



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

Διπλωματική Εργασία

**Δημιουργία σουίτας εφαρμογών αξιοποιώντας την χρήση της Επεξεργασίας
Φυσικής Γλώσσας
(Natural Language Processing)**



Κορτσαρίδης Γεώργιος

Επιβλέπων : Σαρηγιαννίδης Παναγιώτης, Επίκουρος καθηγητής Π.Δ.Μ

Εικόνα εξωφύλλου : BrainTag, Κορτσαρίδης Γεώργιος



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

Diploma Thesis

Application Development using Natural Language Processing



Kortsaridis George

Supervisor: Sarigiannidis Panagiotis, Assistant Professor U.O.W.M

Book Cover: BrainTag, Kortsaridis George

Περίληψη

Η παρούσα διπλωματική εργασία εξετάζει τα εργαλεία μηχανικής μάθησης που μπορούν να συμβάλλουν στη μελέτη, κατανόηση και επεξεργασία της φυσικής γλώσσας (Natural Language Process, συντομογραφία *NLP*).

Σε μία προσπάθεια ενοποίησης των κύριων εννοιών της καθημερινότητας ‘Μάθηση’, ‘Εκπαίδευση’ και ‘Διασκέδαση’, αξιοποιήθηκε ένα από τα πλέον πιο διαδεδομένα εργαλεία μελέτης φυσικής γλώσσας (Stanford Natural Language Process Library). Δημιουργήθηκε η κατάλληλη υποδομή, ώστε να χρησιμοποιηθεί από δύο εφαρμογές, οι οποίες έχουν ως στόχο την ευχάριστη εκμάθηση του συντακτικού της αγγλικής γλώσσας.

Οι εφαρμογές αποφασίστηκε να φέρουν την ονομασία ‘BrainTag’. Η ονομασία αυτή συνδέει δύο λέξεις. Brain (μυαλό - εγκέφαλος), γιατί αυτός είναι και το κύριο στόχος της εφαρμογής, η εκμάθηση. Tag (ετικέτα), γιατί κατά τη διάρκεια του παιχνιδιού, η κάθε λέξη έχει τη δική της ετικέτα με την πληροφορία της συντακτικής της ανάλυσης. Επίσης, η επιλογή μιας σύντομης λέξης μπορεί να βοηθήσει στην πιο εύκολη αποτύπωση και αφομοίωση του ονόματος της εφαρμογής από τους χρήστες.

Το πρώτο βήμα της διπλωματικής εργασίας ήταν η εξερεύνηση και η εκμάθηση της σουίτας Stanford Natural Language Process Library. Χρειάστηκε να γίνουν πειραματισμοί με τις διάφορες γλώσσες υποστήριξης, μέχρι να παρθεί η απόφαση ότι εν τέλει θα χρησιμοποιηθεί η γλώσσα προγραμματισμού Java. Η επιλογή της συγκεκριμένης γλώσσας έγινε λόγω του ότι μπορεί με ευκολία να εκτελεστεί από όλα τα λειτουργικά συστήματα. Ακόμα, μπορεί να επικοινωνήσει με την γλώσσα NodeJs, η οποία χρησιμοποιήθηκε για την δημιουργία του διαδικτυακού κορμού της υπηρεσίας εφαρμογών (backend).

Η δημιουργία του διαδικτυακού κορμού (backend) έγινε με κύριο στόχο την επαναχρησιμοποίηση και την ευκολία χρήσης του από έναν τρίτο προγραμματιστή. Δημιουργήθηκαν ξεχωριστά τερματικά σημεία (http endpoints) για κάθε μία από τις βασικές λειτουργίες της πλατφόρμας, όπως σύνδεση χρήστη, εγγραφή χρήστη, επεξεργασία φυσικού κειμένου, καταχώρηση αποτελεσμάτων, κ.α.

Μετά την επιτυχημένη δημιουργία του εξυπηρετητή, άρχισε ο προγραμματισμός της πρώτης τερματικής εφαρμογής, η οποία προορίστηκε για την κινητή πλατφόρμα Android[1]. Στοχεύοντας ως κύρια ομάδα χρηστών, αγγλόφωνα παιδιά ηλικιών 11-17 (αντίστοιχες ηλικίες του ελληνικού Γυμνασίου - Λυκείου), χρησιμοποιήθηκαν κυρίως φωτεινά και έντονα χρώματα. Σκοπός ήταν η αύξηση του ενδιαφέροντος, κάτι που αποσκοπεί σε περισσότερες ώρες χρήσης της εφαρμογής.

Στη συνέχεια, έχοντας ως κύρια γραμμή το γραφικό περιβάλλον της εφαρμογής Android, πάρθηκε η απόφαση να δημιουργηθεί η αντίστοιχη εφαρμογή για τα λειτουργικά συστήματα Windows[2], MacOS[3] και Linux[4]. Ο βασικός σκοπός της επιλογής δημιουργίας εφαρμογών για αυτά τα λειτουργικά συστήματα είναι ότι στοχεύουν σε μικρές, κυρίως, ηλικίες, όπου ακόμα η χρήση υπολογιστή και κινητών τηλεφώνων πιθανά είναι περιορισμένη και υπό την εποπτεία γονέων. Ο γονέας νιώθει πιο ασφαλής όταν το παιδί του βρίσκεται σε μια κλειστή εφαρμογή στον υπολογιστή, σε σύγκριση με το να βρίσκεται σε μια ιστοσελίδα στο ίντερνετ, όπου μπορεί ανά πάσα στιγμή να μεταφερθεί σε κάποιον αμφιβόλου ποιότητας χώρο.

Οι δύο εφαρμογές/παιχνίδια έχουν τον ίδιο στόχο. Ο χρήστης ξεκινώντας το παιχνίδι χρησιμοποιεί μια τυχαία παράγραφο κειμένου, η οποία προέρχεται από την ιστοσελίδα της Wikipedia, είτε από τίτλους άρθρων των ειδησεογραφικών γραφείων Techcrunch[5], BBC news[6], Google News[7], CNN[8], National Geographic[9], The Verge[10], USA Today[11]. Εφόσον το κείμενο υποστεί την κατάλληλη συντακτική ανάλυση, εμφανίζεται στον χρήστη χωρισμένο σε κομμάτια, ανά προτάσεις. Στο πάνω μέρος της οθόνης εμφανίζεται η πρόταση, και στο κάτω μέρος η συντακτική ανάλυση της κάθε λέξης με τυχαία σειρά. Ο χρήστης καλείται να σύρει με το ποντίκι ή με το χέρι του την ετικέτα με τη σωστή συντακτική ανάλυση πάνω από την

κάθε λέξη. Στο τέλος κάθε πρότασης, ο χρήστης έχει τη δυνατότητα να δει τις σωστές και τις λανθασμένες επιλογές του.

Λέξεις - Κλειδιά: BrainTag, Natural Language Processing, NLP, Android, Windows, MacOS, Linux, Stanford, επεξεργασία φυσικού κειμένου, εκμάθηση, διασκέδαση

Abstract

This Diploma Thesis examines engineering learning tools that can contribute to the study, understanding and processing of natural language (Natural Language Process, abbreviation NLP).

In an attempt to unify the main concepts of everyday learning education and entertainment, Stanford Natural Language Process Library, one of the most famous NLP tools was used. Appropriate infrastructure has been created to be used by two applications, aiming at a efficient and friendly learning of the English language syntax.

Applications were decided to be named 'BrainTag'. This name associates two words. Brain, because learning is the main goal of application. Tag because, during the game, each word has its own tag with the information of the syntax analysis. Also, choosing a short word can help make it easier for users to capture and assimilate the application name.

The first step in this Diploma Thesis was the exploration and learning of the Stanford Natural Language Process Library suite. Experiments with various programming languages needed to be made until the the Java programming language decided that would eventually be used. The motivation behind the selection of this language is attached with the high compatibility with all operating systems. Still, it can communicate with the NodeJs language, which was used to create the backend's backbone.

The backend was created with the main purpose of reuse and ease of use by a third developer. Separate endpoints have been created for each of the platform's basic functions, such as user logging, user registration, physical text processing, listing of results, and so on.

After the successful creation of the server, the first terminal application, which was designed for the Android mobile platform, began to be programmed. Targeting as the main group of users, English-speaking children ages 11-17 (corresponding ages of the Greek High School -

Lyceum), mainly bright and intense colors were used. The goal was to increase interest, which is aimed at more hours of use of the application.

Then, having the Android desktop graphical interface, the decision was made to create the corresponding application for Windows, MacOS, and Linux operating systems. The main purpose of choosing to create applications for these operating systems is that they target small, mainly, ages, where the use of computer and mobile phones is still limited and under the supervision of parents. The parent feels more secure when their child is in a closed application on the computer, compared to being on an internet site where it can be transferred at any time to some dubious quality space.

The two applications / games have the same goal. The user starts the game uses a random text paragraph that comes from the Wikipedia site, or from Techcrunch news articles, BBC news, Google News, CNN, National Geographic, The Verge, USA Today. Once the text has the proper editorial resolution, it appears to the user divided into pieces by suggestions. At the top of the screen appears the sentence, and at the bottom the syntactic analysis of each word in random order. The user is required to draw with the mouse or manually the tag with the correct syntactic resolution above each word. At the end of each proposal, the user can see the right and wrong choices.

Keywords: BrainTag, Natural Language Processing, NLP, Android, Windows, MacOS, Linux, Stanford, Natural Text Processing, Learning, Fun

Ευχαριστίες

Ευχαριστώ θερμά τον επίκουρο καθηγητή του Πανεπιστημίου Δυτικής Μακεδονίας, κύριο Σαρηγιαννίδη Παναγιώτη για την επίβλεψη της διπλωματικής μου εργασίας με θέμα την δημιουργία σουίτας εφαρμογών αξιοποιώντας την χρήση της Επεξεργασίας Φυσικής Γλώσσας. Τον ευχαριστώ θερμά για την εμπιστοσύνη που μου έδειξε σε αυτό το θέμα. Αυτή η εργασία μου έδωσε την ευκαιρία να ασχοληθώ με έναν συναρπαστικό τομέα της πληροφορικής, την μηχανική μάθηση, και να τον χρησιμοποιήσω στη δημιουργία ενός διδακτικού εργαλείου.

Λίστα εικόνων

ΕΙΚΟΝΑ 1: ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΚΑΤΗΓΟΡΙΑΣ ΜΟΡΦΗΜΑΤΩΝ.....	30
ΕΙΚΟΝΑ 2 : ΚΑΘΟΡΙΣΜΟΣ ΓΡΑΜΜΑΤΙΚΟΥ ΔΕΝΤΡΟΥ	31
ΕΙΚΟΝΑ 3 : ΠΑΡΑΔΕΙΓΜΑ ΑΝΑΓΝΩΡΙΣΗΣ ΛΕΞΕΩΝ ΩΣ ΜΕΡΟΣ ΤΟΥ ΛΟΓΟΥ ΚΑΙ ΚΑΘΟΡΙΣΜΟΣ ΠΡΟΤΑΣΗΣ(S)	32
ΕΙΚΟΝΑ 4 : ΠΑΡΑΔΕΙΓΜΑ ΚΑΝΟΝΩΝ ΚΑΙ ΑΠΟΣΠΑΣΗΣ ΡΙΖΑΣ ΛΕΞΕΩΝ.....	32
ΕΙΚΟΝΑ 5 : ΠΑΡΑΔΕΙΓΜΑ ΑΝΑΓΝΩΡΙΣΗΣ ΚΑΙ ΔΙΑΧΩΡΙΣΜΟΥ ΛΕΞΕΩΝ ΑΠΟ ΚΟΜΜΑΤΙ ΚΕΙΜΕΝΟΥ	33
ΕΙΚΟΝΑ 6 : ΠΑΡΑΔΕΙΓΜΑ ΑΝΑΓΝΩΡΙΣΗΣ ΥΠΟ-ΜΟΝΑΔΑΣ ΛΕΞΕΩΝ ΠΟΥ ΔΗΛΩΝΟΥΝ ΧΡΩΜΑ	34
ΕΙΚΟΝΑ 7 : ΒΗΜΑΤΑ ΜΗΧΑΝΙΚΗΣ ΜΕΤΑΦΡΑΣΗΣ ΚΕΙΜΕΝΟΥ	34
ΕΙΚΟΝΑ 8 : ΑΝΑΓΝΩΡΙΣΗ ΟΝΤΟΤΗΤΩΝ ΟΠΩΣ ΥΠΟΘΕΣΙΑ, ΧΡΟΝΟΣ, ΑΤΟΜΟ, ΟΡΓΑΝΙΣΜΟΣ Κ.Α.	35
ΕΙΚΟΝΑ 9 : ΣΧΗΜΑΤΙΚΗ ΔΙΑΦΟΡΑ ΜΕΤΑΞΥ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΚΑΤΑΝΟΗΣΗΣ ΦΥΣΙΚΗΣ ΓΛΩΣΣΑΣ.....	37
ΕΙΚΟΝΑ 10 : ΟΠΤΙΚΗ ΑΝΑΓΝΩΡΙΣΗ ΧΑΡΑΚΤΗΡΩΝ ΜΕΣΩ ΒΙΒΛΙΟΘΗΚΗΣ MATLAB	38
ΕΙΚΟΝΑ 11: ΣΧΗΜΑΤΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ ΔΙΑΔΙΚΑΣΙΑΣ ΑΥΤΟΜΑΤΗΣ ΑΠΑΝΤΗΣΗΣ ΕΡΩΤΗΣΕΩΝ	39
ΕΙΚΟΝΑ 12 : ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ ΑΝΑΛΥΣΗΣ ΣΥΝΑΙΣΘΗΜΑΤΟΣ ΣΤΑ ΚΟΙΝΩΝΙΚΑ ΔΙΚΤΥΑ	40
ΕΙΚΟΝΑ 13 : ΔΙΑΓΡΑΜΜΑ ΔΙΑΔΙΚΑΣΙΑΣ ΑΝΑΓΝΩΡΙΣΗΣ ΦΩΝΗΣ, ΤΗΣ ΕΤΑΙΡΕΙΑΣ GOOGLE.....	44
ΕΙΚΟΝΑ 14: ΓΡΑΦΙΚΟ ΠΑΡΑΔΕΙΓΜΑ ΑΝΑΓΝΩΡΙΣΗΣ ΚΑΙ ΑΝΑΘΕΣΗΣ ΛΕΞΕΩΝ ΩΣ ΜΕΡΗ ΤΟΥ ΛΟΓΟΥ.....	47
ΕΙΚΟΝΑ 15 : ΑΠΕΙΚΟΝΙΣΗ ΑΝΑΘΕΣΗΣ ΛΕΞΕΩΝ ΩΣ ΜΕΡΗ ΤΟΥ ΛΟΓΟΥ.....	48
ΕΙΚΟΝΑ 16 : ΠΛΗΡΩΣ ΑΝΑΛΥΜΕΝΗ ΠΡΟΤΑΣΗ ΣΕ ΕΤΙΚΕΤΕΣ UPENN TREEBANK II	51
ΕΙΚΟΝΑ 17: ΕΝΑΛΛΑΚΤΙΚΕΣ ΑΝΑΛΥΣΕΙΣ ΠΡΟΤΑΣΗΣ, ΠΕΡΑ ΑΠΟ ΤΗΝ ΥΠΗΡΕΣΙΑ PART-OF-SPEECH	52
ΕΙΚΟΝΑ 18 : ΓΡΑΜΜΗ ΕΝΤΟΛΩΝ, ΕΚΔΟΣΗ JAVA	56
ΕΙΚΟΝΑ 19 : ΣΕΛΙΔΑ ORACLE JAVA	57
ΕΙΚΟΝΑ 20 : ΣΕΛΙΔΑ ΛΗΨΗΣ JAVA JDK	57
ΕΙΚΟΝΑ 21 : ΣΕΛΙΔΑ ΛΗΨΗΣ STANFORD CORENLP	58
ΕΙΚΟΝΑ 22 : ΓΚΑΜΑ ΣΥΣΚΕΥΩΝ ΓΙΑ ΤΙΣ ΟΠΟΙΕΣ ΕΙΝΑΙ ΔΙΑΘΕΣΙΜΗ Η ΕΦΑΡΜΟΓΗ BRAINTAG	63
ΕΙΚΟΝΑ 23 : ΟΘΟΝΕΣ ΕΓΓΡΑΦΗΣ ΚΑΙ ΕΙΣΟΔΟΥ ΤΗΣ ΕΦΑΡΜΟΓΗΣ BRAINTAG	64
ΕΙΚΟΝΑ 24 : ΑΡΧΙΚΗ ΟΘΟΝΗ ΤΩΣ ΕΦΑΡΜΟΓΩΝ.....	65
ΕΙΚΟΝΑ 25 : ΟΘΟΝΗ ΠΛΗΡΟΦΟΡΙΩΝ ΕΦΑΡΜΟΓΗΣ BRAINTAG	66
ΕΙΚΟΝΑ 26 : ΕΤΙΚΕΤΕΣ ΣΥΝΤΑΚΤΙΚΟΥ PENN TREEBANK ΕΦΑΡΜΟΓΗΣ BRAINTAG.....	67
ΕΙΚΟΝΑ 27 : ΑΝΑΛΥΣΗ ΠΑΡΑΓΡΑΦΟΥ ΕΦΑΡΜΟΓΗΣ BRAINTAG	68
ΕΙΚΟΝΑ 28 : ΔΙΑΔΙΚΑΣΙΑ ΠΑΙΧΝΙΔΙΟΥ ΕΦΑΡΜΟΓΗΣ BRAINTAG	69
ΕΙΚΟΝΑ 29 : ΚΑΤΗΓΟΡΙΕΣ ΠΑΙΧΝΙΔΙΟΥ ΕΦΑΡΜΟΓΗΣ BRAINTAG	70
ΕΙΚΟΝΑ 30 : ΟΘΟΝΗ ΕΙΣΑΓΩΓΗΣ ΚΕΙΜΕΝΟΥ ΕΦΑΡΜΟΓΗΣ BRAINTAG	71
ΕΙΚΟΝΑ 31 : PROJECT STRUCTURE BRAINTAG BACKEND.....	80
ΕΙΚΟΝΑ 32 : NATIVE APPLICATION PROJECT STRUCTURE.....	146

Λίστα Πινάκων

ΠΙΝΑΚΑΣ 1 : ΒΑΣΙΚΗ ΛΙΣΤΑ ΕΤΙΚΕΤΩΝ ΠΡΟΣ ΧΡΗΣΙΜΟΠΟΙΗΣΗ ΚΑΤΑ ΤΗΝ ΔΙΕΡΓΑΣΙΑ POS TAGGING	49
ΠΙΝΑΚΑΣ 2 : ΛΙΣΤΑ ΕΤΙΚΕΤΩΝ UPENN TREEBANK II	51
ΠΙΝΑΚΑΣ 3 : ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΟΙ ΟΠΟΙΕΣ ΜΠΟΡΟΥΝ ΝΑ ΧΡΗΣΙΜΟΠΟΙΗΣΟΥΝ ΤΗ ΒΙΒΛΙΟΘΗΚΗ STANFORD CORENLP	55
ΠΙΝΑΚΑΣ 4 : ΕΠΕΞΗΓΗΣΗ ΕΝΤΟΛΗΣ ΓΡΑΜΜΗΣ ΕΝΤΟΛΩΝ ΤΟΥ STANFORD CORENLP	60
ΠΙΝΑΚΑΣ 5 : ΠΙΝΑΚΑΣ USERS.....	74
ΠΙΝΑΚΑΣ 6 : ΠΙΝΑΚΑΣ WRONG_ANSWERS.....	76
ΠΙΝΑΚΑΣ 7 : ΕΓΚΑΤΑΣΤΑΣΗ ΑΠΑΡΑΙΤΗΩΝ ΠΑΚΕΤΩΝ ΣΤΟ ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ.....	79
ΠΙΝΑΚΑΣ 8 : HTTP ΘΥΡΕΣ ΕΞΥΠΗΡΕΤΗΤΗ.....	87

Πίνακας Περιεχομένων

Περίληψη	5
Abstract.....	8
Ευχαριστίες.....	10
Λίστα εικόνων	11
Λίστα Πινάκων.....	12
Πίνακας Περιεχομένων	13
1. Εισαγωγή	17
1.1 Προβλήματα και κίνητρα για τη συγγραφή της εργασίας	17
1.2 Συμβολή Εργασίας.....	20
1.3 Συνοπτική Παρουσίαση και Δομή Εφαρμογών	22
1.3.1 Εισαγωγή.....	22
1.3.2 Κορμός Εφαρμογής.....	22
1.3.3 Android Εφαρμογή.....	23
1.3.4 Εφαρμογή Υπολογιστών	23
1.4 Σχετική Βιβλιογραφία - Ιστορική Αναδρομή	24
2. Επεξεργασία Φυσικής Γλώσσας.....	28
2.1 Τι είναι η Επεξεργασία Φυσικής Γλώσσας.....	28
2.2 Κατηγορίες Επεξεργασίας Φυσικής Γλώσσας.....	29
2.2.1 Λημματοποίηση, και Μορφολογική Τμηματοποίηση	29
2.2.2 Αναγνώριση Λέξης ως Μέρος του Λόγου.....	30
2.2.3 Ανάλυση Κειμένου.....	30
2.2.4 Διαχωρισμός Προτάσεων	31
2.2.5 Απόσπαση Ρίζας Λέξεων.....	32
2.2.6 Κατάτμηση Λέξεων.....	33
2.2.7 Λεξιλογική Σημασιολογία	33
2.2.8 Μηχανική Μετάφραση	34
2.2.9 Αναγνώριση Οντοτήτων.....	35
2.2.10 Παραγωγή Φυσικής Γλώσσας	36
2.2.11 Κατανόηση Φυσικής Γλώσσας.....	36
2.2.12 Οπτική Αναγνώριση Χαρακτήρων	37

2.2.13	Απάντηση Ερωτήσεων	38
2.2.14	Κειμενική Συνεπαγωγή	39
2.2.15	Αναγνώριση Σχέσεων.....	39
2.2.16	Ανάλυση Συναισθήματος	40
2.2.17	Αυτόματη Περίληψη Κειμένου	40
2.2.18	Ανάλυση Λόγου	42
2.2.19	Αναγνώριση Ομιλίας.....	42
2.2.20	Τμηματοποίηση Ομιλίας	44
2.2.21	Σύνθεση Ομιλίας	45
2.3	Αναγνώριση Λέξεων ως Μέρος του Λόγου	47
2.3.1	Που χρησιμεύει το Part-Of-Speech Tagging	48
2.3.2	Επιλογή λίστας ετικετών	49
2.4	Stanford Natural Language Processing Library	53
2.4.1	Συνοπτική Εικόνα.....	53
2.4.2	Διαθεσιμότητα Συστημάτων και Προγραμματιστικών Γλωσσών	54
2.4.3	Οδηγίες Χρήσης Γραμμής Εντολών	55
2.4.3.1	Εγκατάσταση Java Environment	56
2.4.3.2	Λήψη Βιβλιοθήκης Stanford CoreNLP	58
2.4.3.3	Χρήση Stanford CoreNLP μέσω γραμμής εντολών	59
3.	BrainTag	63
3.1	Οδηγός Χρήσης Εφαρμογής	63
3.1.1	Λήψη και εγκατάσταση εφαρμογής	63
3.1.2	Εγγραφή και είσοδος.....	64
3.1.3	Αρχικό μενού.....	64
3.1.4	Πληροφορίες Εφαρμογής	66
3.1.5	Ετικέτες συντακτικού Penn TreeBank	67
3.1.6	Ανάλυση Παραγράφου.....	68
3.1.7	Διαδικασία Παιχνιδιού	69
3.1.8	Επιλογή Κατηγοριών Παιχνιδιού	70
3.1.9	Εισαγωγή Κειμένου.....	71
3.1.10	Λοιπές επιλογές παιχνιδιού	72
4.	Βάση Δεδομένων & Εξυπηρετητής Δεδομένων	73
4.1	Βάση Δεδομένων	73
4.1.1	Πίνακας Users	73
4.1.2	Πίνακας wrong_answers	74
4.1.3	Σχηματική Δημιουργία Βάσης Δεδομένων	77
4.2	Εξυπηρετητής Δεδομένων	78
4.2.1	Λειτουργικό Σύστημα.....	78
4.2.3	Κώδικας Εξυπηρετητή	79
5.	Εφαρμογή κινητών, λειτουργικού συστήματος Android	90

5.1 Κώδικας Εφαρμογής	90
5.1.1 AndroidManifest.xml	90
5.1.2 colors.xml & styles.xml.....	92
5.1.3 strings.xml και dims.xml	92
5.1.4 navigation_drawer_items.xml	93
5.1.5 enter_from_left.xml , enter_from_right.xml , exit_to_left.xml , exit_to_right.xml	94
5.1.6 Helpers/CategoriesExpandableListAdapter.java.....	95
5.1.7 Helpers/helper.java.....	96
5.1.8 Helpers/MyDragListener.java	99
5.1.9 Helpers/MyTouchListener.java	100
5.1.10 Helpers/PennPosTagAdapter.java	101
5.1.11 Helpers/PredicateLayout.java.....	102
5.1.12 Helpers/ScoreboardAdapter.java	104
5.1.13 MainActivity.java & activity_main.xml.....	106
5.1.14 CategoriesActivity.java & activity_categories.xml	108
5.1.15 GameActivity.java & activity_game.xml	111
5.1.16 GameResultActivity.java & activity_game_result.xml	116
5.1.17 LoginActivity.java & activity_login.xml	121
5.1.18 ParagraphReviewActivity.java & activity_paragraph_review.xml	125
5.1.19 RegisterActivity.java & activity_register.xml	128
5.1.20 TypeActivity.java & activity_type.xml	132
5.1.21 InfoFragment.java & fragment_info.xml	134
5.1.22 MainMenuFragment.java & fragment_main_menu.xml	135
5.1.23 PenPosTagListFragment.java & fragment_pen_pos_tag_list.xml.....	139
5.1.24 ScoreboardFragment.java & fragment_scoreboard.xml	140
5.1.25 build.gradle (Module : app).....	143
6. Εφαρμογή Υπολογιστών (OS Windows, OSX, Linux)	145
6.1 Παράθεση Κώδικα	145
6.1.1 Project Structure	145
6.1.2 package.json	147
6.1.3 css/style.css	147
6.1.4 html/categories.html & js/categories.js	150
6.1.5 html/game.html & js/game.js	151
6.1.6 html/index.html & js/index.js	155
6.1.7 html/info.html & js/info.js	158
6.1.8 html/login.html	159
6.1.9 html/paragraph_review.html & js/paragraph_review.js	162
6.1.10 html/register.html & js/register.js	164
6.1.11 html/tags.html.....	166
6.1.12 html/type.html & js/type.js	168
6.1.13 js/main.js	169

7. Συμπεράσματα	171
Βιβλιογραφία.....	172

1. Εισαγωγή

1.1 Προβλήματα και κίνητρα για τη συγγραφή της εργασίας

Η γλώσσα αποτελεί κομμάτι της επικοινωνίας, με άλλα λόγια είναι ένα καίριο συστατικό για την συνύπαρξη και συχνά για την επιβίωση των ανθρώπων. Η επικοινωνία βασίζεται σε άτυπους κανόνες τους οποίους χρησιμοποιούμε στην καθημερινότητά μας, χωρίς να έχουμε συνείδηση της πράξης αυτής. Θα ήταν εξάλλου φοβερά κουραστικό να σκεφτόμασταν κάθε φορά τον κατάλληλο τρόπο δημιουργίας μιας και μόνο πρότασης, πόσο μάλλον την κατασκευή ενός διαλόγου. Σπάνιες είναι οι φορές που οι περιστάσεις υποδεικνύουν αυστηρά προσεγμένη ομιλία ή γραπτή επικοινωνία. Δεν είναι τυχαίο το γεγονός ότι από πολύ νωρίς, στα πρώτα κιόλας στάδια της ζωής, ο άνθρωπος κατακτά ασυνείδητα τις δομικές λειτουργίες του λόγου, κάτι που αργότερα το σχολείο συμπληρώνει και τελειοποιεί.

Η τυπική εκπαίδευση συμπεριλαμβάνει, εκτός των άλλων, την άριστη γνώση γραμματικών κανόνων και συντακτικών εννοιών. Στις πρώτες τάξεις του Δημοτικού οι μαθητές καλούνται να κατανοήσουν βασικές χρήσεις της γλώσσας τους και να αποδώσουν τις νεοαποκτηθείσες γνώσεις στον γραπτό και προφορικό λόγο. Στόχος της συγκεκριμένης πρακτικής είναι η κατανόηση από την πλευρά των μαθητών της πολυπλοκότητας που κρύβει η επικοινωνία. Γλωσσολόγοι και καθηγητές Πανεπιστημίου εργάζονται ακατάπαυστα με μοναδικό σκοπό την ανάδειξη της ορθής χρήσης της γλώσσας και την εύρεση του καταλληλότερου τρόπου διδασκαλίας αυτής.

Ωστόσο, παρά την προσπάθεια ειδικών και εκπαιδευτικών, στη σύγχρονη βιβλιογραφία παρατηρείται μία απαξίωση από μέρους των νέων ως προς τις δεξιότητες της επικοινωνίας. Σήμερα, είναι ευρέως αποδεκτό το γεγονός ότι τα άτομα νέων ηλικιών απομακρύνονται όλο και περισσότερο από τις τυπικές μεθόδους μάθησης και εκπαίδευσης. Εργαλεία, που παλαιότερα ήταν χρήσιμα για τον δάσκαλο και τον μαθητή, όπως για παράδειγμα η γραμματική και το συντακτικό, στις μέρες μας τείνουν ολοένα να υποβαθμίζονται. Η αιτία των παραπάνω φαινομένων δε φαίνεται να εστιάζεται σε μία μόνο πηγή.

Αρχικά, οι απαιτητικοί ρυθμοί της σύγχρονης αστικής κοινωνίας προωθούν τη σύντομη επικοινωνία, καθιστώντας την, μάλιστα, απαραίτητη. Αποτέλεσμα όλων αυτών είναι τα νεαρά άτομα να ξεχνούν, ή πολλές φορές να μην αφομοιώνουν καν, τους συντακτικούς και γραμματικούς κανόνες που υπαγορεύει η εκάστοτε γλώσσα. Η χρήση γραμματικο-συντακτικών κανόνων περιορίζεται αυστηρά στα πλαίσια του σχολείου. Πρακτικά, αυτό σημαίνει, ότι οι άνθρωποι που καλούνται να απαρτίσουν τις κοινωνίες του μέλλοντος δεν γνωρίζουν πώς να γράφουν και ποιος είναι ο επίσημος ενδεδειγμένος τρόπος επικοινωνίας. Η επικοινωνία τους και ο τρόπος τους να μοιραστούν ιδέες ακολουθεί μια νέα τάση, που θα μπορούσαμε κωμικά να περιγράψουμε με την φράση “όσο πιο απλό, τόσο πιο κατανοητό”. Αποτέλεσμα όλων αυτών είναι ο πλούτος των γλωσσών, που αποτελούν πολιτιστική κληρονομιά των λαών, να θυσιάζεται σταδιακά στο όνομα της ευκολίας και της οικονομίας του χρόνου.

Επιπλέον, ο νέος τρόπος ζωής έχει απομακρύνει κυρίως τον νεανικό πληθυσμό από την κλασική έννοια του διαβάσματος. Ένα καλογραμμένο βιβλίο έχει σήμερα σημαντικά μικρότερη αξία σε σύγκριση με το παρελθόν. Τα παιδιά προτιμούν πολύ περισσότερο να ασχοληθούν με τεχνολογικά προϊόντα. Η εικόνα και η ψηφιοποίηση των μέσων τραβούν επιτακτικά την προσοχή του ανθρώπου οποιασδήποτε ηλικίας. Ιδιαίτερα στον δυτικό πολιτισμό, η εξοικείωση με τα τεχνολογικά μέσα είναι μεγάλη και θεωρείται πια δεδομένη και απαραίτητη. Όπως είναι φυσικό, η παραδοσιακή έννοια της μελέτης ανήκει στο παρελθόν. Νέες εκπαιδευτικές τάσεις που επωφελούνται από την απήχηση της τεχνολογίας ξεπροβάλλουν και στοχεύουν στην πρόκληση του ενδιαφέροντος των μαθητών. Οι παρουσιάσεις με τη μορφή Power Point που χρησιμοποιούνται πλέον ευρέως σε όλες τις βαθμίδες της εκπαίδευσης είναι χαρακτηριστικό παράδειγμα του γεγονότος αυτού. Ακολουθώς, ο μαθητής χάνει την έννοια της συνοχής του κειμένου και συνηθίζει σε ένα πιο απλό και σχηματικό μοτίβο μελέτης, με αποτέλεσμα φυσικά να εξαλείφονται και όλες οι συντακτικές ιδιότητες της γλώσσας.

Ωστόσο, η καλπάζουσα πρόοδος του είδους μας και οι αλλαγές που αυτή συνεπάγεται δεν είναι εφικτό να σταματήσουν, ακόμα και αν υπάρχουν τέτοιου είδους απώλειες. Ο σοφός άνθρωπος, εξάλλου, δεν στέκεται ποτέ αντίθετα στις εξελίξεις. Η τεχνολογία έχει εισβάλλει δυναμικά στις ζωές όλων μας και η παρουσία της είναι πολλά υποσχόμενη. Κάθε στάδιο της

ανθρώπινης ζωής έχει επηρεαστεί από τις τεχνολογικές προόδους Βρέφη και παιδιά νηπιακής και προσχολικής ηλικίας εξιτάρονται στη θέα μιας οθόνης με κινούμενη εικόνα, ενώ από πολύ νωρίς παιδιά σχολικής ηλικίας γνωρίζουν να χειρίζονται άψογα τον υπολογιστή, το κινητό και άλλα τεχνολογικά εργαλεία.

Όπως αναφέρθηκε και νωρίτερα η μάθηση σε οποιαδήποτε μορφή και ηλικία αξιοποιεί τα νέα μέσα. Το διαδίκτυο βρίθει από εφαρμογές και παιχνίδια που αναφέρονται σε άτομα σε ποικίλα στάδια ανάπτυξης. Τα παιδιά απολαμβάνουν μέσω παιχνιδιού, μιας εύκολης, διασκεδαστικής και χαλαρωτικής διαδικασίας να έρχονται σε επαφή με τα γράμματα και τις νέες γνώσεις. Η μάθηση παύει να νοηματοδοτείται με αρνητικούς όρους, μπορεί να λαμβάνει χώρα σε οποιοδήποτε μέρος, οποιαδήποτε στιγμή της ημέρας και να βοηθάει τα νεαρά άτομα να περνούν την ώρα τους εποικοδομητικά. Κάποιες εφαρμογές, επιπλέον, παρέχουν την επιλογή της συνεργατικής ενασχόλησης με αντικείμενα μάθησης, μέσω της οποίας προωθείται το συνεργατικό πνεύμα και αναπτύσσονται οι αλληλοδιδασκτικές ικανότητες. Εξάλλου, είναι χαρακτηριστικό το παράδειγμα των Βόρειων Ευρωπαϊκών χωρών, που έχουν εντάξει ήδη στο εκπαιδευτικό τους σύστημα τη χρήση Ηλεκτρονικών Υπολογιστών και άλλων ψηφιακών μέσων. Τα μέσα αυτά βοηθούν το δάσκαλο να οπτικοποιεί τα δεδομένα, να διευκολύνει τη διδασκαλία και να διατηρεί το ενδιαφέρον των μαθητών αμείωτο.

Ωστόσο, δεν θα ήταν σωστό να παραλείψουμε να τονίσουμε ότι, όπως κάθε θετική κατάσταση κρύβει και κάποια αρνητική χροιά, έτσι και εδώ η χρήση των τεχνολογικών επιτευγμάτων ενέχει τον κίνδυνο του εθισμού. Πάντα με την κατάλληλη προστασία και επιτήρηση τα παιδιά, ιδιαίτερα αυτά των μικρότερων ηλικιών, είναι απαραίτητο να αναπτύξουν σταδιακά ενός είδους αυτοκυριαρχία και να κατανοήσουν ότι τα τεχνολογικά μέσα δεν είναι άλλο παρά βοηθητικά εργαλείο στη διαδικασία. Η ιδέα ότι η μάθηση περνά απαραίτητα μέσα από το σχολείο, ο δάσκαλος είναι ο μόνος που καταφέρνει να μεταδώσει πέρα από γνώσεις και μια σειρά από απαραίτητες αρχές και ότι το βιβλίο παραμένει ένας πιστός σύντροφος είναι κεντρικής σημασίας.

1.2 Συμβολή Εργασίας

Η ιδέα της δημιουργίας της συγκεκριμένης εφαρμογής στηρίχθηκε στα όσα έχουν ειπωθεί παραπάνω. Ωστόσο, η εφαρμογή αυτή στοχεύει στο να εξυπηρετήσει και κάποιους σκοπούς, να συμβάλλει στην πρόοδο κάποιων τομέων και να βοηθήσει στην ανάδειξη νέων τεχνολογικών μέσων. Η συμβολή της εργασίας είναι μεγάλη και διευρυμένη σε ποικίλους τομείς.

Ο πρώτος στόχος που φαίνεται να ικανοποιεί η δημιουργία του “Brain Tag” είναι ο συνδυασμός μάθησης και παιχνιδιού. Σε αντίθεση με άλλες εφαρμογές που προσφέρουν μόνο διασκέδαση, χωρίς κανένα επιπλέον όφελος, το “BrainTag” επικεντρώνεται στην ποιοτική ψυχαγωγία, όπου η χαλάρωση συνοδεύει τη μάθηση. Συγκρινόμενη με άλλες παρόμοιες, η συγκεκριμένη εφαρμογή, προσεγγίζει ένα ευρείας ηλικίας κοινό, το οποίο έχει τη δυνατότητα πέρα από το να μάθει ή να ανανεώσει τις συντακτικές του γνώσεις, να διασκεδάσει και να περάσει ευχάριστα το χρόνο του. Η μάθηση μετατρέπεται σε μία εύκολη διαδικασία, αγαπητή από τους χρήστες. Σταδιακά, οι συντακτικοί κανόνες εσωτερικεύονται και τότε ο παίκτης είναι σε θέση να τους χρησιμοποιήσει και να τους εφαρμόσει και εκτός παιχνιδιού, σε περισσότερο σημαντικές περιστάσεις της καθημερινής ζωής, όπως για παράδειγμα στη συγγραφή ενός κειμένου για το σχολείο ή το Πανεπιστήμιο, ή σε μία επιστολή.

Δεν θα μπορούσε να παραλειφθεί το γεγονός, ότι η σύλληψη και η δημιουργία της εφαρμογής “BrainTag” δύναται να επηρεάσει τη δουλειά και άλλων προγραμματιστών. Σε μία εποχή που γίνονται γιγαντιαίες προσπάθειες για τη διάσωση των ανθρωπιστικών αξιών, οι εκπρόσωποι της τεχνολογίας δεν θα πρέπει να παραβλέπουν αυτόν τον αγώνα. Εφαρμογές, όπως αυτή, που στηρίζουν θεμελιώδεις λειτουργίες της κοινωνίας, θα συνέβαλαν στην προώθηση των αξιών και των άγραφων κανόνων. Μερικά παραδείγματα θα μπορούσαν να είναι εφαρμογές για άτομα με ειδικές σωματικές ή πνευματικές ανάγκες, εφαρμογές με εκπαιδευτικό ή ενημερωτικό περιεχόμενο, εφαρμογές που θα απευθύνονται σε άτομα τρίτης ηλικίας κ.ο.κ. Τα προϊόντα αυτά θα διεύρυναν, επίσης, την ποικιλία των εφαρμογών που βρίσκονται ήδη στην κυκλοφορία. Το “Brain Tag” έχει ως στόχο να εμπνεύσει τους ενδιαφερόμενους και να στηρίξει την διάσωση της γλώσσας, μέρος της πολιτισμικής μας κληρονομιάς.

Μια ακόμα αναμφίβολα μεγάλη συμβολή της συγκεκριμένης δημιουργίας είναι η εξοικείωση του ελληνικού απασχολούμενου με τα νέα τεχνολογικά μέσα πληθυσμού με τον τομέα της Επεξεργασίας Φυσικής Γλώσσας. Ακριβώς επειδή στον ελληνικό χώρο, ο συγκεκριμένος τομέας δεν είναι διαδεδομένος, η δημοσίευση σχετικών εργασιών, θα αναδείκνυε τα οφέλη που αυτός μπορεί να προσφέρει. Στόχος είναι η ενασχόληση όλο και περισσότερων προγραμματιστών με την επεξεργασία της φυσικής γλώσσας, ένας τομέας που έχει τη δυνατότητα να συμβάλλει τα μέγιστα στην εξέλιξη του προγραμματισμού. Ακόμα, όσο ταχύτερα διαδίδεται η χρήση της επεξεργασίας φυσικού κειμένου, τόσο περισσότερο ανταγωνιστικός γίνεται ο κλάδος και τόσο περισσότερες ευφάνταστες και προγραμματιστικά καινοτόμες εφαρμογές προκύπτουν. Μακροπρόθεσμα, ενδέχεται η πολυπλοκότητα των εφαρμογών να προσελκύσει και άτομα που έως τώρα δίσταζαν να επιδοθούν με την ενασχόληση των τεχνολογικών μέσων, να κάνουν τα πρώτα τους βήματα στο χώρο.

