τα εν τοις επομένοις οριζόμενα.

2. Η κατά τα άνω αίτησις των ενδιαφερομένων υποβάλλεται εις το Υπουργείον Γεωργίας μετά την αναγνώρισιν αυτών ως δικαιούχων κατά την διαδικασίαν του άρθ. 246 του παρόντος Κώδικος.

"Προκειμένου περί πληρούχων αποκατασταθέντων κατά την εποικιστικήν εν γένει Νομοθεσίαν η αναγνώρισις αυτών ως δικαιούχων επί των κατά την προηγουμένην παράγραφον απαλλοτριουμένων γαιών γίνεται επί τη βάσει του οριστικού παραχωρητηρίου συνοδευομένου υπό βεβαιώσεως του οικείου μεταγραφοφύλακος εμφανιούσης τας τυχόν επί του κλήρου επελθούσας μεταβολάς".

3. Μετά την υποβολήν της αιτήσεως ταύτης διαχωρίζεται ίσης αξίας έκτασις εκ των κατά την παράγραφον 1 γαιών υπό τριμελούς Επιτροπής αποτελουμένης εκ του αρμοδίου Οικονομικού Εφόρου, ή του νομίμου αναπληρωτού του, ενός αναπληρωτού του, ενός υπαλλήλου της οικείας Διευθύνσεως Γεωργίας και ενός υπαλλήλου της Τοπογραφικής Υπηρεσίας ή μηχανικού δημοσίας υπηρεσίας ή μηχανικού Δήμων και Κοινοτήτων, των τελευταίων οριζομένων υπό του Υπουργού της Γεωργίας ή των Γενικών Διοικητών. Η Επιτροπή αύτη βεβαίει το ισάξιον των προς ανταλλαγίν εκτάσεων, καταρτίζει δε και τοπογραφικόν σχεδιάγραμμα αυτών.

Εικόνα 59 Άρθρο 263 του Αγροτικού Κώδικα

Βαθμολογία: 2

Μπορείς να μου αναφέρεις ποιά τιμωρία ορίζεται για την παράβαση του άρθρου 281 που αφορά αγροτικά αδικήματα;

Λυπάμαι, αλλά δεν μπορώ να βρω πληροφορίες για το άρθρο 281 που αφορά αγροτικά αδικήματα στο παρεχόμενο κείμενο.
page: 209 document name: pdfs\Ποινικός-Κώδικας.pdf
page: 85 document name: pdfs\Ποινικός-Κώδικας.pdf
page: 85 document name: pdfs\Αγροτικός-Κώδικας.pdf
page: 68 document name: pdfs\Αγροτικός-Κώδικας.pdf

Εικόνα 60 Έκτη ερώτηση & απάντηση

68 / 91 | 189% +

Άρθρο: 281
Ημ/νία: 06.12.1949
Ημ/νία Ισχύος: 06.12.1949
Περιγραφή όρου θησαυρού: ΑΓΡΟΤΙΚΑ ΑΔΙΚΗΜΑΤΑ

