Πανεπιστήμιο Πειραιώς Τμήμα Πληροφορικής Ακαδημαϊκό Έτος 2021-2022

Απαλλακτική Εργασία στο Μάθημα της Επεξεργασίας Φυσικής Γλώσσας 6ου Εξαμήνου

Επιμέλεια: Βαλή Σαράφογλου Ευρυδίκη Π19023 (evrivali@gmail.com)

Επιβλέπων Καθηγητής: Θεμιστοκλής Παναγιωτόπουλος

Περιεχόμενα

Περιγραφή Εργασίας

Η παρούσα εργασία αποτελείται από 4 ασκήσεις που αφορούν στην ανάπτυξη Λεκτικού, Συντακτικού και Σημασιολογικού Αναλυτή, σε πρώτο λόγο, στην ενημέρωση μιας σχετικής Βάσης Γνώσης σε δεύτερο λόγο, και στην πραγματοποίηση ερωταποκρίσεων στην Βάση Γνώσης σχετικά με το εν λόγω κείμενο. Από τα διαθέσιμα θέματα εμείς επιλέξαμε την εργασία 2 : Ανάπτυξη & Τεκμηρίωση Λεκτικού, Συντακτικού και Σημασιολογικού Αναλυτή και Διαχείριση Βάσης Γνώσης σε άλλη γλώσσα Προγραμματισμού. Η επιλεγμένη γλώσσα προγραμματισμού είναι η **PYTHON**. Η λύση μιας άσκησης αποτελείται από τον πηγαίο κώδικα και τα συνοδευτικά αρχεία καθώς και την επεξήγηση που περιγράφει την διαδικασία υλοποίησης.

Θέμα 1

Εκφώνηση

Ανάπτυξη Λεκτικού Αναλυτή σε άλλη γλώσσα Προγραμματισμού. Αναζητείστε στο διαδίκτυο ή αναπτύξτε εσείς Λεκτικό Αναλυτή που διαβάζει μια μικρή ιστορία και μπορεί να παράγει μια λίστα από προτάσεις, κάθε μια από τις οποίες περιέχει μια λίστα από λέξεις. Τεκμηριώστε πειστικά τον κώδικά σας.

Λύση

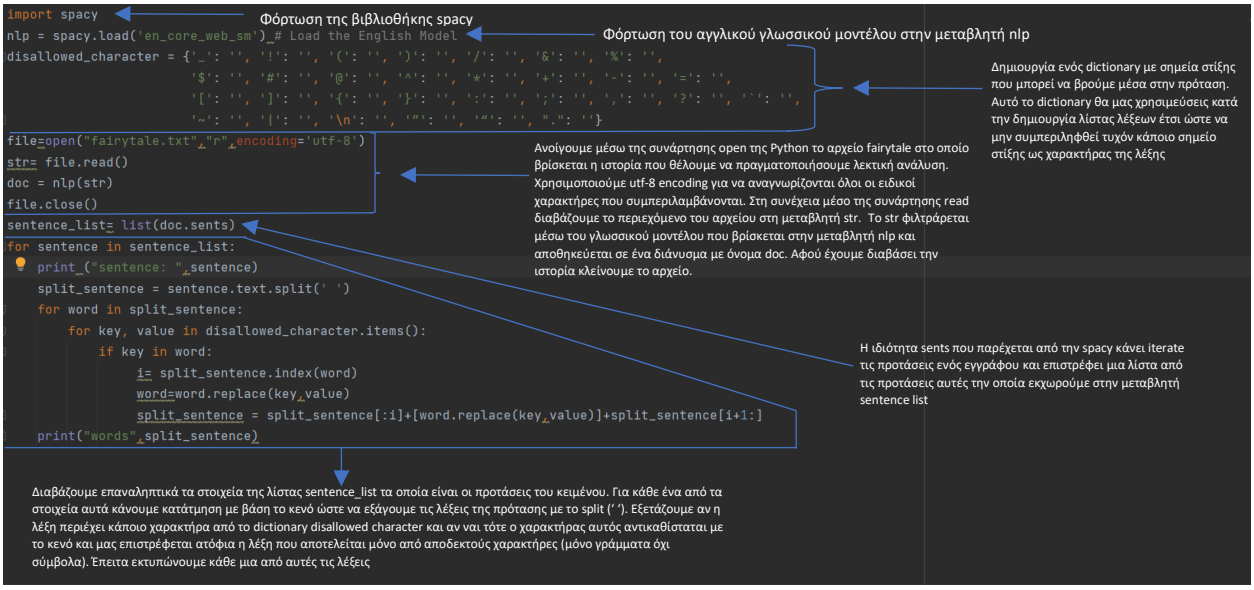
Για την υλοποίηση του θέματος 1 χρησιμοποιήθηκε η γλώσσα προγραμματισμού **PYTHON** και η βιβλιοθήκη **spaCy**.

Λίγα λόγια για την **Spacy**:

Η spaCy είναι μια βιβλιοθήκη λογισμικού ανοιχτού κώδικα για προηγμένη επεξεργασία φυσικής γλώσσας, γραμμένη στις γλώσσες προγραμματισμού Python και Cython. Η spaCy μεταξύ άλλων λειτουργιών, εκτελεί την τμηματοποίηση προτάσεων με πολύ μεγαλύτερη ακρίβεια σε σχέση με άλλες μεθόδος (π.χ split).