Ωστόσο, η δημιουργία της συγκεκριμένης εφαρμογής είχε ως επιπλέον στόχο την ανέλιξη σε προσωπικό επίπεδο. Η σύλληψη της ιδέας στάθηκε αρκετά δύσκολη, αφού σκοπός δεν ήταν η ανάπτυξη ενός προϊόντος πανομοιότυπου με αυτά που ήδη κυκλοφορούν μαζικά στο διαδίκτυο, αλλά η οικοδόμηση μιας εφαρμογής που θα προσφέρει καινούριες δυνατότητες. Οι περιορισμοί όμως αυτοί είχαν προεκτάσεις και στην προσωπική εργασία, όπου αρχικά σπαταλήθηκε ένα μεγάλο μέρος του χρόνου για να βρεθεί ακριβώς αυτή η καινοτόμα ιδέα, ενώ αργότερα χρειάστηκε να κατακτηθεί και να κατανοηθεί απόλυτα η λειτουργία και η χρήση των εργαλείων που απαιτούσε η υλοποίηση της ιδέας. Υπήρξε τριβή με αντικείμενα που δεν ήταν γνωστικά εύκολα και προέκυψαν δυσκολίες που επιζητούσαν άμεση παρέμβαση και επίλυση. Όλη αυτή η διαδικασία αποτέλεσε μία χρήσιμη εμπειρία, που συνέβαλε στη διεύρυνση των γνώσεων και των ικανοτήτων του δημιουργού, μια εμπειρία γεμάτη εφόδια που ως τώρα δεν υπήρξε η ευκαιρία να αποκτηθούν διαφορετικά.

1.3 Συνοπτική Παρουσίαση και Δομή Εφαρμογών

1.3.1 Εισαγωγή

Ο προγραμματισμός της παρούσας εργασίας κινήθηκε εξ ολοκλήρου γύρω από την ιδέα δημιουργίας μιας υπηρεσίας, και όχι μιας εφαρμογής/παιχνιδιού. Η υπηρεσία είχε δύο σκοπούς. Ο πρώτος και κύριος σκοπός, ήταν η προσφορά της καλύτερης εμπειρίας χρήσης στον τελικό χρήστη, και της εκμάθησης του συντακτικού στον υψηλότερο δυνατό βαθμό. Αυτό καλύπτει την επιτυχημένη δημιουργία της εφαρμογής/παιχνιδιού. Γνωρίζοντας και χρησιμοποιώντας την δύναμη της Επεξεργασίας Φυσικού Κειμένου, πάντα στα πλαίσια της λογικής του ανοιχτού λογισμικού, προέκυψε και ο ορισμός ενός δεύτερου σκοπού. Η δημιουργία κορμού της εφαρμογής, ο οποίος θα είναι ανοιχτός να χρησιμοποιηθεί από τους τρίτους προγραμματιστές, απευθείας μέσω των μηχανημάτων του πανεπιστημίου, είτε η δυνατότητα επαναχρησιμοποίησης του υπάρχοντος κώδικα, για δημιουργία νέων εργασιών. Αυτό έφερε τις εξής αλλαγές και προσθήκες στην δημιουργία του κώδικα της εργασίας:

Αρχικά, όλη η επεξεργασία φυσικού κειμένου γίνεται από τον κύριο κορμό, και όχι από τις εφαρμογές ξεχωριστά. Χρησιμοποιώντας τη γλώσσα προγραμματισμού Java για τη συγκεκριμένη διεργασία, αυτή θα μπορούσε να γίνει τοπικά στις ίδιες τις τελικές συσκευές των χρηστών, χωρίς την ανάγκη σύνδεσης στο ίντερνετ. Ωστόσο, θεωρήθηκε προτιμότερο η υπηρεσία να αξιοποιείται και από τις ίδιες τις εφαρμογές, εφόσον αυτή θα είναι ανοιχτή προς όλον τον κόσμο και συγκεκριμένα στους προγραμματιστές προς αξιοποίηση.

1.3.2 Κορμός Εφαρμογής

Ο κορμός, δημιουργήθηκε σε γλώσσα NodeJs σε ένα ποσοστό 90%, με τη βοήθεια της γλώσσας Java[12] σε ποσοστό 10% αποκλειστικά για την επεξεργασία φυσικού κειμένου. Υπάρχουν τερματικά σημεία για τις εξής διεργασίες:

- Εγγραφή χρήστη
- Σύνδεση χρήστη
- Λήψη πίνακα σκορ, και θέσης κατάταξης χρήστη
- Συντακτική ανάλυση κειμένου
- Λήψη τυχαίας παραγράφου από την ιστοσελίδα της Wikipedia και συντακτική ανάλυση του κειμένου
- Λήψη κατηγοριών παιχνιδιού
- Λήψη πίνακα με τη συχνότητα εμφάνισης λαθών, ανά συντακτική ετικέτα
- Προσθήκη βαθμού στον πίνακα των σκορ

1.3.3 Android Εφαρμογή

Η πρώτη εφαρμογή που θα χρησιμοποιεί τον κορμό, αναπτύχθηκε για το κινητό λειτουργικό σύστημα Android. Η απόφαση αυτής της ενέργειας ήταν στρατηγική για τους παρακάτω λόγους. Το προγραμματιστικό περιβάλλον του Android προσφέρει αρκετές βοήθειες στον προγραμματιστή, με αποτέλεσμα να αυξάνει την ταχύτητα δημιουργίας της πρώτης δοκιμαστικής έκδοσης του παιχνιδιού. Με την ταχύτερη δημιουργία και την ευκολότερη διόρθωση σφαλμάτων, που προέκυπταν από τις πρώτες δοκιμαστικές χρήσεις από τον δημιουργό και τρίτα πρόσωπα, επιτεύχθηκε η οριστικοποίηση του βασικού γραφικού κορμού των εφαρμογών. Ο κορμός αυτός περιλαμβάνει τα βασικά χρώματα και χρωματισμούς των εφαρμογών, τα γραφικά στοιχεία, όπως κουμπιά και πίνακες, και τη θέση τους στην εκάστοτε οθόνη. Ένας ακόμα λόγος που πρώτα δημιουργήθηκε η κινητή εφαρμογή, ήταν συγκέντρωση περισσότερων σχολίων, μέσω της ευκολότερης επίδειξης της σε τρίτα άτομα.

1.3.4 Εφαρμογή Υπολογιστών

Το τελευταίο βασικό μέρος του προγραμματιστικού κομματιού της εργασίας, ήταν η δημιουργία αντίστοιχης εφαρμογής για σταθερούς και κινητούς υπολογιστές. Οι επαναλήψεις

και οι διορθώσεις ήταν εμφανώς λιγότερες σε αριθμό, εφόσον υπήρχε έτοιμος από τα προηγούμενα βήματα ο βασικός γραφικός και λειτουργικός κορμός. Η εφαρμογή έχει βελτιστοποιηθεί για να εκτελείται σε όλα τα γνωστά λειτουργικά συστήματα (Microsoft Windows, Apple MacOSX, Εκδόσεις Linux) έτσι ώστε να καλύπτει όσο το δυνατόν μεγαλύτερο ποσοστό της αγοράς υπολογιστών. Η κάλυψη όλων των λειτουργικών συστημάτων επιτεύχθηκε με τη χρήση της τεχνικής προγραμματισμού πολλαπλών περιβαλλόντων (Cross Platform Development). Χρησιμοποιήθηκε η υπηρεσία electron, η οποία προσφέρει τη δημιουργία φυσικών εφαρμογών υπολογιστών (native desktop programs), με τη χρήση κυρίως τεχνικών προγραμματισμού του διαδικτύου, με την βοήθεια των γλωσσών HTML [13] , CSS [14] και Javascript [15].

1.4 Σχετική Βιβλιογραφία - Ιστορική Αναδρομή

Η ιστορία της επεξεργασίας φυσικής γλώσσας επίσημα ξεκινάει τη δεκαετία του 1950. Ο Alan Turing δημοσίευσε το άρθρο με τίτλο “Computing Machinery and Intelligence”[16] το οποίο έθεσε τις βάσεις για αυτό που σήμερα ονομάζουμε ‘Turing Test’[17]. Το Turing Test είναι το βασικό τεστ ευφυΐας των μηχανών.

Το 1954, αναπτύχθηκε ένα πείραμα στο πανεπιστήμιο του Georgetown, το οποίο αποσκοπούσε στην αυτοματοποιημένη μετάφραση περισσότερων από 60 Ρωσικών προτάσεων στην αγγλική γλώσσα. Οι εισηγητές του πειράματος ισχυρίστηκαν ότι στα επόμενα τρία με πέντε χρόνια η μηχανική μετάφραση[18] δε θα αποτελούσε τεχνικό πρόβλημα. Ωστόσο η πραγματική πρόοδος αποδείχθηκε αρκετά πιο χρονοβόρα. Το 1966, η έκθεση ALPAC[19] ανακάλυψε ότι μετά από δέκα χρόνια ερευνών η μηχανική μετάφραση δεν κάλυπτε τις αρχικές προσδοκίες, το οποίο είχε ως αποτέλεσμα τη δραματική μείωση των χρηματοδοτήσεων της στον κλάδο. Τα επόμενα χρόνια δεν σημειώθηκε περαιτέρω έρευνα στην μηχανική μετάφραση, μέχρι και τα τέλη του 1980, όπου αναπτύχθηκαν τα πρώτα μηχανήματα στατιστικής μετάφρασης.

Δύο ιδιαίτερα επιτυχή συστήματα επεξεργασίας φυσικής γλώσσας αναπτύχθηκαν τη δεκαετία του 1960. Το πρώτο σύστημα ονομάστηκε SHRDLU[20], σύστημα το οποίο δούλευε αποκλειστικά με “μπλοκ λέξεων” και περιορισμένο λεξιλόγιο. Το δεύτερο σύστημα ήταν η ELIZA. Η ELIZA δρούσε σαν προσομοιωτής ενός ψυχιάτρου, ο οποίος λειτουργεί με την τεχνική Rogerian[21] και δημιουργήθηκε από τον Joseph Weizenbaum ανάμεσα στο 1964 και 1966. Έχοντας μηδαμινή γνώση πάνω στην ανθρώπινη σκέψη και το συναίσθημα, αρκετές φορές κατάφερε να αλληλεπιδράσει σε ανθρώπινο επίπεδο. Όταν ο «ασθενής» ξεπερνούσε την αρκετά περιορισμένη βάση γνώσεών της, η ELIZA απαντούσε με μια γενικευμένη απάντηση. Για παράδειγμα, η απάντηση της στην πρόταση “Πονάει το κεφάλι μου” θα ήταν “Γιατί θεωρείς ότι πονάει το κεφάλι σου;”.

Κατά τη δεκαετία του 1970, αρκετοί προγραμματιστές άρχισαν να γράφουν «εννοιολογικές οντολογίες», οι οποίες δόμησαν πληροφορίες του πραγματικού κόσμου σε δεδομένα κατανοητά από τους υπολογιστές. Τέτοια παραδείγματα είναι τα: MARGIE (Schank, 1975), SAM (Cullingford, 1978), PAM (Wilensky, 1978), TaleSpin (Meehan, 1976), και QUALM (Lehnert, 1977). Αυτήν τη χρονική περίοδο, δημιουργήθηκαν αρκετά αυτοματοποιημένα ρομπότ συνομιλίας (chatterbots - chat bots)[22]. Παραδείγματα γνωστών συστημάτων ρομπότ συνομιλίας ήταν τα: PARRY, Racter, και το Jabberwacky.

Μέχρι τη δεκαετία του 1980, τα περισσότερα συστήματα επεξεργασίας φυσικής γλώσσας βασίζονταν σε πολύπλοκα σύνολα χειρόγραφων κανόνων. Στα τέλη της δεκαετίας του 1980, ωστόσο, υπήρξε μια επανάσταση στον τομέα με την εισαγωγή των αλγορίθμων μηχανικής μάθησης για την επεξεργασία της γλώσσας. Αυτό οφείλεται τόσο στην σταθερή αύξηση της υπολογιστικής ισχύος (βλέπε νόμο του Moore) και τη σταδιακή αποδυνάμωση της κυριαρχίας των Chomskyan θεωριών[23] της γλωσσολογίας (π.χ. μετασχηματιστική γραμματική), των οποίων το θεωρητικό υπόβαθρο αποθάρρυνε το είδος της γλωσσολογίας σωμάτων (corpus linguistics), που κρύβεται πίσω από την προσέγγιση μηχανικής μάθησης στην επεξεργασία της γλώσσας. Μερικοί από τους πρώτους αλγορίθμους μηχανικής μάθησης που χρησιμοποιήθηκαν, όπως τα δέντρα απόφασης, παράχθηκαν από συστήματα αν-τότε κανόνων (if-then rules) παρόμοιους με τους ισχύοντες χειρόγραφους κανόνες. Εντούτοις, η αναγνώριση λέξεων ως μέρος του λόγου (Part-of-Speech tagging, συντομογραφία POS) εισήγαγε όλο και περισσότερο

τη χρήση των κρυφών μοντέλων Markov[24] στην επεξεργασία φυσικής γλώσσας. Επίσης, οι έρευνες άρχισαν να επικεντρώνονται σε στατιστικά μοντέλα, τα οποία χρησιμοποιούν πιθανολογικές αποφάσεις, με βάση τη σύνδεση πραγματικών τιμών-βαρών με τα χαρακτηριστικά που συνθέτουν τα δεδομένα εισόδου. Τα μοντέλα γλώσσας προσωρινής μνήμης (Cache Language models)[25], πάνω στα οποία βασίζονται πολλά συστήματα αναγνώρισης ομιλίας, είναι παραδείγματα τέτοιων στατιστικών μοντέλων. Τέτοια μοντέλα είναι γενικά πιο ισχυρά όταν τους χορηγείται άγνωστη είσοδος, ειδικά είσοδος που περιέχει σφάλματα (όπως είναι πολύ συνηθισμένο σε δεδομένα πραγματικού κόσμου). Παράγουν περισσότερο αξιόπιστα αποτελέσματα όταν ενσωματώνονται σε ένα μεγαλύτερο σύστημα το οποίο περιλαμβάνει πολλαπλές δευτερεύουσες εργασίες.

Πολλές από τις σημαντικές πρώτες επιτυχίες που σημειώθηκαν στον τομέα της αυτόματης μετάφρασης οφείλονται κυρίως στην εργασία της IBM Research. Εκεί, αναπτύχθηκαν διαδοχικά πιο περίπλοκα στατιστικά μοντέλα[26]. Τα συστήματα αυτά ήταν σε θέση να επωφεληθούν από τα υπάρχοντα πολύγλωσσα σώματα κειμένων, που είχαν παραχθεί από το Κοινοβούλιο του Καναδά και της Ευρωπαϊκής Ένωσης, ως αποτέλεσμα νόμων, ζητώντας τη μετάφραση όλων των κυβερνητικών διαδικασιών σε όλες τις επίσημες γλώσσες των αντίστοιχων συστημάτων διακυβέρνησης. Ωστόσο, τα περισσότερα άλλα συστήματα εξαρτιόνταν από σώματα που είχαν αναπτυχθεί ειδικά για τις εργασίες αυτές, κάτι το οποίο ήταν (και συχνά εξακολουθεί να είναι) ένας σημαντικός περιορισμός για την επιτυχία των συστημάτων αυτών. Ως αποτέλεσμα, ένα μεγάλο μέρος της έρευνας έχει πάει σε μεθόδους πιο αποτελεσματικής μάθησης από περιορισμένες ποσότητες δεδομένων.

Πρόσφατες έρευνες επικεντρώθηκαν περισσότερο στους αλγορίθμους μάθησης χωρίς επίβλεψη, όπως και στους ημι-εποπτευόμενους αλγορίθμους. Τέτοιοι αλγόριθμοι μπορούν να μάθουν από δεδομένα που δεν είναι στοιχισμένα με τις επιθυμητές απαντήσεις ή χρησιμοποιούν ένα συνδυασμό από σχολιασμένα και μη-σχολιασμένα δεδομένα. Σε γενικές γραμμές, το έργο αυτό είναι πολύ πιο δύσκολο από την εποπτευόμενη μάθηση και συνήθως παράγει λιγότερο ακριβή αποτελέσματα για μια συγκεκριμένη ποσότητα δεδομένων εισόδου. Ωστόσο, υπάρχει ένας τεράστιος όγκος διαθέσιμων μη σχολιασμένων δεδομένων (μεταξύ άλλων, το σύνολο των

περιεχομένων του παγκόσμιου ιστού), ο οποίος μπορεί συχνά να αντισταθμίσει τα κατώτερα αποτελέσματα.

Τα τελευταία χρόνια, υπήρξε μια αναταραχή, καθώς αποτελέσματα έδειξαν ότι οι τεχνικές βαθιάς μάθησης (Deep Learning Techniques)[27][28] κατάφεραν να επιτύχουν άριστα αποτελέσματα σε πολλές εργασίες φυσικής γλώσσας. Τέτοια παραδείγματα υπάρχουν στη μοντελοποίηση γλώσσας (Language modeling)[29], στην ανάλυση γλώσσας (Language Parsing)[30][31], και σε πολλούς άλλους τομείς.

2. Επεξεργασία Φυσικής Γλώσσας

2.1 Τι είναι η Επεξεργασία Φυσικής Γλώσσας

Η επεξεργασία Φυσικής Γλώσσας (NLP)[32][33][34][35] είναι για τους υπολογιστές ένας τρόπος ανάλυσης, κατανόησης και άντλησης πληροφορίας από την ανθρώπινη γλώσσα με ένα έξυπνο και χρήσιμο τρόπο. Με τη χρήση της, οι προγραμματιστές μπορούν να οργανώνουν και να δομούν την υπάρχουσα γνώση για την εκτέλεση διαφόρων καθηκόντων.

Εκτός από τις κοινές λειτουργίες επεξεργασίας κειμένου, που αντιμετωπίζουν το κείμενο σαν μια απλή ακολουθία συμβόλων, η Επεξεργασία Φυσικού Κειμένου ιεραρχεί τη δομή της γλώσσας: πολλά λόγια κάνουν μια φράση, αρκετές φράσεις κάνουν μια πρόταση και, εν τέλει, οι προτάσεις μεταφέρουν ιδέες. Ο John Rehling, εμπειρογνώμονας του NLP και μέλος της ομάδας Meltwater Group, δήλωσε σε ομιλία με θέμα “Πώς η Επεξεργασία Φυσικής Γλώσσας βοηθά την αποκάλυψη συναισθημάτων από τα Social Media”[36][37][38]. *“Με την ανάλυση γλώσσας για το νόημά της, τα συστήματα Επεξεργασίας Φυσικής Γλώσσας έχουν εδώ και καιρό καλύπτει χρήσιμους ρόλους, όπως τη διόρθωση της γραμματικής, τη μετατροπή ομιλίας σε κείμενο και την αυτόματη μετάφραση μεταξύ γλωσσών.”*

Η Επεξεργασία Φυσικής Γλώσσας χρησιμοποιείται για την ανάλυση κειμένου και επιτρέπει τις μηχανές να κατανοήσουν πώς μιλούν οι άνθρωποι. Αυτή η αλληλεπίδραση ανθρώπου-υπολογιστή επιτρέπει τη δημιουργία εφαρμογών πραγματικού κόσμου, όπως η αυτόματη περίληψη, η ανάλυση συναισθήματος, η εξόρυξη θέματος, η αναγνώριση λέξεων-μέρους του λόγου, η εξόρυξη σχέσεων λέξεων, και πολλά άλλα. Η Επεξεργασία Φυσικής Γλώσσας χρησιμοποιείται συνήθως για την εξόρυξη κειμένου, την αυτόματη μετάφραση, και την αυτοματοποιημένη απάντηση ερώτησης.

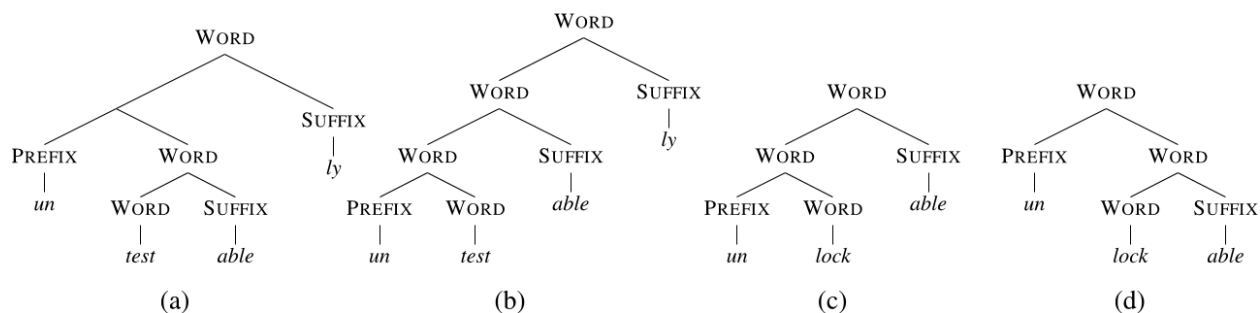
Πλέον χαρακτηρίζεται ως ένα πρόβλημα στην επιστήμη των υπολογιστών. Η ανθρώπινη γλώσσα μπορεί να είναι ιδιαίτερα ακριβής ή απλή στην ομιλία. Για να κατανοήσουμε την ανθρώπινη γλώσσα πρέπει να κατανοήσουμε όχι μόνο τα λόγια, αλλά και τις έννοιες, καθώς και το πώς και αν συνδέονται μεταξύ τους για να δημιουργήσουν νόημα. Παρότι η εκμάθηση της γλώσσας είναι ένα από τα ευκολότερα πράγματα για τους ανθρώπους, η ασάφεια της είναι αυτό που κάνει την Επεξεργασία Φυσικής Γλώσσας ένα δύσκολο εγχείρημα για τους υπολογιστές.

2.2 Κατηγορίες Επεξεργασίας Φυσικής Γλώσσας

Παρακάτω παρατίθεται μια λίστα με τις πιο συχνές χρήσεις της Επεξεργασίας Φυσικής Γλώσσας. Σημειώστε ότι κάποια από αυτά τα καθήκοντα έχουν άμεσες εφαρμογές στον πραγματικό κόσμο, ενώ άλλα χρησιμεύουν πιο συχνά ως δευτερεύουσες εργασίες, που έχουν χρησιμοποιηθεί για να βοηθήσουν στην επίλυση μεγαλύτερων προβλημάτων. Αν και όλες οι χρήσεις της Επεξεργασίας Φυσικής Γλώσσας είναι προφανώς πολύ στενά δεμένες μεταξύ τους, συχνά για λόγους ευκολίας, υποδιαιρούνται σε κατηγορίες.

2.2.1 Λημματοποίηση, και Μορφολογική Τμηματοποίηση

Η Λημματοποίηση[39] και Μορφολογική Τμηματοποίηση[40] είναι κύριος κλάδος του συντακτικού. Ορίζεται ως ο διαχωρισμός λέξεων σε επιμέρους μορφήματα (morphemes) και ως ο προσδιορισμός της κατηγορίας των μορφημάτων. Η δυσκολία αυτής της εργασίας εξαρτάται σε μεγάλο βαθμό από την πολυπλοκότητα της μορφολογίας (δηλαδή, της δομής των λέξεων) της γλώσσας που επεξεργάζεται. Η αγγλική γλώσσα έχει αρκετά απλή μορφολογία, και έτσι είναι συχνά δυνατό να αγνοηθεί αυτή η διαδικασία εντελώς, και απλά να μοντελοποιηθούν όλες οι πιθανές μορφές μιας λέξης (π.χ. “ανοιχτό, ανοίγει, άνοιγμα, κλπ”) ως ξεχωριστές λέξεις. Σε γλώσσες όπως η τουρκική[41] ή η ελληνική, μια τέτοια προσέγγιση δεν είναι δυνατή, δεδομένου ότι κάθε καταχώρηση λεξικού έχει χιλιάδες πιθανές μορφές λέξης.



Εικόνα 1: Προσδιορισμός κατηγορίας μορφημάτων

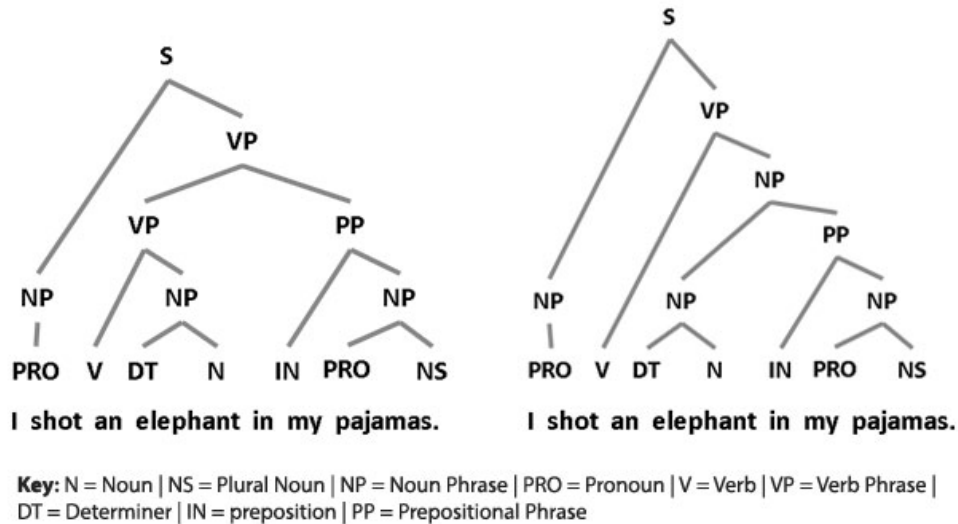
2.2.2 Αναγνώριση Λέξης ως Μέρος του Λόγου

Η Αναγνώριση Λέξης ως Μέρος του Λόγου είναι ένας από τους κλάδους του συντακτικού. Λαμβάνοντας υπόψη μια ολόκληρη πρόταση, καθορίζεται τι μέρος του λόγου είναι η κάθε λέξη[42][43]. Αρκετές λέξεις, ειδικά οι πιο κοινές, μπορεί να χρησιμεύουν ως πολλαπλά μέρη του λόγου. Για παράδειγμα στα Ελληνικά, το ‘τραπέζι’ μπορεί να είναι ένα ουσιαστικό (“το τραπέζι στην κουζίνα”) ή ρήμα (“σήμερα τραπεζώνω την οικογένεια”). Μερικές γλώσσες έχουν περισσότερη ασάφεια από άλλες. Γλώσσες με μικρή κλιτική μορφολογία, όπως η αγγλική δεν είναι ιδιαίτερα επιρρεπείς σε μια τέτοια ασάφεια. Η κινέζικη ωστόσο είναι αρκετά επιρρεπής, επειδή είναι μια τονική γλώσσα κατά τη διάρκεια της ομιλίας. Αυτό το είδος γλώσσας δεν μεταφέρεται εύκολα σε οντότητες που χρησιμοποιούνται κατά την επεξεργασία για να μεταδοθεί το επιδιωκόμενο νόημα.

2.2.3 Ανάλυση Κειμένου

Ως Ανάλυση Κειμένου[44] ορίζεται ο καθορισμός του γραμματικού δένδρου μιας συγκεκριμένης πρότασης. Η γραμματική για φυσικές γλώσσες είναι ασαφής και τυπικά αρκετές προτάσεις μπορούν να έχουν πολλαπλές πιθανές αναλύσεις. Στην πραγματικότητα, ίσως αναμενόμενο, για μια τυπική φράση είναι το γεγονός ότι μπορεί να υπάρχουν χιλιάδες δυνητικές αναλύσεις (οι περισσότερες από τις οποίες μπορεί να φαίνονται εντελώς παράλογες σε έναν άνθρωπο).

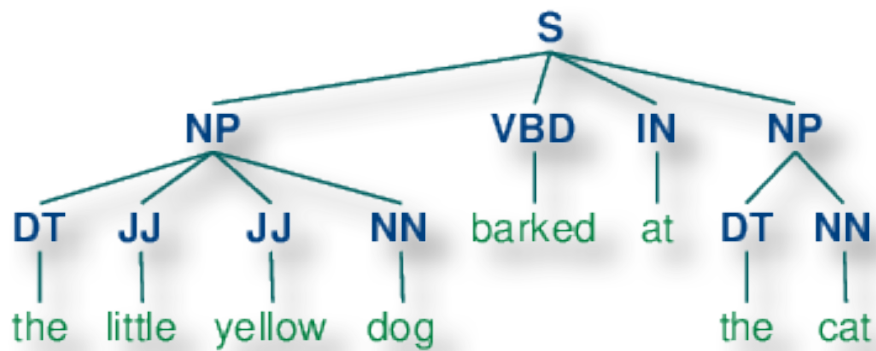
How Parse Trees Work



Εικόνα 2 : Καθορισμός γραμματικού δέντρου

2.2.4 Διαχωρισμός Προτάσεων

Ο Διαχωρισμός Προτάσεων[45], γνωστός και ως ανίχνευση ορίων προτάσεων, είναι ακόμα ένα από τα προβλήματα στο οποίο η επεξεργασία φυσικής γλώσσας μπορεί να δώσει λύση. Η βασική λειτουργία είναι η διάκριση της έναρξης και λήξης μιας πρότασης. Τα εργαλεία Επεξεργασίας Φυσικής Γλώσσας απαιτούν τη συμβολή της κατά την διαίρεση των προτάσεων. Αυτό συμβαίνει για τους παρακάτω λόγους. Η αναγνώριση ορίων φράσης είναι επίπονη διαδικασία, επειδή τα σημεία στίξης είναι συχνά ασαφή. Για παράδειγμα, μια περίοδος μπορεί να υποδηλώνεται από μια συντομογραφία, υποδιαστολή, αποσιωπητικά, ή μια διεύθυνση ηλεκτρονικού ταχυδρομείου. Άρα βλέπουμε ότι μια τελεία δεν δηλώνει πάντα το τέλος μιας πρότασης. Περίπου το 47% των περιόδων στα άρθρα της Wall Street Journal είναι συντομογραφίες. Επίσης, ερωτηματικά και θαυμαστικά μπορούν να εμφανιστούν σε ενσωματωμένα αποσπάσματα, όπως emoticons, κώδικα υπολογιστή, και την καθημερινή αργκό.



Εικόνα 3 : Παράδειγμα αναγνώρισης λέξεων ως μέρος του λόγου και καθορισμός πρότασης(S)

2.2.5 Απόσπαση Ρίζας Λέξεων

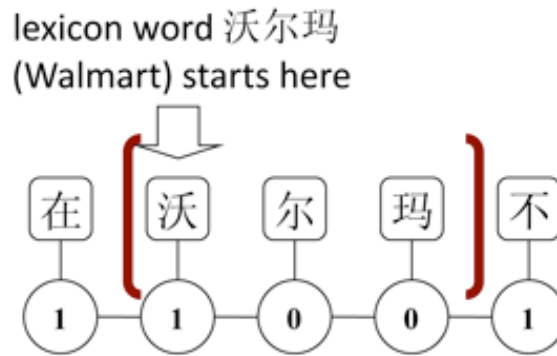
Στη γλωσσολογική μορφολογία και την ανάκτηση πληροφοριών, η απόσπαση ρίζας μιας λέξης[46] είναι η διαδικασία της μείωσης των κλητών (ή μερικές φορές προερχόμενων από άλλες λέξεις) λέξεων, στην βάση τους ή στην ριζική τους μορφή. Το στέλεχος δεν χρειάζεται να είναι ταυτόσημο με την μορφολογική ρίζα της λέξης. Συνήθως αρκεί οι λέξεις να χαρτογραφούνται στο ίδιο στέλεχος, έστω και αν αυτό στέλεχος δεν είναι από μόνο του μια έγκυρη ρίζα. Οι αλγόριθμοι για Απόσπασης Ρίζας Λέξεων έχουν μελετηθεί στην επιστήμη των υπολογιστών από το 1960.

(F)	Rule	Example
	SSES → SS	caresses → caress
	IES → I	ponies → poni
	SS → SS	caress → caress
	S →	cats → cat

Εικόνα 4 : Παράδειγμα κανόνων και απόσπασης ρίζας λέξεων

2.2.6 Κατάτμηση Λέξεων

Η Κατάτμηση Λέξεων[47] ορίζεται ως ο διαχωρισμός ενός μεγάλου κομματιού συνεχούς κειμένου σε ξεχωριστές λέξεις. Για μια γλώσσα, όπως τα Αγγλικά ή τα Ελληνικά, αυτό είναι αρκετά ασήμαντο, δεδομένου ότι οι λέξεις συνήθως χωρίζονται με κενά. Ωστόσο, ορισμένες γλώσσες, όπως τα Κινέζικα, τα Ιαπωνικά ή τα Ταϊλανδέζικα δε σηματοδοτούν τα όρια λέξεων με τέτοιο τρόπο. Σε αυτά η τμηματοποίηση λέξεων κειμένου είναι ένα δύσκολο έργο το οποίο απαιτεί τη γνώση του λεξιλογίου και της μορφολογίας των λέξεων της γλώσσας.

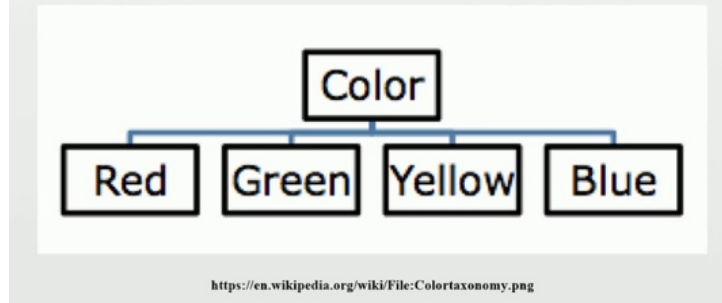


Εικόνα 5 : Παράδειγμα αναγνώρισης και διαχωρισμού λέξεων από κομμάτι κειμένου

2.2.7 Λεξιλογική Σημασιολογία

Λεξική Σημασιολογία (επίσης γνωστή και ως lexicosemantics[48]), είναι ένα υποπεδίο της γλωσσικής σημασιολογίας. Οι μονάδες ανάλυσης στη λεξική σημασιολογία είναι λεκτικές μονάδες, οι οποίες περιλαμβάνουν, όχι μόνο λόγια, αλλά και υπο-λέξεις ή υπο-μονάδες, όπως προσφύματα ή σύνθετες λέξεις και φράσεις. Λεκτικές μονάδες συνθέτουν τον κατάλογο των λέξεων σε μια γλώσσα, το λεξιλόγιο. Η λεξιλογική σημασιολογία εξετάζει το πώς η έννοια των λεκτικών μονάδων συσχετίζεται με τη δομή της γλώσσας ή τη σύνταξη. Αυτό αναφέρεται ως συντακτική-σημασιολογική διεπαφή.

Lexical semantics



Εικόνα 6 : Παράδειγμα αναγνώρισης υπο-μονάδας λέξεων που δηλώνουν χρώμα

2.2.8 Μηχανική Μετάφραση

Η Μηχανική Μετάφραση[49] ορίζεται ως η αυτόματη μετάφραση κειμένου από μια ανθρώπινη γλώσσα στην άλλη. Αυτό είναι ένα από τα πιο δύσκολα προβλήματα τεχνητής νοημοσύνης, και είναι μέλος μιας κατηγορίας προβλημάτων που ονομάζονται "AI-Complete", δηλαδή απαιτούν όλα τα διαφορετικά είδη της γνώσης που κατέχει ο άνθρωπος (γραμματική, σημασιολογία, στοιχεία σχετικά με τον πραγματικό κόσμο, κ.λπ.), προκειμένου να λυθούν σωστά.

Input sentence:	Translation (PBMT):	Translation (GNMT):	Translation (human):
李克強此行將啟動中加總理年度對話機制，與加拿大總理杜魯多舉行兩國總理首次年度對話。	Li Keqiang premier added this line to start the annual dialogue mechanism with the Canadian Prime Minister Trudeau two prime ministers held its first annual session.	Li Keqiang will start the annual dialogue mechanism with Prime Minister Trudeau of Canada and hold the first annual dialogue between the two premiers.	Li Keqiang will initiate the annual dialogue mechanism between premiers of China and Canada during this visit, and hold the first annual dialogue with Premier Trudeau of Canada.

Εικόνα 7 : Βήματα μηχανικής μετάφρασης κειμένου

2.2.9 Αναγνώριση Οντοτήτων

Λαμβάνοντας υπόψη ένα κείμενο, η Αναγνώριση Οντοτήτων[50] καθορίζει τα βασικά στοιχεία του κειμένου, όπως κύρια ονόματα, άτομα ή μέρη, και το είδος τους (π.χ. πρόσωπο, θέση, οργάνωση). Σημειώστε ότι, αν και η κεφαλαιοποίηση μπορεί να βοηθήσει στην αναγνώριση των οντοτήτων, σε γλώσσες όπως τα αγγλικά τα στοιχεία αυτά δε μπορούν να βοηθήσουν στον προσδιορισμό του είδους των οντοτήτων, και σε μερικές περιπτώσεις είναι συχνά ανακριβή ή ανεπαρκή. Για παράδειγμα, η πρώτη λέξη μιας πρότασης γράφεται με κεφαλαία γράμματα, όπως και τα ονόματα-οντότητες, ωστόσο συχνά τα τελευταία απλώνονται σε περισσότερες από μία λέξεις και μόνο μερικά γράφονται με κεφαλαία. Επιπλέον, πολλές άλλες μη-δυτικές γλώσσες (π.χ. Κινέζικα ή Αραβικά) δεν έχουν κανένα κεφαλαίο γράμμα σε ολόκληρες προτάσεις. Ακόμα και γλώσσες με κεφαλαιοποίηση δε μπορούν να τη χρησιμοποιήσουν με συνέπεια για τη διάκριση των ονομάτων. Για παράδειγμα, η γερμανική γλώσσα χρησιμοποιεί κεφαλαία σε όλα τα ουσιαστικά, ανεξάρτητα από το αν αναφέρονται σε ονόματα. Σε αντίθεση τα Γαλλικά και τα Ισπανικά δε χρησιμοποιούν κεφαλαία στις λέξεις που χρησιμεύουν ως επίθετα.

In 1917, Einstein applied the general theory of relativity to model the large-scale structure of the universe. He was visiting the United States when Adolf Hitler came to power in 1933 and did not go back to Germany, where he had been a professor at the Berlin Academy of Sciences. He settled in the U.S., becoming an American citizen in 1940. On the eve of World War II, he endorsed a letter to President Franklin D. Roosevelt alerting him to the potential development of "extremely powerful bombs of a new type" and recommending that the U.S. begin similar research. This eventually led to what would become the Manhattan Project. Einstein supported defending the Allied forces, but largely denounced using the new discovery of nuclear fission as a weapon. Later, with the British philosopher Bertrand Russell, Einstein signed the Russell-Einstein Manifesto, which highlighted the danger of nuclear weapons. Einstein was affiliated with the Institute for Advanced Study in Princeton, New Jersey, until his death in 1955.

Tag colours:

LOCATION TIME PERSON ORGANIZATION MONEY PERCENT DATE

Εικόνα 8 : Αναγνώριση οντοτήτων όπως υποθεσία, χρόνος, άτομο, οργανισμός κ.α.

2.2.10 Παραγωγή Φυσικής Γλώσσας

Παραγωγή Φυσικής Γλώσσας (Natural Language Generation, συντομογραφία NLG[51]) είναι ο τομέας επεξεργασίας φυσικής γλώσσας ο οποίος ασχολείται με τη δημιουργία φυσικής γλώσσας από ένα σύστημα αναπαράστασης μηχανής όπως π.χ. μια βάση δεδομένων ή μία λογική μορφή (logical form). Οι γλωσσολόγοι προτιμούν τον όρο «Παραγωγή Γλώσσας».

Θα μπορούσε να πει κανείς ότι ένα σύστημα παραγωγής φυσικής γλώσσας είναι σαν ένας μετατροπέας που μετατρέπει τα δεδομένα σε μια φυσική αναπαράσταση γλώσσας. Ωστόσο, οι μέθοδοι για την παραγωγή του τελικού κειμένου είναι διαφορετικές από εκείνες ενός μεταγλωττιστή, λόγω της εγγενούς εκφραστικότητας των φυσικών γλωσσών.

Ένα σύστημα παραγωγής φυσικής γλώσσας μπορεί να θεωρηθεί ως το αντίθετο του συστήματος κατανόησης φυσικής γλώσσας. Τα συστήματα κατανόησης φυσικής γλώσσας αποσαφηνίζουν το κείμενο εισόδου για την παραγωγή δεδομένων σε μορφή που είναι κατανοητή από έναν υπολογιστή. Αντιθέτως, το σύστημα παραγωγής φυσικής γλώσσας πρέπει να λάβει αποφάσεις σχετικά με το πώς να μετατρέψει μια έννοια σε λέξεις.