Λήμματα
 Αγροτικός Κώδικας, Πώληση εργαλείων επαγγέλματος, Κυρώσεις

Κείμενο Άρθρου

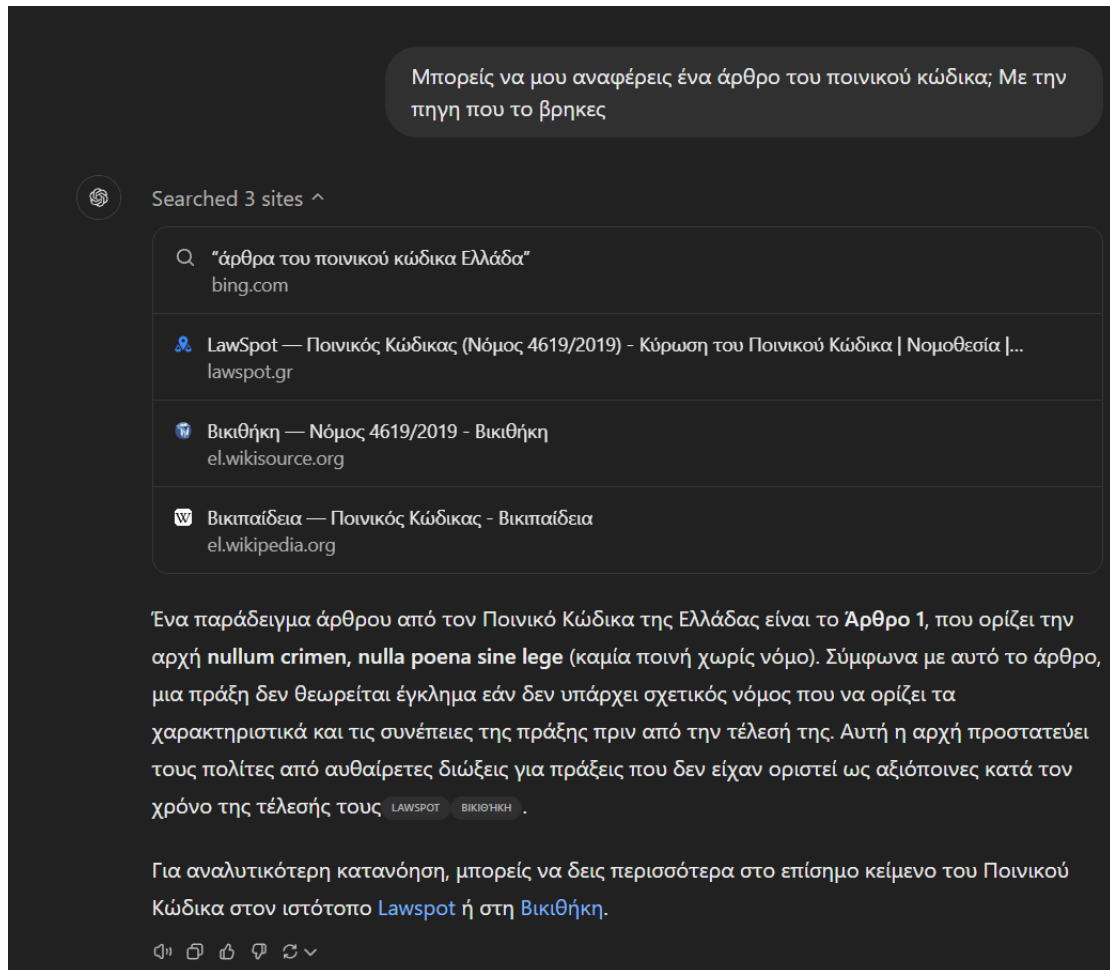
Πρόσφυγες ή γηγενείς πωλούντες άνευ εγγράφου αδείας του Γραφείου Γεωργικής Περιφέρειας, της εγκαταστάσεώς των, τα αγοραζόμενα διά χρημάτων προερχομένων εκ δανείου του Κράτους ή της τέως Επιτροπής Αποκαταστάσεως Προσφύγων ή Νομικών Προσώπων τη εγγυήσει του Κράτους αροτριώντα και λοιπά εφόδια αναγκαία διά την εξάσκησιν του επαγγέλματός των διώκονται και τιμωρούνται κατά τα περί υπεξαίρεσεως διατάξεις του Ποινικού Νόμου.