Κατασκευή Λεξικού αναλυτή:

Ο λεξικός αναλυτής θα δέχεται ως είσοδο ένα κείμενο το οποίο θα βρίσκεται μέσα σε ένα txt αρχείο και θα δίνει ως έξοδο το κείμενο χωρισμένο σε προτάσεις και αντίστοιχα τις προτάσεις χωρισμένες σε λέξεις. Η πραγματοποίηση της λεξικής ανάλυσης θα γίνει με τη βοήθεια της βιβλιοθήκης spaCy μέσω της οποίας θα γίνει η φόρτωση του γλωσσικού μοντέλου και (λεξιλόγιο, σύνταξη, οντότητες κλπ). Στη συνέχεια το κείμενο αυτό θα αποθηκευτεί σε μια μεταβλητή και πάλι με τη βοήθεια της βιβλιοθήκης spaCy θα χωριστεί σε προτάσεις (θεωρούμε πρόταση μια ακολουθία χαρακτήρων/λέξεων μέχρι να συναντήσουμε τελεία) και έπειτα αυτές οι προτάσεις θα χωριστούν σε μια λίστα από λέξεις. Ας δούμε αναλυτικά τη διαδικασία υλοποίησης μέσα από τον κώδικα:



Εκτέλεση κώδικα και αναφορά ελέγχου αποτελεσμάτων:

Είσοδος:

fairytale.txt

*fairytale.txt - Notepad

File Edit Format View Help

```
SHREK is grumpy, smelly, and an ugly ogre living peacefully in a swamp.
One night he suddenly finds his land has been inhabited by a mass of fairy-tale creatures (Pinocchio, the three little pigs, Peter Pan, Snow White, and Cinderella among others), who have been banished by the evil Lord Farquaad.
One of these is Donkey (who just won't shut up).
Accompanied by him, Shrek confronts Lord Farquaad,demanding his land back.
Farquaad gives him a deal, telling him to rescue Princess Fiona from a dragon-guarded castle, to get his swamp back.
Shrek and Donkey go and rescue the Princess, narrowly avoiding being burned by the Dragon, who tries to seduce Donkey (being a girl Dragon!).
Fiona is disgruntled about being rescued by an ugly ogre, rather than Prince Charming.
However, despite their differences, she and Shrek grow fond of each other.
On the second night of their return journey, Fiona hides in a windmill.
Donkey finds her, and discovers that she's turned into an ogress.
She explains that she's under a spell, which can only be broken by true love's first kiss.
Shrek overhears part of this, but misunderstands and thinks she doesn't love him because he's ugly.
Just as she decides to tell Shrek the truth, the sun rises and she becomes a beautiful princess again.
Lord Farquaad arrives and takes Fiona to his castle, and Shrek returns to his swamp. Both are miserable.
Later, Shrek, Donkey, and Dragon head to Farquaad's castle, to try to stop Fiona marrying him. He tells Fiona of Farquaad's true intentions of marrying her.
When they arrive, the sun sets and Fiona becomes an ogress again.
Farquaad doesn't want to marry her, and sets the guards on everybody, but Dragon comes to the rescue and eats Farquaad up. Shrek and Fiona kiss.
Therefore, Fiona stays ugly (in conformity with the spell: "you find true love's first kiss and then take love's true form"). Fiona will forever be an ugly ogre like Shrek.
She marries Shrek in a big wedding with all the fairy tale creatures, and they live "ugly" ever after.
```

Έξοδος:

```
C:\Users\evriv\PycharmProjects\NLP\venv\Scripts\python.exe C:/Users/evriv/PycharmProjects/NLP/lexical_analysis.py
sentence:  SHREK is grumpy, smelly, and an ugly ogre living peacefully in a swamp.
words ['SHREK', 'is', 'grumpy', 'smelly', 'and', 'an', 'ugly', 'ogre', 'living', 'peacefully', 'in', 'a', 'swamp']
sentence:  One night he suddenly finds his land has been inhabited by a mass of fairy-tale creatures (Pinocchio, the three little pigs, Peter Pan, Snow White, and Cinderella
 among others), who have been banished by the evil Lord Farquaad.
words ['One', 'night', 'he', 'suddenly', 'finds', 'his', 'land', 'has', 'been', 'inhabited', 'by', 'a', 'mass', 'of', 'fairytale', 'creatures', 'Pinocchio', 'the', 'three',
 'little', 'pigs', 'Peter', 'Pan', 'Snow', 'White', 'and', 'Cinderella', 'among', 'others', 'who', 'have', 'been', 'banished', 'by', 'the', 'evil', 'Lord', 'Farquaad']
sentence:  One of these is Donkey (who just won't shut up).
words ['One', 'of', 'these', 'is', 'Donkey', 'who', 'just', 'won't', 'shut', 'up']
sentence:  Accompanied by him, Shrek confronts Lord Farquaad,demanding his land back.
words ['Accompanied', 'by', 'him', 'Shrek', 'confronts', 'Lord', 'Farquaaddemanding', 'his', 'land', 'back']
sentence:  Farquaad gives him a deal, telling him to rescue Princess Fiona from a dragon-guarded castle, to get his swamp back.
words ['Farquaad', 'gives', 'him', 'a', 'deal', 'telling', 'him', 'to', 'rescue', 'Princess', 'Fiona', 'from', 'a', 'dragonguarded', 'castle', 'to', 'get', 'his', 'swamp',
 'back']
sentence:
Shrek and Donkey go and rescue the Princess, narrowly avoiding being burned by the Dragon, who tries to seduce Donkey (being a girl Dragon!).
words ['Shrek', 'and', 'Donkey', 'go', 'and', 'rescue', 'the', 'Princess', 'narrowly', 'avoiding', 'being', 'burned', 'by', 'the', 'Dragon', 'who', 'tries', 'to', 'seduce',
 'Donkey', 'being', 'a', 'girl', 'Dragon']
sentence:  Fiona is disgruntled about being rescued by an ugly ogre, rather than Prince Charming.
words ['Fiona', 'is', 'disgruntled', 'about', 'being', 'rescued', 'by', 'an', 'ugly', 'ogre', 'rather', 'than', 'Prince', 'Charming']
sentence:  However, despite their differences, she and Shrek grow fond of each other.
words ['However', 'despite', 'their', 'differences', 'she', 'and', 'Shrek', 'grow', 'fond', 'of', 'each', 'other']
sentence:
On the second night of their return journey, Fiona hides in a windmill.
words ['On', 'the', 'second', 'night', 'of', 'their', 'return', 'journey', 'Fiona', 'hides', 'in', 'a', 'windmill']
sentence:  Donkey finds her, and discovers that she's turned into an ogress.
words ['Donkey', 'finds', 'her', 'and', 'discovers', 'that', 'she's', 'turned', 'into', 'an', 'ogress']
sentence:  She explains that she's under a spell, which can only be broken by true love's first kiss.
words ['She', 'explains', 'that', 'she's', 'under', 'a', 'spell', 'which', 'can', 'only', 'be', 'broken', 'by', 'true', 'love's', 'first', 'kiss']
sentence:  Shrek overhears part of this, but misunderstands and thinks she doesn't love him because he's ugly.
```