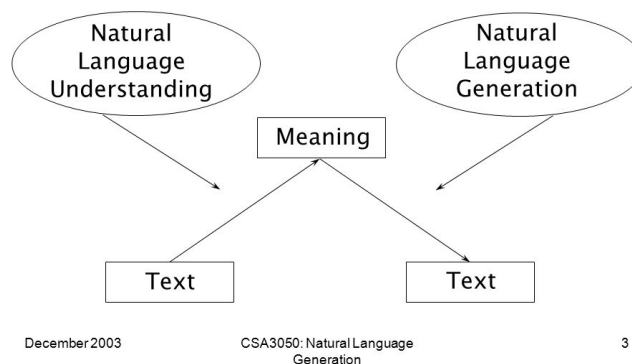
Απλά παραδείγματα τέτοιων συστημάτων είναι αυτά που δημιουργούν μορφές επιστολών. Αυτά δεν περιλαμβάνουν τυπικά τους κανόνες γραμματικής, αλλά μπορούν να δημιουργήσουν ένα γράμμα σε έναν καταναλωτή, π.χ. δηλώνοντας ότι ξεπεράστηκε ένα όριο δαπανών πιστωτικής κάρτας. Απλά συστήματα χρησιμοποιούν ένα πρότυπο κειμένου που δεν διαφέρει από τα πρότυπα των πιο γνωστών κειμενογράφων. Πιο πολύπλοκα συστήματα δημιουργίας φυσικής γλώσσας δημιουργούν δυναμικά το κείμενο. Αυτό μπορεί να γίνει είτε με ρητά μοντέλα της γλώσσας (π.χ., γραμματικά και συντακτικά μοντέλα), ή με τη χρήση στατιστικών μοντέλων που προέρχονται από την ανάλυση γραπτών κειμένων.

2.2.11 Κατανόηση Φυσικής Γλώσσας

Κατανόηση φυσικής γλώσσας[52] είναι η μετατροπή κομματιών φυσικού κειμένου σε παραστάσεις, οι οποίες είναι κατανοητές από τον υπολογιστή, όπως λογικές δομές πρώτης

τάξης, που είναι πιο εύκολο για τα προγράμματα ηλεκτρονικών υπολογιστών να υποστούν επεξεργασία. Η κατανόηση φυσικής γλώσσας προϋποθέτει την αναγνώριση της προβλεπόμενης σημασιολογίας από τις πολλαπλές πιθανές σημασιολογίες οι οποίες μπορεί να προέρχονται από μια έκφραση της φυσικής γλώσσας. Αυτό συνήθως παίρνει τη μορφή οργανωμένων συμβολισμών των φυσικών εννοιών.

$$\text{NLP} = \text{NLU} + \text{NLG}$$



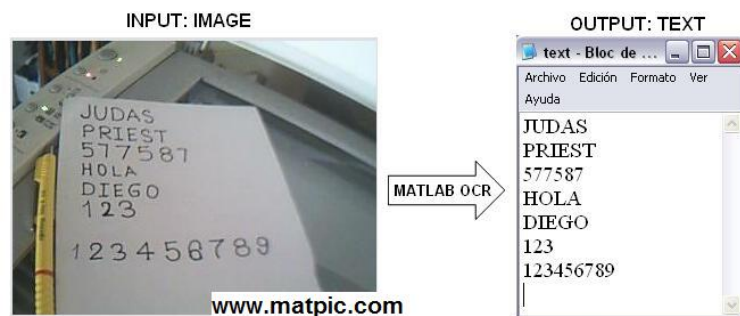
Εικόνα 9 : Σχηματική διαφορά μεταξύ παραγωγής και κατανόησης φυσικής γλώσσας

2.2.12 Οπτική Αναγνώριση Χαρακτήρων

Η Οπτική Αναγνώριση Χαρακτήρων (αλλιώς ανάγνωση χαρακτήρων ή OCR[53]) είναι η μηχανική ή ηλεκτρονική μετατροπή εικόνων δακτυλογραφημένου, χειρόγραφου, τυπωμένου ή οποιαδήποτε άλλης μορφής κειμένου σε κείμενο αναγνωρίσιμο από μία μηχανή. Χρησιμοποιείται ευρέως ως μορφή εισαγωγής πληροφοριών από αρχεία έντυπου χαρτιού, όπως έγγραφα διαβατηρίου, τιμολόγια, τραπεζικές δηλώσεις, ηλεκτρονικές αποδείξεις, επαγγελματικές κάρτες, αλληλογραφία, εκτυπώσεις στατικών δεδομένων ή οποιαδήποτε άλλο έγγραφο. Πρόκειται για μια κοινή μέθοδο ψηφιοποίησης των τυπωμένων κειμένων, ώστε να μπορούν να επεξεργάζονται ηλεκτρονικά, να αναζητούνται, να αποθηκεύονται πιο συμπαγώς, να εμφανίζονται ηλεκτρονικά και να χρησιμοποιούνται σε διαδικασίες, όπως η γνωσιακή υπολογιστική (cognitive computing), η μηχανική μετάφραση (machine translation), η εξαγωγή κειμένου σε ομιλία (text-to-speech) και η εξόρυξη κειμένου (text mining). Η Οπτική

Αναγνώριση Κειμένου είναι πεδίο έρευνας για την αναγνώριση προτύπων και την τεχνητή νοημοσύνη.

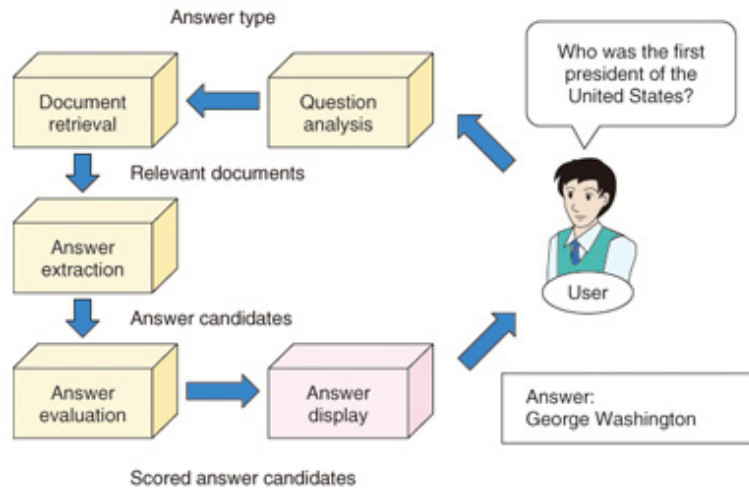
Οι πρώτες εκδόσεις έπρεπε να εκπαιδευτούν με εικόνες από κάθε χαρακτήρα και να δουλεύουν σε μία γραμματοσειρά τη φορά. Τα προηγμένα συστήματα μπορούν να παράγουν υψηλό βαθμό ακρίβειας αναγνώρισης για τις περισσότερες γραμματοσειρές και υποστηρίζουν διάφορες μορφές αρχείων ψηφιακών εικόνων ως είσοδο. Ορισμένα συστήματα είναι σε θέση να αναπαράγουν μορφοποιημένο κείμενο, που προσεγγίζει πολύ την αρχική σελίδα, συμπεριλαμβανομένων εικόνων, στηλών και άλλων μη κειμενικών στοιχείων.



Εικόνα 10 : Οπτική αναγνώριση χαρακτήρων μέσω βιβλιοθήκης Matlab

2.2.13 Απάντηση Ερωτήσεων

Η Απάντηση Ερωτήσεων[54] είναι η διαδικασία κατά την οποία λαμβάνοντας ένα ερώτημα σε φυσική γλώσσα, καθορίζεται η απάντησή του. Οι συνήθεις ερωτήσεις έχουν μια συγκεκριμένη σωστή απάντηση (όπως "Ποια είναι η πρωτεύουσα της Ελλάδος;"), αλλά μερικές φορές εξετάζονται και ανοικτά ερωτήματα (όπως "Ποια είναι η έννοια της ζωής;"). Πρόσφατες εκδόσεις τέτοιων μηχανών εξετάζουν ακόμη πιο πολύπλοκα ερωτήματα.



Εικόνα 11: Σχηματική αναπαράσταση διαδικασίας αυτόματης απάντησης ερωτήσεων

2.2.14 Κειμενική Συνεπαγωγή

Η Συνεπαγωγή Κειμένου στην επεξεργασία της φυσικής γλώσσας[55] είναι μια κατευθυντική σχέση μεταξύ των τμημάτων κειμένου. Η σχέση ισχύει κάθε φορά που ένα τμήμα κειμένου προκύπτει από ένα άλλο τμήμα του. Στο πλαίσιο της ΤΕ, τα συνεπαγόμενα κείμενα ονομάζονται κείμενο (t) και υπόθεση (h). Η συνεπαγωγή του κειμένου δεν είναι η ίδια με την καθαρή λογική συνέπεια - έχει έναν πιο χαλαρό ορισμό: "t συνεπάγεται h" ($t \Rightarrow h$) σημαίνει συνήθως ότι σε μια ανθρώπινη ανάγνωση του t θα συμπεράνει ότι το h πιθανότατα είναι αλήθεια. Η σχέση είναι κατευθυντική επειδή ακόμα και αν "t συνεπάγεται h", το αντίστροφο "h συνεπάγεται t" είναι λιγότερο σίγουρο.

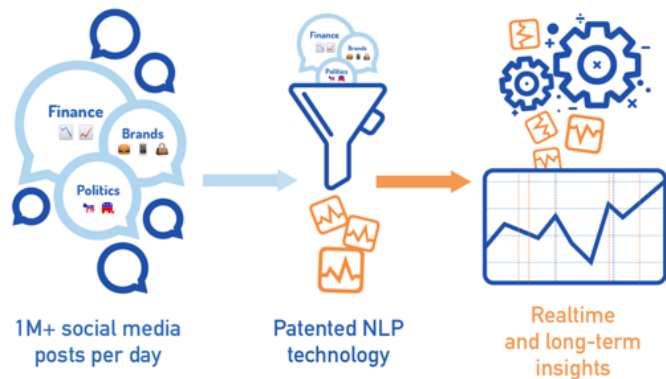
2.2.15 Αναγνώριση Σχέσεων

Η διαδικασία Εξαγωγής Σχέσης[56] απαιτεί την ανίχνευση και την ταξινόμηση των σημασιολογικών σχέσεων που αναφέρονται μέσα σε ένα σύνολο αντικειμένων, συνήθως από κείμενο ή έγγραφο τύπου XML. Το έργο είναι παρόμοιο με αυτό της εξόρυξης πληροφοριών, αν και η εξόρυξη πληροφορικών απαιτεί επιπλέον την αφαίρεση επαναλαμβανόμενων σχέσεων (αποσαφήνιση) και γενικά αναφέρεται στην εξαγωγή πολλών διαφορετικών σχέσεων.

2.2.16 Ανάλυση Συναισθήματος

Η Ανάλυση Συναισθήματος[57], μερικές φορές γνωστή ως «εξόρυξη γνώμης» (opinion mining) ή «τεχνική νοημοσύνη συναισθήματος» (emotion AI), αναφέρεται στη χρήση της επεξεργασίας φυσικής γλώσσας, της ανάλυσης κειμένου, της υπολογιστικής γλωσσολογίας και της βιομετρίας για τον συστηματικό εντοπισμό, την εξαγωγή, την ποσοτικοποίηση και τη μελέτη των συναισθημάτων και των καταστάσεων. Η ανάλυση του συναισθήματος εφαρμόζεται ευρέως στη φωνή των υλικών του πελάτη, όπως κριτικές και απαντήσεις ερευνών, σε κοινωνικά μέσα και υλικά υγειονομικής περίθαλψης. Έχει εφαρμογές που κυμαίνονται από το μάρκετινγκ μέχρι την εξυπηρέτηση πελατών στην κλινική ιατρική.

Σε γενικές γραμμές, η Ανάλυση Συναισθήματος έχει ως στόχο να καθορίσει τη στάση ενός ομιλητή, συγγραφέα ή άλλου ατόμου σε σχέση με κάποιο θέμα. Επίσης, καθορίζει τη συνολική συμφραζόμενη πολικότητα ή συναισθηματική αντίδραση σε ένα έγγραφο, ή γεγονός. Η στάση μπορεί να είναι μια κρίση ή αξιολόγηση.



Εικόνα 12 : Παράδειγμα χρήσης ανάλυσης συναισθήματος στα κοινωνικά δίκτυα

2.2.17 Αυτόματη Περίληψη Κειμένου

Η Αυτόματη Σύνοψη[58] είναι η διαδικασία μείωσης ενός εγγράφου κειμένου με πρόγραμμα υπολογιστή, προκειμένου να δημιουργηθεί μια περίληψη που διατηρεί τα πιο σημαντικά σημεία του πρωτότυπου εγγράφου. Οι τεχνολογίες που μπορούν να κάνουν μια συνεκτική περίληψη λαμβάνουν υπόψη μεταβλητές, όπως το μήκος, το στυλ γραφής και τη

σύνταξη. Η αυτόματη σύνοψη δεδομένων είναι μέρος της εκμάθησης μηχανών (machine learning) και της εξόρυξης δεδομένων (data mining). Η κύρια ιδέα της σύνοψης είναι να βρεθεί ένα αντιπροσωπευτικό υποσύνολο των δεδομένων, το οποίο περιέχει τις πληροφορίες ολόκληρου του συνόλου. Οι τεχνολογίες περίληψης χρησιμοποιούνται σε μεγάλο αριθμό τομέων στη βιομηχανία σήμερα. Ένα παράδειγμα της χρήσης της τεχνολογίας σύνοψης είναι μηχανές αναζήτησης, όπως το Google. Άλλα παραδείγματα περιλαμβάνουν συνοπτική παρουσίαση εγγράφων, σύνοψη συλλογής εικόνων και συνοπτική παρουσίαση βίντεο. Η περίληψη εγγράφων, προσπαθεί να δημιουργήσει αυτόματα μια αντιπροσωπευτική σύνοψη ή περίληψη ολόκληρου του εγγράφου, βρίσκοντας τις πιο ενημερωτικές προτάσεις. Ομοίως, στην περίληψη εικόνων το σύστημα βρίσκει τις πιο αντιπροσωπευτικές και σημαντικές εικόνες. Ομοίως, στα βίντεο των καταναλωτών γίνεται αφαίρεση των βαρετών ή επαναλαμβανόμενων σκηνών και εξαγάγεται μια πολύ πιο σύντομη και συνοπτική έκδοση του βίντεο. Αυτό είναι σημαντικό, στον έλεγχο μακροχρόνιων βίντεο, όπως τα βίντεο παρακολούθησης, όπου κάποιος μπορεί να θέλει να εξάγει μόνο σημαντικά γεγονότα στο εγγεγραμμένο βίντεο, αφού το μεγαλύτερο μέρος του βίντεο μπορεί να περιέχει άχρηστη πληροφορία. Καθώς αυξάνεται το πρόβλημα της υπερφόρτωσης πληροφοριών και καθώς αυξάνεται το μέγεθος των δεδομένων, αυξάνεται και το ενδιαφέρον για αυτόματη σύνοψη.

Γενικά, υπάρχουν δύο προσεγγίσεις για την αυτόματη σύνοψη. Η εξόρυξη και η αφαίρεση. Οι μέθοδοι εξόρυξης λειτουργούν επιλέγοντας ένα υποσύνολο υφιστάμενων λέξεων, φράσεων ή προτάσεων στο αρχικό κείμενο για να σχηματίσουν τη σύνοψη. Αντίθετα, οι αφαιρετικές μέθοδοι οικοδομούν μια εσωτερική σημασιολογική αναπαράσταση και στη συνέχεια χρησιμοποιούν τεχνικές φυσικής παραγωγής γλωσσών για να δημιουργήσουν μια περίληψη που είναι πιο κοντά σε αυτό που μπορεί να δημιουργήσει ένας άνθρωπος. Μια τέτοια περίληψη μπορεί να περιέχει λέξεις που δεν υπάρχουν ρητά στο πρωτότυπο. Η έρευνα σχετικά με τις αφαιρετικές μεθόδους είναι ένας όλο και πιο σημαντικός και ενεργός χώρος έρευνας, ωστόσο, λόγω περιορισμών πολυπλοκότητας, η έρευνα μέχρι σήμερα έχει επικεντρωθεί κυρίως σε μεθόδους εξόρυξης.

2.2.18 Ανάλυση Λόγου

Η Ανάλυση Λόγου ή αλλιώς «μελέτη λόγου» είναι ένας γενικός όρος για μια σειρά προσεγγίσεων για την ανάλυση της γραπτής, φωνητικής και νοηματικής χρήσης οποιουδήποτε σημαντικού σημειωτικού συμβάντος.

Τα αντικείμενα της ανάλυσης του λόγου (λόγος, γραφή, συζήτηση) ορίζονται με διαφορετικό τρόπο από την άποψη συνεκτικών ακολουθιών φράσεων, προτάσεων, ομιλίας και παραγράφων. Σε αντίθεση με την πλειονότητα των τομέων της παραδοσιακής γλωσσολογίας, οι αναλυτές του λόγου όχι μόνο μελετούν τη χρήση της γλώσσας «πέρα από το όριο των προτάσεων», αλλά προτιμούν να αναλύουν τη «γλωσσική χρήση». Η γλωσσολογία των κειμένων είναι ένας στενά συνδεδεμένος τομέας. Η ουσιαστική διαφορά μεταξύ της ανάλυσης του λόγου και της γλωσσολογίας κειμένων είναι ότι η ανάλυση του λόγου αποσκοπεί στην αποκάλυψη κοινωνικο-ψυχολογικών χαρακτηριστικών ενός προσώπου ή προσώπων και όχι σε μια δομή κειμένου.

Η ανάλυση λόγου υιοθετήθηκε σε ποικίλους τομείς των ανθρωπιστικών και κοινωνικών επιστημών, όπως η γλωσσολογία, η εκπαίδευση, η κοινωνιολογία, η ανθρωπολογία, η κοινωνική εργασία, η γνωστική ψυχολογία, η κοινωνική ψυχολογία, οι μελέτες περιοχής, οι πολιτιστικές σπουδές, οι διεθνείς σχέσεις, οι βιβλικές μελέτες και οι μελέτες μετάφρασης. Κάθε ένας από αυτούς τους τομείς υπόκειται στις δικές της υποθέσεις, τις διαστάσεις της ανάλυσης και τις μεθοδολογίες.

2.2.19 Αναγνώριση Ομιλίας

Η Αναγνώριση Ομιλίας (Speech Recognition[59]) είναι το διεπιστημονικό υποπεδίο της υπολογιστικής γλωσσολογίας που αναπτύσσει μεθοδολογίες και τεχνολογίες που επιτρέπουν την αναγνώριση και τη μετάφραση της ομιλούμενης γλώσσας σε κείμενο με τη χρήση υπολογιστών. Είναι επίσης γνωστή ως «αυτόματη αναγνώριση ομιλίας» (Automatic Speech Recognition) ή

απλά «ομιλία σε κείμενο» (Speech-to-text). Ενσωματώνει γνώσεις και έρευνες στους τομείς της γλωσσολογίας, της πληροφορικής και της ηλεκτρολογίας.

Ορισμένα συστήματα αναγνώρισης ομιλίας χρησιμοποιούν «εκπαίδευση δεδομένων» (data training), όπου ένας μεμονωμένος ομιλητής διαβάζει κείμενο ή απομονωμένο λεξιλόγιο στο σύστημα. Το σύστημα αναλύει τη συγκεκριμένη φωνή του ατόμου και τη χρησιμοποιεί για να τελειοποιήσει την αναγνώριση της ομιλίας, με αποτέλεσμα την αύξηση της ακρίβειας. Τα συστήματα που δεν χρησιμοποιούν εκπαίδευση ονομάζονται συστήματα «ανεξάρτητα από ομιλητή». Τα συστήματα που χρησιμοποιούν την εκπαίδευση ονομάζονται «εξαρτώμενα από ομιλητή».

Οι εφαρμογές αναγνώρισης ομιλίας περιλαμβάνουν διεπαφές φωνητικών φράσεων για ενέργειες, όπως η φωνητική κλήση (π.χ. "κλήση στο σπίτι"), η δρομολόγηση κλήσεων (π.χ. "Θα ήθελα να κάνω ομαδική κλήση"), ο έλεγχος οικιακών συσκευών (π.χ. "Άνοιξε το φως του σαλονιού"), η αναζήτηση (π.χ. βρείτε ένα podcast όπου χρησιμοποιεί συγκεκριμένες λέξεις), η απλή εισαγωγή δεδομένων (π.χ. εισαγωγή αριθμού πιστωτικής κάρτας), η προετοιμασία δομημένων εγγράφων (π.χ. έκθεση ακτινολογίας) και άλλες ενέργειες.

Ο όρος αναγνώριση ομιλητή αναφέρεται στην αναγνώριση του ομιλητή και όχι σε αυτό που λέει. Η αναγνώριση του ομιλητή μπορεί να απλοποιήσει το έργο της μετάφρασης της ομιλίας σε συστήματα που έχουν εκπαιδευτεί στη φωνή ενός συγκεκριμένου προσώπου ή μπορεί να χρησιμοποιηθεί για την επαλήθευση της ταυτότητας ενός ατόμου ως μέρος μιας διαδικασίας ασφαλείας.

Από άποψη τεχνολογίας, η αναγνώριση ομιλίας έχει μακρά ιστορία με πολλά κύματα σημαντικών καινοτομιών. Πιο πρόσφατα, ο τομέας έχει επωφεληθεί από τις προόδους στη βαθιά εκμάθηση (deep learning) και τα μεγάλα δεδομένα (big data). Η πρόοδος αποδεικνύεται όχι μόνο από την αύξηση των ακαδημαϊκών εγγράφων που δημοσιεύονται στον τομέα, αλλά κυρίως από την υιοθέτηση από την παγκόσμια βιομηχανία, μιας ποικιλίας μεθόδων βαθιάς μάθησης για το σχεδιασμό και την ανάπτυξη συστημάτων αναγνώρισης ομιλίας. Οι κύριες επιχειρήσεις στον κλάδο της αναγνώρισης ομιλίας περιλαμβάνουν τις Google, Microsoft, IBM, Baidu, Apple,

Amazon, Nuance, SoundHound, IflyTek, CDAC, πολλές από τα οποίες έχουν δημοσιοποιήσει την βασική τεχνολογία στα συστήματα αναγνώρισης ομιλίας βασισμένα στη βαθιά μάθηση.



Εικόνα 13 : Διάγραμμα διαδικασίας αναγνώρισης φωνής, της εταιρείας Google

2.2.20 Τμηματοποίηση Ομιλίας

Η Τμηματοποίηση της Ομιλίας ή αλλιώς «κατακερματισμός του λόγου»[60] είναι η διαδικασία ταυτοποίησης των ορίων μεταξύ λέξεων, συλλαβών ή φωνημάτων σε προφορικές φυσικές γλώσσες. Ο όρος ισχύει τόσο για τις διανοητικές διαδικασίες που χρησιμοποιούνται από τον άνθρωπο, όσο και για τις τεχνητές διαδικασίες επεξεργασίας φυσικής γλώσσας.

Η κατάτμηση του λόγου είναι ένα υποπεδίο της γενικής αντίληψης ομιλίας και ένα σημαντικό υποπρόβλημα του τεχνολογικά εστιασμένου πεδίου της αναγνώρισης ομιλίας, το οποίο δεν μπορεί να λυθεί επαρκώς μεμονωμένα. Όπως και στα περισσότερα προβλήματα φυσικής επεξεργασίας γλώσσας, πρέπει να ληφθεί υπόψη το πλαίσιο, η γραμματική, η σημασιολογία, και μάλιστα το αποτέλεσμα είναι συχνά μια πιθανοτική και όχι μια κατηγορηματική διαίρεση. Η συνωμοσία (coarticulation) - ένα φαινόμενο που μπορεί να συμβεί μεταξύ γειτονικών λέξεων εξίσου εύκολα όπως και μέσα σε μία μόνο λέξη - παρουσιάζει την κύρια πρόκληση στην κατάτμηση ομιλίας σε όλες τις γλώσσες.

Το πρόβλημα αυτό επικαλύπτεται σε κάποιο βαθμό από το πρόβλημα της τμηματοποίησης κειμένων που συμβαίνει σε ορισμένες γλώσσες στις οποίες η παραδοσιακή γραφή λέξεων γίνεται χωρίς διαλείμματα μεταξύ τους (όπως τα Κινέζικα και τα Ιαπωνέζικα). Ωστόσο, ακόμη και για αυτές τις γλώσσες, ο κατακερματισμός κειμένου είναι συχνά πολύ πιο εύκολος από την κατάτμηση ομιλίας, επειδή η γραπτή γλώσσα συνήθως έχει μικρή παρέμβαση μεταξύ παρακείμενων λέξεων και συχνά περιέχει επιπλέον ενδείξεις που δεν υπάρχουν στην ομιλία (όπως η χρήση κινεζικών χαρακτήρων).

2.2.21 Σύνθεση Ομιλίας

Η Σύνθεση του Λόγου[61] είναι η τεχνητή παραγωγή ανθρώπινης ομιλίας. Ένα σύστημα υπολογιστή που χρησιμοποιείται για το σκοπό αυτό ονομάζεται υπολογιστής ομιλίας ή συνθέτης ομιλίας και μπορεί να εφαρμοστεί σε προϊόντα λογισμικού ή και φυσικού υλικού.

Η σύνθετη ομιλία μπορεί να δημιουργηθεί συνδυάζοντας κομμάτια εγγραφής που είναι αποθηκευμένα σε μια βάση δεδομένων. Τα συστήματα διαφέρουν ως προς το μέγεθος των αποθηκευμένων μονάδων ομιλίας. Ένα σύστημα που αποθηκεύει συλλαβές και δίφθογγα παρέχει μεγαλύτερο εύρος λέξεων εξόδου, αλλά μπορεί να μην υπάρχει σαφήνεια. Για συγκεκριμένους τομείς χρήσης, η αποθήκευση ολόκληρων λέξεων ή προτάσεων επιτρέπει την παραγωγή υψηλής ποιότητας εξόδου. Εναλλακτικά, ένας συνθέτης μπορεί να ενσωματώσει ένα μοντέλο της φωνητικής οδού και άλλα χαρακτηριστικά ανθρώπινης φωνής για να δημιουργήσει μια εντελώς "σύνθετη" έξοδο φωνής.

Η ποιότητα ενός συνθέτη ομιλίας κρίνεται από την ομοιότητά του με την ανθρώπινη φωνή και από την ικανότητά της να γίνεται ξεκάθαρα κατανοητή. Ένα κατανοητό πρόγραμμα μετατροπής κειμένου σε ομιλία επιτρέπει σε άτομα με προβλήματα όρασης ή άτομα με αναπηρία να διαβάζουν γραπτές λέξεις σε έναν οικιακό υπολογιστή. Πολλά λειτουργικά συστήματα υπολογιστών έχουν συμπεριλάβει συνθέτες ομιλίας από τις αρχές της δεκαετίας του 1990.

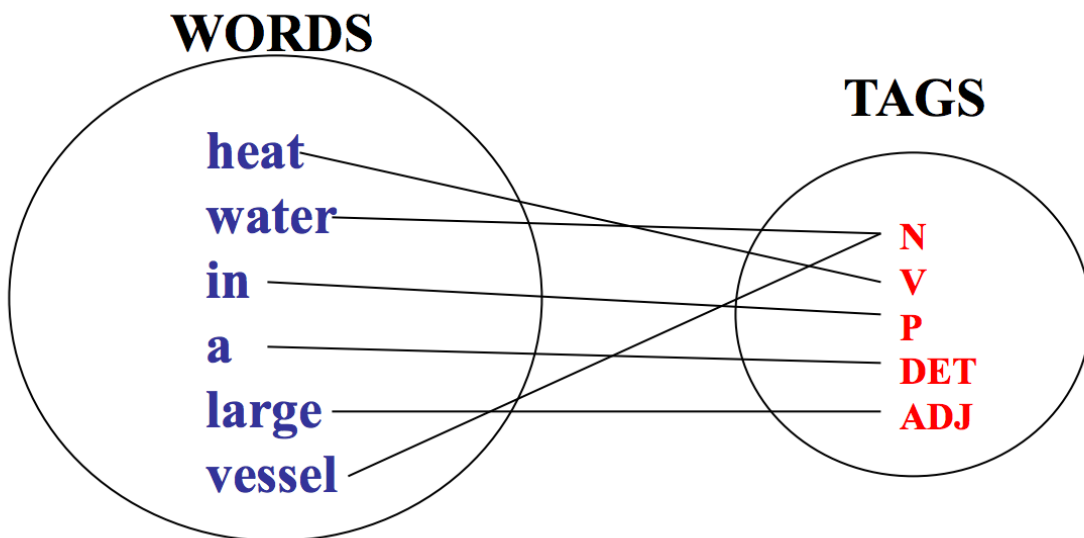
Ένα σύστημα μετατροπής κειμένου σε ομιλία αποτελείται από δύο μέρη. Το εμπρός και το πίσω μέρος. Το εμπρός μέρος (front-end) έχει δύο σημαντικά καθήκοντα. Κατ' αρχάς, μετατρέπει το ακατέργαστο κείμενο που περιέχει σύμβολα, όπως αριθμούς και συντομογραφίες, στο ισοδύναμο των γραπτών λέξεων. Αυτή η διαδικασία ονομάζεται συχνά εξομάλυνση κειμένου ή προεπεξεργασία. Στη συνέχεια, εκχωρεί φωνητικές μεταγραφές σε κάθε λέξη, διαιρεί και επισημαίνει το κείμενο σε βασικές

μονάδες, όπως φράσεις και προτάσεις. Η διαδικασία εκχώρησης των φωνητικών μεταγραφών σε λέξεις ονομάζεται μετατροπή κειμένου σε φωνή ή γραφή-προς-φωνή. Οι φωνητικές μεταγραφές και οι πληροφορίες για την φράση, μαζί αποτελούν τη συμβολική γλωσσική αναπαράσταση που εξάγεται από το front-end. Το πίσω μέρος (back-end) -που συχνά αναφέρεται ως συνθέτης-μετατρέπει έπειτα τη συμβολική γλωσσική αναπαράσταση στον ήχο. Σε ορισμένα συστήματα, αυτό το μέρος περιλαμβάνει τον υπολογισμό της χροιάς της φωνής, που στη συνέχεια επιβάλλεται στην ομιλία εξόδου.

2.3 Αναγνώριση Λέξεων ως Μέρους του Λόγου

Όπως μπορούμε να παρατηρήσουμε από το προηγούμενο κεφάλαιο, η χρήση της επεξεργασίας φυσικής γλώσσας μπορεί να χρησιμεύσει σε αρκετούς κλάδους του συντακτικού, της σημασιολογίας και της γλωσσολογίας. Ωστόσο, η συγκεκριμένη εργασία ειδικεύεται στον τομέα της αναγνώρισης λέξης ως μέρος του λόγου (Part-Of-Speech Tagging, συντομογραφία POS Tagging). Χρησιμοποιώντας τις δυνατότητες του εργαλείου Stanford Natural Language Processing Group, γίνεται χειρισμός κειμένου, χωρισμένο σε προτάσεις ή παραγράφους, ως προς το μέρος του λόγου των λέξεων. Έτσι δίνεται η δυνατότητα χρήσης του αποτελέσματος αυτής της επεξεργασίας για την κύρια λειτουργία της εργασίας.

Κατά την επεξεργασία του κειμένου, ανατίθεται ένα συγκεκριμένο μέρος του λόγου σε κάθε λέξη της πρότασης



Εικόνα 14: Γραφικό παράδειγμα αναγνώρισης και ανάθεσης λέξεων ως μέρη του λόγου

<u>Word</u>	<u>Tag</u>
heat	verb (noun)
water	noun (verb)
in	prep (noun, adv)
a	det (noun)
large	adj (noun)
vessel	noun

Εικόνα 15 : Απεικόνιση ανάθεσης λέξεων ως μέρη του λόγου

2.3.1 Που χρησιμεύει το Part-Of-Speech Tagging

Η διαδικασία αναγνώρισης λέξεων ως μέρη του λόγου είναι πολύ χρήσιμη στις εξής διεργασίες:

- Ανάκτηση δεδομένων από μεγάλο όγκο κειμένων
- Μετατροπή κειμένου σε ομιλία.
 - π.χ. Object (Noun) VS Object (Verb)
 - π.χ. Discount (Noun) VS Discount (Verb)
- Αποσαφηνισμός λέξεων και φράσεων
- Απαραίτητο προπαρασκευαστικό βήμα για την διαδικασία ανάλυσης κειμένου. Όταν κάθε λέξη μίας πρότασης είναι αποσαφηνισμένη ως προς τα μέρη του λόγου, χρειάζονται λιγότερες επαναλήψεις για την πλήρη ανάλυση της πρότασης.

2.3.2 Επιλογή λίστας ετικετών

Βασικό μέρος μιας πετυχημένης εφαρμογής που χρησιμοποιεί το συγκεκριμένο κομμάτι επεξεργασίας φυσικού κειμένου, είναι η επιλογή της λίστας των ετικετών που θα χρησιμοποιηθούν. Εφόσον επιλεγθούν οι ετικέτες για τα μέρη του λόγου που θέλουμε να χρησιμοποιήσουμε, τότε το περιβάλλον επεξεργασίας θα αναθέσει στις λέξεις του κειμένου μια από αυτές τις ετικέτες.

Η αλλαγή στο πλήθος των ετικετών, προφανώς και έχει μεγάλο αντίκτυπο στην πολυπλοκότητα του συστήματος και στον χρόνο περάτωσης της συγκεκριμένης διεργασίας. Μια επιλογή, η οποία αποδίδει σωστά αποτελέσματα με τεράστιο βαθμό γενίκευσης, είναι η επιλογή της εξής λίστας ετικετών:

TAG	DESCRIPTION
N	Noun
V	Verb
ADJ	Adjective
ADV	Adverb
PREP	Preposition

Πίνακας 1 : Βασική λίστα ετικετών προς χρησιμοποίηση κατά την διεργασία POS Tagging

Ωστόσο, ένα δείγμα ετικετών τόσο μικρό όσο αυτό, δεν μπορεί να αποδώσει με την κατάλληλη ακρίβεια τη συντακτική ανάλυση της κάθε λέξης. Για αυτόν τον λόγο, έχουν δημιουργηθεί πιο ανεπτυγμένα και πλήρη λεξικά με λίστες ετικετών. Με αυτόν τον τρόπο υπάρχει ένα διεθνές πλαίσιο το οποίο χρησιμοποιείται για την ίδια διεργασία και έχει ως αποτέλεσμα ένα βασικό σώμα ετικετών το οποίο μπορεί να δημιουργήσει σωστά και λεπτομερή αποτελέσματα. Το πιο γνωστό σώμα ετικετών, το οποίο αποτελείται από 36 ετικέτες, είναι το “UPenn TreeBank II”. Παρακάτω παρατίθενται οι ετικέτες με την ερμηνεία τους.

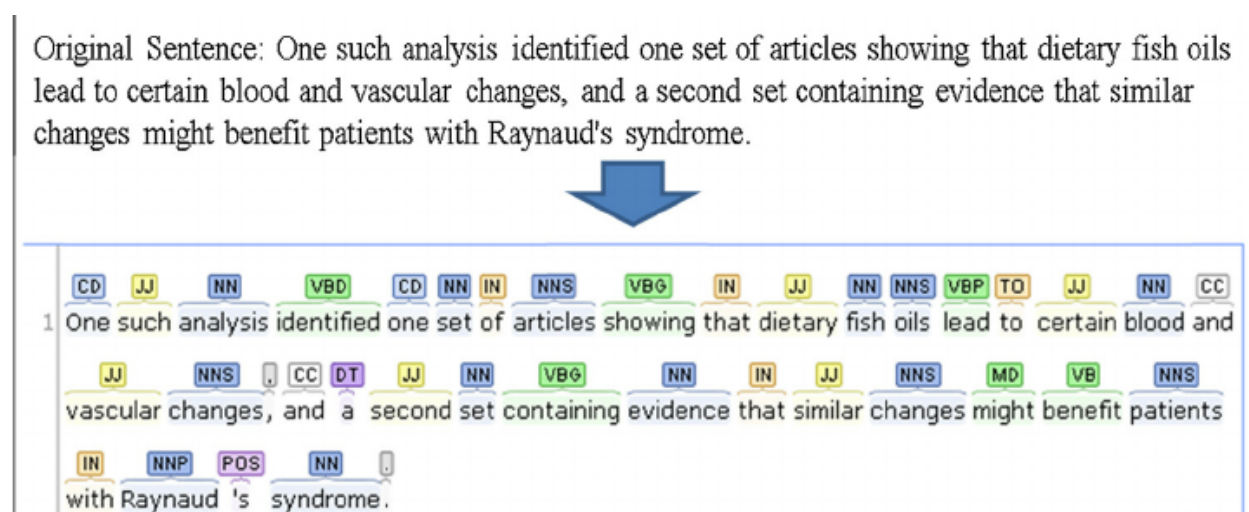
UPenn TreeBank II

TAG	DESCRIPTION
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun
PRP\$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	to
UH	Interjection

VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VCN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Wh-determiner
WP	Wh-pronoun
WP\$	Possessive wh-pronoun
WRB	Wh-adverb

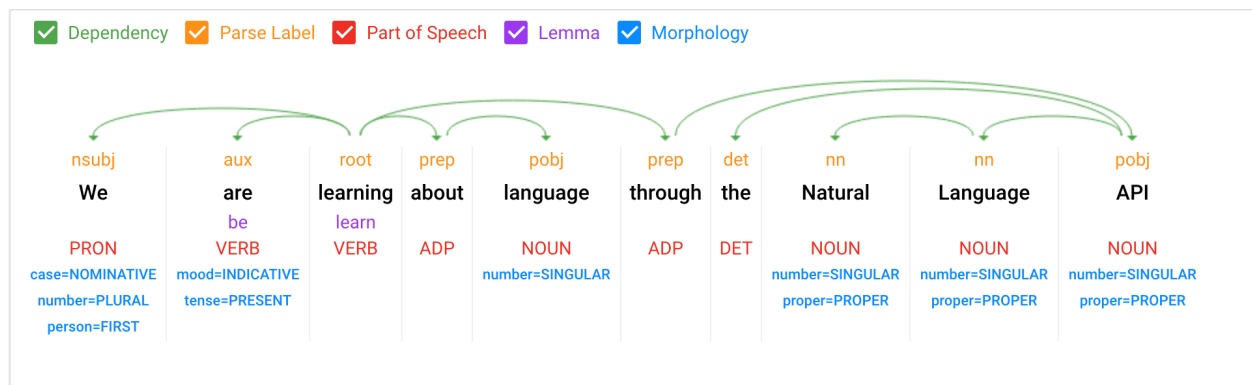
Πίνακας 2 : Λίστα ετικετών UPenn TreeBank II

Παρακάτω μπορούμε να δούμε μια ολοκληρωμένη πρόταση η οποία έχει αναλυθεί σε λέξεις με ετικέτες μορφής UPenn TreeBank II



Εικόνα 16 : Πλήρως αναλυμένη πρόταση σε ετικέτες UPenn TreeBank II

Στην παρακάτω φωτογραφία μπορούμε να παρατηρήσουμε ότι πέρα από την υπηρεσία part-of-speech tagging, η επεξεργασία φυσικής γλώσσας δίνει και άλλες εναλλακτικές επιλογές στην επεξεργασία κειμένου.



Εικόνα 17: Εναλλακτικές αναλύσεις πρότασης, πέρα από την υπηρεσία Part-Of-Speech

2.4 Stanford Natural Language Processing Library

2.4.1 Συνοπτική Εικόνα

Η Ομάδα επεξεργασίας φυσικής γλώσσας του Πανεπιστημίου του Stanford είναι μια ομάδα καθηγητών, μεταπτυχιακών φοιτητών, προγραμματιστών και προπτυχιακών φοιτητών, οι οποίοι εργάζονται πάνω σε αλγόριθμους που επιτρέπουν στους υπολογιστές να επεξεργάζονται και να κατανοούν τις ανθρώπινες γλώσσες. Το έργο τους κυμαίνεται από τη βασική έρευνα στην υπολογιστική γλωσσολογία έως τις βασικές εφαρμογές της τεχνολογίας στην ανθρώπινη γλώσσα και καλύπτει τομείς, όπως η κατανόηση των προτάσεων, η αυτόματη απάντηση ερωτήσεων, η μηχανική μετάφραση, η συντακτική ανάλυση, η ανάλυση συναισθημάτων και τα μοντέλα κειμένων και οπτικών σκηνών, ως εφαρμογές της επεξεργασίας φυσικής γλώσσας στις ψηφιακές ανθρωπιστικές επιστήμες και στις υπολογιστικές κοινωνικές επιστήμες.