Εικόνα 61 Άρθρο 281 του Αγροτικού Κώδικα

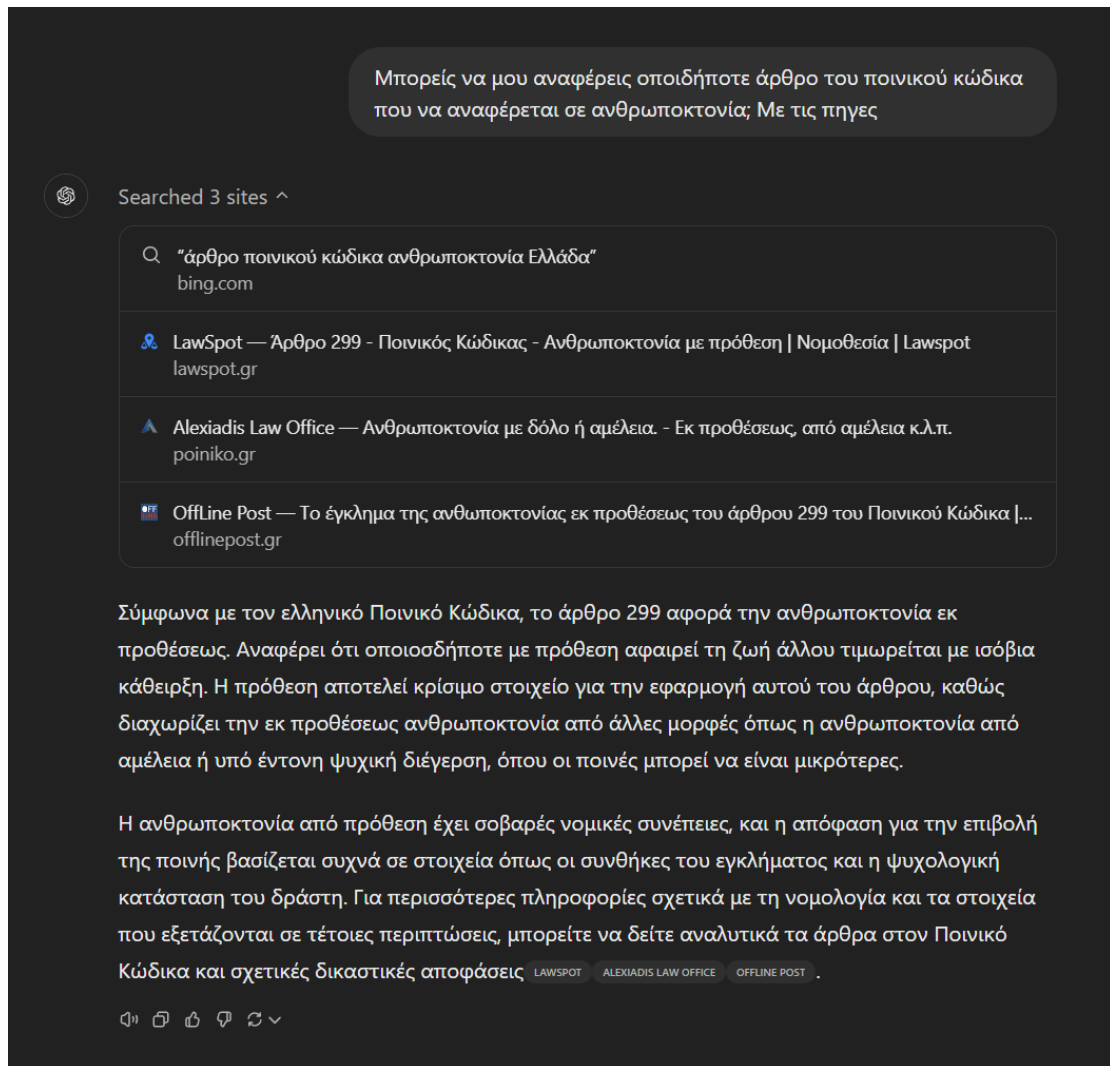
Βαθμολογία: 0

Ερώτηση	Βαθμολογία
Μπορείς να μου αναφέρεις ένα άρθρο του ποινικού κώδικα;	2
Μπορείς να μου αναφέρεις οποιοδήποτε άρθρο του ποινικού κώδικα που να αναφέρεται σε ανθρωποκτονία;	2
Μπορείς να μου αναφέρεις την προβλεπόμενη τιμωρία για την παράβαση του άρθρου 303 που αφορά την παιδοκτονία;	2
Μπορείς να μου αναφέρεις ένα οποιοδήποτε άρθρο του αγροτικού κώδικα;	3
Μπορείς να μου αναφέρεις οποιοδήποτε άρθρο του αγροτικού κώδικα που να αναφέρεται στον όρο απαλλοτρίωση;	2
Μπορείς να μου αναφέρεις ποιά τιμωρία ορίζεται για την παράβαση του άρθρου 281 που αφορά αγροτικά αδικήματα;	0

7.1 Παρόμοιες ερωτήσεις στο ChatGPT



Εικόνα 62 Ερώτηση 1 στο ChatGPT



Εικόνα 63 Ερώτηση 2 στο ChatGPT

Παρατηρείτε πως το ChatGPT δεν αντλεί πληροφορίες από επίσημες πηγές αλλά από άρθρα που κυκλοφορούνε στο διαδίκτυο, με αποτέλεσμα να μην παρουσιάζονται οι νόμοι ή τα νομικά άρθρα αυτολεξεί, με πιθανότητα παραπληροφόρησης ή μη σωστής ερμηνείας.