Όπως βλέπουμε το κείμενο έχει χωριστεί σε προτάσεις. Κάθε πρόταση εμφανίζεται ως “sentence:” και μετά ακολουθεί το κείμενο της πρότασης (κόκκινο πλαίσιο). Ακριβώς από κάτω βρίσκεται μια λίστα από τις λέξεις της εκάστοτε πρότασης (μπλε πλαίσιο). Όπως παρατηρούμε κάποιες λέξεις περιέχουν ειδικά σύμβολα (παρενθέσεις, σημεία στίξης) (πράσινο πλαίσιο). Τα σύμβολα αυτά όμως έχουν αφαιρεθεί στην προβαλλόμενη λίστα με τις λέξεις ακολουθώντας τη διαδικασία που έχει περιγραφεί παραπάνω. Ο λεκτικός αναλυτής επομένως λειτουργεί ορθά αφού πραγματοποιεί τις ζητούμενες λειτουργίες.

Θέμα 2

Εκφώνηση

Ανάπτυξη Συντακτικού Αναλυτή σε άλλη γλώσσα Προγραμματισμού. Αναζητείστε στο διαδίκτυο ή αναπτύξτε εσείς Συντακτικό Αναλυτή που με βάση τους κανόνες συντακτικής ανάλυσης της πρότυπης λύσης σε Prolog που σας δόθηκε παράγει το συντακτικό δένδρο της πρότασης. Τεκμηριώστε πειστικά τον κώδικά σας.

Λύση

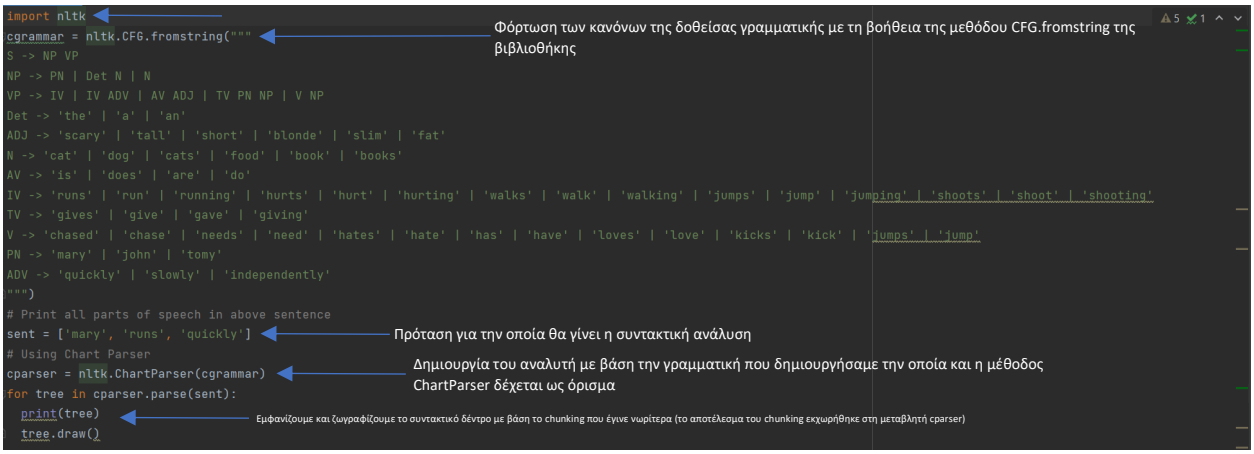
Για την υλοποίηση του θέματος 2 χρησιμοποιήθηκε η γλώσσα προγραμματισμού **PYTHON** και η βιβλιοθήκη **ntlk**.

Λίγα λόγια για την **nltk (Natural Language Toolkit):**

Το Natural Language Toolkit, ή πιο κοινά NLTK, είναι μια σουίτα βιβλιοθηκών και προγραμμάτων για συμβολική και στατιστική επεξεργασία φυσικής γλώσσας για την αγγλική, γραμμένη στη γλώσσα προγραμματισμού Python. Η nltk μεταξύ άλλων λειτουργιών εκτελεί την συντακτική ανάλυση προτάσεων με βάση δοσμένους γραμματικούς κανόνες καθώς και την παραγωγή συντακτικών δέντρων.