Ένα ξεχωριστό χαρακτηριστικό του ομίλου Stanford NLP είναι ο αποτελεσματικός συνδυασμός της εξελιγμένης και βαθιάς γλωσσικής μοντελοποίησης της ανάλυσης δεδομένων με καινοτόμες πιθανοτικές μηχανικές μάθησης, και προσεγγίσεις βαθιάς εκμάθησης στην επεξεργασία φυσικής γλώσσας. Η έρευνά του ομίλου, έχει οδηγήσει στην ανάπτυξη τεχνολογιών για ισχυρή, ευρεία κάλυψη του τομέα επεξεργασίας φυσικής γλώσσας σε αρκετές γλώσσες. Αυτή η έρευνα είχε ως αποτέλεσμα την παροχή μιας ευρέως χρησιμοποιούμενης και ολοκληρωμένης εργαλειοθήκης NLP, με επωνομασία “Stanford CoreNLP”. Οι υπηρεσίες του Stanford CoreNLP είναι διαθέσιμες για την επεξεργασία του αραβικού, κινεζικού, γαλλικού, γερμανικού και ισπανικού κειμένου.

Ο όμιλος NLP του Stanford περιλαμβάνει μέλη τόσο του τμήματος γλωσσολογίας όσο και του τμήματος πληροφορικής και είναι μέλος του εργαστηρίου Stanford AI Lab.

2.4.2 Διαθεσιμότητα Συστημάτων και Προγραμματιστικών Γλωσσών

Η βιβλιοθήκη Stanford CoreNLP είναι γραμμένη σε γλώσσα προγραμματισμού Java. Οι τελευταίες εκδόσεις απαιτούν για την σωστή της λειτουργία να είναι εγκαταστημένο το πακέτο της Java 1.8+, και μπορεί να χρησιμοποιηθεί σε όλα τα γνωστά λειτουργικά συστήματα (Microsoft Windows, Linux και Apple OSX).

Η κύρια διεπαφή της βιβλιοθήκης είναι η γραμμή εντολών (command-line) μέσω των Java APIs που παρέχονται. Στην πάροδο των ετών, έχουν δημιουργηθεί APIs για άλλες γλώσσες πέραν της Java. Παρακάτω παρατίθεται ένας πίνακας με τις προγραμματιστικές γλώσσες για τις οποίες υπάρχουν διαθέσιμα APIs της βιβλιοθήκης.

Γλώσσα Προγραμματισμού	Όνομα API
C#/F#/.NET	Stanford CoreNLP for .NET by Sergey Tihon
Clojure	Clojure wrapper for CoreNLP by Cory Giles, Hans Engel, and Damien Stanton. Incomplete. Currently only a tagger and parser
Docker	corenlp-docker A Dockerfile for Stanford CoreNLP Server by motiz88
Erlang/Elixir	corenlp by Robert Bates provides a thin client interface in Erlang to a CoreNLP server. GitHub
Java	DKPro Core is a collection of NLP components, wrapped as UIMA components. It includes the Stanford CoreNLP components, and there is a tutorial on how to use them in the DKPro Core documentation. DKPro Core is part of the DKPro community. It is well-maintained and our recommended way of using Stanford CoreNLP within UIMA. DKPro Core was principally developed by Richard Eckart de Castilho at the Ubiquitous Processing Lab (UKP) at the Technische Universität Darmstadt

JavaScript (node.js)	stanford-simple-nlp is a node.js CoreNLP wrapper by xissy
Lua	CoreNLP.lua. Lua client for StanfordCoreNLPServer by Victor Zhong
Perl	Perl wrapper by Kalle Raeisaenen
PHP	php-stanford-corenlp-adapter by Dennis De Swart. Simple connection to Stanford CoreNLP server
Python	pycorenlp, A Python wrapper for Stanford CoreNLP by Smitha Milli that uses the new CoreNLP v3.6.0 server. Available on PyPI
R (CRAN)	coreNLP: Wrappers Around Stanford CoreNLP Tools by Taylor Arnold and Lauren Tilton. Supports CoreNLP version $\geq 3.5.2$
Ruby	Stanford CoreNLP Ruby bindings by Louis Mullie
Scala	CoreNLP wrapper for Spark v0.1 by Xiangrui Meng of Databricks
Thrift server	Apache Thrift server for Stanford CoreNLP by Diane Napolitano. (Written in Java, but usable from many languages.)
ZeroMQ/ØMQ servers	stanford-0mq by Diane Napolitano. An implementation of a server for Stanford's CoreNLP suite using Ømq and a basic client/server/JSON requests configuration

Πίνακας 3 : Γλώσσες προγραμματισμού οι οποίες μπορούν να χρησιμοποιήσουν τη βιβλιοθήκη Stanford CoreNLP

2.4.3 Οδηγίες Χρήσης Γραμμής Εντολών

Σε αυτό το κεφάλαιο υπάρχουν οι οδηγίες για την επιτυχημένη ανάπτυξη και χρήση της βιβλιοθήκης Stanford CoreNLP. Απαιτούμενες προϋποθέσεις είναι είναι ένας υπολογιστής λειτουργικού συστήματος Microsoft Windows, Linux ή Apple OSX. Βασικές γνώσεις χρήσης

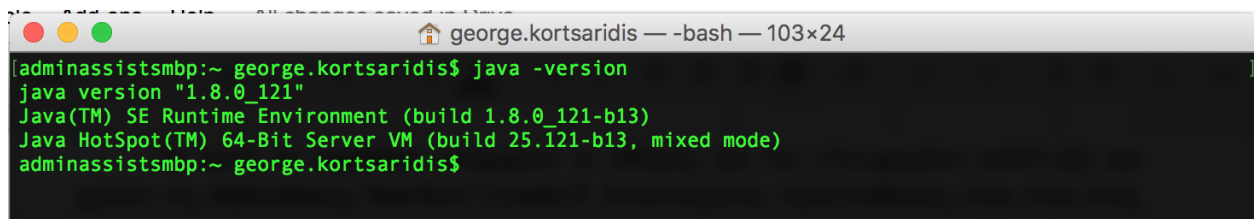
γραμμής εντολών σε οποιοδήποτε από αυτά τα λειτουργικά συστήματα θα βοηθήσουν αρκετά στην κατανόηση της διαδικασίας.

2.4.3.1 Εγκατάσταση Java Environment

Βασική προϋπόθεση για την εκτέλεση της βιβλιοθήκης είναι να υπάρχει εγκατεστημένη η έκδοση του Java Environment 1.8+. Μπορούμε να ελέγξουμε αν είναι ήδη εγκατεστημένη, ανοίγοντας ένα παράθυρο γραμμής εντολών και γράφοντας την εξής εντολή:

```
java -version
```

Αν υπάρχει εγκατεστημένη, θα υπάρχει ένα αποτέλεσμα σαν της παρακάτω φωτογραφίας, το οποίο γνωστοποιεί την έκδοση της Java που υπάρχει στο σύστημα.

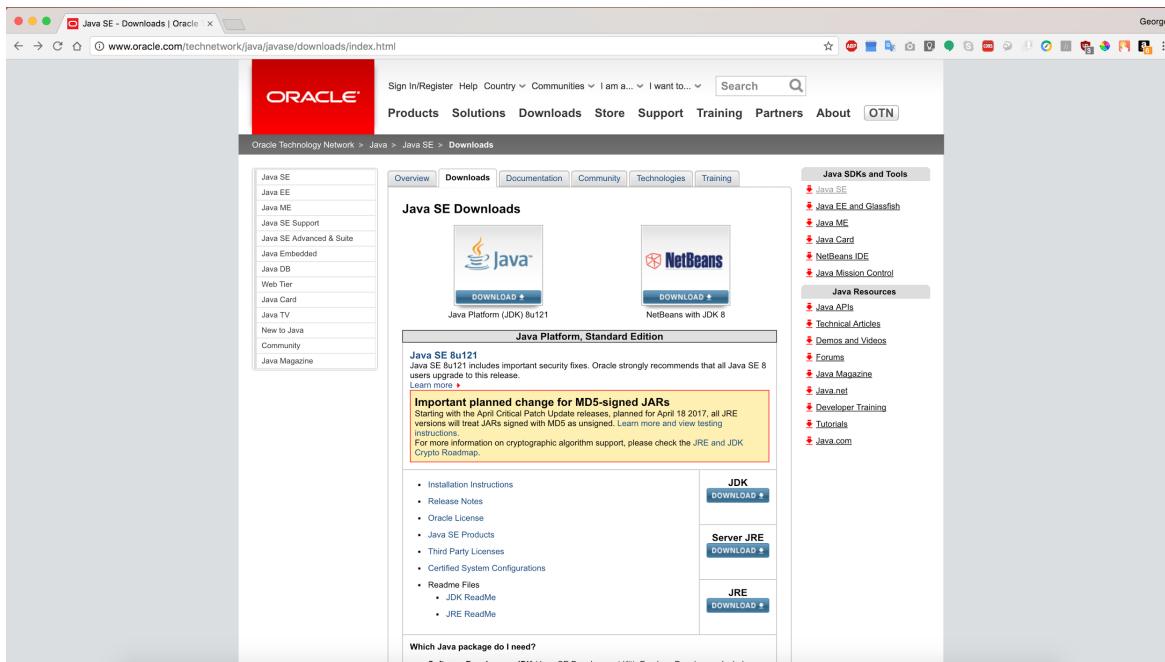
A screenshot of a terminal window on a Mac. The title bar shows 'george.kortsaridis — -bash — 103x24'. The terminal text is as follows:

```
[adminassistsmbp:~ george.kortsaridis$ java -version  
java version "1.8.0_121"  
Java(TM) SE Runtime Environment (build 1.8.0_121-b13)  
Java HotSpot(TM) 64-Bit Server VM (build 25.121-b13, mixed mode)  
adminassistsmbp:~ george.kortsaridis$
```

Εικόνα 18 : Γραμμή εντολών, έκδοση java

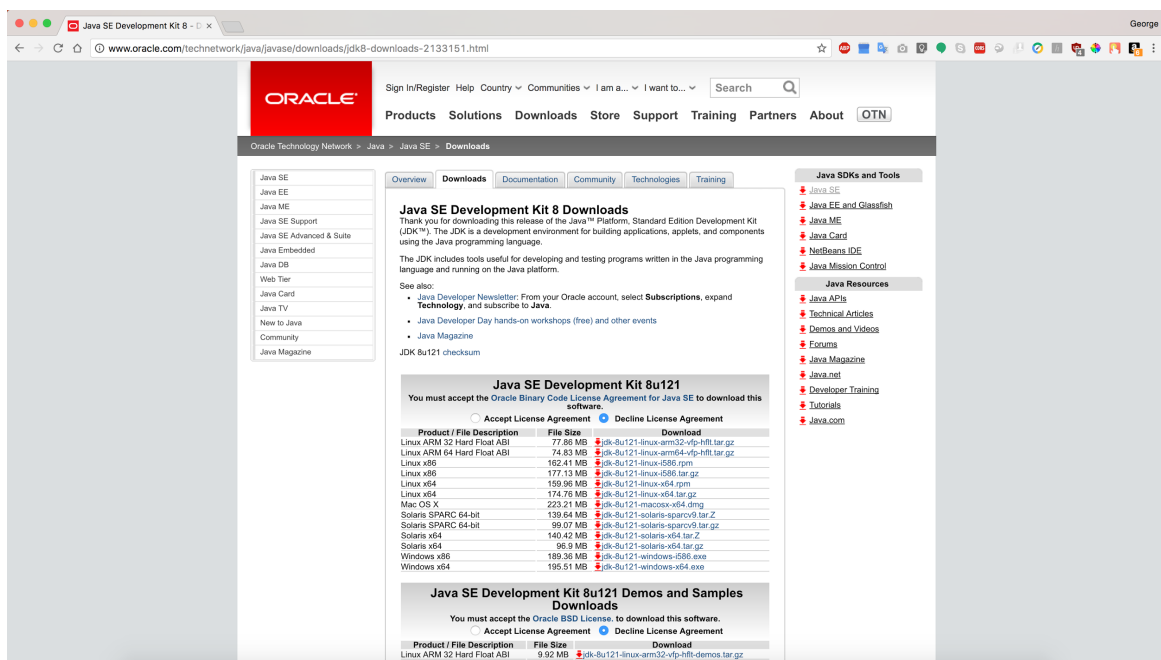
Εδώ θα πρέπει να βεβαιωθείτε ότι η έκδοση της java είναι μεγαλύτερη της 1.8. Αν το σύστημα πληρεί αυτή την προϋπόθεση, μπορείτε να προχωρήσετε στο υποκεφάλαιο 2.4.3.2 Λήψη Βιβλιοθήκης Stanford CoreNLP. Σε αντίθετη περίπτωση θα πρέπει να πλοηγηθείτε στην εξής σελίδα :

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>



Εικόνα 19 : Σελίδα Oracle Java

Από την αρχική σελίδα, επιλέγεται το κουμπί με ονομασία “JDK”. Στην συνέχεια θα πρέπει να επιλέξετε το κουμπί “Accept licence agreement”, και να κάνετε κλικ στον σύνδεσμο που αντιστοιχεί στο λειτουργικό σύστημα του μηχανήματος σας.



Εικόνα 20 : Σελίδα λήψης Java JDK

Εφόσον πραγματοποιηθεί επιτυχημένη λήψη του πακέτου Oracle Java, ανοίγετε το εκτελέσιμο και ακολουθείτε τις οδηγίες για να ολοκληρώσετε την εγκατάσταση.

2.4.3.2 Λήψη Βιβλιοθήκης Stanford CoreNLP

Για την λήψη της βιβλιοθήκης Stanford CoreNLP, αρκεί να περιηγηθείτε στην εξής σελίδα :

<https://stanfordnlp.github.io/CoreNLP/index.html>

The screenshot shows the Stanford CoreNLP website. On the left is a navigation menu with options like Overview, Usage, Download / Maven, etc. The main content area has a 'Download' section with a large red button labeled 'Download CoreNLP 3.7.0'. Below this is a table listing available language models.

LANGUAGE	MODEL JAR	VERSION
Arabic	download	3.7.0
Chinese	download	3.7.0
English	download	3.7.0
English (KBP)	download	3.7.0

Εικόνα 21 : Σελίδα λήψης Stanford CoreNLP

Από αυτήν την σελίδα χρειάζεται να βρούμε το κουμπί “Download CoreNLP X.X.X” (όπου X.X.X η πιο πρόσφατη έκδοση της βιβλιοθήκης). Αυτό θα πραγματοποιήσει λήψη ενός συμπίεσμένου .zip αρχείου με όλα τα αρχεία που χρειάζεται η βιβλιοθήκη για να λειτουργήσει. Εδώ να σημειωθεί ότι η βιβλιοθήκη δε βρίσκεται σε εκτελέσιμη μορφή, αλλά ως φάκελος με τα απαραίτητα java αρχεία χρήσης.

2.4.3.3 Χρήση Stanford CoreNLP μέσω γραμμής εντολών

Ο πιο εύκολος τρόπος χρησιμοποίησης της βιβλιοθήκης Stanford CoreNLP για παραγωγή γρήγορων αποτελεσμάτων είναι μέσω της γραμμής εντολών του λειτουργικού συστήματος. Οι εντολές που θα χρησιμοποιηθούν είναι εντολές της γλώσσας Java, οπότε είναι αδιάφορες του λειτουργικού συστήματος στο οποίο εκτελούνται. Ο χρήστης μπορεί να εκτελέσει την βιβλιοθήκη με την εξής εντολή στη γραμμή εντολών. Σε αυτό το σημείο να διευκρινιστεί ότι ο ενεργός φάκελος στην γραμμή εντολών πρέπει να είναι ο φάκελος του Stanford CoreNLP του προηγούμενου βήματος.

```
java -cp "*" -Xmx2g edu.stanford.nlp.pipeline.StanfordCoreNLP -  
annotators tokenize,ssplit,pos,lemma,ner,parse,dcoref -file  
input.txt
```

Η εντολή που μόλις γράψαμε χωρίζεται στα εξής κομμάτια:

Εντολή	Εξήγηση
java	Η βιβλιοθήκη είναι γραμμένη σε γλώσσα προγραμματισμού java, οπότε χρειάζεται η αντίστοιχη εντολή για να εκκινήσει την εκτέλεση
-cp "*"	Δηλώνει ότι πρέπει να εκτελεστούν όλα τα .jar αρχεία που βρίσκονται στον φάκελο
-Xmx2g	Δηλώνει το μέγεθος της RAM που μπορεί να χρησιμοποιήσει η βιβλιοθήκη για την εκτέλεση της. Σε ένα μηχάνημα 64bit τυπικά

	απαιτούνται 2 gigabytes μνήμης. Σε μηχανήμα 32bit όμως, δεν μπορούν να χρησιμοποιηθούν 2 gigabytes, οπότε αντικαθιστούμε αυτό το κομμάτι με την εντολή -Xmx1800m για 1800 megabytes RAM ή την εντολή -Xmx1500m για 1500 megabytes αντίστοιχα
edu.stanford.nlp.pipeline.StanfordCoreNLP	Αυτό είναι το path της βασικής κλάσης η οποία πρέπει να κληθεί
-annotators tokenize,ssplit,pos,lemma,ner,parse,dcoref	Ως annotators ορίζουμε τις κατηγορίες NLP όπου χρειάζονται για την επεξεργασία του κειμένου (Βλέπε ενότητα 2.2)
-file input.txt	Ορισμός του path με το αρχείο κειμένου προς ανάλυση

Πίνακας 4 : Επεξήγηση εντολής γραμμής εντολών του Stanford CoreNLP

Η παραπάνω εντολή θα επεξεργαστεί το κείμενο που βρίσκεται στο αρχείο input.txt, και θα δημιουργήσει το αρχείο εξόδου input.txt.xml με τα αποτελέσματα της επεξεργασίας σε μορφή xml. Παρακάτω παρατίθεται κείμενο εισόδου μαζί με το αποτέλεσμα μετά την επεξεργασία.

Είσοδος: Stanford University is located in California.

Έξοδος:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="CoreNLP-to-HTML.xsl" type="text/xsl"?>
<root>
  <document>
    <sentences>
      <sentence id="1">
        <tokens>
```

```

<token id="1">
  <word>Stanford</word>
  <lemma>Stanford</lemma>
  <CharacterOffsetBegin>0</CharacterOffsetBegin>
  <CharacterOffsetEnd>8</CharacterOffsetEnd>
  <POS>NNP</POS>
  <NER>ORGANIZATION</NER>
  <Speaker>PER0</Speaker>
</token>
<Ομοίως όλα τα υπόλοιπα tokens>
<token id="7">
  <word>.</word>
  <lemma>.</lemma>
  <CharacterOffsetBegin>44</CharacterOffsetBegin>
  <CharacterOffsetEnd>45</CharacterOffsetEnd>
  <POS>.</POS>
  <NER>O</NER>
  <Speaker>PER0</Speaker>
</token>
</tokens>
<parse>(ROOT (S (NP (NNP Stanford) (NNP University)) (VP (VBZ is) (ADJP (JJ
located) (PP (IN in) (NP (NNP California))))) (. .))) </parse>
<dependencies type="basic-dependencies">
  <dep type="root">
    <governor idx="0">ROOT</governor>
    <dependent idx="4">located</dependent>
  </dep>
  <dep type="compound">
    <governor idx="2">University</governor>
    <dependent idx="1">Stanford</dependent>
  </dep>
  <dep type="nsubj">
    <governor idx="4">located</governor>
    <dependent idx="2">University</dependent>
  </dep>
  <dep type="cop">
    <governor idx="4">located</governor>
    <dependent idx="3">is</dependent>
  </dep>

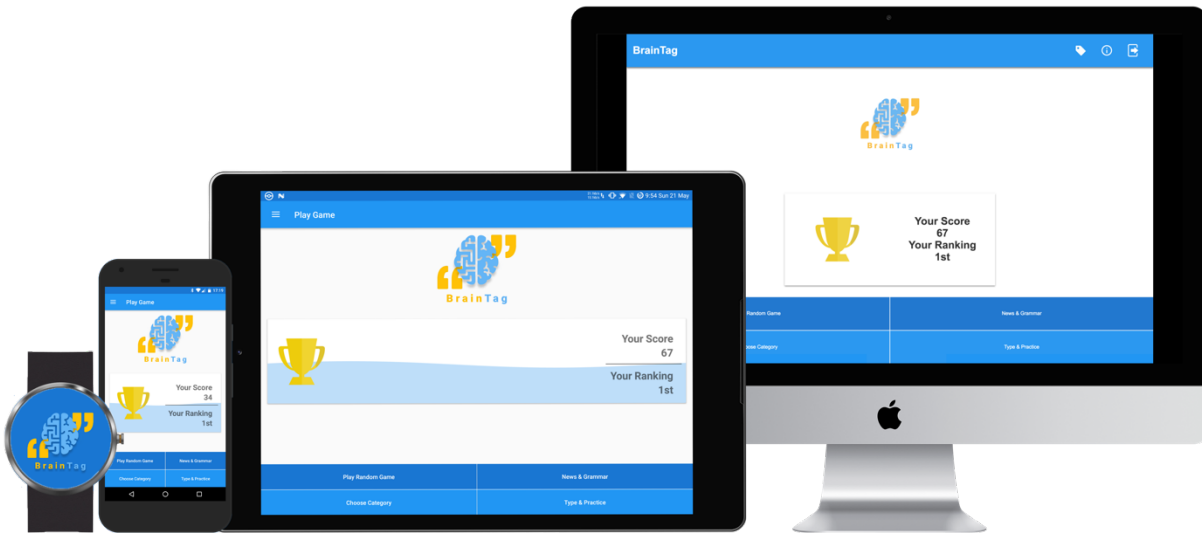
```

```

    </dep>
    <dep type="case">
      <governor idx="6">California</governor>
      <dependent idx="5">in</dependent>
    </dep>
    <dep type="nmod">
      <governor idx="4">located</governor>
      <dependent idx="6">California</dependent>
    </dep>
    <dep type="punct">
      <governor idx="4">located</governor>
      <dependent idx="7">.</dependent>
    </dep>
  </dependencies>
  <Ομοίως collapsed-ccprocessed-dependencies >
  <Ομοίως collapsed-dependencies >
  <Ομοίως enhanced-dependencies >
  <Ομοίως enhanced-plus-plus-dependencies >
</sentence>
</sentences>
</document>
</root>

```

3. BrainTag



Εικόνα 22 : Γκάμα συσκευών για τις οποίες είναι διαθέσιμη η εφαρμογή BrainTag

3.1 Οδηγός Χρήσης Εφαρμογής

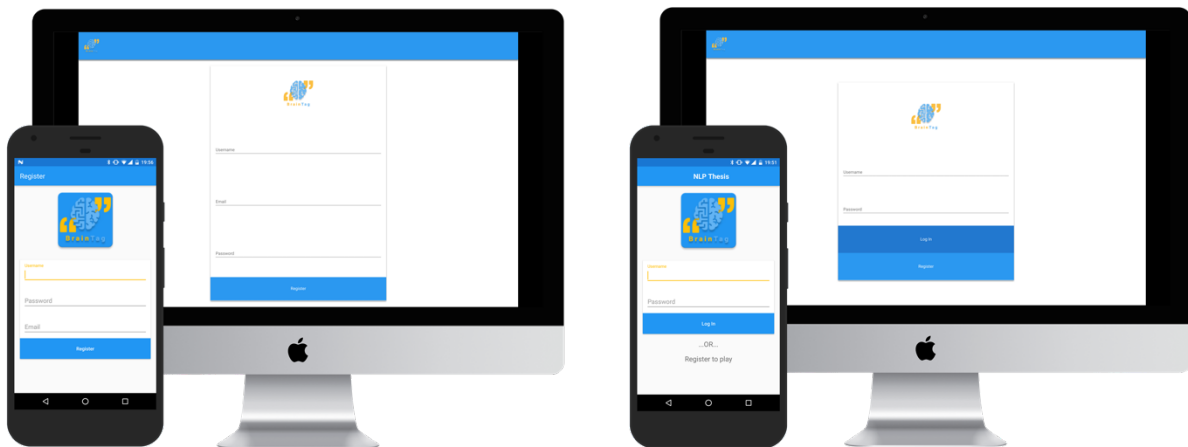
3.1.1 Λήψη και εγκατάσταση εφαρμογής

Ο χρήστης μπορεί να λάβει δωρεάν την εφαρμογή BrainTag σε οποιαδήποτε συσκευή λειτουργικού συστήματος Android, μέσω του Google Playstore. Μέσω του συγκεκριμένου καταστήματος, μπορεί να βρεθεί η εφαρμογή κάνοντας αναζήτηση με το όρο 'BrainTag' και στη συνέχεια λήψη και εγκατάσταση.

Στην περίπτωση χρήσης της υπηρεσίας από υπολογιστή, ο χρήστης μπορεί να λάβει το εκτελέσιμο αρχείο του προγράμματος από την ιστοσελίδα <http://83.212.118.131/braintag/computer>. Μετά την επιτυχή λήψη, το πρόγραμμα είναι έτοιμο προς εκτέλεση. Δεν υπάρχει ανάγκη εγκατάστασης.

3.1.2 Εγγραφή και είσοδος

Η εγγραφή είναι ένα απαραίτητο βήμα για την χρήση της υπηρεσίας BrainTag. Ο χρήστης την πρώτη φορά που θα εκκινήσει την εφαρμογή σε μια συσκευή, θα βρεθεί στην οθόνη εισόδου. Εκεί μπορεί να συμπληρώσει το όνομα χρήστη και τον προσωπικό του κωδικό, αν έχουν δημιουργηθεί σε άλλη συσκευή, ώστε να πραγματοποιήσει είσοδο. Αν δεν έχει πραγματοποιηθεί εγγραφή, υπάρχει κατάλληλο κουμπί που μεταφέρει τον χρήστη στην κατάλληλη οθόνη. Εκεί συμπληρώνοντας τα στοιχεία του και πατώντας το κουμπί 'Register' δημιουργεί τον προσωπικό του λογαριασμό. Στην συνέχεια μπορεί να πραγματοποιήσει είσοδο χρήστη στην υπηρεσία.

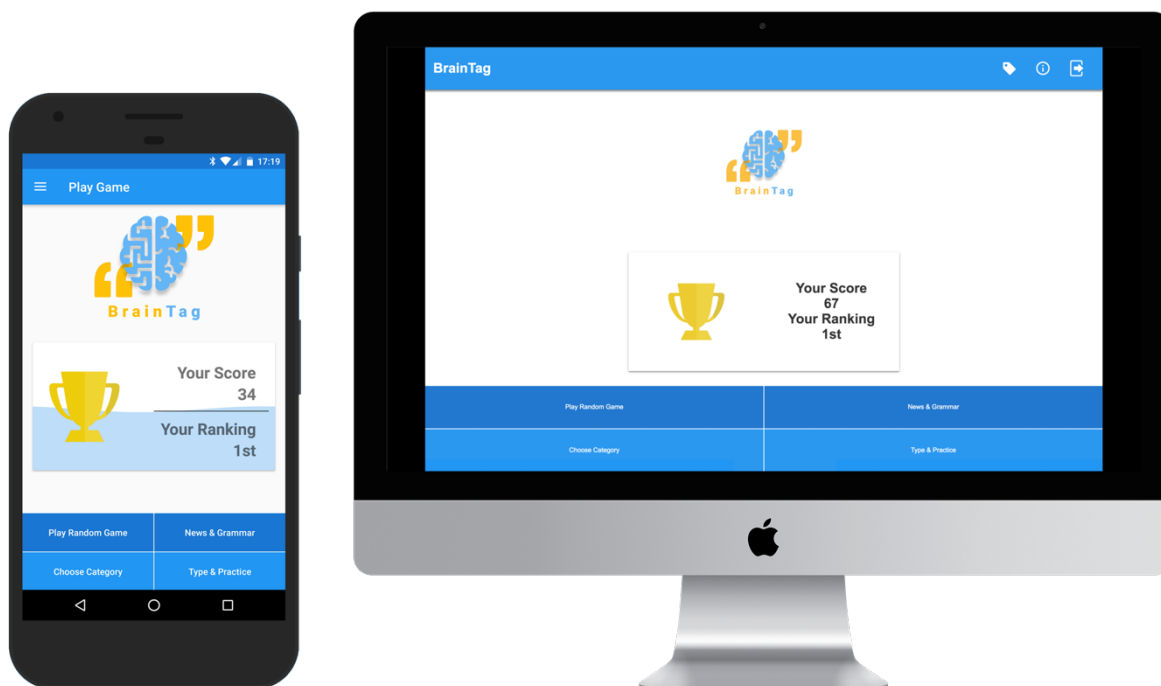


Εικόνα 23 : Οθόνες εγγραφής και εισόδου της εφαρμογής BrainTag

3.1.3 Αρχικό μενού

Το αρχικό μενού είναι η οθόνη στην οποία μεταφέρεται ο χρήστης κατά την εκκίνηση της εφαρμογής, μετά την επιτυχημένη είσοδο. Περιέχει πληροφορίες για το προσωπικό σκορ του

παίκτη, όπως και την θέση του στην παγκόσμια κατάταξη χρηστών της εφαρμογής. Από εκεί ο χρήστης μπορεί να ξεκινήσει ένα παιχνίδι τυχαίας κατηγορίας, να επιλέξει κατηγορία παιχνιδιού, να παίξει χρησιμοποιώντας τίτλους ειδήσεων ή και δικό του κείμενο. Επίσης μπορεί να μεταφερθεί στις πληροφορίες της εφαρμογής, και στην οθόνη με τα ονόματα των ετικετών.



Εικόνα 24 : Αρχική οθόνη της εφαρμογής

3.1.4 Πληροφορίες Εφαρμογής

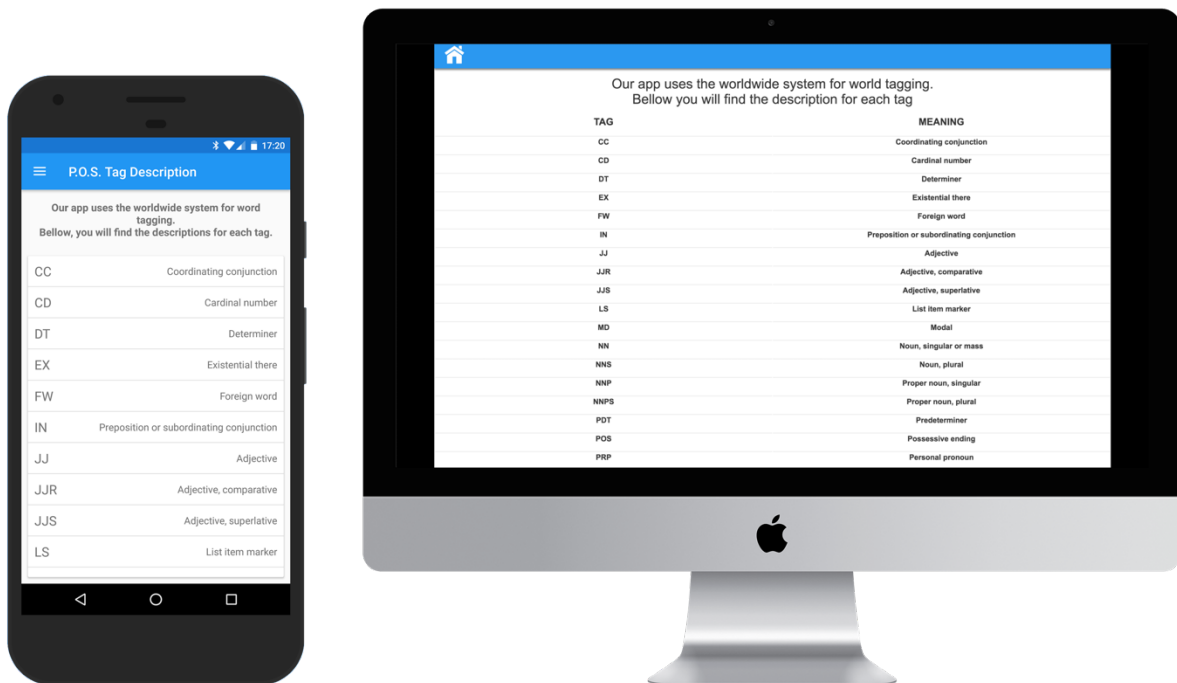
Από την αρχική οθόνη, ο χρήστης έχει την δυνατότητα να μεταφερθεί στο τμήμα των πληροφοριών της εφαρμογής. Εκεί ο χρήστης μπορεί να δει πληροφορίες για τον προγραμματιστή της εφαρμογής, όπως και τις εξωτερικές βιβλιοθήκες που χρησιμοποιήθηκαν κατά την προγραμματιστική διαδικασία.



Εικόνα 25 : Οθόνη πληροφοριών εφαρμογής BrainTag

3.1.5 Ετικέτες συντακτικού Penn TreeBank

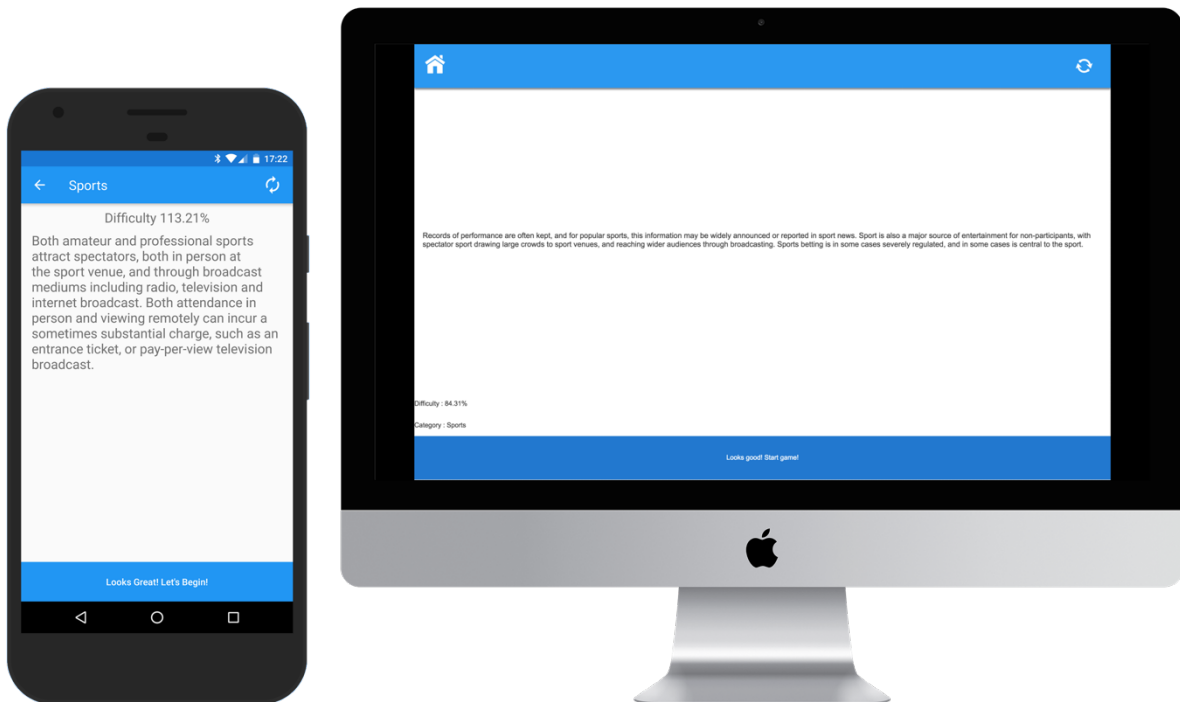
Ο χρήστης από την αρχική οθόνη μπορεί να μεταφερθεί στο τμήμα των ετικετών συντακτικού Penn TreeBank. Εκεί υπάρχει μια λίστα με όλες τις ετικέτες συντακτικού, έχοντας την συντόμευση της ετικέτας αλλά και το πλήρες όνομα. Αυτό έχει ως αποτέλεσμα, ο χρήστης να μπορεί να γνωρίσει και να μάθει πιο εύκολα τις συντομογραφίες των ετικετών.



Εικόνα 26 : Ετικέτες συντακτικού Penn Treebank εφαρμογής BrainTag

3.1.6 Ανάλυση Παραγράφου

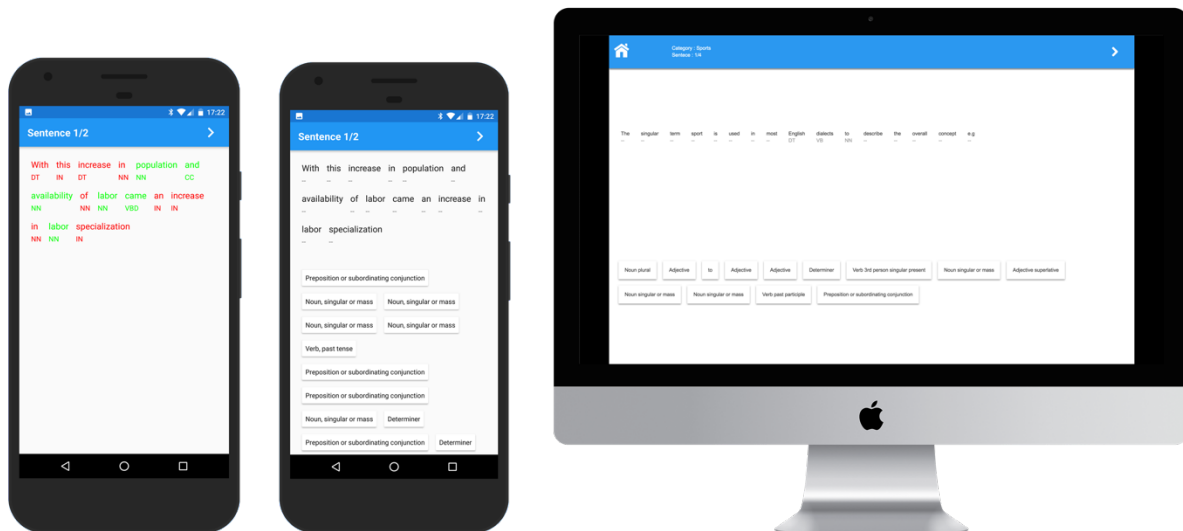
Πριν την εκκίνηση της διαδικασίας κάθε παιχνιδιού, ο χρήστης έχει την δυνατότητα να δει ολόκληρη την παράγραφο στην οποία θα αγωνιστεί. Από εκεί, μπορεί να μάθει την δυσκολία της παραγράφου, και αν επιθυμεί να λάβει καινούρια παράγραφο. Όταν αποφασίσει ότι η παράγραφος που έχει λάβει είναι ικανοποιητική, ξεκινάει την διαδικασία του παιχνιδιού με το πάτημα ενός κουμπιού.



Εικόνα 27 : Ανάλυση παραγράφου εφαρμογής BrainTag

3.1.7 Διαδικασία Παιχνιδιού

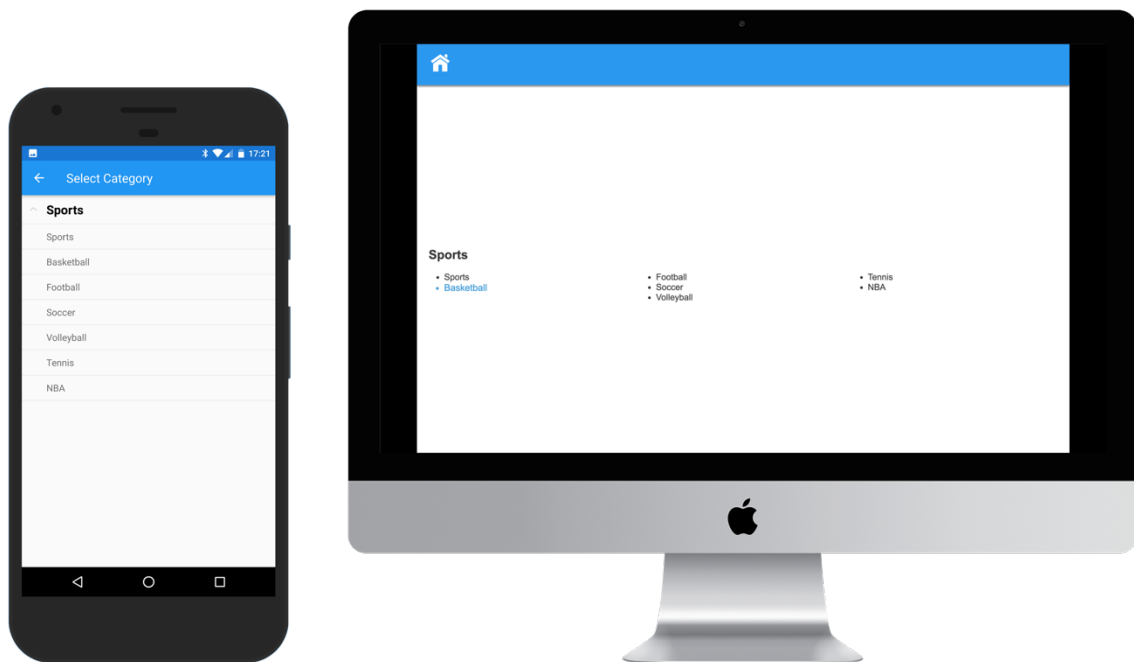
Η διαδικασία παιχνιδιού ξεκινάει εφόσον ο χρήστης επιβεβαιώσει την παράγραφο στην οποία θα αγωνιστεί. Κατά αυτήν την διαδικασία, χωρίζεται η παράγραφος σε προτάσεις, και εμφανίζονται μια-μια στον χρήστη. Σε κάθε πρόταση, στο κάτω μέρος της οθόνης υπάρχουν όλες οι ετικέτες των λέξεων του εμφανίζονται. Ο σκοπός του παιχνιδιού είναι ο χρήστης να σύρει με το ποντίκι ή το δάχτυλο του την σωστή ετικέτα από το κάτω μέρος της οθόνης, στην αντίστοιχη λέξη στο πάνω μέρος της οθόνης. Εφόσον αντιστοιχήσει κάθε λέξη της προτάσεως με τις ετικέτες, ο χρήστης επιλέγει το βέλος που αντιστοιχεί στην επόμενη πρόταση. Τότε φαίνονται οι σωστές και οι λανθασμένες επιλογές του χρήστη για 4 δευτερόλεπτα, και στην συνέχεια εμφανίζεται η επόμενη πρόταση.



Εικόνα 28 : Διαδικασία παιχνιδιού εφαρμογής BrainTag

3.1.8 Επιλογή Κατηγοριών Παιχνιδιού

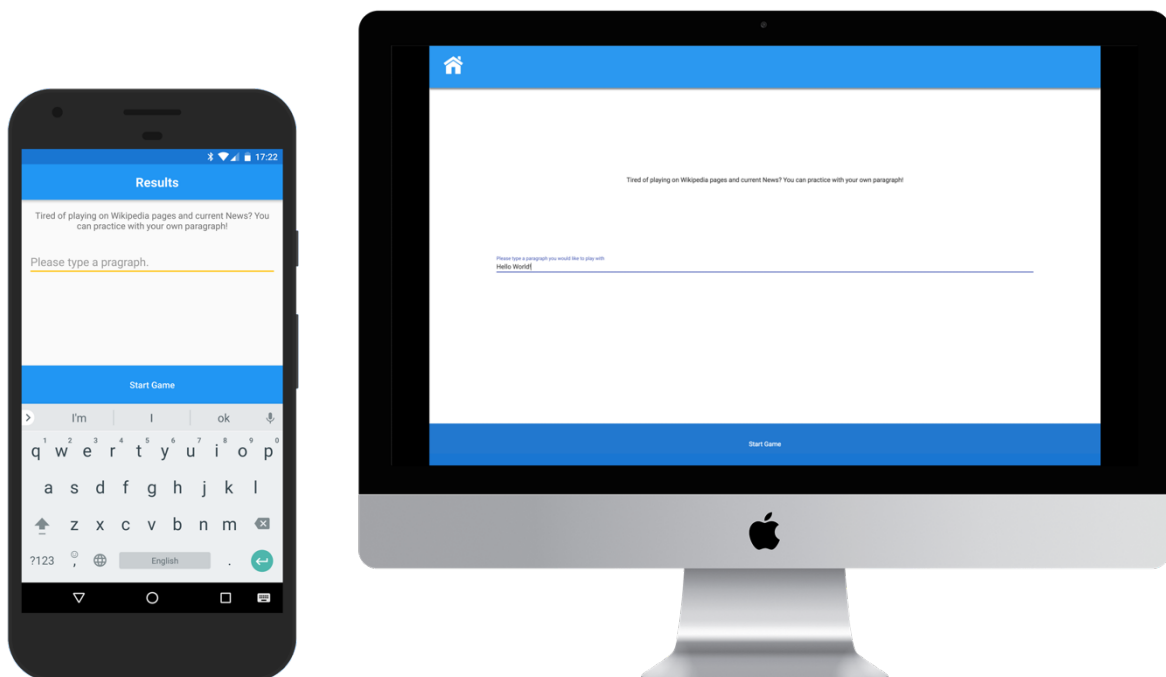
Ένας από τους τρόπους επιλογής παιχνιδιού είναι μετά από επιλογή κατηγοριών. Από την αρχική οθόνη, ο χρήστης μετά από κλικ στο κουμπί «Categories», εμφανίζει όλες τις πιθανές κατηγορίες παιχνιδιού. Οι κατηγορίες εμφανίζονται σε μορφή κύριας κατηγορίας και υποκατηγοριών. Π.Χ βασική κατηγορία ‘Αθλητικά’ με υποκατηγορίες ‘Ποδόσφαιρο’ , ‘Μπάσκετ’ , ‘Τέννις’ , κ.λ.π. Εφόσον ο χρήστης επιλέξει κάποια υποκατηγορία, θα μεταφερθεί στην οθόνη ανάλυσης παραγράφου, με παραγράφους μόνο από την συγκεκριμένη υποκατηγορία.



Εικόνα 29 : Κατηγορίες παιχνιδιού εφαρμογής BrainTag

3.1.9 Εισαγωγή Κειμένου

Η υπηρεσία BrainTag δίνει την δυνατότητα στους χρήστες να προπονούνται στην εκμάθηση συντακτικού, σε δικές τους προτάσεις και παραγράφους. Για να το πετύχουν αυτό, πρέπει να επιλέξουν την επιλογή ‘Type and Play’ από την αρχική οθόνη. Στην συνέχεια υπάρχει πληκτρολόγηση τη παραγράφου προς εξάσκηση, και κλικ στο κουμπί ‘Play Game’



Εικόνα 30 : Οθόνη εισαγωγής κειμένου εφαρμογής BrainTag

3.1.10 Λοιπές επιλογές παιχνιδιού

3.1.10.1 Τυχαίο Παιχνίδι

Μία από τις 4 επιλογές παιχνιδιού είναι το τυχαίο παιχνίδι. Το τυχαίο παιχνίδι μπορεί να ξεκινήσει από την αρχική οθόνη με το κουμπί ‘Random Game’. Εκεί ο χρήστης δέχεται μια τυχαία παράγραφο από τυχαία κατηγορία της ιστοσελίδας wikipedia. Αν επιλέξει αλλαγή παραγράφου, η διαδικασία θα ξεκινήσει από την αρχή, και θα βρεθεί καινούρια κατηγορία και παράγραφος.

3.1.10.2 Παιχνίδι άρθρων ειδήσεων

Η τελευταία επιλογή παιχνιδιού της υπηρεσίας BrainTag είναι το παιχνίδι μέσω άρθρων ειδήσεων. Ο χρήστης μπορεί να εκκινήσει αυτή τη διαδικασία μέσω του κουμπιού ‘News & Grammar’ της αρχικής οθόνης. Η συγκεκριμένη λειτουργία λαμβάνει τίτλους άρθρων από τα ειδησεογραφικά πρακτορία ‘BBC News’ , ‘TechCrunch’ , ‘Google News’ , ‘CNN News’ , ‘National Geographic’ , ‘The Verge’ και ‘USA Today’.

4. Βάση Δεδομένων & Εξυπηρετητής Δεδομένων

4.1 Βάση Δεδομένων

Η βάση δεδομένων της πλατφόρμας BrainTag δημιουργήθηκε με το σκεπτικό να είναι απλή, μικρή και με δυνατότητες επέκτασης. Στην βασική της έκδοση, χρησιμοποιείται μία βάση δεδομένων, με δύο πίνακες.

4.1.1 Πίνακας Users

Ο πρώτος πίνακας, ονομασίας Users, χρησιμοποιείται για να αποθηκεύει τα χρήσιμα στοιχεία των χρηστών. Αυτά είναι το όνομα χρήστη (πεδίο Username) , ο κωδικός χρήστη (πεδίο Password) , η διεύθυνση ηλεκτρονικού ταχυδρομείου (πεδίο Email) και το σκορ το οποίο έχει μαζέψει ως τώρα στο παιχνίδι (πεδίο Score). Με αυτόν τον τρόπο δεν ζητούνται από τον νέο χρήστη πολλές προσωπικές πληροφορίες κατά την εγγραφή, πράγμα που θα μπορούσε να τον αποτρέψει από την ολοκλήρωση της διαδικασίας, με αποτέλεσμα την μη χρήση της εφαρμογής. Παρακάτω παρατίθεται εικόνα με το σχεδιάγραμμα του πίνακα Users.

users		
ID	int	PK
Username	varchar(100)	
Password	varchar(100)	
Email	varchar(100)	
Score	int	

Πίνακας 5 : Πίνακας Users

Ο πίνακας αυτός χρησιμοποιείται κατά την διαδικασία της δημιουργίας καινούριου παίκτη, όπου προστίθενται οι καινούριες εγγραφές. Απαραίτητο ρόλο προφανώς έχει και στην διαδικασία εισόδου υπαρχόντων χρηστών στην εφαρμογή για την εξακρίβωση των στοιχείων. Τελευταία του χρήση είναι κατά το τέλος του κάθε γύρου των παιχνιδιών όπου αυξάνεται το σκορ του αντίστοιχου παίκτη.

4.1.2 Πίνακας wrong_answers

Πέρα του πίνακα με τις πληροφορίες χρηστών, το σύστημα απαιτεί την ύπαρξη ενός δεύτερου, βοηθητικού, πίνακα που περιέχει τις πληροφορίες των λανθασμένων συντακτικών επιλογών όλων των ολοκληρωμένων παιχνιδιών από κάθε λειτουργικό σύστημα και πλατφόρμα (υπολογιστή και εφαρμογή κινητού).

Πιο συγκεκριμένα, ο πίνακας περιέχει μια μοναδική εγγραφή, και ξεχωριστά πεδία για την κάθε ετικέτα συντακτικού, σύμφωνα με τον πίνακα UPenn TreeBank II. Στο τέλος κάθε παιχνιδιού, στέλνεται από την συσκευή του παίκτη προς την βάση δεδομένων ένας πίνακας σε μορφή δεδομένων JSON, με την συχνότητα εμφάνισης λαθών της κάθε ετικέτας. Με αυτόν τον τρόπο το σύστημα έχει πρόσβαση σε πληροφορίες όπως ποια ετικέτα έχει απαντηθεί λάθος τις

πιο πολλές φορές, και για κάποια συγκεκριμένη ετικέτα, ποιος είναι ο συνολικός αριθμός εμφάνισης λαθών. Αυτό είναι απαραίτητο στην διαδικασία εύρεσης του βαθμού δυσκολίας της κάθε παραγράφου, ο οποίος εμφανίζεται στον χρήστη πριν την εκκίνηση του παιχνιδιού, και χρησιμοποιείται ως είσοδος στον αλγόριθμο ο οποίος αποφασίζει το τελικό σκορ του παίκτη. Παρακάτω παρατίθεται εικόνα με το σχεδιάγραμμα του πίνακα `wrong_answers`.

wrong_answers	
CC	int
CD	int
DT	int
EX	int
FW	int
IN_	int
JJ	int
JJR	int
JJS	int
LS	int
MD	int
NN	int
NNS	int
NNP	int
NNPS	int
PDT	int
POS	int
PRP	int
PRP_	int
RB	int
RBS	int
RP	int
SYM	int
TO_	int
UH	int
VB	int
VBD	int
VBG	int
VCN	int
VBP	int
VBZ	int
WDT	int
WP	int
WP_	int
WRB	int

Πίνακας 6 : Πίνακας wrong_answers

4.1.3 Σχηματική Δημιουργία Βάσης Δεδομένων

Για την δημιουργία ξεχωριστής βάσης δεδομένων, μπορείτε να χρησιμοποιήσετε τον εξής mysql κώδικα.

