Εθνικο Μετσοβιο Πολυτεχνειο
Τμημα Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Αποσαφήνιση Οντοτήτων σε Κείμενο με χρήση Γράφου Γνώσης και Σημασιολογικής Εγγύτητας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
του
Αλέξιου Μανδαλιού

Επιβλέπων: Γιώργος Στάμου
Αναπληρωτής Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2017

Εθνικο Μετσοβιο Πολυτεχνειο
Τμημα Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Αποσαφήνιση Οντοτήτων σε Κείμενο με χρήση Γράφου Γνώσης και Σημασιολογικής Εγγύτητας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
του
Αλέξιου Μανδαλιού

Επιβλέπων: Γιώργος Στάμου
Αναπληρωτής Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 20η Ιουλίου 2017.

(υπογραφή)	*(υπογραφή)*	*(υπογραφή)*
.....	.....	.....
Γιώργος Στάμου	Δημήτριος Φωτάκης	Ανδρέας-Γεώργιος Σταφυλοπάτης
Αναπληρωτής Καθηγητής Ε.Μ.Π.	Επίκουρος Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2017

(υπογραφή)

.....

Αλέξιος Μανδαλιός
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Ένα κείμενο μπορεί να περιέχει αναφορές σε φυσικά πρόσωπα, τοποθεσίες, οργανισμούς, ταινίες, μάρκες προϊόντων και άλλους τύπους οντοτήτων. Οι αναφορές αυτές είναι συχνά αμφίσημες ως προς τις οντότητες του κόσμου που αναπαριστούν, πλην όμως η ανθρώπινη νοημοσύνη έχει τη δεξιότητα να τις αποσαφηνίζει με επιτυχία στις περισσότερες περιπτώσεις. Στόχος της παρούσας εργασίας είναι η αναγνώριση των αναφορών αυτών σε κάποιο κείμενο και η αποσαφήνισή τους μέσω αντιστοίχισης με οντότητες που βρίσκονται σε μια βάση γνώσης. Αυτή η διαδικασία είναι γνωστή ως αναγνώριση και αποσαφήνιση ονοματικών οντοτήτων. Για να επιτευχθεί αυτός ο στόχος χρησιμοποιείται μια γραφοθεωρητική προσέγγιση. Αυτή περιλαμβάνει ανάλυση του κειμένου με μεθόδους επεξεργασίας φυσικής γλώσσας, καθώς και χρήση σύγχρονων τεχνικών και εργαλείων. Ως βασικές πηγές γνώσης αξιοποιούνται ο Γράφος Γνώσης της Google και η Wikipedia.

Ιδιαίτερο βάρος δίνεται στην ανάλυση των βασικών αποφάσεων που πρέπει να ληφθούν κατά τη σχεδίαση ενός συστήματος αποσαφήνισης οντοτήτων σε κείμενο, οι οποίες θα καθορίσουν την ποιότητα του αποτελέσματος και το πεδίο εφαρμογής. Η εργασία αυτή επικεντρώνεται σε μικρού και μεσαίου μεγέθους κείμενα, τα οποία αναφέρονται σε σημασιολογικά συναφείς οντότητες από σχετικά λίγες και σχετιζόμενες θεματικές ενότητες. Το σύστημα αξιολογείται πειραματικά σε δύο σύνολα κειμένων, το πρώτο με μικρά κείμενα και το δεύτερο με μεσαίου μεγέθους κείμενα. Τα αποτελέσματα της αξιολόγησης αυτής υποδεικνύουν ότι το σύστημα που αναπτύχθηκε είναι ανταγωνιστικό και η απόδοσή του είναι συγκρίσιμη με αυτήν των πιο επιτυχημένων συστημάτων αποσαφήνισης οντοτήτων.

Λέξεις Κλειδιά: αναγνώριση ονοματικών οντοτήτων, αποσαφήνιση ονοματικών οντοτήτων, Γράφος Γνώσης Google, Wikipedia, k-partite γράφος, k-clique μέγιστου βάρους, ευριστική τεχνική αφαίρεσης χειρότερου στοιχείου

Abstract

A document may include mentions about people, locations, organizations, films, product brands and other kinds of entities. Those mentions are often ambiguous and there is no obvious way to map them to real world entities. However, in most cases, the human cognitive ability is capable of disambiguating them successfully. This thesis aims to recognize those mentions in unstructured text and proceed to disambiguate them by mapping them to entities stored in a knowledge base. This process is known as Named Entity Recognition & Disambiguation (NERD) or Entity Linking. The aforementioned goal is achieved via a graph-based approach, that leverages natural language processing methods and state-of-the art techniques and tools. Google's Knowledge Graph and Wikipedia are the main sources of knowledge in this project.