Κατασκευή Συντακτικού αναλυτή:
Ο συντακτικός αναλυτής θα δέχεται ως είσοδο ένα σύνολο κανόνων γραμματικής και μια πρόταση και με θα πραγματοποιεί την συντακτική ανάλυση με βάση τους δοσμένους κανόνες. Στη συνέχεια θα παράγει και θα εμφανίζει το συντακτικό δέντρο που αναπαριστά γραφικά την συντακτική ανάλυση για την δοσμένη πρόταση. Η πραγματοποίηση της συντακτικής ανάλυσης γίνεται με την βοήθεια της βιβλιοθήκης ntlk μέσω της οποίας θα γίνει η φόρτωση της γραμματικής, γράφοντας τους κανόνες σε string μορφή με το αποδεκτό από την βιβλιοθήκη

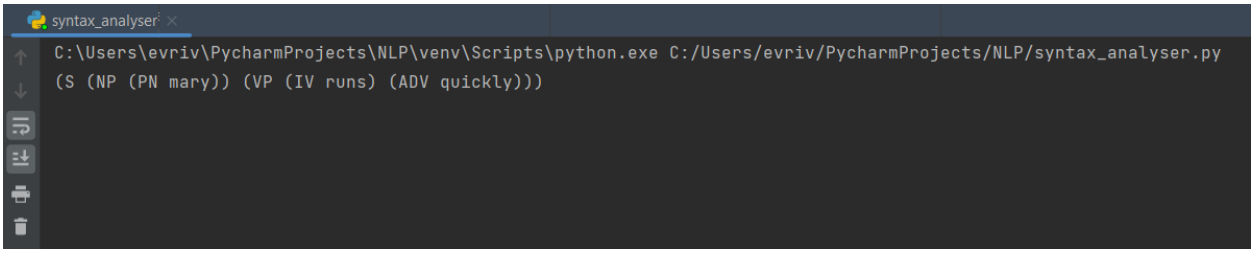
format. Έπειτα με βάση τη γραμματική αυτή δημιουργούμε τον αναλυτή με τη μέθοδο nltk.ChartParser και με τη βοήθεια του κάνουμε ανάλυση της ζητούμενης πρότασης. Τέλος εμφανίζουμε το συντακτικό δέντρο. Ας δούμε αναλυτικά τη διαδικασία υλοποίησης μέσα από τον κώδικα:

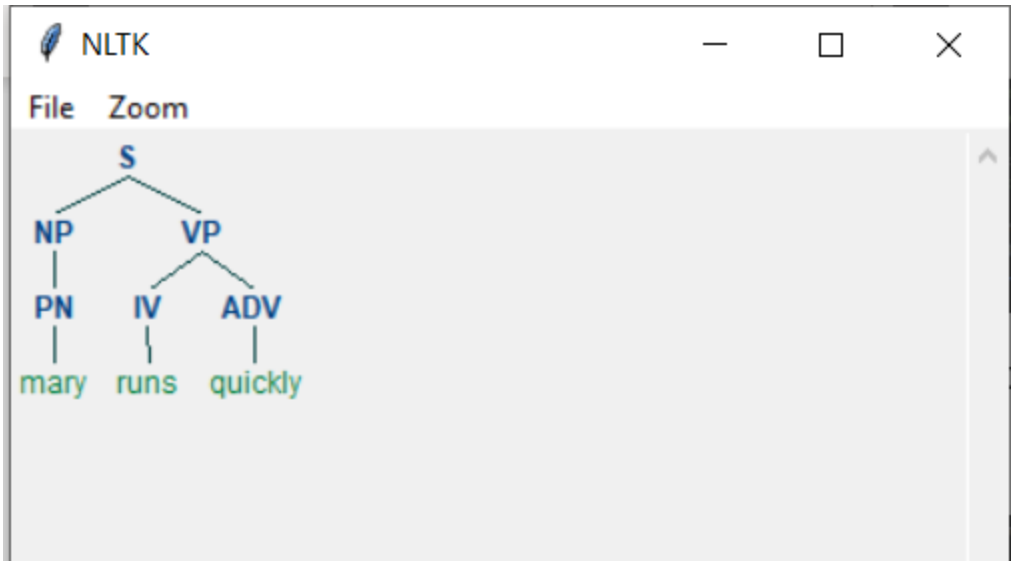


Εκτέλεση κώδικα και αναφορά ελέγχου αποτελεσμάτων:

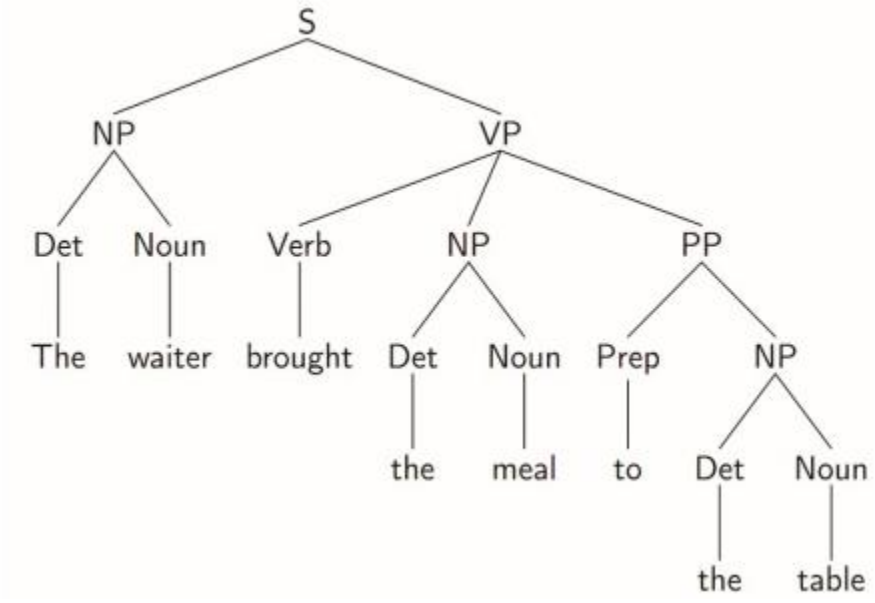
Είσοδος:

Έξοδος:





Όπως βλέπουμε η πρόταση έχει αναλυθεί συντακτικά με βάση του κανόνες που δόθηκαν και με βάση την πρότυπη λύση της άσκησης που φαίνεται παρακάτω



Επομένως η έξοδος του προγράμματος είναι αποδεκτή με βάση τα ζητούμενα της άσκησης

Θέμα 3

Εκφώνηση

Ανάπτυξη Σημασιολογικού Αναλυτή σε άλλη γλώσσα Προγραμματισμού. Αναζητείστε στο διαδίκτυο ή αναπτύξτε εσείς Σημασιολογικό Αναλυτή που με βάση τους κανόνες σημασιολογικής ανάλυσης της πρότυπης λύσης σε Prolog που σας δόθηκε παράγει τα σημαινόμενα της πρότασης (σχέσεις μεταξύ ρημάτων, ουσιαστικών, επιθέτων, κ.λπ). Τεκμηριώστε πειστικά τον κώδικά σας.

Λύση

Για την υλοποίηση του θέματος 3 χρησιμοποιήθηκε η γλώσσα προγραμματισμού **PYTHON** η βιβλιοθήκη **ntlk** και η βιβλιοθήκη **sqlite3**.

Για την βιβλιοθήκη nltk μιλήσαμε παραπάνω. Στην συγκεκριμένη περίπτωση θα εισάγουμε τα εργαλεία grammar και parse για την σημασιολογική ανάλυση.

Λίγα λόγια για την sqlite3:

Η βιβλιοθήκη sqlite3 μας επιτρέπει να εκτελούμε querries πάνω σε μια sqlite βάση χωρίς την χρήση κάποιου συστήματος διαχείρισης βάσης δεδομένων μέσα από τον κώδικα της python. Στην συγκεκριμένη περίπτωση την χρησιμοποιούμε για να κάνουμε τη συντακτική ανάλυση συνδυάζοντας του κανόνες που έχουν δοθεί και κάποιες προτάσεις που είναι αποθηκευμένες σε μια βάση γνώσης που δημιουργήσαμε. Η βάση δημιουργήθηκε με τη βοήθεια του περιβάλλοντος της εφαρμογής: DB Browser for SQLite

Κατασκευή σημασιολογικού αναλυτή

Ο σημασιολογικός αναλυτής θα δέχεται ως είσοδο κάποιους σημασιολογικούς κανόνες και ένα ερώτημα σε φυσική γλώσσα το οποίο το πρόγραμμα το επεξεργάζεται σε query της sql και δίνει την απάντηση του ερωτήματος εξάγοντας ταυτόχρονα τα σημαινόμενα της πρότασης. Ο αναλυτής αποτελείται από μια συνάρτηση **execute_query(conn,q)** για την εκτέλεση sqlite queries και το κύριο πρόγραμμα που δέχεται τους κανόνες και το ερώτημα και παράγει τη ζητούμενη έξοδο. Εισάγουμε σε μια μεταβλητή τους σημασιολογικούς κανόνες σε string μορφή σε αποδεκτό format. Με τη μέθοδο grammar.FeatureGrammar.fromstring μετατρέπουμε τους κανόνες από string σε grammar μορφή. Φτιάχνουμε τον αναλυτή σύμφωνα με τη γραμματική (που πλέον είναι σε grammar μορφή) με τη μέθοδο parse.FeatureEarleyChartParser. Κάνουμε ανάλυση της πρότασης με τη βοήθεια του αναλυτή που φτιάξαμε. Αυτό που μένει είναι να πάρουμε την απάντηση στην ερώτηση με βάση τα δεδομένα της βάσης γνώσης σε συνδυασμό με την ανάλυση. Επομένως μετατρέπουμε τη σημασιολογική ανάλυση που θα μας επιστραφεί από τον αναλυτή σε query το οποίο θα τρέξουμε με τη βοήθεια του execute_query και θα πάρουμε τα δεδομένα που ζητάμε. Στη συνέχεια θα εμφανίζουμε όλα αυτά τα στοιχεία στην κονσόλα για τον χρήστη. Ας δούμε αναλυτικά τη διαδικασία υλοποίησης μέσα από τον κώδικα:

```
import sqlite3
from nltk import grammar, parse
```

Φόρτωση των απαραίτητων βιβλιοθηκών

```
def execute_query(conn,q):
 cur = conn.cursor()
 cur.execute(q)
 rows = cur.fetchall()
 for row in rows:
  print(row)
```