```
--  
-- Database: `braintag`  
--  
CREATE DATABASE IF NOT EXISTS `braintag` DEFAULT CHARACTER SET latin1 COLLATE  
latin1_swedish_ci;  
USE `braintag`;  
  
-----  
  
--  
-- Table structure for table `users`  
--  
CREATE TABLE IF NOT EXISTS `users` (  
  `ID` int(11) NOT NULL AUTO_INCREMENT,  
  `Username` varchar(100) NOT NULL,  
  `Password` varchar(100) NOT NULL,  
  `Email` varchar(100) NOT NULL,  
  `Score` int(11) NOT NULL,  
  PRIMARY KEY (`ID`)  
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;  
  
-----  
  
--  
-- Table structure for table `wrong_answers`  
--  
CREATE TABLE IF NOT EXISTS `wrong_answers` (  
  `CC` int(11) NOT NULL,  
  `CD` int(11) NOT NULL,  
  `DT` int(11) NOT NULL,  
  `EX` int(11) NOT NULL,  
  `FW` int(11) NOT NULL,  
  `IN` int(11) NOT NULL,  
  `JJ` int(11) NOT NULL,  
  `JJR` int(11) NOT NULL,  
  `JJS` int(11) NOT NULL,  
  `LS` int(11) NOT NULL,  
  `MD` int(11) NOT NULL,  
  `NN` int(11) NOT NULL,  
  `NNS` int(11) NOT NULL,  
  `NNP` int(11) NOT NULL,  
  `NNPS` int(11) NOT NULL,  
  `PDT` int(11) NOT NULL,  
  `POS` int(11) NOT NULL,  
  `PRP` int(11) NOT NULL,  
  `PRP` int(11) NOT NULL,
```

```
`RB` int(11) NOT NULL,  
`RBR` int(11) NOT NULL,  
`RBS` int(11) NOT NULL,  
`RP` int(11) NOT NULL,  
`SYM` int(11) NOT NULL,  
`TO_` int(11) NOT NULL,  
`UH` int(11) NOT NULL,  
`VB` int(11) NOT NULL,  
`VBD` int(11) NOT NULL,  
`VBG` int(11) NOT NULL,  
`VBN` int(11) NOT NULL,  
`VBP` int(11) NOT NULL,  
`VBZ` int(11) NOT NULL,  
`WDT` int(11) NOT NULL,  
`WP` int(11) NOT NULL,  
`WP_` int(11) NOT NULL,  
`WRB` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4.2 Εξυπηρετητής Δεδομένων

4.2.1 Λειτουργικό Σύστημα

Ο εξυπηρετητής δεδομένων (main server) εκτελείται σε ένα εικονικό μηχάνημα λειτουργικού συστήματος Ubuntu Server LTS 14.4. Η επιλογή του εξής λειτουργικού έγινε με την λογική ότι για την εξής διεργασία δεν χρειάζεται εξυπηρετητής με γραφικό περιβάλλον. Επίσης η εμπειρία και η εξοικείωση με την γραμμή εντολών του περιβάλλοντος linux, βοήθησε στην επιτάχυνση της διαδικασίας προγραμματισμού.

Για την επιτυχή εκτέλεση του εξυπηρετητή της πλατφόρμας BrainTag, χρειάζεται η εγκατάσταση των εξής προγραμμάτων στο λειτουργικό σύστημα.

Πακέτο	Εντολή εγκατάστασης
NodeJS	sudo apt-get install nodejs
NPM	sudo apt-get install npm
MySQLServer	sudo apt-get install mysql-server
phpMyAdmin	sudo apt-get install php5-mysql phpmyadmin
Java Development Kit	sudo apt-get install openjdk-7-jdk

Πίνακας 7 : Εγκατάσταση απαραίτητων πακέτων στο λειτουργικό σύστημα

4.2.3 Κώδικας Εξυπηρετητή

Ολόκληρος ο κώδικας του εξυπηρετητή είναι γραμμένος σε γλώσσες Javascript και NodeJS. Αυτή η επιλογή έγινε γιατί μέσω των πακέτων NodeJS παρέχονται αρκετές βιβλιοθήκες για την γρήγορη και εύκολη ολοκλήρωση των απαραίτητων διεργασιών. Ο κώδικας του εξυπηρετητή φιλοξενείται στην ιστοσελίδα του github.com. Βρίσκεται διαθέσιμος σε όλους προς κατέβασμα, από την ιστοσελίδα

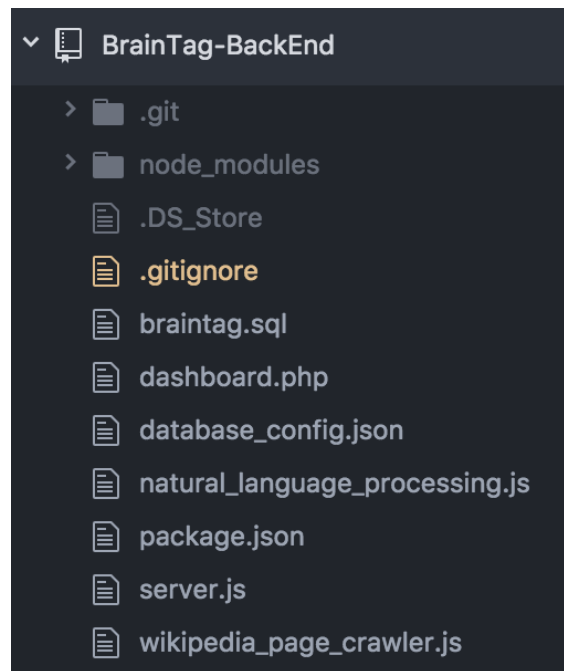
<https://github.com/gkortsaridis/BrainTag-BackEnd>

Για αυτούς που προτιμούν την γραμμή εντολών και το git, ο κώδικας είναι διαθέσιμος με την εντολή

```
git clone https://github.com/gkortsaridis/BrainTag-BackEnd.git
```

4.2.3.1 Project Structure

Για την εξοικείωση του προγραμματιστή με όλα τα αρχεία της πλατφόρμας, παρακάτω παρατίθεται η δομή του φακέλου, και εξηγείται η χρήση και η λειτουργία του κάθε αρχείου ξεχωριστά.



Εικόνα 31 : Project Structure BrainTag BackEnd

Μετά την λήψη του φακέλου είτε μέσω της ιστοσελίδας του Github, είτε μέσω της εντολής γραμμής εντολών, θα λείπει το αρχείο `.DS_Store`, όπως και ο φάκελος `node_modules`. Το αρχείο `.DS_Store`, δημιουργήθηκε λόγω του λειτουργικού συστήματος Apple OSX, κατά την δημιουργία των αρχείων. Δεν έχει κανένα ρόλο στην λειτουργία της πλατφόρμας. Ο φάκελος `node_modules`, είναι φάκελος ο οποίος αποθηκεύει όλες τις βιβλιοθήκες που είναι απαραίτητες για την εκτέλεση του κώδικα. Οι λόγοι που δεν υπάρχει έτοιμος από την στιγμή που κατεβαίνουν τα αρχεία είναι οι εξής. Ο βασικότερος λόγος είναι γιατί οι εξωτερικές βιβλιοθήκες που χρησιμοποιούνται στο πρόγραμμα, αναβαθμίζονται με την πάροδο του καιρού, οπότε είναι

σωστή τεχνική όταν κάποιος προσπαθήσει να εκτελέσει τον κώδικα, να έχει πάντα τις τελευταίες εκδόσεις αυτών των βιβλιοθηκών. Ο δεύτερος λόγος είναι για μείωση του μεγέθους χώρου στην σελίδα του Github. Ο φάκελος `node_modules` μπορεί να δημιουργηθεί με ευκολία, έχοντας τις τελευταίες εκδόσεις των εξωτερικών βιβλιοθηκών, μέσω της γραμμής εντολών. Πηγαίνοντας στον φάκελο λήψης του κώδικα με όνομα BrainTag, εκτελείται η εξής εντολή

```
npm install
```

Το πακέτο `npm`, θα βρει όλες τις απαραίτητες βιβλιοθήκες που χρειάζονται για την εκτέλεση του κώδικα, θα τις κατεβάσει, και θα τις τοποθετήσει στον φάκελο αυτόματα.

4.2.3.2 database_config.json

Το αρχείο `database_config.json` είναι ένα πολύ μικρό, αλλά χρήσιμο αρχείο, γιατί κρατάει τις πληροφορίες των στοιχείων σύνδεσης της βάσης δεδομένων. Σε αυτό το αρχείο, ο προγραμματιστής πρέπει να δηλώσει που βρίσκεται η βάση δεδομένων (πέδίο `host`, δίνεται είτε IP διεύθυνση, είτε `'localhost'` για τοπική βάση δεδομένων), όνομα χρήστη για σύνδεση στη βάση δεδομένων (πέδίο `user`), κωδικό σύνδεσης στην βάση δεδομένων (πέδίο `password`) και το όνομα της βάσης δεδομένων (πέδίο `database`). Παρακάτω παρατίθεται ο κώδικας του αρχείου `database_config.json`

```
{
  "host"      : "localhost",
  "user"      : "dbUsername",
  "password"  : "dbPassword",
  "database"  : "braintag"
}
```

4.2.3.3 package.json

Το αρχείο package.json , χρησιμοποιείται από το πακέτο NPM, και περιέχει τις βασικές πληροφορίες του φακέλου εργασίας. Πληροφορίες που παρέχονται είναι το όνομα, έκδοση και περιγραφή του project, το αρχικό αρχείο προς εκτέλεση (server.js), συγγραφέα και άδεια χρήσης του κώδικα. Επίσης υπάρχουν οι εξαρτήσεις από εξωτερικές βιβλιοθήκες. Με την εκτέλεση της εντολής npm install, γίνεται λήψη των βιβλιοθηκών που υπάρχουν στο “dependencies”. Παρακάτω παρατίθεται ο κώδικας του αρχείου package.json

```
{
  "name": "BrainTag-BackEnd",
  "version": "1.0.0",
  "description": "All the backend structure for the BraiTag platform",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "dependencies": {
    "express": ">= 4.14.1",
    "body-parser": ">= 1.16.0",
    "mysql": ">= 2.13.0",
    "sync-request" : ">=4.0.1",
    "encodeurl" : ">=1.0.1",
    "cheerio": ">=0.22.0"
  },
  "author": "George Kortsaridis",
  "license": "ISC"
}
```

4.2.3.4 wikipedia_page_crawler.js

Το αρχείο wikipedia_page_crawler.js χρησιμοποιείται ξεχωριστό module στην εκτέλεση του κώδικα. Η δουλειά που εκτελεί είναι η αναζήτηση μιας συγκεκριμένης σελίδας της εγκυκλοπαίδειας wikipedia, η λήψη ολόκληρου του κειμένου από το αποτέλεσμα αναζήτησης, η μορφοποίηση του σε αποδεκτή μορφή δεδομένων JSON, και η επιστροφή του αποτελέσματος

στο κυρίως πρόγραμμα. Παρακάτω παρατίθεται ο κώδικας του αρχείου wikipedia_page_crawler.js

```
/******BrainTag*****  
  Author : George Kortsaridis  
  Email  : gkortsaridis@gmail.com  
******/  
  
var request = require('sync-request');  
var cheerio = require('cheerio');  
var encodeUrl = require('encodeurl')  
  
module.exports = {  
  //Crawls a specific section of wikipedia based on language, and returns back an array of  
  //text paragraphs.  
  crawl : function(language,section){  
    //Make HTTP call on the wiki page.  
    var link = encodeUrl('https://'+language+'.wikipedia.org/wiki/'+section);  
    var res = request('GET', link);  
    var wiki_paragraphs = [];  
    var $ = cheerio.load(String(res.getBody()));  
  
    //Use Cheerio, to get all <p> tags from the response.  
    $('p').each(function(i, element){  
      var par = $(this).text();  
      if(par != ""){ wiki_paragraphs.push($(this).text()); }  
    });  
  
    return this.removeNumbers(wiki_paragraphs);  
  },  
  
  //Removes numbers from text formatted text. From "this is [87] my text [88] to format  
  // [89]" -> to "this is my text to format".  
  removeNumbers : function (text){  
    for(var j=0; j<text.length; j++){  
      for(var i=0; i<999; i++){  
        text[j] = text[j].replace("[ "+i+" ]", "");  
      }  
    }  
    return text;  
  }  
};
```

4.2.3.5 natural_language_processing.js

Το αρχείο `natural_language_processing.js` λειτουργεί ως ξεχωριστό module. Θα μπορούσε να θεωρηθεί ίσως και η καρδιά ολόκληρης της διπλωματικής εργασίας. Η δουλειά που εκτελεί είναι η πλήρη συντακτική ανάλυση οποιοδήποτε κειμένου και η επιστροφή του σε δομημένη JSON μορφή. Παρακάτω παρατίθεται ο κώδικας του αρχείου `natural_language_processing.js`

```

/*****BrainTag*****/
  Author : George Kortsaridis
  Email  : gkortsaridis@gmail.com
  *****/

var request = require('sync-request');
var encodeUrl = require('encodeurl');

module.exports = {
  tag : function(language, text, wrongs){
    text = this.removeNumbers(text);

    //Splitting the paragraph into sentences.
    var splited_sentences = this.split_sentences(text);
    return this.tag_en_stanford(text,splited_sentences,wrongs);
  },

  get_total_recorded_wrongs : function(wrongs){

    var cnt = 0;
    for (var p in wrongs) {
      if( wrongs.hasOwnProperty(p) ) {
        cnt = cnt + wrongs[p];
      }
    }
    return cnt;
  },

  get_empty_pos_json : function(){
    var json = { "CC": 0, "CD": 0, "DT": 0, "EX": 0, "FW": 0, "IN_": 0, "JJ": 0, "JJR": 0, "JJS": 0, "LS": 0, "MD": 0, "NN": 0, "NNS": 0, "NNP": 0, "NNPS": 0, "PDT": 0, "POS": 0, "PRP": 0, "PRP_": 0, "RB": 0, "RBR": 0, "RBS": 0, "RP": 0, "SYM": 0, "TO_": 0, "UH": 0, "VB": 0, "VBD": 0, "VBG": 0, "VBN": 0, "VBP": 0, "VBZ": 0, "WDT": 0, "WP": 0, "WP_": 0, "WRB": 0 };

    return json;
  },

  tag_en_stanford : function(paragraph,splited_sentences,wrongs){

    var http_response = new Object();
    http_response.paragraph = paragraph;

    var totalRecordedErrors = this.get_total_recorded_wrongs(wrongs);
    var empty_words_json_pos = this.get_empty_pos_json();
  },
}
```

```

var sentence_array = [];

for(var i=0; i<splited_sentences.length; i++){

    var text = splited_sentences[i].replace('\n/g', '');

    var link = encodeUrl('http://localhost:9000/?properties={"annotators": "pos",
"outputFormat": "json"}');
    var res = request('POST', link , { json : text });
    var tagged_stanford = JSON.parse(res.getBody()).sentences[0].tokens;

    //console.log("tagged -->" + JSON.stringify(tagged_stanford));
    //console.log("wrongs -->" + JSON.stringify(wrongs));
    var sentence_object = new Object();
    sentence_object.sentence = splited_sentences[i];
    var response = [];

    for(var j=0; j<tagged_stanford.length; j++){
        var word_element = new Object();
        if(tagged_stanford[j].pos != undefined){
            if(wrongs[tagged_stanford[j].pos] != undefined){
                empty_words_json_pos[tagged_stanford[j].pos]++;
            }
        }

        word_element['textpos'] = tagged_stanford[j].pos;
        word_element['word'] = tagged_stanford[j].originalText;

        response.push(word_element);
    }
    sentence_object.posTagging = response;

    sentence_array.push(sentence_object);
}

var paragraph_word_count = 0;
for (var p in empty_words_json_pos) {
    if( empty_words_json_pos.hasOwnProperty(p) ) {
        paragraph_word_count = paragraph_word_count +
empty_words_json_pos[p];
    }
}

var difficulty = 0;
for (var p in empty_words_json_pos) {
    if( empty_words_json_pos.hasOwnProperty(p) ) {
        difficulty = difficulty +
empty_words_json_pos[p]/paragraph_word_count * wrongs[p]/totalRecordedErrors;
    }
}
http_response.difficulty = difficulty*10;
http_response.sentences = sentence_array;

return http_response;
},

split_seneces : function(text){
    var splited = text.split(' ');
    return splited;
}

```

```

    },
    removeNumbers : function (text){
        for(var i=0; i<999; i++){
            text = text.replace("[+i+]", "");
        }
        return text;
    }
};

```

4.2.3.6 server.js

Ο κώδικας του αρχείου server.js ενεργοποιεί τις θύρες http επικοινωνίας του συστήματος. Με αυτόν τον τρόπο μπορούν οι εξωτερικές εφαρμογές (εφαρμογή κινητών Android και εφαρμογή υπολογιστών) να διαβάζουν και να στέλνουν δεδομένα στο σύστημα. Η επικοινωνία γίνεται μέσω της TCP πόρτας 3000, το οποίο μπορεί να αλλάξει. Οι θύρες http επικοινωνίας που ανοίγουν είναι οι εξής :

HTTP Θύρα	Τρόπος Επικοινωνίας	Είσοδος	Έξοδος
/	HTTP POST	<ul style="list-style-type: none"> • Γλώσσα Wikipedia • Κατηγορία Wikipedia 	<ul style="list-style-type: none"> • Πλήρως επεξεργασμένη παράγραφος
/tag/	HTTP POST	<ul style="list-style-type: none"> • Παράγραφος προς επεξεργασία • Πίνακας λαθών 	<ul style="list-style-type: none"> • Πλήρως επεξεργασμένη παράγραφος
/wrongs/	HTTP GET	-	<ul style="list-style-type: none"> • Πίνακας λαθών
/categories/	HTTP GET	-	<ul style="list-style-type: none"> • Πίνακας κατηγοριών

/scoreboard/	HTTP POST	<ul style="list-style-type: none"> • ID χρήστη 	<ul style="list-style-type: none"> • Πίνακας σκορ
/register/	HTTP POST	<ul style="list-style-type: none"> • Στοιχεία νέου χρήστη 	<ul style="list-style-type: none"> • Αποτέλεσμα εγγραφής
/login/	HTTP POST	<ul style="list-style-type: none"> • Στοιχεία εισόδου χρήστη 	<ul style="list-style-type: none"> • ID χρήστη ή μήνυμα μη επιτυχούς εισόδου
/game_end/	HTTP POST	<ul style="list-style-type: none"> • ID χρήστη • Σκόρ παιχνιδιού 	<ul style="list-style-type: none"> • Αποτέλεσμα εισαγωγής σκορ
/wiki/	HTTP POST	<ul style="list-style-type: none"> • Γλώσσα Wikipedia • Κατηγορία Wikipedia 	<ul style="list-style-type: none"> • Τυχαία παράγραφος της Wikipedia

Πίνακας 8 : HTTP θύρες εξυπηρετητή

Παρακάτω παρατίθεται συνοπτικά ο κώδικας του αρχείου server.js. Λόγω του μεγάλου του μεγέθους, ο πλήρης κώδικας μπορεί να βρεθεί στην ιστοσελίδα του github.

```

/*****BrainTag*****/
  Author : George Kortsaridis
  Email  : gkortsaridis@gmail.com
  *****/

//To persistently StanfordCoreNLPServer run the next command from inside stanford directory
//nohup java -mx4g -cp "*" edu.stanford.nlp.pipeline.StanfordCoreNLPServer &

const express = require('express');
var bodyParser = require('body-parser');
var mysql = require('mysql');

var config = require('./database_config.json');
var nlp_pos_tagging = require("./natural_language_processing.js");
var wiki = require("./wikipedia_page_crawler");

const app = express();
const port = 3000;

app.use(bodyParser.urlencoded({ extended: true }));
app.use(bodyParser.json());
app.use(function(req, res, next) {
  res.header("Access-Control-Allow-Origin", "*");
  res.header("Access-Control-Allow-Headers", "Origin, X-Requested-With, Content-Type, Accept");

```

```

    next();
  });

app.post('/', (request, response) => {
  console.log("Request : / , for Category : "+request.body.wiki_category);

  var category = request.body.wiki_category;
  var language = request.body.wiki_language;
  var wrongs = request.body.wrongs;

  var wiki_paragraphs = wiki.crawl(language,category);
  var paragraph_to_tag = wiki_paragraphs[getRandomInt(0, wiki_paragraphs.length-1)];

  var tagged = nlp_pos_tagging.tag(language, paragraph_to_tag, wrongs);

  response.send(tagged);
});

app.post('/tag/', (request, response) => {
  var par = request.body.paragraph;
  var wrongs = request.body.wrongs;

  console.log("Request : /tag/ for paragraph");
  var tagged = nlp_pos_tagging.tag("en", par, wrongs);
  response.send(tagged);
});

app.get('/wrongs/', (request, response) => {

  var connection = mysql.createConnection(config);
  var wrongs_json;
  connection.connect();
  connection.query("SELECT * FROM wrong_answers", function (error, results, fields) {
    if (error) {
      throw error;
    } else {
      var wrongs_json = JSON.parse(JSON.stringify(results));
      response.status(200).send(wrongs_json[0]);
    }
  });
  connection.end();
});

app.get('/categories/', (request, response) => {
  console.log("Request : /categories/ ");

  var respo = new Object();

  var resp = new Array();

  var sports = new Object();
  sports.name = "Sports";
  var sportsCategories = new Array();
  sportsCategories.push("Sports");
  sportsCategories.push("Basketball");
  sportsCategories.push("Football");
  sportsCategories.push("Soccer");
  sportsCategories.push("Volleyball");

```



```

    sportsCategories.push("Tennis");
    sportsCategories.push("NBA");
    sports.categories = sportsCategories;
    resp.push(sports);

    respo.categories = resp;
    respo.category = "Sports";
    response.status(200).send(respo);
  });

app.post('/scoreboard/', (request, response) => {
  κώδικας διαχείρισης scoreboard
});

app.post('/register/', (request, response) => {
  κώδικας διαχείρισης εγγραφής νέου χρήστη
});

//Checked out
app.post('/login/', (request, response) => {
  κώδικας διαχείρισης εισόδου χρήστη
});

//Checked out
app.post('/game_end/', (request, response) => {
  κώδικας διαχείρισης τέλους παιχνιδιού
});

app.post('/wiki/', (request, response) => {
  κώδικας για την λήψη κειμένου από την ιστοσελίδα της wikipedia
});

app.listen(port, (err) => {
  if (err) {
    return console.log('something bad happened', err)
  }

  console.log(`server is listening on ${port}`)
});

function getRandomInt(min, max) {
  min = Math.ceil(min);
  max = Math.floor(max);
  return Math.floor(Math.random() * (max - min)) + min;
}

```

5. Εφαρμογή κινητών, λειτουργικού συστήματος Android

5.1 Κώδικας Εφαρμογής

Για την δημιουργία του κώδικα της κινητής εφαρμογής BrainTag του λειτουργικού συστήματος Android, χρησιμοποιήθηκε το ολοκληρωμένο προγραμματιστικό περιβάλλον Android Studio. Αυτή η επιλογή επιτάχυνε την προγραμματιστική διαδικασία αρκετά σε σχέση με άλλα προγραμματιστικά εργαλεία όπως το περιβάλλον Eclipse, γιατί προσφέρει αρκετές βοήθειες στον προγραμματιστή, όπως αυτόματη συμπλήρωση κειμένου (autocomplete), γρήγορος προσομοιωτής συσκευών Android (Android emulator) και ταχύτατη δημιουργία εκτελέσιμων προγραμμάτων (fast compiling). Παρακάτω παρατίθεται ολόκληρος ο κώδικας που χρησιμοποιήθηκε για την δημιουργία της εφαρμογής. Ο κώδικας του εξυπηρετητή φιλοξενείται στην ιστοσελίδα του github.com. Βρίσκεται διαθέσιμος σε όλους προς κατέβασμα, από την ιστοσελίδα

<https://github.com/gkortsaridis/BrainTag-Android>

Για αυτούς που προτιμούν την γραμμή εντολών και το git, ο κώδικας είναι διαθέσιμος με την εντολή

```
git clone https://github.com/gkortsaridis/BrainTag-Android.git
```

5.1.1 AndroidManifest.xml

Το αρχείο AndroidManifest.xml μπορεί να βρεθεί στην ιεραρχία φακέλων, στο μονοπάτι : app -> manifests -> AndroidManifest.xml. Είναι ένα αρκετά σημαντικό αρχείο μιας εφαρμογής Android. Δημιουργείται αυτόματα με την έναρξη του project κώδικα, αλλά είναι στην διάθεση του προγραμματιστή να το τροποποιήσει κατάλληλα. Οι βασικές πληροφορίες που

φέρει το συγκεκριμένο αρχείο ορίζουν τις άδειες της συσκευής που μπορεί να χρησιμοποιήσει η συγκεκριμένη εφαρμογή (permissions), όπως και την λίστα των activities που υπάρχουν στην εφαρμογή.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.gkortsaridis.braintag">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainActivity"
            android:configChanges="keyboardHidden|orientation|screenSize">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".GameActivity"
            android:configChanges="keyboardHidden|orientation|screenSize" />
        <activity
            android:name=".ParagraphReviewActivity"
            android:configChanges="keyboardHidden|orientation|screenSize" />
        <activity
            android:name=".GameResultActivity"
            android:configChanges="keyboardHidden|orientation|screenSize" />
        <activity
            android:name=".LoginActivity"
            android:configChanges="keyboardHidden|orientation|screenSize" />
        <activity android:name=".CategoriesActivity" />
        <activity android:name=".RegisterActivity" />
        <activity android:name=".TypeActivity"></activity>
    </application>

</manifest>
```

5.1.2 colors.xml & styles.xml

Τα αρχεία colors.xml και styles.xml μπορούν να βρεθούν στην ιεραρχία φακέλων, στο μονοπάτι app → res → values → colors.xml και app → res → values → styles.xml. Περιέχουν λεπτομέρειες για τα βασικά χρώματα τις εφαρμογής.

Colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#2196F3</color>
    <color name="colorPrimaryDark">#1976D2</color>
    <color name="colorAccent">#FFC107</color>

    <color name="white">#ffffff</color>
    <color name="black">#212121</color>
</resources>
```

Styles.xml

```
<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>
</resources>
```

5.1.3 strings.xml και dims.xml

Τα αρχεία strings.xml και dims.xml υπάρχουν στην ιεραρχία φακέλων, στο μονοπάτι app → res → values → strings.xml και app → res → values → dims.xml. Περιέχουν πληροφορίες για κάποιες σταθερές που χρησιμοποιούνται στο πρόγραμμα για κείμενο (strings.xml) και κάποια σταθερά μεγέθη (dims.xml)

strings.xml

```
<resources>
    <string name="app_name">BrainTag</string>
```

```

<string name="drawer_open">open</string>
<string name="drawer_close">close</string>

<string name="server_url">http://83.212.118.131:3000/</string>
<string name="newsapi_url">https://newsapi.org/v1/articles</string>
<string name="newsapi_apiKey">50d716cb112e4a5db187990b0c011b96</string>
</resources>

```

dimens.xml

```

<resources>
    <!-- Default screen margins, per the Android Design guidelines. -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
    <dimen name="toolbar_elevation">5dp</dimen>
    <dimen name="scroll_by">-100px</dimen>
</resources>

```

5.1.4 navigation_drawer_items.xml

Το αρχείο navigation_drawer_items.xml βρίσκεται στην ιεραρχία φακέλων, στο μονοπάτι app → res → menu → navigation_drawer_items.xml. Το συγκεκριμένο αρχείο ορίζει πόσες επιλογές θα υπάρχουν στο navigation drawer της εφαρμογής.

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <group android:checkableBehavior="single">
        <item android:icon="@drawable/play_game" android:id="@+id/navItem1"
android:title="Play Game" />
        <item android:icon="@drawable/trophy" android:id="@+id/navItem2"
android:title="Scoreboard" />
        <item android:icon="@drawable/info" android:id="@+id/navItem3" android:title="Info" />
        <item android:icon="@drawable/tag" android:id="@+id/navItem4" android:title="P.O.S.
Tag Description" />
        <item android:icon="@drawable/logout" android:id="@+id/navItem5" android:title="Log
Out" />
    </group>
</menu>

```

5.1.5 enter_from_left.xml , enter_from_right.xml , exit_to_left.xml , exit_to_right.xml

Τα αρχεία αυτά βρίσκονται στην ιεραρχία φακέλων, μέσα στον φάκελο app → res → anim. Είναι αρχεία τα οποία χρησιμοποιούνται για την εμφάνιση animations στην οθόνη. Τα συγκεκριμένα animations εκτελούνται όταν επιλέγεται ένα αντικείμενο του navigation drawer, ώστε να εμφανιστεί στην οθόνη με φυσικό τρόπο.

enter_from_left.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:shareInterpolator="false">
    <translate
        android:fromXDelta="-100%" android:toXDelta="0%"
        android:fromYDelta="0%" android:toYDelta="0%"
        android:duration="300"/>
</set>
```

enter_from_right.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:shareInterpolator="false">
    <translate
        android:fromXDelta="100%" android:toXDelta="0%"
        android:fromYDelta="0%" android:toYDelta="0%"
        android:duration="300" />
</set>
```

exit_to_left.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:shareInterpolator="false">
    <translate
        android:fromXDelta="0%" android:toXDelta="-100%"
        android:fromYDelta="0%" android:toYDelta="0%"
        android:duration="300"/>
</set>
```

exit_to_right.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:shareInterpolator="false">
    <translate
        android:fromXDelta="0%" android:toXDelta="100%"
        android:fromYDelta="0%" android:toYDelta="0%"
        android:duration="300" />
</set>
```

5.1.6 Helpers/CategoriesExpandableListViewAdapter.java

Το συγκεκριμένο αρχείο χρησιμεύει στην οπτικοποίηση μιας επεκτάσιμης λίστας και μπορεί να βρεθεί στην ιεραρχία φακέλων, στο μονοπάτι app → java → com.gkortsaridis.braintag → Helpers → CategoriesExpandableListViewAdapter.java. Δέχεται ως είσοδο μια λίστα με ονόματα κατηγοριών και υποκατηγοριών, με σκοπό να τα εμφανίσει στην οθόνη.