8 Συμπεράσματα και Προοπτικές

8.1 Συμπεράσματα επί των αποτελεσμάτων

Η παρούσα διπλωματική εργασία έχει ως στόχο την ανάπτυξη ενός εξειδικευμένου GPT chatbot για την ελληνική νομοθεσία. Η συνεισφορά της συγκεκριμένης έρευνας είναι πολύπλευρη. Αρχικά, αναπτύχθηκε ένα πρωτοποριακό εργαλείο που μπορεί να κατανοεί και να απαντά σε ερωτήματα σχετικά με την ελληνική νομοθεσία, γεγονός που βελτιώνει την πρόσβαση σε νομικές πληροφορίες τόσο για επαγγελματίες όσο και για το ευρύ κοινό.

Η έρευνα αυτή πέτυχε την ανάπτυξη του μοντέλου GPT με εξειδίκευση στην ελληνική γλώσσα και νομοθεσία, καθώς και τη δημιουργία μιας λειτουργικής διεπαφής χρήστη, η οποία επιτρέπει την αλληλεπίδραση με το bot και την ανάκτηση σχετικών νομικών πληροφοριών.

Ωστόσο, υπήρχαν και ορισμένες προκλήσεις. Σε κάποιες περιπτώσεις, η αναγνώριση πιο περίπλοκων νομικών όρων και φράσεων δεν επιτεύχθηκε με την επιθυμητή ακρίβεια. Επιπλέον, παρατηρήθηκαν περιορισμοί στην απόδοση του μοντέλου λόγω της πολυπλοκότητας της ελληνικής γλώσσας και της ποικιλίας των νομικών εγγράφων.

Τέλος, θα πρέπει να σημειωθεί ότι η ανάγκη για μεγάλα σύνολα δεδομένων για την εκπαίδευση του μοντέλου αποτέλεσε έναν σημαντικό περιοριστικό παράγοντα. Επίσης, η περιορισμένη πρόσβαση σε πλήρως ενημερωμένα και αξιόπιστα νομικά κείμενα καθιστούσε τη διαδικασία πιο απαιτητική.

8.2 Συμπεράσματα επί της μεθοδολογίας

Στα συμπεράσματα επί της μεθοδολογίας, αναγνωρίζονται ορισμένες αδυναμίες όσο αναφορά την προσέγγιση που χρησιμοποιήθηκε στην ανάπτυξη του chatbot. Μία από τις κύριες προκλήσεις ήταν η πολυπλοκότητα της επεξεργασίας φυσικής γλώσσας (NLP) για την ελληνική γλώσσα, η οποία απαιτούσε επιπλέον προσπάθεια και εξειδικευμένες λύσεις. Η πολυπλοκότητα αυτή επηρέασε την απόδοση του μοντέλου και την ακρίβεια των απαντήσεων του chatbot.

Επιπλέον, η ανάγκη για εξειδικευμένα δεδομένα αποτέλεσε έναν ακόμη σημαντικό περιορισμό. Η δυσκολία στην «αλλίευση» αυτών των δεδομένων, τα οποία είναι απαραίτητα για την αποτελεσματική εκπαίδευση του μοντέλου, είχε αντίκτυπο στην απόδοση του συστήματος. Η έλλειψη πρόσβασης σε πλούσια και ενημερωμένα σύνολα δεδομένων δυσχεραίνει την πλήρη αξιοποίηση των δυνατοτήτων της μεθοδολογίας.

8.3 Συμπεράσματα επί των εργαλείων

Τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη του chatbot, όπως το OpenAI API και το Gradio, ανταποκρίθηκαν επαρκώς στις απαιτήσεις της έρευνας και της υλοποίησης του συστήματος. Το OpenAI API παρείχε τις απαραίτητες δυνατότητες για την ανάπτυξη του chatbot, ενώ το Gradio προσέφερε μια εύκολη και γρήγορη λύση για τη δημιουργία της διεπαφής χρήστη.

Ωστόσο, υπήρξαν συγκεκριμένοι περιορισμοί στα εργαλεία αυτά. Το OpenAI API παρουσίασε εν γένει προκλήσεις, όπως το κόστος και η οριοθετημένη πρόσβαση σε προηγμένα χαρακτηριστικά, γεγονός που επηρέασε την έκταση των πειραμάτων που μπορούσαν να διεξαχθούν. Παρόλο που το Gradio ήταν εύχρηστο, παρουσίασε περιορισμένες δυνατότητες στην προσαρμογή της διεπαφής χρήστη, κάτι που κατέστειλε την ευελιξία στην τελική παρουσίαση του chatbot.