Συνάρτηση που δέχεται ως όρισμα το string σύνδεσης σε μια βάση δεδομένων και το query προς εκτέλεση και εκτελεί το ζητούμενο ερώτημα επιστρέφοντας την απάντηση

```
g = """
% start S
# ###################
# Grammar Productions
# ###################
S[SEM=('SELECT'+?q + 'FROM knowledge WHERE' + ?p)] -> Q[SEM=?q] P[SEM=?p]
P[SEM=(?v+' AND '+?n)] -> V[SEM=?v] N[SEM=?n]
P[SEM=?v] -> V[SEM=?v]
P[SEM=?adj] -> ADJ[SEM=?adj]
P[SEM=?iv] -> IV[SEM=?iv]
P[SEM=(?iv+' AND '+?adv)] -> IV[SEM=?iv] ADV[SEM=?adv]
P[SEM=?tv] -> TV[SEM=?tv]
P[SEM=(?tv+' AND '+?n)] -> TV[SEM=?tv] N[SEM=?n]
P[SEM=(?tv+' AND '+?r)] -> TV[SEM=?tv] R[SEM=?r]
# ###################
# Lexical Productions
# ###################
Q[SEM='Name'] -> 'who'
Q[SEM='Noun'] -> 'what' | 'what is'
Q[SEM='Adverb'] -> 'how'
Q[SEM='Receiver'] -> 'who is'
R[SEM=("Receiver='john'")] -> 'to john'
R[SEM=("Receiver='mary'")] -> 'to mary'
R[SEM=("Receiver='tomy'")] -> 'to tomy'
N[SEM=("Noun='dog'")]-> 'dog' | 'dogs'
N[SEM=("Noun='food'")]-> 'food' | 'foods'
N[SEM=("Noun='cat'")]-> 'cat' | 'cats'
N[SEM=("Noun='book'")]-> 'book' | 'books'
V[SEM="Verb='need'"]-> 'need' | 'needs'
V[SEM="Verb='hate'"]-> 'hate' | 'hates'
V[SEM="Verb='chase'"]-> 'chase' | 'chases'
V[SEM="Verb='love'"]-> 'love' | 'loves'
V[SEM="Verb='have'"]-> 'have' | 'has'
ADJ[SEM="Adjective='scary'"]-> 'scary'
ADJ[SEM="Adjective='tall'"]-> 'tall'
ADJ[SEM="Adjective='short'"]-> 'short'
ADJ[SEM="Adjective='blonde'"]-> 'blonde'
ADJ[SEM="Adjective='slim'"]-> 'slim'
ADJ[SEM="Adjective='fat'"]-> 'fat'
AV[SEM='-'] -> 'are' | 'is' | 'do' | 'does'| 'to' | 'a' | 'is mary' | 'is john' | 'is tomy' | 'the' | 'does mary'
IV[SEM="Intransitive_Verb='run'"]-> 'run' | 'runs' | 'running'
TV[SEM="Transitive_Verb='give'"]-> 'give' | 'gives' | 'giving' | 'gave'
ADV[SEM="Adverb='quickly'"]-> 'quickly'
 """
```

Φόρτωση των κανόνων του σημασιολογικού αναλυτή με βάση την δοσμένη πρότυπη λύση σε prolog

```
grammar = grammar.FeatureGrammar.fromstring(g)
tokens= 'who gives dog'.split()
parser = parse.FeatureEarleyChartParser(grammar)
trees = list(parser.parse(tokens))
if (len(trees)==0):
 print("There's no answer for your question")
```

Δημιουργία του αναλυτή με βάση τους κανόνες που βρίσκονται στην μεταβλητή g. Σπάσιμο ερωτήματος εισαγωγή στον αναλυτή και δημιουργία δέντρου με βάση την γραμματική που έχω ορίσει. Έλεγχος του μήκους του δέντρου για την παραγωγή αντίστοιχης εξόδου αν δεν υπάρχει απάντηση για το ερώτημα

```
else:
 answer = trees[0].label()['SEM']
 answer = [s for s in answer if s]
 q = ' '.join(answer)
 print(q)
 execute_query(sqlite3.connect('semantic_analysis.db'),q)
```

Σπάω το δέντρο που εξήχθη από την κλήση του σημασιολογικού αναλυτή για να το μετατρέψω σε query και να το υποβάλω στη βάση γνώσης. Κάνω τις απαραίτητες τροποποιήσεις και καλώ την κατάλληλη συνάρτηση

Εκτέλεση κώδικα και αναφορά ελέγχου αποτελεσμάτων:

Είσοδος:

```
g = """
% start S
# ###################
# Grammar Productions
# ###################
S[SEM=('SELECT'+?q + 'FROM knowledge WHERE' + ?p)] -> Q[SEM=?q] P[SEM=?p]
P[SEM=(?v+' AND '+?n)] -> V[SEM=?v] N[SEM=?n]
P[SEM=?v] -> V[SEM=?v]
P[SEM=?adj] -> ADJ[SEM=?adj]
P[SEM=?iv] -> IV[SEM=?iv]
P[SEM=(?iv+' AND '+?adv)] -> IV[SEM=?iv] ADV[SEM=?adv]
P[SEM=?tv] -> TV[SEM=?tv]
P[SEM=(?tv+' AND '+?n)] -> TV[SEM=?tv] N[SEM=?n]
P[SEM=(?tv+' AND '+?r)] -> TV[SEM=?tv] R[SEM=?r]
# ###################
# Lexical Productions
# ###################
Q[SEM='Name'] -> 'who'
Q[SEM='Noun'] -> 'what' | 'what is'
Q[SEM='Adverb'] -> 'how'
Q[SEM='Receiver'] -> 'who is'
R[SEM=("Receiver='john'")] -> 'to john'
R[SEM=("Receiver='mary'")] -> 'to mary'
R[SEM=("Receiver='tomy'")] -> 'to tomy'
N[SEM=("Noun='dog'")]-> 'dog' | 'dogs'
N[SEM=("Noun='food'")]-> 'food' | 'foods'
N[SEM=("Noun='cat'")]-> 'cat' | 'cats'
N[SEM=("Noun='book'")]-> 'book' | 'books'
V[SEM="Verb='need'"]-> 'need' | 'needs'
V[SEM="Verb='hate'"]-> 'hate' | 'hates'
```

```
V[SEM="Verb='chase'"]-> 'chase' | 'chases'
V[SEM="Verb='love'"]-> 'love' | 'loves'
V[SEM="Verb='have'"]-> 'have' | 'has'
ADJ[SEM="Adjective='scary'"]-> 'scary'
ADJ[SEM="Adjective='tall'"]-> 'tall'
ADJ[SEM="Adjective='short'"]-> 'short'
ADJ[SEM="Adjective='blonde'"]-> 'blonde'
ADJ[SEM="Adjective='slim'"]-> 'slim'
ADJ[SEM="Adjective='fat'"]-> 'fat'
AV[SEM='-'] -> 'are' | 'is' | 'do' | 'does'| 'to' | 'a' | 'is mary' | 'is john' | 'is tomy' | 'the' | 'does mary'
IV[SEM="Intransitive_Verb='run'"]-> 'run' | 'runs' | 'running'
TV[SEM="Transitive_Verb='give'"]-> 'give' | 'gives' | 'giving' | 'gave'
ADV[SEM="Adverb='quickly'"]-> 'quickly'
"""
```

```
tokens= 'who gives dog'.split()
```