```
public class CategoriesExpandableListViewAdapter extends BaseExpandableListAdapter{

    private Context context;
    private List<String> expandableListTitle;
    private HashMap<String, List<String>> expandableListDetail;

    public CategoriesExpandableListViewAdapter(Context context, List<String>
expandableListTitle, HashMap<String, List<String>> expandableListDetail) {
        this.context = context;
        this.expandableListTitle = expandableListTitle;
        this.expandableListDetail = expandableListDetail;
    }

    @Override
    public Object getChild(int listPosition, int expandedListPosition) {
        return
this.expandableListDetail.get(this.expandableListTitle.get(listPosition)).get(expandedListPosi
tion);
    }

    @Override
    public long getChildId(int listPosition, int expandedListPosition) {
        return expandedListPosition;
    }

    @Override
    public View getChildView(int listPosition, final int expandedListPosition, boolean
isLastChild, View convertView, ViewGroup parent) {

        final String expandedListText = (String) getChild(listPosition, expandedListPosition);
        if (convertView == null) {
            LayoutInflater inflater = (LayoutInflater)
this.context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            convertView = inflater.inflate(R.layout.expandable_listview_item, null);
        }
        TextView expandedListTextView = (TextView)
convertView.findViewById(R.id.expandedListItem);
        expandedListTextView.setText(expandedListText);
        return convertView;
    }

    @Override
    public int getChildrenCount(int listPosition) {
        return
this.expandableListDetail.get(this.expandableListTitle.get(listPosition)).size();
    }
}
```

```

@Override
public Object getGroup(int listPosition) {
    return this.expandableListTitle.get(listPosition);
}

@Override
public int getGroupCount() {
    return this.expandableListTitle.size();
}

@Override
public long getGroupId(int listPosition) {
    return listPosition;
}

@Override
public View getGroupView(int listPosition, boolean isExpanded, View convertView, ViewGroup
parent) {
    String listTitle = (String) getGroup(listPosition);
    if (convertView == null) {
        LayoutInflater inflater = (LayoutInflater)
this.context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        convertView = inflater.inflate(R.layout.expandable_listview_group_header,
null);
    }
    TextView listTitleTextView = (TextView) convertView.findViewById(R.id.listTitle);
    listTitleTextView.setTypeface(null, Typeface.BOLD);
    listTitleTextView.setText(listTitle);
    return convertView;
}

@Override
public boolean hasStableIds() {
    return false;
}

@Override
public boolean isChildSelectable(int listPosition, int expandedListPosition) {
    return true;
}
}

```

5.1.7 Helpers/helper.java

Το αρχείο helper.java έχει συμπυκνωμένο κώδικα ο οποίος χρησιμοποιείται επανειλημμένα σε αρκετά αρχεία. Μπορεί να βρεθεί στην ιεραρχία φακέλων, στο μονοπάτι app → java → com.gkortsaridis.braintag → Helpers → helper.java. Παρακάτω παρατείνεται μια

σύντομη έκδοση του κώδικα με τις βασικές μεθόδους. Το ολοκληρωμένο αρχείο βρίσκεται στην ιστοσελίδα του Github.

```
public class Helper {

    private static ArrayList<String> shortTags;
    private static ArrayList<String> longTags;

    public static String getTag() {
        String tag = "";
        final StackTraceElement[] ste = Thread.currentThread().getStackTrace();
        for (int i = 0; i < ste.length; i++) {
            if (ste[i].getMethodName().equals("getTag")) {
                tag = "(" + ste[i + 1].getFileName() + ":" + ste[i + 1].getLineNumber() + ")";
            }
        }
        return tag;
    }

    public static String getRandomNewsSource(){
        String sources[] = {"techcrunch", "bbc-news", "google-news", "cnn", "national-geographic", "the-verge", "usa-today"};
        int rnd = new Random().nextInt(sources.length);
        return sources[rnd];
    }

    private static void getTags(){

        shortTags = new ArrayList<>();
        longTags = new ArrayList<>();

        shortTags.add("CC");
        προσθήκη των υπόλοιπων short tags στην λίστα

        longTags.add("Coordinating conjunction");
        προσθήκη των υπόλοιπων long tags στην λίστα
    }

    public static ArrayList<String> getShortTags(){
        getTags();
        return shortTags;
    }

    public static ArrayList<String> getLongTags(){
        getTags();
        return longTags;
    }

    public static String ordinal(int i) {
        String[] suffixes = new String[] { "th", "st", "nd", "rd", "th", "th", "th", "th", "th", "th" };
        switch (i % 100) {
            case 11:
            case 12:
            case 13:
                return i + "th";
            default:
                return i + suffixes[i % 10];
        }
    }
}
```

```

        return i + suffixes[i % 10];
    }
}

public static boolean determineWords(String word){
    ArrayList<String> unclickableWords = new ArrayList<>();
    unclickableWords.add("\");
    unclickableWords.add(",");
    unclickableWords.add(".");
    unclickableWords.add("'");
    unclickableWords.add(":");
    unclickableWords.add("?");
    unclickableWords.add("!");
    unclickableWords.add("(");
    unclickableWords.add(")");
    unclickableWords.add("[");
    unclickableWords.add("]");
    unclickableWords.add("{");
    unclickableWords.add("}");
    unclickableWords.add("-");
    unclickableWords.add("/");

    for(int i=0; i<unclickableWords.size(); i++){
        if(unclickableWords.get(i).equals(word)) return true;
    }

    return false;
}

public static JSONArray reformatJson(JSONArray jsonArray){
    JSONArray jsonArray1 = jsonArray;
    try {
        JSONObject sentenceObj = jsonArray1.getJSONObject(0);
        //Log.i("Sentence",sentenceObj.getString("sentence"));

        JSONArray sentencePosTagging = sentenceObj.getJSONArray("posTagging");
        sentencePosTagging.remove(0);
        sentencePosTagging.remove(sentencePosTagging.length()-1);

        for(int i=0; i<sentencePosTagging.length(); i++) {
            String word = sentencePosTagging.getJSONObject(i).getString("word");
            String pos = sentencePosTagging.getJSONObject(i).getString("textpos");
            //Log.i("!", "Word : "+word+" pos : "+pos);
            if(pos.equals("")){
                sentencePosTagging.remove(i);
            }else if(word.equals("/")){
                sentencePosTagging.remove(i);
            }else if(word.equals("")){
                sentencePosTagging.remove(i);
            }else if(word.equals("\\")){
                sentencePosTagging.remove(i);
            }else if(word.equals("\\")){
                sentencePosTagging.remove(i);
            }else if(word.equals("'")){
                sentencePosTagging.remove(i);
            }else if(word.equals(".")){
                sentencePosTagging.remove(i);
            }else if(word.equals(";")){
                sentencePosTagging.remove(i);
            }else if(word.equals(":")){
                sentencePosTagging.remove(i);
            }
        }
    }
}

```

```

        sentencePosTagging.remove(i);
    }else if(word.equals("\\\\\\")){
        sentencePosTagging.remove(i);
    }else if(word.equals("-")){
        sentencePosTagging.remove(i);
    }
}

} catch (JSONException e) {
    e.printStackTrace();
}

return jsonArray1;
}

public static String getFullPosName(String shortName){
    getTags();

    for(int i=0; i<shortTags.size(); i++){
        if(shortTags.get(i).equals(shortName)){
            return longTags.get(i);
        }
    }

    return null;
}

public static String getShortPosName(String longName){
    getTags();

    for(int i=0; i<longTags.size(); i++){
        if(longTags.get(i).equals(longName)){
            return shortTags.get(i);
        }
    }

    return null;
}
}

```

5.1.8 Helpers/MyDragListener.java

Το αρχείο MyDragListener.java μπορεί να βρεθεί στην ιεραρχία φακέλων, στο μονοπάτι app → java → com.gkortsaridis.braintag → Helpers → MyDragListener.java. Εκτελείται όταν κάποια καρτέλα ,από την οθόνη του παιχνιδιού, κουνηθεί. Χρησιμεύει στην ομορφότερη γραφική απεικόνιση του παιχνιδιού και την καλύτερη εμπειρία χρήσης

```

public class MyDragListener implements View.OnDragListener {

    Context context;
    Drawable enterShape;

    public MyDragListener(Context context){
        this.context = context;
        enterShape = context.getResources().getDrawable(R.drawable.shape_droptarget);
    }

    @Override
    public boolean onDrag(View v, DragEvent event) {
        int action = event.getAction();
        switch (event.getAction()) {
            case DragEvent.ACTION_DRAG_STARTED:
                // do nothing
                break;
            case DragEvent.ACTION_DRAG_ENTERED:
                v.setBackgroundDrawable(enterShape);
                break;
            case DragEvent.ACTION_DRAG_EXITED:
                v.setBackgroundDrawable(null);
                break;
            case DragEvent.ACTION_DROP:
                // Dropped, reassign View to ViewGroup
                View view = (View) event.getLocalState();
                String pos = ((TextView)view.findViewById(R.id.answer)).getText().toString();

                if(((TextView)v.findViewById(R.id.pos)).getText().toString().equals("--")){
                    ((TextView)v.findViewById(R.id.pos)).setText(Helper.getShortPosName(pos));
                    ViewGroup owner = (ViewGroup) view.getParent();
                    owner.removeView(view);
                }

                break;
            case DragEvent.ACTION_DRAG_ENDED:
                v.setBackgroundDrawable(null);
            default:
                break;
        }
        return true;
    }
}

```

5.1.9 Helpers/MyTouchListener.java

Το αρχείο MyTouchListener.java μπορεί να βρεθεί στην ιεραρχία φακέλων, στο μονοπάτι app → java → com.gkortsaridis → Helpers → MyTouchListener.java. Σε συνδιασμό με το αρχείο

MyDragListener.java, χρησιμοποιούνται για την σωστή κύλιση των καρτελών κατά την διάρκεια του παιχνιδιού, και την ευχάριστη εμπειρία χρήσης της εφαρμογής.

```
public class MyTouchListener implements View.OnTouchListener {
    public boolean onTouch(View view, MotionEvent motionEvent) {
        if (motionEvent.getAction() == MotionEvent.ACTION_DOWN) {
            ClipData data = ClipData.newPlainText("", "");
            View.DragShadowBuilder shadowBuilder = new View.DragShadowBuilder(view);
            view.startDrag(data, shadowBuilder, view, 0);
            return true;
        } else {
            view.setVisibility(View.VISIBLE);
            return false;
        }
    }
}
```

5.1.10 Helpers/PennPosTagAdapter.java

Το αρχείο PennPosTagAdapter.java μπορεί να βρεθεί στην ιεραρχία φακέλων, στο μονοπάτι app → java → com.gkortsaridis.braintag → Helpers → PennPosTagAdapter.java. Δέχεται ως είσοδο δύο λίστες κειμένου με την συντομογραφία και το πλήρες όνομα των ετικετών συντακτικού PennPos. Ως έξοδο, δημιουργεί ένα πίνακα με την δυνατότητα κύλισης (scroll) για την πληροφόρηση του χρήστη και την εκμάθηση των ετικετών.

```
public class PennPosTagAdapter extends BaseAdapter {

    private static LayoutInflater inflater=null;
    Context context;
    ArrayList<String> shortNames, longNames;

    public PennPosTagAdapter(Context context, ArrayList<String> shortNames, ArrayList<String> longNames){
        this.context = context;
        this.shortNames = shortNames;
        this.longNames = longNames;

        inflater =
        (LayoutInflater )context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    }

    @Override
```

```

    public int getCount() {
        return shortNames.size();
    }

    @Override
    public Object getItem(int i) {
        return i;
    }

    @Override
    public long getItemId(int i) {
        return i;
    }

    class ListItem{
        TextView tag,description;
    }

    @Override
    public View getView(int i, View view, ViewGroup viewGroup) {
        ListItem listItem = new ListItem();
        View rowView = inflater.inflate(R.layout.listview_item_tag_description, null);

        listItem.tag = (TextView) rowView.findViewById(R.id.tagTV);
        listItem.description = (TextView) rowView.findViewById(R.id.descriptionTV);

        listItem.description.setText(longNames.get(i));
        listItem.tag.setText(shortNames.get(i));

        return rowView;
    }
}

```

5.1.11 Helpers/PredicateLayout.java

Το αρχείο PredicateLayout.java μπορεί να βρεθεί στην ιεραρχία φακέλων, στο μονοπάτι app → java → com.gkortsaridis.braintag → Helpers → PredicateLayout.java. Αυτό το αρχείο εξυπηρετεί σε ένα μοναδικό σκοπό. Τοποθετεί τις κάρτες στην οθόνη του παιχνιδιού οριζόντια στην οθόνη, και όταν το οριζόντιο μήκος δεν είναι αρκετό αποφασίζει την αλλαγή γραμμής.

```

public class PredicateLayout extends ViewGroup {

    private int line_height;

    public static class LayoutParams extends ViewGroup.LayoutParams {
        public final int horizontal_spacing;
        public final int vertical_spacing;
    }
}

```

```

    /**
     * @param horizontal_spacing Pixels between items, horizontally
     * @param vertical_spacing Pixels between items, vertically
     */
    public LayoutParams(int horizontal_spacing, int vertical_spacing) {
        super(0, 0);
        this.horizontal_spacing = horizontal_spacing;
        this.vertical_spacing = vertical_spacing;
    }
}

public PredicateLayout(Context context) {
    super(context);
}

public PredicateLayout(Context context, AttributeSet attrs){
    super(context, attrs);
}

@Override
protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {
    assert(MeasureSpec.getMode(widthMeasureSpec) != MeasureSpec.UNSPECIFIED);

    final int width = MeasureSpec.getSize(widthMeasureSpec) - getPaddingLeft() -
getPaddingRight();
    int height = MeasureSpec.getSize(heightMeasureSpec) - getPaddingTop() -
getPaddingBottom();
    final int count = getChildCount();
    int line_height = 0;

    int xpos = getPaddingLeft();
    int ypos = getPaddingTop();

    for (int i = 0; i < count; i++) {
        final View child = getChildAt(i);
        if (child.getVisibility() != GONE) {
            final LayoutParams lp = (LayoutParams) child.getLayoutParams();
            child.measure(
                MeasureSpec.makeMeasureSpec(width, MeasureSpec.AT_MOST),
                MeasureSpec.makeMeasureSpec(height, MeasureSpec.UNSPECIFIED));

            final int childw = child.getMeasuredWidth();
            line_height = Math.max(line_height, child.getMeasuredHeight() +
lp.vertical_spacing);

            if (xpos + childw > width) {
                xpos = getPaddingLeft();
                ypos += line_height;
            }

            xpos += childw + lp.horizontal_spacing;
        }
    }
    this.line_height = line_height;

    if (MeasureSpec.getMode(heightMeasureSpec) == MeasureSpec.UNSPECIFIED){
        height = ypos + line_height;
    }else if (MeasureSpec.getMode(heightMeasureSpec) == MeasureSpec.AT_MOST){
        if (ypos + line_height < height){

```

```

        height = ypos + line_height;
    }
    }
    setMeasuredDimension(width, height);
}

@Override
protected ViewGroup.LayoutParams generateDefaultLayoutParams() {
    return new LayoutParams(1, 1); // default of 1px spacing
}

@Override
protected boolean checkLayoutParams(ViewGroup.LayoutParams p) {
    if (p instanceof LayoutParams) {
        return true;
    }
    return false;
}

@Override
protected void onLayout(boolean changed, int l, int t, int r, int b) {
    final int count = getChildCount();
    final int width = r - l;
    int xpos = getPaddingLeft();
    int ypos = getPaddingTop();

    for (int i = 0; i < count; i++) {
        final View child = getChildAt(i);
        if (child.getVisibility() != GONE) {
            final int childw = child.getMeasuredWidth();
            final int childh = child.getMeasuredHeight();
            final LayoutParams lp = (LayoutParams) child.getLayoutParams();
            if (xpos + childw > width) {
                xpos = getPaddingLeft();
                ypos += line_height;
            }
            child.layout(xpos, ypos, xpos + childw, ypos + childh);
            xpos += childw + lp.horizontal_spacing;
        }
    }
}
}
}

```

5.1.12 Helpers/ScoreboardAdapter.java

Το αρχείο ScoreboardAdapter.java μπορεί να βρεθεί στην ιεραρχία φακέλων, στο μονοπάτι app → java → com.gkortsaridis.braintag → Helpers → ScoreboardAdapter.java. Είναι αρχείο εμφάνισης συρόμενης λίστας. Το συγκεκριμένο δέχεται ως είσοδο δύο λίστες κειμένου με τα ονόματα των παικτών, και τα σκορ παιχνιδιού. Ως έξοδο εμφανίζει την λίστα των μεγαλύτερων σκόρερ του παιχνιδιού.


```

public class ScoreboardAdapter extends BaseAdapter {

    private static LayoutInflater inflater=null;
    Context context;
    ArrayList<String> usernames, scores;

    public ScoreboardAdapter(Context context, ArrayList<String> usernames, ArrayList<String>
scores){
        this.context = context;
        this.usernames = usernames;
        this.scores = scores;

        inflater =
( LayoutInflater )context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    }

    @Override
    public int getCount() {
        return usernames.size();
    }

    @Override
    public Object getItem(int i) {
        return i;
    }

    @Override
    public long getItemId(int i) {
        return i;
    }

    class ListItem{
        TextView username,score;
    }

    @Override
    public View getView(int i, View view, ViewGroup viewGroup) {
        ListItem listItem = new ListItem();
        View rowView = inflater.inflate(R.layout.listview_item_scoreboard, null);

        listItem.username = (TextView) rowView.findViewById(R.id.usernameTV);
        listItem.score = (TextView) rowView.findViewById(R.id.scoreTV);

        listItem.username.setText(usernames.get(i));
        listItem.score.setText(scores.get(i));

        return rowView;
    }
}

```

5.1.13 MainActivity.java & activity_main.xml

Τα αρχεία MainActivity.java και activity_main.xml δημιουργούν το MainActivity. Μπορούν να βρεθούν στην ιεραρχία φακέλων στα μονοπάτια app → java → com.gkortsaridis.braintag → MainActivity.java και app → res → layout → activity_main.xml αντίστοιχα. Είναι το πρώτο Activity που εκτελείται κατά την εκκίνηση της εφαρμογής, και χρησιμοποιείται για να στεγάσει τον κώδικα της αρχικής οθόνης, της λίστας των μεγαλύτερων σκόρερ, των πληροφοριών, της λίστας των ετικετών PennPos, και της εξόδου. Λόγω του μεγάλου μεγέθους του αρχείου, παρακάτω παρατείνεται ένα ενδεικτικό κομμάτι του κώδικα. Ο πλήρης κώδικας μπορεί να βρεθεί στην ιστοσελίδα του Github.

MainActivity.java

```
public class MainActivity extends AppCompatActivity {

    Toolbar toolbar;
    DrawerLayout drawerLayout;
    String what;
    ActionBarDrawerToggle mDrawerToggle;
    NavigationView view;
    ArrayList<String> history;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        history = new ArrayList<>();

        this.drawerLayout = (DrawerLayout) findViewById(R.id.drawer_layout);
        view = (NavigationView) findViewById(R.id.navigation_view);

        toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        //Αρχικοποίηση mDrawerToggle
        mDrawerToggle = new ActionBarDrawerToggle(...);

        getSupportActionBar().setDisplayHomeAsUpEnabled(false);
        getSupportActionBar().setHomeButtonEnabled(true);

        view.getMenu().getItem(0).setChecked(true);
        MainActivity.this.what = "Play Game";
        setFragment("Play Game");

        //Αρχικοποίηση NavigationItemSelectedListener
        view.setNavigationItemSelectedListener(...)

        public void setFragment(String what) {
            FragmentTransaction ft = getSupportFragmentManager().beginTransaction();
```

```

        ft.setCustomAnimations(R.anim.enter_from_left, R.anim.exit_to_right,
        R.anim.enter_from_right, R.anim.exit_to_left);

        if (what.equals("Play Game")) {
            ft.replace(R.id.your_placeholder, new MainMenuFragment());
            view.setCheckedItem(R.id.navItem1);
        } else if (what.equals("Scoreboard")) {
            ft.replace(R.id.your_placeholder, new ScoreboardFragment());
            view.setCheckedItem(R.id.navItem2);
        } else if (what.equals("Info")) {
            ft.replace(R.id.your_placeholder, new InfoFragment());
            view.setCheckedItem(R.id.navItem3);
        } else if (what.equals("P.O.S. Tag Description")){
            ft.replace(R.id.your_placeholder, new PennPosTagListFragment());
            view.setCheckedItem(R.id.navItem4);
        } else if (what.equals("Log Out")){
            //Κώδικας για εμφάνιση μηνύματος εξόδου, και έξοδο από την εφαρμογή
        }

        ft.commit();
        getSupportActionBar().setTitle(what);
        history.add(what);
    }
}

```

activity_main.xml

```

<android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true">

    <LinearLayout
        android:id="@+id/content"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <android.support.v7.widget.Toolbar
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="@color/colorPrimary"
            android:id="@+id/toolbar"
            app:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"
            android:elevation="5dp"/>

        <FrameLayout
            android:id="@+id/your_placeholder"
            android:layout_width="match_parent"
            android:layout_height="match_parent">
        </FrameLayout>

    </LinearLayout>

    <android.support.design.widget.NavigationView
        android:id="@+id/navigation_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"

```

```

        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        app:headerLayout="@layout/nav_header"
        app:menu="@menu/navigation_drawer_items"/>

```

```
</android.support.v4.widget.DrawerLayout>
```

5.1.14 CategoriesActivity.java & activity_categories.xml

Τα αρχεία CategoriesActivity.java και activity_categories.xml δημιουργούν το CategoriesActivity. Μπορούν να βρεθούν στην ιεραρχία φακέλων, στο μονοπάτι app → java → com.gkortsaridis.braintag → CategoriesActivity.java και app → res → layout → activity_categories.xml. Το συγκεκριμένο activity, εμφανίζει μια λίστα με όλες τις κατηγορίες του παιχνιδιού, τις οποίες τις λαμβάνει από το ίντερνετ, μετά από επικοινωνία με τον κύριο κορμό της εφαρμογής.

CategoriesActivity.java

```

public class CategoriesActivity extends AppCompatActivity {

    ExpandableListView expandableListView;
    ExpandableListAdapter expandableListAdapter;
    List<String> expandableListTitle;
    HashMap<String, List<String>> expandableListDetail;
    Toolbar toolbar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_categories);

        toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        if (getSupportActionBar() != null){
            toolbar.setTitleTextColor(Color.WHITE);

            final Drawable upArrow = ContextCompat.getDrawable(getBaseContext(),
R.drawable.abc_ic_ab_back_material);
            upArrow.setColorFilter(ContextCompat.getColor(getBaseContext(), R.color.white),
PorterDuff.Mode.SRC_ATOP);

            getSupportActionBar().setHomeAsUpIndicator(upArrow);
            getSupportActionBar().setHomeButtonEnabled(true);
            getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        }

        expandableListView = (ExpandableListView) findViewById(R.id.expandableListView);

```

```

        getCategories(getBaseContext());
    }

    public void getCategories(final Context context) {
        final AsyncHttpClient client = new AsyncHttpClient();
        client.addHeader("Accept", "application/json");
        client.addHeader("Content-Type", "application/json");

        expandableListDetail = new HashMap<>();

        String url = context.getResources().getString(R.string.server_url)+"categories";
        client.get(context, url, new AsyncHttpResponseHandler() {

            @Override
            public void onSuccess(int statusCode, Header[] headers, final byte[] response) {

                try {
                    JSONObject resp = new JSONObject(new String(response));
                    JSONArray categories = resp.getJSONArray("categories");
                    for(int i=0; i<categories.length(); i++){
                        JSONObject parentCategory = categories.getJSONObject(i);
                        JSONArray childrenCategories =
parentCategory.getJSONArray("categories");
                        String parentCategoryName = parentCategory.getString("name");

                        List<String> childrenCategoriesList = new ArrayList<String>();
                        for(int j=0; j<childrenCategories.length(); j++){
                            childrenCategoriesList.add(childrenCategories.get(j).toString());
                        }
                        expandableListDetail.put(parentCategoryName, childrenCategoriesList);
                    }

                    displayCategories();

                    Log.i("RESPO",resp.toString());
                } catch (JSONException e) {
                    e.printStackTrace();
                }

            }

            @Override
            public void onFailure(int statusCode, Header[] headers, byte[] errorResponse,
Throwable e) {
                // called when response HTTP status is "4XX" (eg. 401, 403, 404)
                Log.i(helper.getTag(),"FAIL "+statusCode);
            }

        });
    }

    public void displayCategories(){
        expandableListTitle = new ArrayList<String>(expandableListDetail.keySet());
        expandableListAdapter = new CategoriesExpandableListAdapter(this,
expandableListTitle, expandableListDetail);
        expandableListView.setAdapter(expandableListAdapter);

        expandableListView.setOnChildClickListener(new
ExpandableListView.OnChildClickListener() {
            @Override

```

```

        public boolean onChildClick(ExpandableListView parent, View v, int groupPosition,
int childPosition, long id) {
            Intent intent = new Intent(CategoriesActivity.this,
ParagraphReviewActivity.class);
            intent.putExtra("fromCategory",true);

intent.putExtra("category",expandableListDetail.get(expandableListTitle.get(groupPosition)).ge
t(childPosition));
            startActivity(intent);

            return false;
        }
    });

    for(int i=0; i<expandableListDetail.size(); i++) {
        expandableListView.expandGroup(i);
    }
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            finish();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
}

```

activity_categories.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:id="@+id/activity_categories"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.gkortsaridis.braintag.CategoriesActivity">

    <android.support.v7.widget.Toolbar
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/toolbar"
        android:background="@color/colorPrimary"
        android:elevation="@dimen/toolbar_elevation">
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_centerVertical="true"
                android:textSize="20sp"
                android:text="Select Category"
                android:textColor="@color/white"/>

```

```

</RelativeLayout>
</android.support.v7.widget.Toolbar>

<ExpandableListView
    android:id="@+id/expandableListView"
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    android:indicatorLeft="?android:attr/expandableListPreferredItemIndicatorLeft"
    android:divider="@android:color/darker_gray"
    android:dividerHeight="0.5dp" />

</LinearLayout>

```

5.1.15 GameActivity.java & activity_game.xml

Τα αρχεία GameActivity.java και activity_game.xml συνθέτουν το GameActivity. Μπορούν να βρεθούν στην ιεραρχία φακέλων, στα μονοπάτια app → java → com.gkortsaridis.braintag → GameActivity.java και app → res → layout → activity_game.xml. Το συγκεκριμένο activity εκτελείται κατά την διαδικασία του παιχνιδιού στον χρήστη. Είναι υπεύθυνο για την σωστή εμφάνιση όλων των λέξεων και καρτελών στην οθόνη, όπως και για τις ενέργειες που εκτελούνται όταν ο χρήστης κυλάει μια ετικέτα σε μια λέξη.

```

GameActivity.java
public class GameActivity extends AppCompatActivity {

    PredicateLayout l1,l2;
    ArrayList<String> answers;
    ScrollView myScrollView;
    TextView sentenceCount;
    JSONArray jsonArray;
    int curSentence = 0;
    Handler h1 = new Handler();

    int wordCount = 0;
    int correctCount = 0;
    String paragraph;
    boolean isChecking;

    ArrayList<String> wrongs;
    boolean practice;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```

setContentView(R.layout.activity_game);

practice = getIntent().getExtras().getBoolean("practice");
paragraph = getIntent().getExtras().getString("paragraph");

answers = new ArrayList<>();
wrongs = new ArrayList<>();

l = (PredicateLayout) findViewById(R.id.predicate_layout);
l2 = (PredicateLayout) findViewById(R.id.predicate_layout_answers);
myScrollView = (ScrollView) findViewById(R.id.myscrollview);
sentenceCount = (TextView) findViewById(R.id.sentenceCount);

try {
    jsonArray = new JSONArray(paragraph);
    jsonArray = Helper.reformatJson(jsonArray);
    displayData(jsonArray, curSentence);
} catch (JSONException e) {
    e.printStackTrace();
}

}

public void nextSentence(View view){

    isChecked = true;

    if(curSentence < jsonArray.length()-1){
        checkAnswers();

        h1.postDelayed(new Runnable(){
            public void run(){
                //do something
                curSentence++;
                displayData(jsonArray, curSentence);
            }
        }, 3000);

    }else{
        checkAnswers();

        h1.postDelayed(new Runnable(){
            public void run(){
                Intent intent = new Intent(GameActivity.this, GameResultActivity.class);
                intent.putExtra("wordCount", wordCount);
                intent.putExtra("correctCount", correctCount);
                intent.putExtra("practice", practice);
                intent.putExtra("wrongs", wrongs.toString());
                startActivity(intent);
            }
        }, 3000);

    }

}

}

public void checkAnswers(){
    for(int i=0; i<l.getChildCount(); i++){

```



```

        View wordView = l.getChildAt(i);

        TextView theWordView = (TextView)wordView.findViewById(R.id.word);
        TextView posView = (TextView)wordView.findViewById(R.id.pos);
        TextView answerView = (TextView)wordView.findViewById(R.id.answer);
        if(posView != null){
            wordCount++;

            String pos = posView.getText().toString();
            String answer = answerView.getText().toString();

            if(!posView.getText().toString().equals(answerView.getText().toString())){
                theWordView.setTextColor(Color.RED);
                posView.setTextColor(Color.RED);
                if(!pos.equals("--")){
                    wrongs.add(answer);
                }
            }else{
                correctCount++;
                theWordView.setTextColor(Color.GREEN);
                posView.setTextColor(Color.GREEN);
            }
        }
    }
}

public void displayData(JSONArray posdata, int cnt){
    isChecking = false;
    l.removeAllViews();
    l2.removeAllViews();
    answers.clear();

    try {
        sentenceCount.setText("Sentence "+(cnt+1)+"/"+posdata.length());

        JSONObject sentenceObj = posdata.getJSONObject(cnt);
        Log.i(Helper.getTag(),"Sentence : "+sentenceObj.getString("sentence"));
        JSONArray sentencePosTagging = sentenceObj.getJSONArray("posTagging");
        for(int i=0; i<sentencePosTagging.length(); i++){
            JSONObject wordObj = sentencePosTagging.getJSONObject(i);
            String word = wordObj.getString("word");
            final String pos = wordObj.getString("textpos");

            if(Helper.determineWords(word)){
                View noClickWord = createNoClickWord(wordObj.getString("word"));
                l.addView(noClickWord);
            }else{
                final View sentenceWord = createSentenceWord(word,pos);
                sentenceWord.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View view) {
                        if(!isChecking) {
                            String tempPos = ((TextView)
sentenceWord.findViewById(R.id.pos)).getText().toString();
                            ((TextView) sentenceWord.findViewById(R.id.pos)).setText("--
");

                            if (!tempPos.equals("--")) {
                                View possibleAnswer = createPossibleAnswer(tempPos);

```

```

        l2.addView(possibleAnswer);
    }
}
});

l1.addView(sentenceWord);

answers.add(pos);
}

}

} catch (JSONException e) {
    e.printStackTrace();
}

addAllPossibleAnswers();
}

public void addAllPossibleAnswers(){
    Collections.shuffle(answers);

    for(int i=0; i<answers.size(); i++) {
        //Create Possible Answer, and add it to second list
        View possibleAnswer = createPossibleAnswer(answers.get(i));
        l2.addView(possibleAnswer);
    }

}

public View createNoClickWord(String word){

    FrameLayout container = new FrameLayout(this);
    final View inflatedLayout;

    inflatedLayout= getLayoutInflater().inflate(R.layout.single_word_unclickable, null,
false);
    ((TextView)inflatedLayout.findViewById(R.id.word)).setText(word);
    container.addView(inflatedLayout);

    return container;
}

public View createSentenceWord(String word, String pos){

    FrameLayout container = new FrameLayout(this);
    final View inflatedLayout;

    inflatedLayout= getLayoutInflater().inflate(R.layout.single_word_two, null, false);
    inflatedLayout.setOnDragListener(new MyDragListener(getBaseContext()));
    ((TextView)inflatedLayout.findViewById(R.id.word)).setText(word);
    ((TextView)inflatedLayout.findViewById(R.id.answer)).setText(pos);
    ((TextView)inflatedLayout.findViewById(R.id.pos)).setText("--");
    container.addView(inflatedLayout);

    return container;
}
}

```

```

public View createPossibleAnswer(String pos){

    View inflatedLayout= getLayoutInflater().inflate(R.layout.answer_item, null, false);
    FrameLayout container = new FrameLayout(getBaseContext());
    inflatedLayout.setOnTouchListener(new MyTouchListener());
    inflatedLayout.setOnDragListener(new View.OnDragListener() {
        @Override
        public boolean onDrag(View view, DragEvent dragEvent) {
            if(dragEvent.getAction() == DragEvent.ACTION_DRAG_STARTED) {
                myScrollView.scrollBy(0,
(int)getResources().getDimension(R.dimen.scroll_by));
            }
            return false;
        }
    });

    ((TextView)inflatedLayout.findViewById(R.id.answer)).setText(Helper.getFullPosName(pos));
    container.removeAllViews();
    container.addView(inflatedLayout);

    return container;
}
}

```

activity_game.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.gkortsaridis.braintag.GameActivity">

    <android.support.v7.widget.Toolbar
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/toolbar"
        android:background="@color/colorPrimary"
        android:elevation="@dimen/toolbar_elevation">
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_centerVertical="true"
                android:id="@+id/sentenceCount"
                android:textColor="#ffffff"
                android:textSize="20sp"
                android:textStyle="bold"/>

            <ImageButton
                android:layout_width="40dp"
                android:layout_height="40dp"
                android:background="@drawable/ic_navigate_next_white_24dp"

```

```

        android:onClick="nextSentence"
        android:layout_alignParentRight="true"
        android:layout_centerVertical="true"
        android:layout_marginRight="16dp"/>

    </RelativeLayout>

</android.support.v7.widget.Toolbar>

<ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/myscrollview">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <com.gkortsaridis.braintag.Helpers.PredicateLayout
            android:id="@+id/predicate_layout"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="16dp"/>

        <com.gkortsaridis.braintag.Helpers.PredicateLayout
            android:id="@+id/predicate_layout_answers"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="16dp"/>

    </LinearLayout>

</ScrollView>

</LinearLayout>

```

5.1.16 GameResultActivity.java & activity_game_result.xml

Τα αρχεία GameResultActivity.java και activity_game_result.xml συνθέτουν το GameResultActivity. Μπορούν να βρεθούν στην ιεραρχία φακέλων, στο μονοπάτι app → java → com.gkortsaridis.braintag → GameResultActivity.java και app → res → layout → activity_game_result.xml αντίστοιχα. Το συγκεκριμένο activity είναι υπεύθυνο για την εμφάνιση του τελικού σκόρ του κάθε παιχνιδιού, όπως και την αποστολή του στον κόρμο της εφαρμογής (back-end) για αποθήκευση.

GameResultActivity.java

```
public class GameResultActivity extends AppCompatActivity {

    TextView result;
    int wordCount, correctCount;
    String wrongs;
    ArrayList<String> wrongs_list;
    JSONObject wrongs_json;
    SharedPreferences sharedPreferences;
    Toolbar toolbar;
    boolean isPractice;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_game_result);

        toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        if (getSupportActionBar() != null){
            toolbar.setTitleTextColor(Color.WHITE);

            final Drawable upArrow = ContextCompat.getDrawable(getBaseContext(),
R.drawable.abc_ic_ab_back_material);
            upArrow.setColorFilter(ContextCompat.getColor(getBaseContext(), R.color.white),
PorterDuff.Mode.SRC_ATOP);

            getSupportActionBar().setHomeAsUpIndicator(upArrow);
            getSupportActionBar().setHomeButtonEnabled(true);
            getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        }

        isPractice = getIntent().getExtras().getBoolean("practice");
        sharedPreferences = getSharedPreferences("preferences", Context.MODE_PRIVATE);

        result = (TextView) findViewById(R.id.result);

        wordCount = getIntent().getExtras().getInt("wordCount");
        correctCount = getIntent().getExtras().getInt("correctCount");
        wrongs = getIntent().getExtras().getString("wrongs");

        result.setText("Correct : "+correctCount+" out of "+wordCount+" words");

        createWrongsJson(wrongs);
        if(!isPractice){
            postGameData();
        }else{
            Toast.makeText(getBaseContext(), "That was a great practice! We hope to see you
again!", Toast.LENGTH_LONG).show();
        }
    }

    public void postGameData(){

        JSONObject jsonHttpBody = new JSONObject();
        try {
            jsonHttpBody.put("user_id", new
JSONObject(sharedPreferences.getString("user", null)).getString("ID"));
```

```

        jsonHttpBody.put("score", correctCount);
        jsonHttpBody.put("wrong_answers", wrongs_json);

        String entityBody = jsonHttpBody.toString();

        AsyncHttpClient client = new AsyncHttpClient();
        client.addHeader("Accept", "application/json");
        client.addHeader("Content-Type", "application/json");
        setRequestedOrientation(getRequestedOrientation());

        StringEntity entity = new StringEntity(entityBody);
        client.post(getBaseContext(),
getResources().getString(R.string.server_url)+"game_end/", entity, "application/json", new
AsyncHttpResponseHandler() {
            ProgressDialog pd;

            @Override
            public void onStart() {
                // called before request is started
                pd = new ProgressDialog(GameResultActivity.this);
                pd.setTitle("Please Wait...");
                pd.setMessage("We are working our magic right now");
                pd.setCancelable(false);
                pd.show();
            }

            @Override
            public void onSuccess(int statusCode, Header[] headers, final byte[] response)
{
                pd.cancel();
                setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_UNSPECIFIED);
                String resp = new String(response);
            }

            @Override
            public void onFailure(int statusCode, Header[] headers, byte[] errorResponse,
Throwable e) {
                // called when response HTTP status is "4XX" (eg. 401, 403, 404)

                pd.cancel();
                Log.i(Helper.getTag(), "FAIL "+statusCode);
                setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_UNSPECIFIED);
            }
        });

    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    } catch (JSONException e) {
        e.printStackTrace();
    }
}

public void createWrongsJson(String wrongs){

    wrongs = wrongs.replace("[", "");
    wrongs = wrongs.replace("]", "");
    String[] strValues = wrongs.split(",");

    wrongs_list = new ArrayList<String>(Arrays.asList(strValues));
    wrongs_json = new JSONObject();

```

```

    try {
        ArrayList<String> shortTags = Helper.getShortTags();
        for(int i=0; i<shortTags.size(); i++){
            wrongs_json.put(shortTags.get(i),0);
        }

        for(int i=0; i<wrongs_list.size(); i++){
            String cur_error = wrongs_list.get(i).trim();
            if(wrongs_json.has(cur_error)) {
                int new_val = wrongs_json.getInt(cur_error) + 1;

                wrongs_json.remove(cur_error);
                wrongs_json.put(cur_error, new_val);
            }
        }

    } catch (JSONException e) {
        e.printStackTrace();
    }
}

@Override
public void onBackPressed() {
    Intent intent = new Intent(this, MainActivity.class);
    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    startActivity(intent);
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            // app icon in action bar clicked; go home
            Intent intent = new Intent(this, MainActivity.class);
            intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(intent);
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
}

```

activity_game_result.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:wave="http://schemas.android.com/apk/res-auto"
    android:id="@+id/activity_main_meny"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.v7.widget.Toolbar
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="@color/colorPrimary"
        android:id="@+id/toolbar"
        android:elevation="@dimen/toolbar_elevation">

```

```

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Results"
        android:textStyle="bold"
        android:textSize="20sp"
        android:textColor="@color/white"
        android:layout_centerVertical="true"/>

    </RelativeLayout>
</android.support.v7.widget.Toolbar>

<ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginBottom="120dp"
    android:layout_below="@+id/toolbar">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <android.support.v7.widget.CardView
            android:layout_width="match_parent"
            android:layout_height="200dp"
            android:elevation="@dimen/toolbar_elevation"
            android:layout_margin="16dp"
            android:layout_below="@+id/appicon">

            <RelativeLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent">

                <TextView
                    android:text="TextView"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_centerVertical="true"
                    android:layout_centerHorizontal="true"
                    android:textSize="20sp"
                    android:textStyle="bold"
                    android:id="@+id/result" />

                </RelativeLayout>
            </android.support.v7.widget.CardView>
        </RelativeLayout>
    </ScrollView>
</RelativeLayout>