Εκτός από τα βασικά αυτά εργαλεία, χρησιμοποιήθηκαν και άλλα εργαλεία, όπως το Langchain, για τη βελτίωση της επεξεργασίας κειμένου και την ενσωμάτωση διανυσματικών αναπαραστάσεων. Αυτά τα εργαλεία συνέβαλαν σημαντικά στην ενίσχυση της λειτουργικότητας και της απόδοσης του chatbot, παρέχοντας επιπλέον δυνατότητες που δεν ήταν διαθέσιμες με τα αρχικά εργαλεία.

8.4 Προοπτικές/ Επόμενα Βήματα

Οι δυνατότητες επαναχρησιμοποίησης και εξέλιξης του chatbot είναι πολυδιάστατες. Αρχικά, η ανάπτυξη του μπορεί να επεκταθεί ώστε να συμπεριλάβει περισσότερα νομικά κείμενα, γεγονός που θα βελτιώσει την ακρίβεια των απαντήσεων. Παράλληλα, η ενσωμάτωση πιο προηγμένων τεχνικών επεξεργασίας φυσικής γλώσσας (NLP) και μηχανικής μάθησης θα μπορούσε να ενισχύσει την κατανόηση και την απόδοση του μοντέλου. Εξίσου σημαντική είναι η διερεύνηση συνεργασιών με νομικούς φορείς και οργανισμούς για την παροχή περισσότερων και πιο εξειδικευμένων δεδομένων, η οποία θα υποστηρίξει την εξέλιξη του συστήματος.

Επιπλέον, η δημιουργία μιας μόνιμης βάσης δεδομένων για την αποθήκευση των embeddings θα μπορούσε να συμβάλει στην αποφυγή της επαναδημιουργίας τους σε κάθε εκτέλεση, αυξάνοντας έτσι την αποδοτικότητα του. Παράλληλα, η επέκταση της δυνατότητας του chatbot να χειρίζεται και άλλους τύπους αρχείων πέρα από τα PDF, όπως DOCX, TXT, και HTML, θα διευρύνει το φάσμα των εφαρμογών του. Τέλος, η διεξαγωγή δοκιμών από δικηγόρους και νομικούς παράγοντες θα βελτιώσει την ακρίβεια και την αποτελεσματικότητα του συστήματος, μέσω πρακτικής χρήσης και ανατροφοδότησης.

Στις προτάσεις για μελλοντική έρευνα περιλαμβάνεται η επέκταση της τελευταίας για την κάλυψη περισσότερων τομέων της ελληνικής νομοθεσίας. Επίσης, η ανάπτυξη νέων μοντέλων και αλγορίθμων, όπως την βελτιστοποίηση της διαίρεσης των κειμένων και της ανάκτησης τους, που θα ανταποκρίνονται καλύτερα στις ανάγκες της νομικής κοινότητας είναι καίριας σημασίας.

Επιπροσθέτως, η διερεύνηση της δυνατότητας ενσωμάτωσης του chatbot σε πραγματικά νομικά συστήματα και πλατφόρμες παροχής νομικών υπηρεσιών μπορεί να προσφέρει μελλοντικά οφέλη. Η χρήση μοντέλων μεγάλης κλίμακας ανοιχτού κώδικα (LLM) για εκπαίδευση στην ελληνική γλώσσα και σε νομικά ζητήματα θα μπορούσε να αμβλύνει το κόστος και να βελτιώσει την ασφάλεια, παρέχοντας τη δυνατότητα χρήσης του chatbot χωρίς την ανάγκη σύνδεσης στο διαδίκτυο ή τη συλλογή ευαίσθητων δεδομένων από τρίτους όπως το OpenAI.

Με αυτές τις κατευθύνσεις, η παρούσα διπλωματική εργασία όχι μόνο επιτυγχάνει τους αρχικούς στόχους της αλλά και θέτει τα θεμέλια για μελλοντική εξέλιξη και βελτίωση στον τομέα της τεχνητής νοημοσύνης και της νομικής επιστήμης.