Έξοδος:

```
C:\Users\evriv\PycharmProjects\NLP\venv\Scripts\python.exe C:/Users/evriv/PycharmProjects/NLP/semantic_analyser.py
SELECT Name FROM knowledge WHERE Transitive_Verb='give'  AND  Noun='dog'
('mary',)

Process finished with exit code 0
```

Όπως βλέπουμε η πρόταση έχει αναλυθεί σημασιολογικά με βάση τους κανόνες που δόθηκαν και αν κοιτάξουμε τη βάση γνώσης που βρίσκεται στο αρχείο semantic_analysis.db θα διαπιστώσουμε πως το mary είναι η απάντηση του υποβεβλημένου ερωτήματος.

	id	Name	Verb	Noun	Adjective	Intransitive_Verb	Transitive_Verb	Adverb	Receiver
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	mary	NULL	dog	NULL	NULL	give	NULL	john
2	2	mary	NULL	book	NULL	NULL	give	NULL	tomy
3	3	mary	NULL	NULL	NULL	run	NULL	quickly	NULL
4	4	mary	NULL	NULL	tall	NULL	NULL	NULL	NULL
5	5	mary	NULL	NULL	slim	NULL	NULL	NULL	NULL
6	6	mary	NULL	NULL	blonde	NULL	NULL	NULL	NULL
7	7	mary	love	book	NULL	NULL	NULL	NULL	NULL
8	8	dog	need	food	NULL	NULL	NULL	NULL	NULL
9	9	cat	have	food	NULL	NULL	NULL	NULL	NULL
10	10	dog	hate	cat	NULL	NULL	NULL	NULL	NULL
11	11	dog	chase	cat	NULL	NULL	NULL	NULL	NULL
12	12	cat	NULL	NULL	scary	NULL	NULL	NULL	NULL

Θέμα 4

Εκφώνηση

Πρόγραμμα στην γλώσσα προγραμματισμού της επιλογής σας, για την ενημέρωση και πραγματοποίηση ερωταποκρίσεων σε Βάση Γνώσης που έχετε αναπτύξει για αυτό τον σκοπό. Οι ερωτήσεις και απαντήσεις θα δίδονται σε φυσική γλώσσα.

Λύση

Για την υλοποίηση του θέματος αυτού χρησιμοποιήθηκε η γλώσσα προγραμματισμού **PYTHON** η βιβλιοθήκη **pandas** και η βιβλιοθήκη **Levenshtein**.

Λίγα λόγια για τις βιβλιοθήκες:

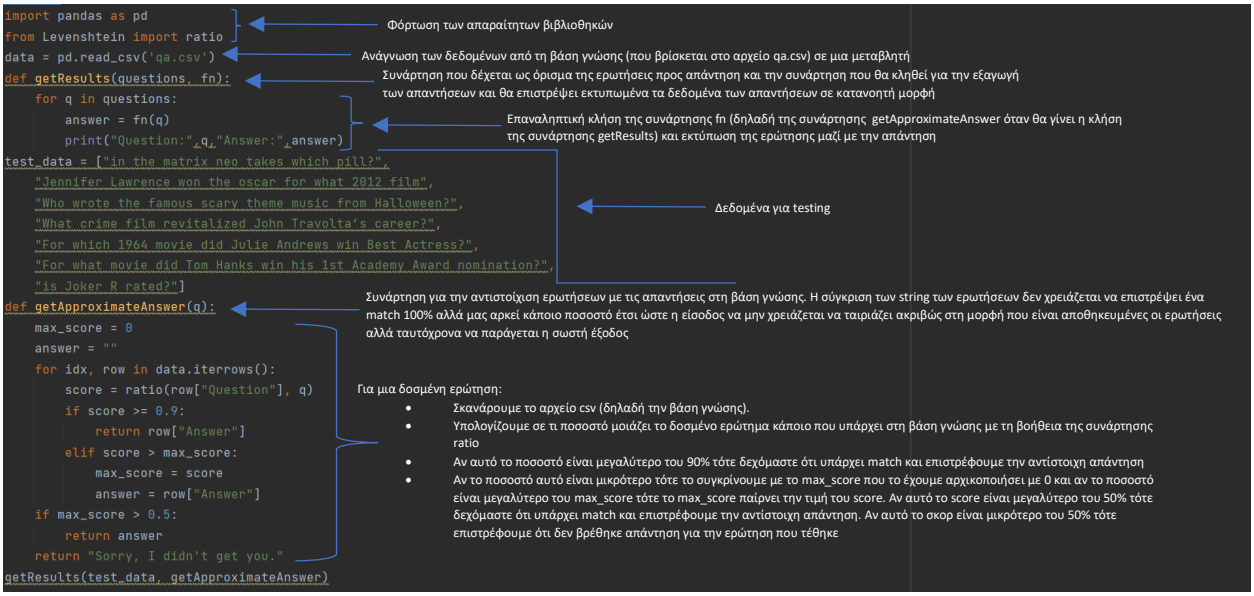
Το **pandas** είναι μια βιβλιοθήκη λογισμικού γραμμένη για τη γλώσσα προγραμματισμού Python για χειρισμό και ανάλυση δεδομένων. Συγκεκριμένα, προσφέρει δομές δεδομένων και λειτουργίες για το χειρισμό αριθμητικών πινάκων και χρονοσειρών. Στην περίπτωση μας τη χρησιμοποιούμε για την ανάγνωση δεδομένων από csv αρχείο στο οποίο είναι αποθηκευμένες και οι ερωταποκρίσεις. Η βιβλιοθήκη Levenshtein περιέχει συναρτήσεις για γρήγορο υπολογισμό των