```


5.1.17 LoginActivity.java & activity_login.xml

Τα αρχεία LoginActivity.java και activity_login.xml συνθέτουν το LoginActivity. Μπορούν να βρεθούν στην ιεραρχία φακέλων, στο μονοπάτι app → java → com.gkortsaridis.braintag → LoginActivity.java και app → res → layout → activity_login.xml αντίστοιχα. Το συγκεκριμένο activity εμφανίζεται μόνο κατά την πρώτη εκκίνηση της εφαρμογής από τον χρήστη, ζητώντας την είσοδο των στοιχείων εισόδου ή προτρέποντας τον να εγγραφεί στην υπηρεσία.

```
LoginActivity.java
public class LoginActivity extends AppCompatActivity {

    EditText username, password;
    SharedPreferences sharedPreferences;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        username = (EditText) findViewById(R.id.username_ET);
        password = (EditText) findViewById(R.id.password_ET);
    }

    public void login(View view){

        JSONObject jsonHttpBody = new JSONObject();
        try {
            jsonHttpBody.put("username",username.getText().toString());
            jsonHttpBody.put("password",password.getText().toString());

            String entityBody = jsonHttpBody.toString();

            AsyncHttpClient client = new AsyncHttpClient();
            client.addHeader("Accept", "application/json");
            client.addHeader("Content-Type", "application/json");
            setRequestedOrientation(getRequestedOrientation());

            StringEntity entity = new StringEntity(entityBody);
            client.post(getBaseContext(),
getResources().getString(R.string.server_url)+"login/", entity, "application/json", new
AsyncHttpResponseHandler() {
                ProgressDialog pd;

                @Override
                public void onStart() {
                    // called before request is started
                    pd = new ProgressDialog(LoginActivity.this);
                    pd.setTitle("Please Wait...");
                    pd.setMessage("We are logging you in!");
                    pd.setCancelable(false);
```

```

        pd.show();
    }

    @Override
    public void onSuccess(int statusCode, Header[] headers, final byte[] response)
    {
        pd.cancel();
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_UNSPECIFIED);
        String resp = new String(response);
        sharedPreferences = getSharedPreferences("preferences",
Context.MODE_PRIVATE);
        SharedPreferences.Editor editor = sharedPreferences.edit();
        editor.putString("user", resp);
        editor.commit();

        Intent intent = new Intent(LoginActivity.this, MainActivity.class);
        startActivity(intent);
    }

    @Override
    public void onFailure(int statusCode, Header[] headers, byte[] errorResponse,
Throwable e) {
        // called when response HTTP status is "4XX" (eg. 401, 403, 404)

        final android.support.v7.app.AlertDialog.Builder alertDialogBuilder = new
android.support.v7.app.AlertDialog.Builder(LoginActivity.this);
        alertDialogBuilder.setTitle("Log In Error");
        alertDialogBuilder.setMessage("No user found with these credentials");
        alertDialogBuilder.setNeutralButton("OK", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
            }
        });
        alertDialogBuilder.create();
        alertDialogBuilder.show();

        pd.cancel();
        Log.i(Helper.getTag(), "FAIL "+statusCode);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_UNSPECIFIED);
    }
    });

    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    } catch (JSONException e) {
        e.printStackTrace();
    }
}
}
}

```

activity_login.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:wave="http://schemas.android.com/apk/res-auto"
    android:id="@+id/activity_main_meny"

```

```

android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.gkortsaridis.braintag.MainMenuFragmentt">

<android.support.v7.widget.Toolbar
    android:layout_width="match_parent"
    android:layout_height="?attr/actionBarSize"
    android:background="@color/colorPrimary"
    android:id="@+id/toolbar"
    android:elevation="@dimen/toolbar_elevation">
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="NLP Thesis"
            android:textStyle="bold"
            android:textSize="20sp"
            android:textColor="@color/white"
            android:layout_centerInParent="true"/>

    </RelativeLayout>
</android.support.v7.widget.Toolbar>

<ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@+id/toolbar">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <ImageView
            android:layout_width="200dp"
            android:layout_height="200dp"
            android:src="@mipmap/ic_launcher"
            android:layout_centerHorizontal="true"
            android:id="@+id/appicon"/>

        <android.support.v7.widget.CardView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:elevation="@dimen/toolbar_elevation"
            android:layout_margin="16dp"
            android:layout_below="@+id/appicon"
            android:id="@+id/cardview">

            <LinearLayout
                android:orientation="vertical"
                android:layout_width="match_parent"
                android:layout_height="match_parent">

                <android.support.design.widget.TextInputLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_margin="10dp">

                    <EditText

```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Username"
        android:id="@+id/username_ET"
        android:inputType="text"
        android:maxLines="1"/>

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:inputType="textPassword"
        android:id="@+id/password_ET"
        android:maxLines="1"/>

</android.support.design.widget.TextInputLayout>

<Button
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:text="Log In"
    android:textAllCaps="false"
    android:background="@color/colorPrimary"
    android:onClick="login"
    android:textColor="@color/white"
    android:layout_alignParentBottom="true"
    android:id="@+id/chooseCategory"/>

</LinearLayout>
</android.support.v7.widget.CardView>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="...OR..."
    android:textSize="20sp"
    android:layout_centerHorizontal="true"
    android:id="@+id/or"
    android:layout_below="@+id/cardview"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Register to play"
    android:textSize="20sp"
    android:layout_margin="16dp"
    android:layout_centerHorizontal="true"
    android:id="@+id/registerTV"
    android:layout_below="@+id/or"/>

</RelativeLayout>
</ScrollView>

```

```
</RelativeLayout>
```

5.1.18 ParagraphReviewActivity.java & activity_paragraph_review.xml

Τα αρχεία ParagraphReview.java και activity_paragraph_review.xml συνθέτουν το ParagraphReviewActivity. Μπορούν να βρεθούν στην ιεραρχία των φακέλων, στο μονοπάτι app → java → com.gkortsaridis.braintag → ParagraphReviewActivity.java και app → res → layout → activity_paragraph_review.xml αντίστοιχα. Το συγκεκριμένο activity, εμφανίζει συνολικά την παράγραφο του παιχνιδιού, πριν ξεκινήσει η διαδικασία ελέγχου των γνώσεων. Με αυτόν τον τρόπο, ο χρήστης μπορεί να αξιολογήσει την παράγραφο μόνος του, και αν προτιμάει να λάβει άλλη παράγραφο για να εξεταστεί στο παιχνίδι. Λόγω του μεγάλου μεγέθους του αρχείου, παρακάτω παρατείνεται ένα ενδεικτικό κομμάτι του κώδικα. Ο πλήρης κώδικας μπορεί να βρεθεί στην ιστοσελίδα του Github.

```
ParagraphReviewActivity.java
public class ParagraphReviewActivity extends AppCompatActivity {

    TextView paragraphTV,difficultyTV,categoryTV;
    JSONArray sentenceJSONArray;
    Toolbar toolbar;

    boolean fromCategories;
    boolean practice;
    String preConfiguredCategory;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_paragraph_review);

        fromCategories = getIntent().getExtras().getBoolean("fromCategory");
        preConfiguredCategory = getIntent().getExtras().getString("category");

        toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        // add back arrow to toolbar
        if (getSupportActionBar() != null){
            toolbar.setTitleTextColor(Color.WHITE);

            final Drawable upArrow = ContextCompat.getDrawable(getBaseContext(),
R.drawable.abc_ic_ab_back_material);
            upArrow.setColorFilter(ContextCompat.getColor(getBaseContext(), R.color.white),
```

```

PorterDuff.Mode.SRC_ATOP);

        getSupportActionBar().setHomeAsUpIndicator(upArrow);
        getSupportActionBar().setHomeButtonEnabled(true);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }

    categoryTV = (TextView) findViewById(R.id.categoryTV);
    paragraphTV = (TextView) findViewById(R.id.paragraph);
    difficultyTV = (TextView) findViewById(R.id.difficultyTV);

    getData();
}

public void getData(){
    if(preConfiguredCategory.equals("_news_")){
        String news = Helper.getRandomNewsSource();
        getNews(news);
        practice = true;
    }else if(preConfiguredCategory.equals("_input_")){
        String input = getIntent().getExtras().getString("input");
        tagData(input);
        practice = true;
    }else{
        practice = false;
        getWikiParagraph();
    }
}

public void getNews(final String source){
    //Κώδικας για την λήψη αρχείου κειμένου τύπου json με τίτλους ειδήσεων.
}

public String beautify(String text){
    return text.replace('_', ' ');
}

public void tagData(final String parData){
    //κώδικας για την συντακτική ανάλυση κειμένου
}

public void getWikiParagraph(){
    //κώδικας για λήψη παραγράφου κειμένου από την wikipedia, και συντακτική ανάλυση του
    κειμένου
}

public void refreshData(View view){
    getData();
}

public void startGame(View view){
    Intent intent = new Intent(ParagraphReviewActivity.this, GameActivity.class);
    intent.putExtra("paragraph",sentenceJsonArray.toString());
    intent.putExtra("practice",practice);
    startActivity(intent);
}
}

activity_paragraph_review.xml
<?xml version="1.0" encoding="utf-8"?>

```

```

<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:id="@+id/activity_paragraph_review"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.gkortsaridis.braintag.ParagraphReviewActivity">

    <android.support.v7.widget.Toolbar
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="@color/colorPrimary"
        android:id="@+id/toolbar"
        android:elevation="@dimen/toolbar_elevation">
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_centerVertical="true"
                android:id="@+id/categoryTV"
                android:textSize="20sp"
                android:textColor="@color/white"/>

            <ImageButton
                android:layout_width="30dp"
                android:layout_height="30dp"
                android:background="@drawable/ic_autorenew_white_24dp"
                android:onClick="refreshData"
                android:layout_alignParentRight="true"
                android:layout_centerVertical="true"
                android:layout_marginRight="16dp"/>

        </RelativeLayout>
    </android.support.v7.widget.Toolbar>

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <Button
            android:layout_width="match_parent"
            android:layout_height="60dp"
            android:background="@color/colorPrimary"
            android:text="Looks Great! Let's Begin!"
            android:layout_alignParentBottom="true"
            android:textAllCaps="false"
            android:textColor="@color/white"
            android:id="@+id/startGameBtn"
            android:onClick="startGame"/>

        <ScrollView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginBottom="60dp">

            <LinearLayout
                android:layout_width="match_parent"

```

```

        android:layout_height="match_parent"
        android:orientation="vertical">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/difficultyTV"
            android:layout_marginTop="8dp"
            android:textSize="20sp"
            android:gravity="center"/>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/paragraph"
            android:layout_marginTop="8dp"
            android:layout_marginLeft="16dp"
            android:layout_marginRight="16dp"
            android:layout_marginBottom="16dp"
            android:paddingBottom="16dp"
            android:textSize="20sp"/>

    </LinearLayout>

</ScrollView>
</RelativeLayout>
</LinearLayout>

```

5.1.19 RegisterActivity.java & activity_register.xml

Τα αρχεία RegisterActivity.java και activity_register.xml συνθέτουν το Register Activity. Μπορούν να βρεθούν στην ιεραρχία φακέλων, στο μονοπάτι app → java → com.gkortsaridis.braintag → RegisterActivity.java και app → res → layout → activity_register.xml αντίστοιχα. Το συγκεκριμένο activity, βοηθάει τον χρήστη να κάνει εγγραφή στην υπηρεσία BrainTag. Εμφανίζει τα κατάλληλα πεδία προς συμπλήρωση, και στέλνει τα δεδομένα στην βάση δεδομένων προς αποθήκευση.

```

RegisterActivity.java
public class RegisterActivity extends AppCompatActivity {

    EditText username,password,email;

    @Override

```



```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_register);

    username = (EditText) findViewById(R.id.username_ET);
    password = (EditText) findViewById(R.id.password_ET);
    email = (EditText) findViewById(R.id.email_ET);
}

public void register(View view){

    JSONObject jsonHttpBody = new JSONObject();
    try {
        jsonHttpBody.put("username",username.getText().toString());
        jsonHttpBody.put("password",password.getText().toString());
        jsonHttpBody.put("email",email.getText().toString());

        String entityBody = jsonHttpBody.toString();

        AsyncHttpClient client = new AsyncHttpClient();
        client.addHeader("Accept", "application/json");
        client.addHeader("Content-Type", "application/json");
        setRequestedOrientation(getRequestedOrientation());

        StringEntity entity = new StringEntity(entityBody);
        client.post(getBaseContext(),
getResources().getString(R.string.server_url)+"register/", entity, "application/json", new
AsyncHttpResponseHandler() {
            ProgressDialog pd;

            @Override
            public void onStart() {
                // called before request is started
                pd = new ProgressDialog(RegisterActivity.this);
                pd.setTitle("Please Wait...");
                pd.setMessage("We are registering you");
                pd.setCancelable(false);
                pd.show();
            }

            @Override
            public void onSuccess(int statusCode, Header[] headers, final byte[] response)
{
                pd.cancel();
                setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_UNSPECIFIED);
                Toast.makeText(getBaseContext(),"You were successfully
registered.",Toast.LENGTH_SHORT).show();
                finish();
            }

            @Override
            public void onFailure(int statusCode, Header[] headers, byte[] errorResponse,
Throwable e) {
                // called when response HTTP status is "4XX" (eg. 401, 403, 404
                pd.cancel();
                Log.i(Helper.getTag(),"FAIL "+statusCode);
                setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_UNSPECIFIED);
            }
        });

    } catch (UnsupportedEncodingException | JSONException e) {

```

```

        e.printStackTrace();
    }

}
}

```

activity_register.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.v7.widget.Toolbar
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/toolbar"
        android:background="@color/colorPrimary"
        android:elevation="@dimen/toolbar_elevation">
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_centerVertical="true"
                android:textSize="20sp"
                android:text="Register"
                android:textColor="@color/white"/>

        </RelativeLayout>
    </android.support.v7.widget.Toolbar>

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <ImageView
                android:layout_width="200dp"
                android:layout_height="200dp"
                android:src="@mipmap/ic_launcher"
                android:layout_centerHorizontal="true"
                android:id="@+id/appicon"/>

            <android.support.v7.widget.CardView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:elevation="@dimen/toolbar_elevation"
                android:layout_margin="16dp"
                android:layout_below="@+id/appicon"
                android:id="@+id/cardview">

```

```

<LinearLayout
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.design.widget.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp">

        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Username"
            android:id="@+id/username_ET"
            android:inputType="text"
            android:maxLines="1"/>

    </android.support.design.widget.TextInputLayout>

    <android.support.design.widget.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp">

        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Password"
            android:inputType="textPassword"
            android:id="@+id/password_ET"
            android:maxLines="1"/>

    </android.support.design.widget.TextInputLayout>

    <android.support.design.widget.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp">

        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Email"
            android:id="@+id/email_ET"
            android:maxLines="1"/>

    </android.support.design.widget.TextInputLayout>

    <Button
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:text="Register"
        android:textAllCaps="false"
        android:background="@color/colorPrimary"
        android:onClick="register"
        android:textColor="@color/white" />

</LinearLayout>
</android.support.v7.widget.CardView>
</RelativeLayout>

```

```
</ScrollView>
</LinearLayout>
```

5.1.20 TypeActivity.java & activity_type.xml

Τα αρχεία TypeActivity.java και activity_type.xml συνθέτουν το Type Activity. Μπορούν να βρεθούν στην ιεραρχία φακέλων, στο μονοπάτι app → java → com.gkortsaridis.braintag → TypeActivity.java και app → res → layout → activity_type.xml. Το συγκεκριμένο activity, λειτουργεί ως είσοδος σε ένα νέο χαρακτηριστικό των εφαρμογών, το οποίο είναι η εκτέλεση του παιχνιδιού με κείμενο από τον χρήστη. Χρησιμεύει στο να εμφανίζει το πεδίο στον χρήστη για την είσοδο του κειμένου, και να το προωθεί στο Paragraph Review Activity.

TypeActivity.java

```
public class TypeActivity extends AppCompatActivity {

    EditText text;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_type);
        text = (EditText) findViewById(R.id.inputText);
    }

    public void startGame(View view){
        Intent intent = new Intent(this, ParagraphReviewActivity.class);
        intent.putExtra("fromCategory", true);
        intent.putExtra("category", "_input_");
        intent.putExtra("input", text.getText().toString());
        startActivity(intent);
    }
}
```

activity_type.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:wave="http://schemas.android.com/apk/res-auto"
    android:id="@+id/activity_main_meny"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.v7.widget.Toolbar
        android:layout_width="match_parent"
```

```

        android:layout_height="?attr/actionBarSize"
        android:background="@color/colorPrimary"
        android:id="@+id/toolbar"
        android:elevation="@dimen/toolbar_elevation">
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Results"
                android:textStyle="bold"
                android:textSize="20sp"
                android:textColor="@color/white"
                android:layout_centerInParent="true"/>

        </RelativeLayout>
    </android.support.v7.widget.Toolbar>

    <Button
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:background="@color/colorPrimary"
        android:text="Start Game"
        android:layout_alignParentBottom="true"
        android:textAllCaps="false"
        android:textColor="@color/white"
        android:id="@+id/startGameBtn"
        android:onClick="startGame"/>

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginBottom="60dp"
        android:layout_below="@+id/toolbar">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="Tired of playing on Wikipedia pages and current News? You can
practice with your own paragraph!"
                android:layout_margin="15dp"
                android:gravity="center_horizontal"
                android:id="@+id/text"/>

            <EditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:hint="Please type a paragraph."
                android:layout_margin="10dp"
                android:id="@+id/inputText"
                android:layout_below="@+id/text"/>

        </RelativeLayout>
    </ScrollView>

```

```
</RelativeLayout>
```

5.1.21 InfoFragment.java & fragment_info.xml

Τα αρχεία InfoFragment.java και φragment_info.xml συνθέτουν το Info Fragment. Μπορούν να βρεθούν στην ιεραρχία φακέλων, στο μονοπάτι app → java → com.gkortsaridis.braintag → InfoFragment.java και app → res → layout → fragment_info.xml. Το συγκεκριμένο fragment εμφανίζει στον χρήστη μια οθόνη με πληροφορίες για την εφαρμογή, και τον δημιουργό της.

InfoFragment.java

```
public class InfoFragment extends Fragment {
    public InfoFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_info, container, false);
        return view;
    }
}
```

fragment_info.xml

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.gkortsaridis.braintag.InfoFragment">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">

            <ImageView
                android:layout_width="200dp"
                android:layout_height="200dp"
                android:src="@drawable/tagme_logo_transp_letters_no_shadow"
                android:layout_gravity="center_horizontal"/>
        </LinearLayout>
    </ScrollView>
</LinearLayout>
```

```

<android.support.v7.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp">
    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/app_name"
            android:textSize="25sp"
            android:textStyle="bold"
            android:layout_gravity="center_horizontal"
            android:layout_margin="16dp"/>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="This app is developed as part of the THESIS statement of
George Kortsaridis.\nIt is supervised by Mr. Panayotis Sarigiannidis."
            android:layout_marginLeft="16dp"
            android:layout_marginRight="16dp"
            android:layout_marginBottom="16dp"
            android:gravity="center_horizontal"/>

    </LinearLayout>
</android.support.v7.widget.CardView>
</LinearLayout>
</ScrollView>
</LinearLayout>

```

5.1.22 MainMenuFragment.java & fragment_main_menu.xml

Τα αρχεία MainMenuFragment.java και fragment_main_menu.xml συνθέτουν το Main Menu Fragment. Μπορούν να βρεθούν στην ιεραρχία φακέλων, στο μονοπάτι app → java → com.gkortsaridis.braintag → MainMenuFragment.java και app → res → layout → fragment_main_menu.xml αντίστοιχα. Το συγκεκριμένο fragment, είναι υπεύθυνο για την εμφάνιση ολόκληρου του αρχικού μενού της εφαρμογής, συμπεριλαμβανομένων του σκόρ του χρήστη, και τα κουμπιά για την εκκίνηση του παιχνιδιού. Λόγω του μεγάλου μεγέθους του αρχείου, παρακάτω παρατείνεται ένα ενδεικτικό κομμάτι του κώδικα. Ο πλήρης κώδικας μπορεί να βρεθεί στην ιστοσελίδα του Github.

MainMenuFragment.java

```
public class MainMenuFragment extends Fragment {

    SharedPreferences sharedPreferences;
    String user;
    TextView myScoreTV, myRankTV;
    Button randomGame, chooseCategory, newsGrammar, typePractice;

    public MainMenuFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_main_menu, container, false);

        myScoreTV = (TextView) view.findViewById(R.id.myscore);
        myRankTV = (TextView) view.findViewById(R.id.myrank);
        randomGame = (Button) view.findViewById(R.id.playRandomGame);
        chooseCategory = (Button) view.findViewById(R.id.chooseCategory);
        newsGrammar = (Button) view.findViewById(R.id.newsGrammar);
        typePractice = (Button) view.findViewById(R.id.typePractice);

        newsGrammar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getActivity(), ParagraphReviewActivity.class);
                intent.putExtra("fromCategory", true);
                intent.putExtra("category", "_news_");
                startActivity(intent);
            }
        });

        typePractice.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getActivity(), TypeActivity.class);
                startActivity(intent);
            }
        });

        randomGame.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(getActivity(), ParagraphReviewActivity.class);
                intent.putExtra("fromCategory", false);
                intent.putExtra("category", "");
                startActivity(intent);
            }
        });

        chooseCategory.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(getActivity(), CategoriesActivity.class);
                startActivity(intent);
            }
        });
    }
}
```



```

        sharedPreferences = getActivity().getSharedPreferences("preferences",
Context.MODE_PRIVATE);
        user = sharedPreferences.getString("user", "");

        if(user.equals("")){
            Intent intent = new Intent(getActivity(), LoginActivity.class);
            startActivity(intent);
        }else{
            try {
                JSONObject userJson = new JSONObject(user);
                getScoreboard(userJson.getString("ID"));
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }

        return view;
    }

    public void getScoreboard(String id){
        //Κώδικας για την λήψη ολόκληρου του πίνακα των σκορ
    }
}

```

fragment_main_menu.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:wave="http://schemas.android.com/apk/res-auto"
    android:id="@+id/activity_main_meny"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="120dp"
        android:orientation="vertical"
        android:layout_alignParentBottom="true">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="60dp"
            android:layout_alignParentBottom="true">

            <View
                android:layout_width="1dp"
                android:layout_height="60dp"
                android:background="@color/white"
                android:layout_centerHorizontal="true"
                android:id="@+id/centerView1"/>

            <Button
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="News & Grammar"
                android:textAllCaps="false"
                android:background="@color/colorPrimaryDark"

```

```

        android:foreground="?android:attr/selectableItemBackground"
        android:textColor="@color/white"
        android:id="@+id/newsGrammar"
        android:layout_alignParentTop="true"
        android:layout_toEndOf="@+id/centerView1" />

<Button
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:text="Play Random Game"
    android:textAllCaps="false"
    android:background="@color/colorPrimaryDark"
    android:foreground="?android:attr/selectableItemBackground"
    android:textColor="@color/white"
    android:id="@+id/playRandomGame"
    android:layout_alignParentTop="true"
    android:layout_toStartOf="@+id/centerView1" />

</RelativeLayout>

<View
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:background="@color/white"/>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:layout_alignParentBottom="true">

    <View
        android:layout_width="1dp"
        android:layout_height="60dp"
        android:background="@color/white"
        android:layout_centerHorizontal="true"
        android:id="@+id/centerView"/>

    <Button
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="Type & Practice"
        android:textAllCaps="false"
        android:background="@color/colorPrimary"
        android:foreground="?android:attr/selectableItemBackground"
        android:textColor="@color/white"
        android:layout_toEndOf="@+id/centerView"
        android:id="@+id/typePractice"/>

    <Button
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:text="Choose Category"
        android:textAllCaps="false"
        android:background="@color/colorPrimary"
        android:foreground="?android:attr/selectableItemBackground"
        android:textColor="@color/white"
        android:layout_toStartOf="@+id/centerView"
        android:id="@+id/chooseCategory"/>

</RelativeLayout>

```

```

</LinearLayout>

<ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginBottom="120dp"
    android:layout_below="@+id/toolbar">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <ImageView
            android:layout_width="200dp"
            android:layout_height="200dp"
            android:src="@drawable/tagme_logo_transp_letters_no_shadow"
            android:layout_centerHorizontal="true"
            android:id="@+id/appicon"/>

    </RelativeLayout>
</ScrollView>
</RelativeLayout>

```

5.1.23 PenPosTagListFragment.java & fragment_pen_pos_tag_list.xml

Τα αρχεία PenPosTagListFragment.java και fragment_pen_pos_tag_list.xml συνθέτουν το PenPos Tag List Fragment. Μπορούν να βρεθούν στην ιεραρχία φακέλων, στο μονοπάτι app → java → com.gkortsaridis.braintag → PenPosTagListFragment.java και app → res → layout → fragment_pen_pos_tag_list.xml αντίστοιχα. Το συγκεκριμένο fragment, βοηθάει τον χρήστη, εμφανίζοντας του την λίστα με όλες τις ετικέτες της λίστας PenPos. Εμφανίζει την συντομογραφία της κάθε ετικέτας μαζί με το πλήρες όνομα σε ένα ολοκληρωμένο πίνακα, ώστε ο χρήστης να το επισκέπτεται σε κάθε ευκαιρία.

```

PenPosTagListFragment.java
public class PenPosTagListFragment extends Fragment {

    ListView listView;
    public PenPosTagListFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        // Inflate the layout for this fragment
    }
}

```

```

        View view = inflater.inflate(R.layout.fragment_penn_pos_tag_list, container, false);
        listView = (ListView) view.findViewById(R.id.listview);
        listView.setAdapter(new PennPosTagAdapter(getActivity(),
Helper.getShortTags(),Helper.getLongTags()));
        return view;
    }
}

fragment_pen_pos_tag_list.xml
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.gkortsaridis.braintag.PennPosTagListFragment">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Our app uses the worldwide system for word tagging.\nBellow, you will
find the descriptions for each tag."
        android:layout_margin="16dp"
        android:textStyle="bold"
        android:textSize="16sp"
        android:gravity="center_horizontal"/>

    <android.support.v7.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="10dp">

        <ListView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:id="@+id/listview"/>

    </android.support.v7.widget.CardView>

</LinearLayout>

```

5.1.24 ScoreboardFragment.java & fragment_scoreboard.xml

Τα αρχεία ScoreboardFragment.java και fragment_scoreboard.xml συνθέτουν το Scoreboard Fragment. Μπορούν να βρεθούν στην ιεραρχία φακέλων, στο μονοπάτι app → java → com.gkortsaridis.braintag -> ScoreboardFragment.java και app → res → layout → fragment_scoreboard.xml αντίστοιχα. Το συγκεκριμένο fragment εμφανίζει ολόκληρο τον πίνακα των σκόρ της υπηρεσίας, ώστε ο χρήστης να μπορεί να δει σε τι θέση βρίσκεται στην παγκόσμια κατάταξη σε σχέση με όλους τους υπόλοιπους παίκτες του παιχνιδιού.

ScoreboardFragment.java

```
public class ScoreboardFragment extends Fragment {

    ListView listView;
    ArrayList<String> usernames,scores;

    public ScoreboardFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_scoreboard, container, false);
        usernames = new ArrayList<>();
        scores = new ArrayList<>();
        listView = (ListView) view.findViewById(R.id.listview);
        getData();
        return view;
    }

    public void getData(){

        usernames.clear();
        scores.clear();

        JSONObject jsonHttpBody = new JSONObject();
        try {
            jsonHttpBody.put("ID","2");

            String entityBody = jsonHttpBody.toString();

            AsyncHttpClient client = new AsyncHttpClient();
            client.addHeader("Accept", "application/json");
            client.addHeader("Content-Type", "application/json");
            getActivity().setRequestedOrientation(getActivity().getRequestedOrientation());

            StringEntity entity = new StringEntity(entityBody);
            client.post(getContext(),
getResources().getString(R.string.server_url)+"scoreboard/", entity, "application/json", new
AsyncHttpResponseHandler() {
                ProgressDialog pd;

                @Override
                public void onStart() {
                    // called before request is started
                    pd = new ProgressDialog(getActivity());
                    pd.setTitle("Please Wait...");
                    pd.setMessage("We are retrieving the scoreboard!");
                    pd.setCancelable(false);
                    pd.show();
                }

                @Override
                public void onSuccess(int statusCode, Header[] headers, final byte[] response)
{
                    pd.cancel();
                }
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}
```

```

getActivity().setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_UNSPECIFIED);
    String resp = new String(response);
    try {
        JSONObject jsonObject = new JSONObject(resp);
        JSONArray scoreboard = jsonObject.getJSONArray("scoreboard");

        for(int i=0; i<scoreboard.length(); i++){
            usernames.add(scoreboard.getJSONObject(i).getString("Username"));
            scores.add(scoreboard.getJSONObject(i).getString("Score"));
        }

        listView.setAdapter(new ScoreboardAdapter(getContext(),
usernames,scores));

        } catch (JSONException e) {
            e.printStackTrace();
        }

    }

    @Override
    public void onFailure(int statusCode, Header[] headers, byte[] errorResponse,
Throwable e) {

        // called when response HTTP status is "4XX" (eg. 401, 403, 404)

        pd.cancel();
        Log.i(helper.getTag(),"FAIL "+statusCode);

getActivity().setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_UNSPECIFIED);
    }
    });

    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    } catch (JSONException e) {
        e.printStackTrace();
    }

}

}
}

```

fragment_scoreboard.xml

```

<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.gkortsaridis.braintag.ScoreboardFragment">

    <android.support.v7.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="16dp">
        <LinearLayout
            android:orientation="vertical"
            android:layout_width="match_parent"
            android:layout_height="match_parent">

```

```

        <ImageView
            android:layout_width="70dp"
            android:layout_height="70dp"
            android:background="@drawable/trophy"
            android:layout_gravity="center_horizontal"
            android:layout_marginTop="10dp"/>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Our greatest ever!"
            android:layout_gravity="center_horizontal"
            android:textSize="20sp"
            android:textStyle="bold"
            android:layout_margin="10dp"/>
    </LinearLayout>
</android.support.v7.widget.CardView>

<ListView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/listview"/>

</LinearLayout>

```

5.1.25 build.gradle (Module : app)

Το αρχείο build.gradle (Module : app) μπορεί να βρεθεί στην ιεραρχία φακέλων, στο μονοπάτι Gradle Scripts → build.gradle (Module:app). Περιέχει βασικές πληροφορίες για την εφαρμογή, όπως το ελάχιστο λειτουργικό σύστημα για να εγκατασταθεί η εφαρμογή, τον αριθμό έκδοσης, και όλες τις βιβλιοθήκες που χρειάζονται για την σωστή λειτουργία του παιχνιδιού.

```

apply plugin: 'com.android.application'

android {
    compileSdkVersion 25
    buildToolsVersion "25.0.2"
    defaultConfig {
        applicationId "com.gkortsaridis.syntaxgame"
        minSdkVersion 19
        targetSdkVersion 25
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {

```

```

        minifyEnabled false
        proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })

    compile 'com.android.support:appcompat-v7:25.2.0'
    compile 'com.loopj.android:android-async-http:1.4.9'
    compile 'com.android.support:cardview-v7:25.2.0'
    compile 'com.android.support:design:25.2.0'
    compile 'com.github.john990:WaveView:v0.9'
    compile 'com.android.support:support-v4:25.2.0'
    compile 'me.relex:circleindicator:1.2.2@aar'
}

```

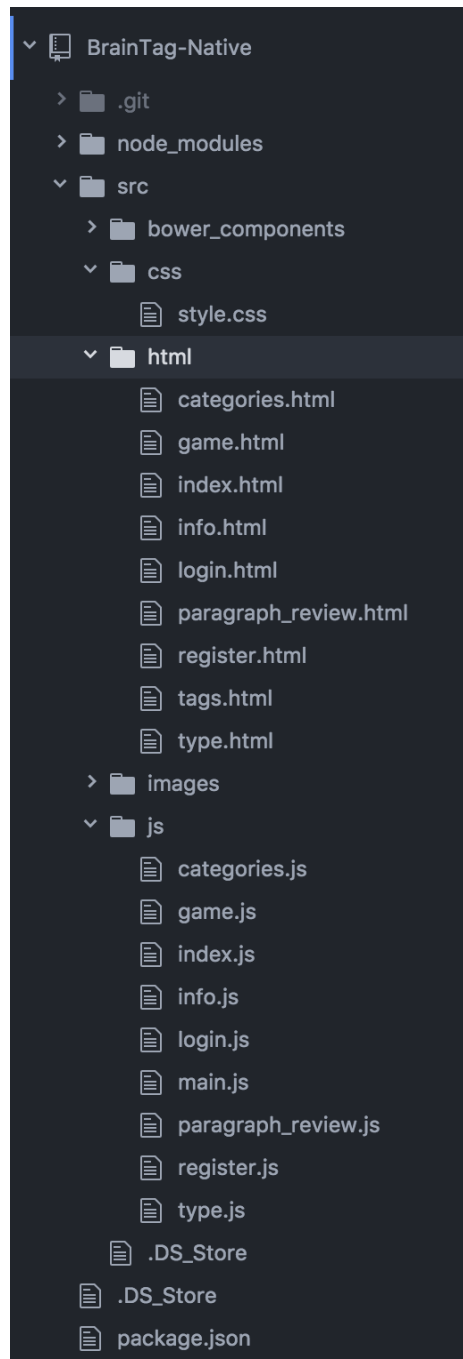

6. Εφαρμογή Υπολογιστών (OS Windows, OSX, Linux)

6.1 Παράθεση Κώδικα

Για την δημιουργία εφαρμογής που είναι ικανή να εκτελείται σε υπολογιστές λειτουργικών συστημάτων Windows, Apple OSX και Linux, χρησιμοποιήθηκε το προγραμματιστικό περιβάλλον Electron. Το Electron επιτρέπει την δημιουργία εφαρμογών υπολογιστών, μέσα από κώδικα που φυσιολογικά προορίζεται για διαδικτυακές εφαρμογές (δηλαδή μέσω γλωσσών όπως HTML, CSS και Javascript). Αυτό βοήθησε σε μεγάλο βαθμό την προγραμματιστική διαδικασία, διότι δεν χρειάστηκαν αλλαγές στον κώδικα για την δημιουργία των εφαρμογών των 3 λειτουργικών συστημάτων.

6.1.1 Project Structure

Για την εξοικείωση του προγραμματιστή με όλα τα αρχεία της πλατφόρμας, παρακάτω παρατίθεται η δομή του φακέλου, και εξηγείται η χρήση και η λειτουργία του κάθε αρχείου ξεχωριστά.



Εικόνα 32 : Native Application Project Structure

6.1.2 package.json

Το αρχείο package.json χρησιμοποιείται για την αρχικοποίηση των βασικών στοιχείων της εφαρμογής, όπως το όνομα, έκδοση, περιγραφή, και άλλα.

```
{
  "name": "braintag-native",
  "version": "1.0.0",
  "description": "",
  "main": "src/js/main.js",
  "scripts": {
    "start" : "electron ."
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "electron": "^1.4.15"
  }
}
```

6.1.3 css/style.css

Το προγραμματιστικό περιβάλλον Electron, χρησιμοποιεί γλώσσες προγραμματισμού διαδικτύου για τη δημιουργία των τελικών εφαρμογών. Ένα πολύ βασικό συστατικό στη σύσταση μιας εφαρμογής διαδικτύου ή Electron είναι τα γραφικά. Για να αποδοιούν χρώματα και γραφικές αλλαγές στην εφαρμογή χρειάζεται το εξής CSS αρχείο.

```
body {
  font-family: Roboto, sans-serif;
  color: #333;
  margin: 0;
}

p2{
  margin-left: 30px;
  color: white;
}
```

```

table.mytable{
    margin: 5px;
    padding: 10px;
    float: left;
}

td.myans{
    color: #9c9c9c;
}

paper-card{
    margin: 10px;
    padding: 20px;
}

div#answers{
    margin: 20px;
}

div#words{
    margin: 20px;
}

div#paragraph{
    font-size: large;
    padding: 20px;
}

td#btn{
    background-color: #1976D2;
}

td#btn:hover{
    background-color: #90CAF9;
}

td#btn:hover>p#btnTxt{
    color:black;
}

td#btn_down{
    background-color: #2196F3;
}

td#btn_down:hover{
    background-color: #90CAF9;
}

td#btn_down:hover>p#btnTxt{
    color:black;
}

p#btnTxt{
    text-align: center;
    line-height: 100%;
    color: white;
}

tr#toolbar{

```

```

        background-color: #2196F3;
        box-shadow: 0px 4px 5px #888888;
    }

    table#mypage{
        border:none;
        border-radius: 0px;
    }

    .loader {
        border: 16px solid #f3f3f3; /* Light grey */
        border-top: 16px solid #3498db; /* Blue */
        border-radius: 50%;
        width: 120px;
        height: 120px;
        animation: spin 2s linear infinite;
    }

    @keyframes spin {
        0% { transform: rotate(0deg); }
        100% { transform: rotate(360deg); }
    }

    div#categories{
        margin: 30px;
        -webkit-column-count: 3; /* Chrome, Safari, Opera */
        -moz-column-count: 3; /* Firefox */
        column-count: 3;
    }

    h2#categName{
        -webkit-column-span: all; /* Chrome, Safari, Opera */
        column-span: all;
    }

    li#category_item:hover{
        color: #3498db;
        font-size: 1.5em;
    }

    li#category_item{
        font-size: 1.4em;
    }

    table#tagtable>tr:hover{
        background-color:#f5f5f5
    }

```

6.1.4 html/categories.html & js/categories.js

Τα δύο αρχεία συνθέτουν την οθόνη των κατηγοριών. Σε αυτήν την οθόνη, γίνεται λήψη των κατηγοριών από τον κορμό της εφαρμογής, και στη συνέχεια εμφανίζονται ώστε να μπορεί ο χρήστης να επιλέξει κατηγορία παιχνιδιού.

html/categories.html

```
<html>
  <head>
    <script type="text/javascript" src="../../js/categories.js"></script>
    <link rel="stylesheet" href="../../css/style.css">
    <title>BrainTag</title>
  </head>
  <body>

    <table id="mypage" height="100%" width="100%" cellpadding="0">
      <tr id="toolbar" height="10%" width="100%">
        <td width="100%">
          <div style="margin-left: 20px">
            <a href="index.html"></a>
          </div>
        </td>
      </tr>
      <tr height="90%" width="100%">
        <td>
          <div style="overflow-y: scroll; top:8%;" id="categories"></div>
        </td>
      </tr>
    </table>
  </body>
</html>
```

js/categories.js

```
const remote = require('electron').remote;

window.onload = function(){
  getCategories();
};

function getCategories(){
  var http = new XMLHttpRequest();
  var url = "http://83.212.118.131:3000/categories";
  http.open("GET", url, true);

  document.getElementById("categories").innerHTML = "<div id='loader' class='loader'
style='display: block; margin: auto;'></div>";
  http.setRequestHeader("Content-type", "application/json");
  http.onreadystatechange = function() { //Call a function when the state changes.
    if(http.readyState == 4 && http.status == 200) {
      var json = JSON.parse(http.responseText);
      console.log(json);
    }
  }
}
```

```

        //<h4>B</h4><ul><li><a href="#">Billy</a></li><li><a href="#">Bob</a></li></ul>
        var myInnerHTML = "";
        for(var i=0; i<json.categories.length; i++){
            var category_name = json.categories[i].name;
            var cur_categories = json.categories[i].categories;
            myInnerHTML += "<div>";
            myInnerHTML += "<h2 id='categName' style='font-size:2em;'>" + beautify(category_name) + "</h2>";
            myInnerHTML += "<ul>";
            for(var j=0; j<cur_categories.length; j++){
                myInnerHTML += "<li id='category_item'
onclick=\"clickedcategory('"+cur_categories[j]+"')\">" + beautify(cur_categories[j]) + "</li>";
            }
            myInnerHTML += "</ul>";
            myInnerHTML += "</div>";

            document.getElementById("categories").innerHTML = myInnerHTML;
        }
    }
}
http.send();
}

function beautify(word){
    word = word.replace("_", " ");
    word = word.capitalizeFirstLetter();
    return word;
}

String.prototype.capitalizeFirstLetter = function() {
    return this.charAt(0).toUpperCase() + this.slice(1);
}

function clickedcategory(category){
    var url = `paragraph_review.html?data=` + encodeURIComponent(category);
    remote.getCurrentWindow().loadURL(`file://${__dirname}/../html/${url}`);
}

```

6.1.5 html/game.html & js/game.js

Τα δύο αρχεία συνθέτουν την οθόνη του παιχνιδιού. Εδώ ο χρήστης μπορεί με το ποντίκι του να σύρει τις ετικέτες συντακτικού, στις αντίστοιχες λέξεις. Λόγω του μεγάλου μεγέθους του αρχείου, παρακάτω παρατείνεται ένα ενδεικτικό κομμάτι του κώδικα. Ο πλήρης κώδικας μπορεί να βρεθεί στην ιστοσελίδα του Github.

```

html/game.html
<html>
    <head>
        <script type="text/javascript" src="../js/game.js"></script>

```

```

        <script src="../../bower_components/webcomponentsjs/webcomponents-
lite.min.js"></script>
        <link rel="import" href="../../bower_components/paper-card/paper-card.html">
        <link rel="stylesheet" href="../../css/style.css">

        <title>BrainTag</title>

    </head>
    <body>

        <table id="mypage" height="100%" width="100%" cellpadding="0">

            <tr id="toolbar" height="10%" width="100%">
                <td width="10%">
                    <a href="index.html"></a>
                </td>

                <td width="85%">
                    <table>
                        <tr><td> <p2 id="category_p">Category : Sports</p2> </td></tr>
                        <tr><td> <p2 id="sentence_p">Sentence : 1/5</p2>

                    </table>
                </td>

                <td width="5%">
                    
                </td>
            </tr>

            <tr><td colspan="3"> <div class="container" id="words"></div> </td></tr>
            <tr><td colspan="3"> <div class="container" id="answers"></div>

        </td></tr>
    </table>
</body>
</html>