9 Αναφορές

- [1] D. M. Katz, M. J. Bommarito, and J. Blackman, “A general approach for predicting the behavior of the Supreme Court of the United States,” *PLOS ONE*, vol. 12, no. 4, p. e0174698, Apr. 2017, doi: 10.1371/journal.pone.0174698.
- [2] “Chroma |  LangChain.” Accessed: Aug. 27, 2024. [Online]. Available: <https://python.langchain.com/v0.1/docs/integrations/vectorstores/chroma/>
- [3] “Cleo | AI meets money.” Accessed: Aug. 27, 2024. [Online]. Available: <https://web.meetcleo.com/>
- [4] “Conversation Buffer |  LangChain.” Accessed: Aug. 27, 2024. [Online]. Available: <https://python.langchain.com/v0.1/docs/modules/memory/types/buffer/>
- [5] “Erica - Virtual Financial Assistant | Bank of America.” Accessed: Aug. 27, 2024. [Online]. Available: <https://promotions.bankofamerica.com/digitalbanking/mobilebanking/erica>
- [6] S. Bolinger, “Everything I’ve Learned Building an AI App,” GoDaddy Blog. Accessed: Aug. 27, 2024. [Online]. Available: <https://www.godaddy.com/resources/news/everything-ive-learned-building-an-ai-app>
- [7] “Get Firefox browser — Mozilla (US),” Mozilla. Accessed: Aug. 27, 2024. [Online]. Available: <https://www.mozilla.org/en-US/firefox/>
- [8] G. Team, “Gradio.” Accessed: Aug. 27, 2024. [Online]. Available: <https://gradio.app>
- [9] G. Team, “Gradio Dropdown Docs.” Accessed: Aug. 27, 2024. [Online]. Available: <https://www.gradio.app/docs/gradio/dropdown>
- [10] G. Team, “Gradio Interface Docs.” Accessed: Aug. 27, 2024. [Online]. Available: <https://www.gradio.app/docs/gradio/interface>
- [11] G. Team, “Gradio Textbox Docs.” Accessed: Aug. 27, 2024. [Online]. Available: <https://www.gradio.app/docs/gradio/textbox>
- [12] Pieces , “How to Build a Langchain PDF Chatbot,” Pieces for Developers. Accessed: Aug. 27, 2024. [Online]. Available: <https://medium.com/getpieces/how-to-build-a-langchain-pdf-chatbot-b407fcd750b9>
- [13] “LangChain.” Accessed: Aug. 27, 2024. [Online]. Available: <https://www.langchain.com/>
- [14] I. R. PhD, “LangChain 101: Part 2ab. All You Need to Know About (Large Language) Models,” Medium. Accessed: Aug. 27, 2024. [Online]. Available: <https://pub.towardsai.net/langchain-101-part-2ab-all-you-need-to-know-about-large-language-models-3512ae41dfc3>
- [15] T. B. Brown *et al.*, “Language Models are Few-Shot Learners,” Jul. 22, 2020, *arXiv*: arXiv:2005.14165. Accessed: Aug. 26, 2024. [Online]. Available: <http://arxiv.org/abs/2005.14165>
- [16] “OpenAI API | OpenAI.” Accessed: Aug. 27, 2024. [Online]. Available: <https://openai.com/index/openai-api/>
- [17] “pip documentation v24.2.” Accessed: Aug. 27, 2024. [Online]. Available: <https://pip.pypa.io/en/stable/>
- [18] P. Lewis *et al.*, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” May 2020.
- [19] “RetrievalTutorials/tutorials/LevelsOfTextSplitting/5_Levels_Of_Text_Splitting.ipynb at main · FullStackRetrieval-com/RetrievalTutorials,” GitHub. Accessed: Aug. 27, 2024. [Online]. Available: https://github.com/FullStackRetrieval-com/RetrievalTutorials/blob/main/tutorials/LevelsOfTextSplitting/5_Levels_Of_Text_Splitting.ipynb
- [20] “Semantic Search — Sentence Transformers documentation.” Accessed: Aug. 27, 2024. [Online]. Available: <https://sbert.net/examples/applications/semantic-search/README.html>
- [21] “VSCodium - Open Source Binaries of VSCode.” Accessed: Aug. 27, 2024. [Online]. Available: <https://vscodium.com/#why>
- [22] “Το Δίκαιο στην εποχή της Τεχνητής Νοημοσύνης,” Lawspot. Accessed: Aug. 26, 2024. [Online]. Available: <https://www.lawspot.gr/node/263273>

[23] R. Caspi and P. D. Karp, "An evaluation of ChatGPT and Bard (Gemini) in the context of biological knowledge retrieval," Access Microbiology, vol. 6, no. 6, Jun. 2024, doi: 10.1099/acmi.0.000790.v3.