One of the most important tasks in the development process is the selection of an appropriate set of features for the disambiguation engine, as those features are inevitably going to determine the resulting system's quality as well as its applications. This thesis focuses on small and medium-sized documents, that refer to coherent entities from one or a few related topics. The developed system is evaluated experimentally using two datasets, the first containing small documents and the second medium-sized documents. The evaluation results suggest that the system is quite competitive and its performance can be compared to that of the most successful NERD systems available today.

Key Words: Named Entity Recognition, NER, Named Entity Disambiguation, NED, NERD, Google Knowledge Graph, Wikipedia, k-partite graph, max weight k-clique, worst out heuristic

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την οικογένειά μου για τη συμπαράστασή τους σε όλη τη διάρκεια των σπουδών μου, αλλά ιδιαίτερα κατά τους μήνες εκπόνησης της παρούσας εργασίας. Ευχαριστώ τον επιβλέποντα καθηγητή κύριο Γιώργο Στάμου για τη δημιουργική ελευθερία που μου προσέφερε. Εκφράζω τις ευχαριστίες μου προς τον κύριο Κωνσταντίνο Τζαμαλούκα, ο οποίος με καθοδήγησε με υπομονή και προσωπική ενασχόληση επί ένα έτος, ώστε να επιτευχθεί ένα ποιοτικό τελικό αποτέλεσμα. Τέλος, ευχαριστώ τον καλό μου φίλο Θοδωρή για τις συζητήσεις που είχαμε πάνω σε θέματα υλοποίησης και τον εξάδελφό μου Μιχάλη για τα σχόλιά του στην τελική έκδοση του κειμένου.

Περιεχόμενα

Κατάλογος Εικόνων **15**

Κατάλογος Πινάκων **17**

Κατάλογος Αλγορίθμων **19**

Κατάλογος Τμημάτων Κώδικα **21**

1 Εισαγωγή **23**
1.1 Χρήσιμοι ορισμοί 23
1.2 Προκλήσεις κατά την αποσαφήνιση οντοτήτων 25
1.3 Κίνητρα για αποσαφήνιση οντοτήτων 26
1.3.1 Ανάκληση πληροφορίας 26
1.3.2 Εξαγωγή χαρακτηριστικών κειμένου 26
1.3.3 Απάντηση ερωτήσεων 28
1.3.4 Εμπλουτισμός βάσης γνώσης 28
1.4 Εύρος της εργασίας 29
1.5 Συνεισφορές της εργασίας 29
1.6 Διάρθρωση του κειμένου 30

2 Θεωρητικό υπόβαθρο και προηγούμενες εργασίες **31**
2.1 Μοντελοποίηση του προβλήματος 31
2.2 Βάσεις γνώσης 31
2.3 Βασικά βήματα κατά την αποσαφήνιση οντοτήτων 33
2.3.1 Επιλογή γλώσσας ενδιαφέροντος 33
2.3.2 Αναγνώριση ονοματικών οντοτήτων 33
2.3.3 Παραγωγή υποψήφιων οντοτήτων 34
2.3.3.1 Χρήση λεξικών ονομάτων 34
2.3.3.2 Χρήση μηχανών αναζήτησης 37
2.3.3.3 Αναζήτηση εναλλακτικών ονομάτων στο κείμενο 38
2.3.4 Ταξινόμηση υποψήφιων οντοτήτων 39
2.3.4.1 Τεχνικές ανεξάρτητες των συμφραζομένων 40
2.3.4.2 Τεχνικές βασιζόμενες στα συμφραζόμενα 41
2.3.5 Αναγνώριση μη αποσαφηνίσιμων αναφορών 44
2.4 Αξιολόγηση συστημάτων αποσαφήνισης 46