```

js/game.js

```

const remote = require('electron').remote;

window.onload = function(){
    var url = window.location.href;
    var captured = /data=([^&]+)/.exec(url)[1];
    var result = captured ? captured : 'myDefaultValue';
    tagData = JSON.parse(decodeURIComponent(result));
    document.getElementById("category_p").innerHTML = "Category : "+tagData.category;

    processSentece(curSentence);
};

function processSentece(cnt){
    var sentence = tagData.sentences[cnt].postTagging;

    wordsArr = [];
    answersArr = [];

    document.getElementById("sentence_p").innerHTML = "Sentece :
" + (cnt+1) + "/" + tagData.sentences.length;

```



```

    }

    words = document.getElementById("words");
    answers = document.getElementById("answers");

    var wordsString = "";
    for(var i=0; i<wordsArr.length; i++){
        wordsString += wordName(i,wordsArr[i],answersArr[i]);
    }

    words.innerHTML = wordsString;

    answersArr = shuffle(answersArr);
    var answersString = "";
    for(var i=0; i<answersArr.length; i++){
        answersString += cardName(i,getTagLongName(answersArr[i]));
    }
    answers.innerHTML = answersString;
}

function cardName(cnt,name){
    return "<paper-card elevation=2 draggable='true' ondragstart='drag(event)'
class='mycardclass' id='"+cnt+"'><div id='content'+cnt+"'>"+name+"</div></paper-card>";
}

function wordName(cnt,name,correct){
    var str = "<table ondrop='drop(event)' onclick='wordClick(this.id)'
ondragover='allowDrop(event)' class='mytable' id='t_"+cnt+"' ><tr><td
id='w_"+cnt+"'>"+name+"</td></tr> <tr><td class='myans' id='a_"+cnt+"'>--</td></tr> <tr><td>
<p id='c_"+cnt+"' hidden>"+correct+"</p> </td></tr></table>";
    return str;
}

function nextSentence(){

    for(var i=0; i<wordsArr.length; i++){
        var word = document.getElementById("w_"+i).innerHTML;
        var answ = document.getElementById("a_"+i).innerHTML;
        var corr = document.getElementById("c_"+i).innerHTML;

        if(answ == corr){
            document.getElementById("w_"+i).style.color = '#CDDC39';
            document.getElementById("a_"+i).style.color = '#CDDC39';
            correctCount++;
        }else{
            document.getElementById("w_"+i).style.color = '#f44336';
            document.getElementById("a_"+i).style.color = '#f44336';
            wrongAnswers.push(corr);
        }
    }

    setTimeout(
function() {
    if(curSentence < tagData.sentences.length-1){
        curSentence++;
        processSentece(curSentence);
    }else{
        finishGame();
    }
}, 3000);

```

```

}

function finishGame(){
    console.log(wrongAnswers);

    var wrongs_json = new Object();

    if (typeof(Storage) !== "undefined") {
        //localStorage.setItem("userid", 6);
        var userid = localStorage.getItem("userid");

        for(var i=0; i<shortTags.length; i++){
            wrongs_json[shortTags[i]] = 0;
        }

        for(var i=0; i<wrongAnswers.length; i++){
            var cur_error = wrongAnswers[i];
            wrongs_json[cur_error] = wrongs_json[cur_error] + 1;
        }

        if(tagData.practice){
            alert("You scored "+correctCount+" point(s).\nThat was a great practice!
We hope to see you again!");
            //Κώδικας για την αποθήκευση και αποστολή του σκορ παιχνιδιού.
        } else {
            alert("Sorry, your browser does not support Local Storage files. Please upgrade your
browser.");
        }
    }
}

function getTagLongName(shortName){
    for(var i=0; i<shortTags.length; i++){
        if(shortName == shortTags[i]){
            return longTags[i];
        }
    }
}

function getTagShortName(longName){
    for(var i=0; i<longTags.length; i++){
        if(longName == longTags[i]){
            return shortTags[i];
        }
    }
}

function shuffle(a) {
    for (let i = a.length; i; i--) {
        let j = Math.floor(Math.random() * i);
        [a[i - 1], a[j]] = [a[j], a[i - 1]];
    }

    return a;
}

```

6.1.6 html/index.html & js/index.js

Τα δύο αρχεία συνθέτουν την αρχική οθόνη. Εδώ ο χρήστης μπορεί να δει το σκορ του, και να επιλέξει οποιαδήποτε άλλη ενέργεια του παιχνιδιού μέσω των κουμπιών που παρέχονται. Λόγω του μεγάλου μεγέθους του αρχείου, παρακάτω παρατίθεται ένα ενδεικτικό κομμάτι του κώδικα. Ο πλήρης κώδικας μπορεί να βρεθεί στην ιστοσελίδα του Github.

html/index.html

```
<html>
  <head>
    <script type="text/javascript" src="../js/index.js"></script>
    <link rel="import" href="../bower_components/paper-card/paper-card.html">
    <link rel="stylesheet" href="../css/style.css">
    <title>BrainTag</title>
  </head>
  <body>

    <table id="mypage" height="100%" width="100%" cellpadding="0">

      <tr id="toolbar" height="10%" width="100%">
        <td width="85%">
          <div style="margin-left: 20px"><h1 style="font-size:2vw;
color: white;display:table-cell; vertical-align:middle;">BrainTag</h1></div>
        </td>

        <td width="5%">
          <div style="margin-right: 20px"></div>
        </td>

        <td width="5%">
          <div style="margin-right: 20px"></div>
        </td>

        <td width="5%">
          <div style="margin-right: 20px"></div>
        </td>
      </tr>

      <tr height="35%" width="100%">
        <td colspan="4">
          
        </td>
      </tr>

      <tr height="35%" width="100%">
        <td colspan="4">
          <paper-card elevation=2 style="height: 80%; width: 40%; margin:
```

```

auto; display: block; padding: 0px;">
    <table width="100%" height="100%">
        <tr>
            <td width="50%">
                
            </td>

            <td width="50%">
                <h1 id="user_data" style="font-size:2vw; display: block; margin: auto;
vertical-align:middle; width : 100%; text-align: center">Your Score<br>75<br>Your
Ranking<br>3rd</h1>
            </td>
        </tr>
    </table>
</paper-card>
</td>

</tr>

    <tr height="10%">
        <td colspan="4" width="100%">

            <table width="100%" height="100%;" style="border-collapse:
collapse;">
                <tr>
                    <td id="btn" width="50%" style="border-
right: 2px solid white;" onclick="startRandomGame()">
                        <p id="btnTxt">Play Random Game</p>
                    </td>
                    <td id="btn" width="50%" style="border-
left: 2px solid white;" onclick="startNews()">
                        <p id="btnTxt">News & Grammar</p>
                    </td>
                </tr>
            </table>

        </td>
    </tr>

    <tr height="10%">
        <td colspan="4" width="100%">

            <table width="100%" height="100%;" style="border-collapse:
collapse;">
                <tr style="border-top: 2px solid white;">
                    <td id="btn_down" width="50%"
style="border-right: 2px solid white;" onclick="showCategories()">
                        <p id="btnTxt">Choose Category</p>
                    </td>
                    <td id="btn_down" width="50%"
style="border-left: 2px solid white;" onclick="startType()">
                        <p id="btnTxt">Type & Practice</p>
                    </td>
                </tr>
            </table>

        </td>
    </tr>
</table>
</body>
</html>

```

```

js/index/js
const remote = require('electron').remote;

window.onload = function(){

    if (typeof(Storage) !== "undefined") {
        if(localStorage.getItem("userid") == -1 || localStorage.getItem("userid") ==
undefined){ logout(); }
        else{ getData(); }
    }else{
        alert("Sorry, your browser does not support Local Storage files. Please upgrade your
browser.");
    }

};

function getData(){
    //Κώδικας για την λήψη δεδομένων παιχνιδιού
}

function startType(){
    remote.getCurrentWindow().loadURL(`file://${__dirname}/../html/type.html`);
}

function logout(){

    if (typeof(Storage) !== "undefined") {
        localStorage.setItem("userid",-1);
        var url = 'login.html';
        remote.getCurrentWindow().loadURL(`file://${__dirname}/../html/login.html`);
    }else{
        alert("Sorry, your browser does not support Local Storage files. Please upgrade your
browser.");
    }

}

function ordinal_suffix_of(i) {
    var j = i % 10,
        k = i % 100;
    if (j == 1 && k != 11) {
        return i + "st";
    }
    if (j == 2 && k != 12) {
        return i + "nd";
    }
    if (j == 3 && k != 13) {
        return i + "rd";
    }
    return i + "th";
}

```

6.1.7 html/info.html & js/info.js

Τα δύο αυτά αρχεία συνθέτουν την οθόνη πληροφοριών. Εμφανίζουν στον χρήστη πληροφορίες για τον δημιουργό της εφαρμογής, και για τις βιβλιοθήκες που βοήθησαν στο τελικό αποτέλεσμα.

html/info.html

```
<html>
  <head>
    <script type="text/javascript" src="../js/info.js"></script>
    <link rel="import" href="../bower_components/paper-card/paper-card.html">
    <link rel="stylesheet" href="../css/style.css">
    <title>BrainTag</title>
  </head>
  <body>

    <table id="mypage" height="100%" width="100%" cellpadding="0">

      <tr id="toolbar" height="10%" width="100%">
        <td width="100%">
          <a href="index.html"></a>
        </td>
      </tr>

      <tr height="25%" width="100%">
        <td >
          
        </td>
      </tr>

      <tr height="65%" width="100%">
        <td>
          <paper-card elevation=2 style="height: 70%; width: 80%;
margin: auto; display: block; padding: 0px;">
            <h4 style="margin: 20px; text-align:
center;">BrainTag<br>Developed by George Kortsaridis<br>Student of University of Western
Macedonia, dept. of Informatics & Telecommunications.<br>Supervised by Sarigiannidis
Panayotis</h4>
            <h4 style="margin: 20px; text-align: center;">This
project would not happen without</h4>
            <ul>
              <li> <a id="extLink1"
href="http://stanfordnlp.github.io/CoreNLP/index.html">Stanford Core NLP library</a> (Natural
Language Processing)
              <li> <a id="extLink2"
href="https://en.wikipedia.org/wiki/Main_Page">Wikipedia</a>, the free encyclopledia (Text
retrieval)
              <li> <a id="extLink3">
```

```

href="https://newsapi.org/">NewsApi.org</a> (News retrieval)
      </ul>
    </paper-card>
  </td>
</tr>
</table>
</body>
</html>

```

js/info.js

```

const remote = require('electron').remote;

window.onload = function(){

  var shell = require('electron').shell;
  //open links externally by default

  var link1 = document.getElementById("extLink1");
  var link2 = document.getElementById("extLink2");
  var link3 = document.getElementById("extLink3");

  link1.addEventListener("click", function(event) {
    event.preventDefault();
    shell.openExternal(this.href);
  });

  link2.addEventListener("click", function(event) {
    event.preventDefault();
    shell.openExternal(this.href);
  });

  link3.addEventListener("click", function(event) {
    event.preventDefault();
    shell.openExternal(this.href);
  });

};

```

6.1.8 html/login.html

Τα δύο αυτά αρχεία συνθέτουν την οθόνη εισόδου της εφαρμογής. Εδώ ο χρήστης χρησιμοποιεί το μοναδικό του όνομα χρήστη και κωδικό, ώστε να εισέλθει στην εφαρμογή.

html/login.html

```

<html>
  <head>
    <script type="text/javascript" src="../js/login.js"></script>
    <link rel="import" href="../../bower_components/paper-card/paper-card.html">

```

```

<link rel="import" href="../../bower_components/paper-input/paper-input.html">
<link rel="stylesheet" href="../../css/style.css">
<title>BrainTag</title>
</head>
<body>

    <table id="mypage" height="100%" width="100%" cellspacing="0" cellpadding="0">

        <tr id="toolbar" height="10%" width="100%">
            <td width="95%">
                
            </td>
        </tr>

        <tr height="90%" width="100%">
            <td>
                <paper-card elevation=2 style="height: 80%; width: 40%; margin:
auto; display: block; padding: 0px;">
                    <table id="mypage" cellspacing="0" cellpadding="0" width="100%" height="100%">
                        <tr height="25%">
                            <td>
                                
                            </td>
                        </tr>

                        <tr height="10%">
                            <td>
                                <paper-input style="margin: 20px" label="Username"
id="username_txt"></paper-input>
                            </td>
                        </tr>

                        <tr height="10%">
                            <td>
                                <paper-input style="margin: 20px" label="Password"
id="password_txt"></paper-input>
                            </td>
                        </tr>

                        <tr height="10%">
                            <td width="100%" id="btn"
onclick="login()">
                                <p id="btnTxt">Log In</p>
                            </td>
                        </tr>

                        <tr height="10%">
                            <td width="100%" id="btn_down"
onclick="register()">
                                <p id="btnTxt">Register</p>
                            </td>
                        </tr>

                    </table>
                </paper-card>
            </td>
        </tr>
    </table>
</body>
</html>

```



```

js/login.js
const remote = require('electron').remote;

window.onload = function(){
};

function register(){
    remote.getCurrentWindow().loadURL(`file://${__dirname}/../html/register.html`);
}

function login(){

    var http = new XMLHttpRequest();
    var url = "http://83.212.118.131:3000/login";
    http.open("POST", url, true);

    var usernameTxt = document.getElementById("username_txt").value;
    var password_txt = document.getElementById("password_txt").value;

    if(usernameTxt == "" || password_txt == ""){
        alert("Please provide both username and password");
    }else{
        var postBody = {
            "username" : usernameTxt,
            "password" : password_txt
        };

        document.getElementById("btnTxt").innerHTML = "<div id='loader' class='loader'
style='display: block; margin: auto;'></div>";
        http.setRequestHeader("Content-type", "application/json");
        http.onreadystatechange = function() { //Call a function when the state changes.
            if(http.readyState == 4 && http.status == 200) {
                var json = JSON.parse(http.responseText);
                console.log(json);
                document.getElementById("btnTxt").innerHTML = "Log In";

                if (typeof(Storage) !== "undefined") {
                    localStorage.setItem("userid", json.ID);

                    remote.getCurrentWindow().loadURL(`file://${__dirname}/../html/index.html`);
                }else{
                    alert("Sorry, your browser does not support Local Storage files. Please
upgrade your browser.");
                }
            }
        }
        http.send(JSON.stringify(postBody));
    }
}

```

6.1.9 html/paragraph_review.html & js/paragraph_review.js

Τα δύο αυτά αρχεία συνθέτουν την οθόνη ελέγχου παραγράφου, πριν την εκκίνηση του παιχνιδιού γνώσεων. Εδώ ο χρήστης μπορεί να ελέγξει την παράγραφο με την οποία θα αγωνιστεί, και αν θέλει να την αλλάξει με κάποια επόμενη. Λόγω του μεγάλου μεγέθους του αρχείου, παρακάτω παρατίθεται ένα ενδεικτικό κομμάτι του κώδικα. Ο πλήρης κώδικας μπορεί να βρεθεί στην ιστοσελίδα του Github.

html/paragraph_review.html

```
<html>
  <head>
    <script type="text/javascript" src="../js/paragraph_review.js"></script>
    <link rel="stylesheet" href="../css/style.css">
    <title>BrainTag</title>
  </head>
  <body>

    <table id="mypage" height="100%" width="100%" cellpadding="0" cellspacing="0">

      <tr id="toolbar" height="10%" width="100%">
        <td width="95%">
          <a href="index.html"></a>
        </td>

        <td width="5%">
          
        </td>
      </tr>

      <tr height="70%" width="100%"><td colspan="2">
        <div class="container" id="paragraph"></div>
      </td></tr>

      <tr height="5%"><td width="100%" colspan="2"><p
id="difficultyTxt">Difficulty : N/A</p></td></tr>
      <tr height="5%"><td width="100%" colspan="2"><p
id="categoryTxt">Category : N/A</p></td></tr>
      <tr height="10%"><td width="100%" id="btn" colspan="2"
onclick="startGame()"><p id="btnTxt">Looks good! Start game!</p></td></tr>
    </table>
  </body>
</html>
```

js/paragraph_review.js

```
const remote = require('electron').remote;

var tagRequests = {
  "wiki_category" : "Technology",
  "wiki_language" : "en",
```

```

        "wrongs" : ""
    };

var tagOnly;

var tagOnlyRequets = {
    "paragraph" : "",
    "wrongs" : "",
    "category" : ""
};

var tagResponse;
var gotEverything = false;

window.onload = function(){
    getCategory();
};

function getCategory(){
    //Κώδικας για την λήψη κατηγοριών και την συντακτική ανάλυση τους
}

function getWrongs(tagOnly){
    //Κώδικας για την λήψη πίνακα με τις μέχρι τώρα λάνθασμένες επιλογές χρηστών στα παιχνίδια
}

function startGame(){
    if(gotEverything == true){
        if(tagOnly){
            tagResponse.category = beautify(tagOnlyRequets.category);
            tagResponse.practice = true;
        }else{
            tagResponse.category = beautify(tagRequests.wiki_category);
            tagResponse.practice = false;
        }
        var array = JSON.stringify(tagResponse);
        var url = 'game.html?data=' + encodeURIComponent(array);
        remote.getCurrentWindow().loadURL(`file://${__dirname}../html/${url}`);
    }
}

function refreshQuestion(){
    getCategory();
}

function beautify(word){
    word = word.replace("_", " ");
    word = word.capitalizeFirstLetter();
    return word;
}

String.prototype.capitalizeFirstLetter = function() {
    return this.charAt(0).toUpperCase() + this.slice(1);
}

function getTagsOnly(){
    //Κώδικας για την λήψη συντακτικής ανάλυσης κειμένου
}

function getTags(){ //Κώδικας για την λήψη κειμένου και την συντακτική ανάλυση του }

```

6.1.10 html/register.html & js/register.js

Τα δύο αυτά αρχεία συνθέτουν την οθόνη εγγραφής του χρήστη στην υπηρεσία braintag. Σε αυτήν την οθόνη ο χρήστης μπορεί να δώσει τα στοιχεία του, με σκοπό να δημιουργηθεί ο λογαριασμός, και να μπορεί να κάνει είσοδο στο παιχνίδι.

```
html/register.html
<html>
  <head>
    <script type="text/javascript" src="../js/register.js"></script>
    <link rel="import" href="../bower_components/paper-card/paper-card.html">
    <link rel="import" href="../bower_components/paper-input/paper-input.html">
    <link rel="stylesheet" href="../css/style.css">
    <title>BrainTag</title>
  </head>
  <body>

    <table id="mypage" height="100%" width="100%" cellpadding="0" cellspacing="0">

      <tr id="toolbar" height="10%" width="100%">
        <td width="95%">
          
        </td>
      </tr>

      <tr height="90%" width="100%">
        <td>
          <paper-card elevation=2 style="height: 95%; width: 40%; margin:
auto; display: block; padding: 0px;">
            <table id="mypage" cellpadding="0" cellspacing="0" width="100%" height="100%">
              <tr>
                <td>
                  
                </td>
              </tr>
              <tr>
                <td>
                  <paper-input style="margin: 20px" label="Username"
id="username_txt"></paper-input>
                </td>
              </tr>
              <tr>
                <td>
                  <paper-input style="margin: 20px" label="Email" id="email_txt"></paper-
input>
                </td>
              </tr>
              <tr>
                <td>
                  <paper-input style="margin: 20px" label="Password"
id="password_txt"></paper-input>
                </td>
              </tr>
            </table>
          </paper-card>
        </td>
      </tr>
    </table>
  </body>
</html>
```

```

        </tr>

        <tr height="10%">
            <td width="100%" id="btn_down"

                <p id="btnTxt">Register</p>
            </td>
        </tr>

    </table>
</paper-card>
</td>
</tr>
</table>
</body>
</html>

js/register.js
const remote = require('electron').remote;

window.onload = function(){
};

function register(){

    var http = new XMLHttpRequest();
    var url = "http://83.212.118.131:3000/register";
    http.open("POST", url, true);

    var usernameTxt = document.getElementById("username_txt").value;
    var password_txt = document.getElementById("password_txt").value;
    var email_txt = document.getElementById("email_txt").value;

    if(usernameTxt == "" || password_txt == "" || email_txt == ""){
        alert("Please provide all credentials");
    }else{
        var postBody = {
            "username" : usernameTxt,
            "password" : password_txt,
            "email" : email_txt
        };

        document.getElementById("btnTxt").innerHTML = "<div id='loader' class='loader'
style='display: block; margin: auto;'></div>";
        http.setRequestHeader("Content-type", "application/json");
        http.onreadystatechange = function() { //Call a function when the state changes.
            if(http.readyState == 4 && http.status == 200) {
                if(http.responseText == "OK"){
                    alert("You are successfully registered.\nNow you can login with your
credentials.");
                }else{
                    alert("We encountered a problem : "+http.responseText);
                }
                remote.getCurrentWindow().loadURL(`file://${__dirname}/../html/login.html`);
            }else if(http.readyState == 4 && http.status != 200){
                alert(http.responseText);
                document.getElementById("btnTxt").innerHTML = "Register";
            }
        }
    }
    http.send(JSON.stringify(postBody));
}

```

```
}  
}
```

6.1.11 html/tags.html

Το αρχείο αυτό δημιουργεί την οθόνη εμφάνισης όλων των ετικετών συντακτικού σε μορφοποιημένο πίνακα. Με αυτόν τον τρόπο, ο χρήστης μπορεί οποιαδήποτε στιγμή να δει τις ετικέτες με την συντομογραφία αλλά και το πλήρες κείμενο τους.

```
<html>  
  <head>  
    <link rel="import" href="../../bower_components/paper-card/paper-card.html">  
    <link rel="stylesheet" href="../../css/style.css">  
    <title>BrainTag</title>  
  </head>  
  <body>  
  
    <table id="mypage" height="100%" width="100%" cellpadding="0" cellspacing="0">  
      <tr id="toolbar" height="10%" width="100%">  
        <td width="100%">  
          <a href="index.html"></a>  
        </td>  
      </tr>  
      <tr height="90%" width="100%">  
        <td><p style="font-size:2vw; text-align: center; margin:20px">Our app uses the  
worldwide system for world tagging.<br>Bellow you will find the description for each tag</p>  
        <table id="tagtable" width="100%">  
          <tr> <th width="50%" style="border-bottom: 1px solid #ddd;"><h2> TAG </h2></th>  
<th width="50%" style="border-bottom: 1px solid #ddd;"><h2> MEANING </h2></th></tr>  
          <tr> <td width="50%" style="border-bottom: 1px solid #ddd;text-align:  
center;"><h3> CC </h3></td> <td width="50%" style="border-bottom: 1px solid #ddd; text-align:  
center;"><h3> Coordinating conjunction </h3></td></tr>  
          <tr> <td width="50%" style="border-bottom: 1px  
solid #ddd;text-align: center;"><h3> CD </h3></td> <td width="50%" style="border-bottom: 1px  
solid #ddd; text-align: center;"><h3> Cardinal number </h3></td></tr>  
          <tr> <td width="50%" style="border-bottom: 1px  
solid #ddd;text-align: center;"><h3> DT </h3></td> <td width="50%" style="border-bottom: 1px  
solid #ddd; text-align: center;"><h3> Determiner </h3></td></tr>  
          <tr> <td width="50%" style="border-bottom: 1px  
solid #ddd;text-align: center;"><h3> EX </h3></td> <td width="50%" style="border-bottom: 1px  
solid #ddd; text-align: center;"><h3> Existential there </h3></td></tr>  
          <tr> <td width="50%" style="border-bottom: 1px  
solid #ddd;text-align: center;"><h3> FW </h3></td> <td width="50%" style="border-bottom: 1px  
solid #ddd; text-align: center;"><h3> Foreign word </h3></td></tr>  
          <tr> <td width="50%" style="border-bottom: 1px  
solid #ddd;text-align: center;"><h3> IN </h3></td> <td width="50%" style="border-bottom: 1px  
solid #ddd; text-align: center;"><h3> Preposition or subordinating conjunction </h3></td></tr>  
          <tr> <td width="50%" style="border-bottom: 1px  
solid #ddd;text-align: center;"><h3> JJ </h3></td> <td width="50%" style="border-bottom: 1px
```



```

        <tr> <td width="50%" style="border-bottom: 1px
solid #ddd;text-align: center;"><h3> VBD </h3></td> <td width="50%" style="border-bottom: 1px
solid #ddd; text-align: center;"><h3> Verb, past tense </h3></td></tr>
        <tr> <td width="50%" style="border-bottom: 1px
solid #ddd;text-align: center;"><h3> VBG </h3></td> <td width="50%" style="border-bottom: 1px
solid #ddd; text-align: center;"><h3> Verb, gerund or present participle </h3></td></tr>
        <tr> <td width="50%" style="border-bottom: 1px
solid #ddd;text-align: center;"><h3> VBN </h3></td> <td width="50%" style="border-bottom: 1px
solid #ddd; text-align: center;"><h3> Verb, past participle </h3></td></tr>
        <tr> <td width="50%" style="border-bottom: 1px
solid #ddd;text-align: center;"><h3> VBP </h3></td> <td width="50%" style="border-bottom: 1px
solid #ddd; text-align: center;"><h3> Verb, non-3rd person singular present </h3></td></tr>
        <tr> <td width="50%" style="border-bottom: 1px
solid #ddd;text-align: center;"><h3> VBZ </h3></td> <td width="50%" style="border-bottom: 1px
solid #ddd; text-align: center;"><h3> Verb, 3rd person singular present </h3></td></tr>
        <tr> <td width="50%" style="border-bottom: 1px
solid #ddd;text-align: center;"><h3> WDT </h3></td> <td width="50%" style="border-bottom: 1px
solid #ddd; text-align: center;"><h3> Wh-determiner </h3></td></tr>
        <tr> <td width="50%" style="border-bottom: 1px
solid #ddd;text-align: center;"><h3> WP </h3></td> <td width="50%" style="border-bottom: 1px
solid #ddd; text-align: center;"><h3> Wh-pronoun </h3></td></tr>
        <tr> <td width="50%" style="border-bottom: 1px
solid #ddd;text-align: center;"><h3> WP$ </h3></td> <td width="50%" style="border-bottom: 1px
solid #ddd; text-align: center;"><h3> Possessive wh-pronoun </h3></td></tr>
        <tr> <td width="50%" style="border-bottom: 1px
solid #ddd;text-align: center;"><h3> WRP </h3></td> <td width="50%" style="border-bottom: 1px
solid #ddd; text-align: center;"><h3> Wh-adverb </h3></td></tr>
    </table>
    </td>
</tr>
</table>
</body>
</html>

```

6.1.12 html/type.html & js/type.js

Τα δύο αυτά αρχεία συνθέτουν την οθόνη εισαγωγής κειμένου. Μια λειτουργία της εφαρμογής, επιτρέπει τον χρήστη να εξετάσει τις γνώσεις του σε κείμενο της δικής του επιλογής. Για να γίνει αυτό, πρέπει πρώτα να γίνει εισαγωγή του κειμένου μέσω αυτής της οθόνης.

```

html/type.html
<html>
  <head>
    <script type="text/javascript" src="../js/type.js"></script>
    <link rel="import" href="../bower_components/paper-input/paper-input.html">
    <link rel="stylesheet" href="../css/style.css">
    <title>BrainTag</title>
  </head>
  <body>

```



```

        <table id="mypage" height="100%" width="100%" cellpadding="0">
            <tr id="toolbar" height="10%" width="100%">
                <td width="100%">
                    <div style="margin-left: 20px">
                        <a href="index.html"></a>
                    </div>
                </td>
            </tr>

            <tr height="80%" width="100%">
                <td>
                    <p style="margin: 20px; text-align: center;">Tired of
playing on Wikipedia pages and current News? You can practice with your own paragraph!</p>
                    <paper-input id="input" style="width:80%; margin:10%"
label="Please type a paragraph you would like to play with"></paper-input>
                </td>
            </tr>

            <tr height="10%"><td width="100%" id="btn" onclick="startGame()"><p
id="btnTxt">Start Game</p></td></tr>

        </table>
    </body>
</html>

```

js/type.js

```

const remote = require('electron').remote;

window.onload = function(){};

function startGame(){
    var text = document.getElementById("input").value;
    console.log(text);
    if (typeof(Storage) !== "undefined") {
        localStorage.setItem("typedata",text);
        var url = `paragraph_review.html?data=` + encodeURIComponent("_typed_");
        remote.getCurrentWindow().loadURL(`file://${__dirname}/../html/${url}`);
    }else{
        alert("Sorry, your browser does not support Local Storage files. Please upgrade your
browser.");
    }
}

```

6.1.13 js/main.js

Το αρχείο αυτό είναι το πρώτο αρχείο που εκτελείται κατά την εκκίνηση της εφαρμογής. Ορίζει το βασικό μέγεθος της εφαρμογής σε pixel οθόνης, όπως και το πρώτο html αρχείο που θα εκτελεστεί για την εμφάνιση εικόνας στην οθόνη.

```
const electron = require('electron');
const app = electron.app;
const BrowserWindow = electron.BrowserWindow;

let windowDimensions = {height : 720, width: 1280, show: true, 'node-integration': false,
'minHeight': 600, 'minWidth': 860};
let mainWindow;

app.on('ready', _ =>{
  mainWindow = new BrowserWindow(windowDimensions);
  mainWindow.loadURL(`file://${__dirname}/../html/index.html`)

  mainWindow.on('closed', _ =>{
    mainWindow = null;
  });
})
```

7. Συμπεράσματα

Κλείνοντας την εργασία ύστερα από την εμπειρία της εκπόνησης αυτής της διπλωματικής, μπορώ να πω ότι η πλατφόρμα που έχει δημιουργηθεί, αν και ακόμα με εμβριακή μορφή, θα μπορούσε να έχει μεγάλο αντίκτυπο στον τομέα της εκπαίδευσης αλλά και στις ψυχαγωγίας.

Η πλατφόρμα μπορεί να χρησιμοποιηθεί σε καθημερινή ή και εβδομαδιαία βάση από καθηγητές, βοηθώντας τους να ελέγχουν την πρόοδο των μαθητών τους με έναν διασκεδαστικό τρόπο. Οι μαθητές μικρών ηλικιών, έχουν στα χέρια τους ένα εργαλείο του οποίου θα μπορούσε μέχρι ενός σημείου να αντικαταστήσει το πατροπαράδοτο τρόπο εξέτασης. Αυτό τους κάνει να απολαμβάνουν αυτήν την διαδικασία σε μεγαλύτερο βαθμό, με αποτέλεσμα να διαβάζουν και περισσότερο.

Κύρια σημεία καινοτομίας των εφαρμογών είναι ότι χρησιμοποιούν ένα πολύ δυνατό εργαλείο, όπως η επεξεργασία φυσικής γλώσσας, για τη δημιουργία ενός προϊόντος που έχει στόχο τελικούς χρήστες παιδιά. Οι πιο γνωστές εφαρμογές και εργαλεία που χρησιμοποιούν αντίστοιχες βιβλιοθήκες και έχουν τις ίδιες δυνατότητες απευθύνονται κυρίως σε αναλυτές και έχουν κυρίως επαγγελματικό ή στατιστικό σκοπό (π.χ. εργαλεία μετάφρασης και διερεύνησης συναισθημάτων πελατών/χρηστών).

Όσον αφορά τους τομείς μελλοντικής εργασίας και τους τρόπους βελτίωσης των εφαρμογών, μπορώ να ξεχωρίσω 2 βασικούς πυλώνες. Ο πρώτος είναι η βελτίωση του γραφικού περιβάλλοντος των εφαρμογών, με σκοπό να γίνουν ακόμα πιο προσίτες σε μικρά παιδιά. Αυτό μπορεί να γίνει με την χρήση κινούμενων εικόνων (animations) όπως και ήχων κατά την διάρκεια του παιχνιδιού. Σαν δεύτερο πυλώνα θα θέσω την δημιουργία αντίστοιχης εφαρμογής για το κινητό λειτουργικό σύστημα iOS της Apple. Στοχεύοντας το ευρύ κοινό, ειδικά στις αγγλόφωνες χώρες, δεν πρέπει να υπάρχει περιορισμός στον χρήστη κατά την επιλογή της συσκευής στην οποία θα γίνει η χρήση της εφαρμογής. Για αυτόν τον λόγο πρέπει και να προσφέρεται σε όλα τα διαθέσιμα λειτουργικά συστήματα.

Βιβλιογραφία

- [1] - <https://www.android.com/> [ηλεκτρονικό]
- [2] - <https://www.microsoft.com/en-us/windows> [ηλεκτρονικό]
- [3] - <https://www.apple.com/lae/macOS/sierra/> [ηλεκτρονικό]
- [4] - <https://www.linux.org/> [ηλεκτρονικό]
- [5] - <https://techcrunch.com/> [ηλεκτρονικό]
- [6] - <http://www.bbc.com/news> [ηλεκτρονικό]
- [7] - <https://news.google.com/> [ηλεκτρονικό]
- [8] - <http://edition.cnn.com/> [ηλεκτρονικό]
- [9] - <http://www.nationalgeographic.com/> [ηλεκτρονικό]
- [10] - <https://www.theverge.com/> [ηλεκτρονικό]
- [11] - <https://www.usatoday.com/> [ηλεκτρονικό]
- [12] - <https://www.java.com/en/> [ηλεκτρονικό]
- [13] - <https://developer.mozilla.org/en-US/docs/Web/HTML> [ηλεκτρονικό]
- [14] - <https://developer.mozilla.org/en-US/docs/Web/CSS> [ηλεκτρονικό]
- [15] - <https://www.javascript.com/> [ηλεκτρονικό]
- [16] - <http://www.abelard.org/turpap/turpap.php> [Ηλεκτρονικό]
- [17] - Michael L. Mauldin “CHATTERBOTS, TINYMUDS, and the Turing Test”
- [18] - Hutchins, J. “The history of machine– translation in a nutshell” , 2005
- [19] - Sergei Nirenburg, H. L. Somers, Yorick Wilks, “Readings in Machine Translation”,
σελίδες 130-135

- [20] - Artificial Intelligence Review, (March 1994), Τόμος 8, Τεύχος 2, σελίδες 189–214
- [21] - Garfield David A. S. M.D. , Rapp Charles , Evens Martha Ph.D. , “Natural Language Processing in Psychiatry: Artificial Intelligence Technology and Psychopathology.” , σε Journal of Nervous & Mental Disease, April 1992
- [22] - Bayan Abu Shawar and Eric Atwell “Machine learning from dialogue corpora to generate chatbots” από School of Computing, University of Leeds
- [23] - Marie-Catherine de Marneffe , Bill MacCartney , Christopher D. Manning “Generating Typed Dependency Parses from Phrase Structure Parses”
- [24] - “Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms” δημοσιεύτηκε σε “Proceedings of the ACL-02 conference on Empirical methods in natural language processing” , τόμος 10, σελίδες 1-8
- [25] - Tonio Wandmacher, Jean-Yves Antoine “Methods to integrate a language model with semantic information for a word prediction component” , Jan 2008
- [26] - Christopher D. Manning and Hinrich Schütze “Foundations of Statistical Natural Language Processing” σε Stanford University και Xerox Palo Alto Research Center
- [27] - Goldberg, Yoav (2016). “A Primer on Neural Network Models for Natural Language Processing.” σε Journal of Artificial Intelligence Research 57, 2016 , σελίδες 345–420
- [28] - Ian Goodfellow, Yoshua Bengio and Aaron Courville. “Deep Learning Book” , σε MIT Press.
- [29] - Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu , “Exploring the Limits of Language Modeling” , 2016
- [30] - Do Kook Choe and Eugene Charniak (EMNLP 2016) , “Parsing as Language Modeling”
- [31] - Vinyals, Oriol, σε NIPS2015 <https://papers.nips.cc/paper/5635-grammar-as-a-foreign-language.pdf> [ηλεκτρονικό]
- [32] - Gobinda G. Chowdhury “Natural language processing” , 2003
- [33] - Elizabeth D. Liddy, “Natural Language Processing” , 2001

- [34] - P.Spyns “Natural Language Processing in Medicine : An Overview” , 1996
- [35] - John Wiley and Sons Ltd. Chichester, UK “Encyclopedia of Computer Science 4th”
σελίδες 1218-1222 , ISBN:0-470-86412-5
- [36] - B. Liu, «Sentiment Analysis and Subjectivity,» σε Handbook of Natural Language Processing, Second Edition, N. I. a. F. J. Damerau, Επιμ., Boca Raton, FL, CRC Press, Taylor and Francis Group, 2010, σελίδες 627-666.
- [37] - B. Pang και L. Lee, «Opinion Mining and Sentiment Analysis,» Foundations and Trends® in Information Retrieval, τόμ. 2, σελίδες 1-135, 2008.
- [38] - E. Kouloumpis, T. Wilson και J. Moore, «Twitter Sentiment Analysis: The Good the Bad and the OMG!»,» σε Proceedings of the Fifth International AAAI Conference on Weblogs and Social Meda, Barcelona, 2011.
- [39] - Tuomo Korenius , Jorma Laurikkala , Kalervo Järvelin ,Martti Juhola “Stemming and lemmatization in the clustering of finnish text documents” σε CIKM '04 Proceedings of the thirteenth ACM international conference on Information and knowledge management, σελίδες 625-633
- [40] - F. Meyer .S. Beucher, “Journal of Visual Communication and Image Representation” Τόμος 1, Τεύχος 1, September 1990, Σελίδες 21-46
- [41] - Kishorjit, N., Vidya Raj RK., Nirmal Y., and Sivaji B. , “Manipuri Morpheme Identification”, Proceedings of the 3rd Workshop on South and Southeast Asian Natural Language Processing (SANLP), σελίδες 95–108, COLING 2012, Mumbai, December 2012
- [42] - Eric Brill , “Part-of-Speech Tagging” σε Handbook of Natural Language Processing, 2000, Κεφάλαιο 17 , ISBN: 978-0-8247-9000-4
- [43] - Tunga Güngör “Part-of-Speech Tagging” σε Handbook of Natural Language Processing, Second Edition, σελίδες 205–235, Κεφάλαιο 10, ISBN: 978-1-4200-8592-1
- [44] - James R. Curran , Stephen Clark , David Vadas , “Multi-tagging for lexicalized-grammar parsing” σε ACL-44 Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics , σελίδες 697-704

- [45] - David D. Palmer , Marti A. Hearst , “Adaptive multilingual sentence boundary disambiguation” σε Computational Linguistics , Τόμος 23 Τεύχος 2, June 1997 , σελίδες 241-267
- [46] - David A. Hull , Stemming Algorithms - A Case Study for Detailed Evaluation , June 1995
- [47] - Pierre Perruchet, Annie Vinter , “Journal of Memory and Language”
Τόμος 39, Τεύχος 2, August 1998, σελίδες 246–263 , σε PARSER: A Model for Word Segmentation
- [48] - Michael Stubbs “WORDS AND PHRASES: CORPUS STUDIES OF LEXICAL SEMANTICS” , October 2001
- [49] - Harold Somers , “Machine Translation” , June 1999, Τόμος 14, Τεύχος 2, σελίδες 113–157
- [50] - Lev Ratinov , Dan Roth , “Design challenges and misconceptions in named entity recognition” σε CoNLL '09 Proceedings of the Thirteenth Conference on Computational Natural Language Learning , σελίδες 147-155
- [51] - John Bateman and Michael Zock , “Natural Language Generation” σε The Oxford Handbook of Computational Linguistics 2nd edition (2 ed.)
- [52] - Allen, James , “Natural Language Understanding” , 1995 , ISBN: 08053033409780805303346
- [53] - Sargur N. Srihari , Ajay Shekhawat , Stephen W. Lam , “Optical character recognition (OCR)” σε Encyclopedia of Computer Science 4th , σελίδες 1326-1333 , ISBN:0-470-86412-5
- [54] - David Elworthy , “Question Answering using a large NLP System” σε Microsoft Research Limited
- [55] - Dagan Ido , Roth Dan , Sammons Mark , “Recognizing textual entailment : models and applications”, 2013. Σελίδα 222
- [56] - Cory B Giles and Jonathan D Wren , “Large-scale directional relationship extraction and resolution” σε BMC Bioinformatics20089(Suppl 9):S11
- [57] - Theresa Wilson , Janyce Wiebe , Paul Hoffmann , “Recognizing contextual polarity in phrase-level sentiment analysis” σε HLT '05 Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing

σελίδες 347-354

[58] - Inderjeet Mani and Mark T. Maybury , “Advances in Automatic Text Summarization”, The MIT Press, 1999, ISBN 0-262-13359-8,

[59] - Lawrence R. Rabiner, B. H. Juang , “Fundamentals of speech recognition” , April 1993

[60] - Mari Ostendorf , Benoit Favre , Ralph Grishman , “Speech segmentation and spoken document processing” σε IEEE Signal Processing Magazine, Τόμος 25, Τεύχος 3, May 2008

[61] - Richard W. Sproat , “Multilingual Text-to-Speech Synthesis” , 1997 , ISBN:0792380274