[24] "ChunkViz." Accessed: Nov. 08, 2024. [Online]. Available: <https://chunkviz.up.railway.app/>

10 Παράρτημα

10.1 Κώδικας και αποτελέσματα

```
1. import os
2. import openai
3. import gradio
4.
5. from apikey import apikey
6. from dikaia import dikaia
7.
8. #αρχικοποίηση κλειδιού
9. openai.api_key = apikey
10.
11. def ask_question_with_instruction(instruction, question):
12.     instruction = "Είσαι δικηγόρος εξειδικευμένος στο " + instruction
13.     # Πρόσθεση ερώτησης με το δίκαιο
14.     prompt = f"{instruction}\n{question}"
15.     # Παραγωγή απάντησης με το GPT
16.     response = openai.ChatCompletion.create(
17.         model = "gpt-3.5-turbo",
18.         messages = [{"role": "user", "content": prompt}],
19.         max_tokens = 1024
20.     )
21.
22.     return response["choices"][0]["message"]["content"].strip()
23.
24. def main():
25.     iface = gradio.Interface(
26.         ask_question_with_instruction,
27.         [
28.             #οι παράμετροι που θα δωθούνε
29.             gradio.Dropdown(choices = dikaia, label = "Διάλεξε ένα δίκαιο"),
30.             gradio.Textbox(lines = 2, label = "Κάνε την ερώτηση σου εδώ:)],
31.             #τι επιστρέφει
32.             "text",
33.             title = "GPT-based Chatbot για την ελληνική νομοθεσία",
34.             description = "Διάλεξε ένα δίκαιο από την λίστα και κάνε την ερώτηση σου",
35.         )
36.     iface.launch()
37.
38. if __name__ == "__main__":
39.     main()
40.
```

GPT-based Chatbot για την ελληνική νομοθεσία

Διάλεξε ένα βιβλίο από την λίστα και κάνε την ερώτησή σου

Διάλεξε ένα βιβλίο

Αλφεινικός Κώδικας

Κάνε την ερώτησή σου εδώ:

Δώσε μου τον πρώτο νόμο του αλφεινικού κώδικα

Clear Submit

4.6s

Flag

Εικόνα 64 Αποτέλεσμα του παραπάνω κώδικα

```

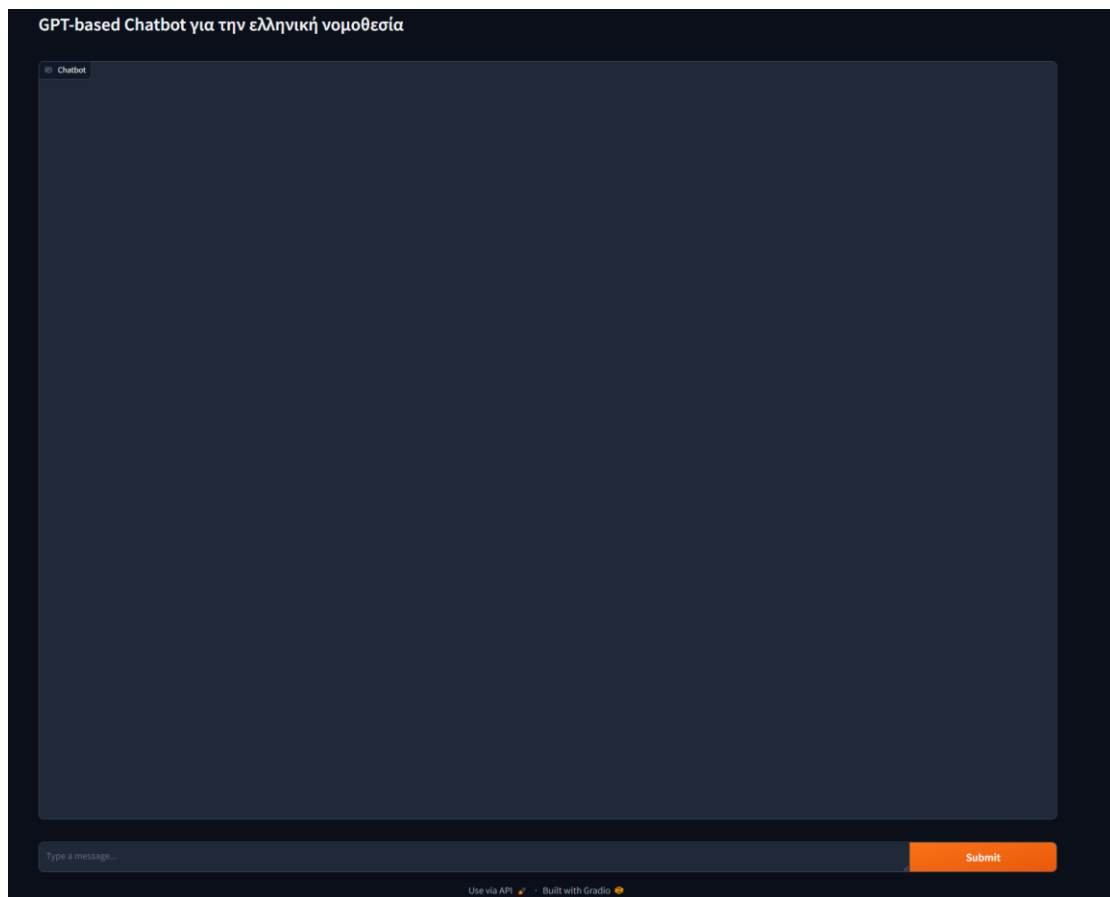
1. import os
2.
3. import openai
4. import gradio as gr
5. import glob
6.
7. from apikey import apikey
8.
9. from langchain_experimental.text_splitter import SemanticChunker
10.
11. from langchain_openai.embeddings import OpenAIEmbeddings
12. from langchain_openai.chat_models import ChatOpenAI
13.
14. from langchain_community.vectorstores import Chroma
15. from langchain_community.document_loaders import DirectoryLoader
16.
17. from langchain.memory import ConversationBufferMemory
18. from langchain.chains import ConversationalRetrievalChain
19.
20. from langchain_community.document_loaders import PyPDFLoader
21.
22.
23. os.environ["OPENAI_API_KEY"] = apikey
24.
25. #συλλογή κειμένου από τα pdf
26. def pdf_read(pdf_path):
27.     pdf_files = glob.glob("%s/*.pdf" % pdf_path)
28.
29.     pages = []
30.
31.     embeddings = OpenAIEmbeddings()
32.     text_splitter = SemanticChunker(embeddings, breakpoint_threshold_type =
"percentile")
33.
34.
35.     for pdf in pdf_files:
36.         print(pdf)
37.         loader = PyPDFLoader(pdf)
38.         pages.extend(text_splitter.split_documents(loader.load()))
39.
40.     return pages
41.
42.
43. #χωρισμός του κειμένου σε chunks
44. def get_chunks(text):
45.     embeddings = OpenAIEmbeddings()
46.     text_splitter = SemanticChunker(embeddings, breakpoint_threshold_type =
"percentile")
47.
48.     chunks = text_splitter.split_text(text)
49.
50.     return chunks
51.
52. #δημιουργία βάσης πληροφοριών για το LLM μοντέλο

```

```

53. def get_vectorstore(text_chunks):
54.     embeddings = OpenAIEmbeddings()
55.     vectorstore = Chroma.from_documents(text_chunks, embeddings)
56.
57.     return vectorstore
58.
59.
60. def conversation_chain(vectorstore):
61.     llm = ChatOpenAI(model = "gpt-4o")
62.     memory = ConversationBufferMemory(memory_key = 'chat_history', return_messages
= True, output_key='answer')
63.
64.     conversationchain = ConversationalRetrievalChain.from_llm(
65.         llm = llm,
66.         retriever = vectorstore.as_retriever(),
67.         memory = memory,
68.         return_source_documents=True,
69.     )
70.
71.     return conversationchain
72.
73. def chat_function(user_input, history):
74.
75.     response = conversation.invoke(user_input)
76.     resources = ''
77.
78.     for page in response["source_documents"]:
79.         for metadata in page:
80.             if(metadata[0] == 'metadata'):
81.                 resources += "page: " + str(metadata[1]['page'] + 1)
82.                 resources += " document name: " + metadata[1]['source'] + "\n"
83.
84.     print(resources)
85.
86.     return response["answer"] + "\n" + resources
87.
88.
89. def main():
90.
91.     chat_history = []
92.
93.     documents = pdf_read("pdfs")
94.     vectorstore = get_vectorstore(documents)
95.
96.     global conversation
97.     conversation = conversation_chain(vectorstore)
98.
99.     gr.ChatInterface(fn = chat_function, retry_btn = None, undo_btn = None,
clear_btn = None, title = "GPT-based Chatbot για την ελληνική νομοθεσία").launch(share =
False)
100.
101.
102. if __name__ == "__main__":
103.     main()
104.

```



Εικόνα 65 Τελικό αποτέλεσμα