- Απόσταση Levenshtein
- Ισότητα συμβολοσειρών κατά προσέγγιση

Εμείς θα την χρησιμοποιήσουμε για να υπολογίσουμε αν δύο συμβολοσειρές είναι ίσες ακόμα και αν δεν έχουν ίδια όλα τα σύμβολα, αλλά κατά προσέγγιση με βάση το σύνολο των χαρακτήρων είναι κοινοί. Αυτό θα χρησιμεύσει ώστε οι ερωτήσεις να μπορούν να δίνονται σε φυσική γλώσσα χωρίς όμως η διατύπωση να πρέπει να είναι αυστηρά ίδια με αυτή που υπάρχει στη βάση γνώσης.

Κατασκευή συστήματος ερωταποκρίσεων:
Το σύστημα ερωταποκρίσεων δέχεται ως είσοδο κάποια/κάποιες ερωτήσεις σε φυσική γλώσσα και παράγει ως έξοδο την απάντηση που της αντιστοιχεί με βάση τη βάση γνώσης. Το πρόγραμμα αποτελείται από τη συνάρτηση getResults, τα test data και τη συνάρτηση getApproximateAnswer. Η πρώτη συνάρτηση δέχεται τα δεδομένα της δεύτερης μαζί με τα test data και εμφανίζει τα αποτελέσματα σε κατανοητή μορφή ώστε να το εκτυπώσουμε στον χρήστη. Τα test data είναι κάποια δεδομένα που εισάγουμε για να ελέγξουμε την ορθότητα λειτουργίας του προγράμματος. Η συνάρτηση getApproximateAnswer δέχεται την ερώτηση από τα test data και υπολογίζει σε τι ποσοστό η ερώτηση αυτή ταιριάζει με κάποια από τις ερωτήσεις της βάσης γνώσης. Σε περίπτωση που δεν βρεθεί κάποιο match για την ερώτηση στη βάση γνώσης τότε εμφανίζεται μήνυμα λάθους.

Ας δούμε αναλυτικά τη διαδικασία υλοποίησης μέσα από τον κώδικα:



Βάση γνώσης:

Question,Answer
In The Matrix does Neo take the blue pill or the red pill?,Red
For what movie did Tom Hanks score his first Academy Award nomination?,Big
What pop vocal group performs at the wedding in Bridesmaids?,Wilson Phillips
For which 1964 musical blockbuster did Julie Andrews win the Academy Award for Best Actress?,Marry Poppins
Jennifer Lawrence won a Best Actress Academy Award for what 2012 romantic comedy drama?,Silver Linnings PlayBook
What is the highest grossing R rated movie of all time?,Joker
Who wrote the famous scary theme music from Halloween?,John Carpenter
What crime film revitalized John Travoltas career?,Pulp Fiction
Who played Juror Number 8 in 12 Angry Men?,Henry Fonda

Είσοδος:

```
"in the matrix neo takes which pill?",
"Jennifer Lawrence won the oscar for what 2012 film",
"Who wrote the famous scary theme music from Halloween?",
"What crime film revitalized John Travolta's career?",
"For which 1964 movie did Julie Andrews win Best Actress?",
"For what movie did Tom Hanks win his 1st Academy Award nomination?",
"is Joker R rated?"
```

Έξοδος:

```
C:\Users\evriv\PycharmProjects\NLP\venv\Scripts\python.exe C:/Users/evriv/PycharmProjects/NLP/QA_system.py
Question: in the matrix neo takes which pill? Answer: Red
Question: Jennifer Lawrence won the oscar for what 2012 film Answer: Silver Linnings PlayBook
Question: Who wrote the famous scary theme music from Halloween? Answer: John Carpenter
Question: What crime film revitalized John Travolta's career? Answer: Pulp Fiction
Question: For which 1964 movie did Julie Andrews win Best Actress? Answer: Marry Poppins
Question: For what movie did Tom Hanks win his 1st Academy Award nomination? Answer: Big
Question: is Joker R rated? Answer: Sorry, I didn't get you.

Process finished with exit code 0
```

Πηγές

- https://www.nltk.org/howto/featgram.html
- https://medium.com/codex/linguistic-modelling-techniques-with-python-de3baf4bb752
- https://www.codegrepper.com/code-examples/python/split+text+into+sentences+python
- https://www.sqlitetutorial.net/sqlite-python/sqlite-python-select/
- https://www.kdnuggets.com/2020/04/simple-question-answering-systems-text-similarity-python.html
- Πρότυπη εργασία που διατέθηκε από τον διδάσκοντα https://unipigr.sharepoint.com/:f:/r/sites/TMC113-/Class%20Materials/Python%20Code?csf=1&web=1&e=WlJyHi