3 Εξέταση APIs **51**
3.1 Google Knowledge Graph Search API 51
3.2 MediaWiki API 61
3.2.1 MediaWiki Links API 61
3.2.2 MediaWiki Redirects API 63
3.2.3 MediaWiki Linkshere API 65
3.3 Stanford CoreNLP 67

4 Ανάλυση και σχεδίαση συστήματος αποσαφήνισης 71
4.1 Αποσαφήνιση στην αγγλική γλώσσα 71
4.2 Αναγνώριση ονοματικών οντοτήτων με το Stanford CoreNLP 71
4.3 Παραγωγή υποψήφιων οντοτήτων με χρήση APIs 72
4.4 Κριτήρια ταξινόμησης οντοτήτων 75
4.4.1 GKG resultScore 75
4.4.2 Binary document similarity 77
4.4.3 Entity relatedness 77
4.5 Κατασκευή γράφου 78
4.6 Επίλυση γράφου 79
4.7 Διαγραμματική παρουσίαση συστήματος αποσαφήνισης 86

5 Ζητήματα υλοποίησης 89
5.1 Γλώσσα προγραμματισμού 89
5.2 Τοπικά αποθηκευμένα δεδομένα 90
5.3 Διαδικτυακή εφαρμογή 91

6 Πειραματική αξιολόγηση συστήματος αποσαφήνισης 99
6.1 Πειραματική αξιολόγηση με μικρά κείμενα 99
6.1.1 Dataset μικρών κειμένων 99
6.1.2 Παραμετροποίηση και πειραματικά αποτελέσματα 100
6.2 Πειραματική αξιολόγηση με το CoNLL dataset 106
6.2.1 Επεξεργασία του CoNLL dataset 106
6.2.2 Παραμετροποίηση και πειραματικά αποτελέσματα 107

7 Επίλογος 111
7.1 Συμπεράσματα 111
7.2 Δυνατές επεκτάσεις 112

Αναφορές 115

Παράρτημα 123

Κατάλογος Εικόνων

1 Εποπτική σύγκριση ανθρώπινης αντίληψης και αντίληψης υπολογιστή όσον αφορά την αναγνώριση και αποσαφήνιση ονοματικών οντοτήτων 27

2 Εποπτική παρουσίαση σημασιολογικής συγγένειας μεταξύ υποψήφιων οντοτήτων 45

3 Κατασκευή γράφου μέγιστης συνεισφοράς για έναν κόμβο ενός απλού γράφου 81

4 Ενημέρωση γράφου μέγιστης συνεισφοράς για έναν κόμβο ενός απλού γράφου 82

5 Activity diagram συστήματος αποσαφήνισης 87

6 Component diagram συστήματος αποσαφήνισης 88

7 Activity diagram συστήματος αποσαφήνισης, με τοπικά αποθηκευμένα δεδομένα 92

8 Component diagram συστήματος αποσαφήνισης, με τοπικά αποθηκευμένα δεδομένα 93

9 Sequence diagram διαδικτυακής εφαρμογής αποσαφήνισης 94

10 Παράδειγμα εισόδου στη διαδικτυακή εφαρμογή αποσαφήνισης 95

11 Παράδειγμα εξόδου που παράγεται από τη διαδικτυακή εφαρμογή αποσαφήνισης 96

12 Πληροφορίες για μια οντότητα, όπως δίνονται από τη διαδικτυακή εφαρμογή αποσαφήνισης 97

13 Γραφική απεικόνιση Micro Average Precision και Macro Average Precision κατά την πειραματική αξιολόγηση με χρήση του dataset 300 μικρών κειμένων . 103

14 Γραφική απεικόνιση Micro Average Recall και Macro Average Recall κατά την πειραματική αξιολόγηση με χρήση του dataset 300 μικρών κειμένων 104

15 Γραφική απεικόνιση Micro Average Accuracy και Macro Average Accuracy κατά την πειραματική αξιολόγηση με χρήση του dataset 300 μικρών κειμένων . 105

16 Γραφική απεικόνιση Micro Average Accuracy και Macro Average Accuracy κατά την πειραματική αξιολόγηση με χρήση του Reuters-231 dataset 108

Κατάλογος Πινάκων

1 Ενδεικτικό τμήμα ενός λεξικού ονομάτων 38

2 Υπολογισμός μετρικής συνοχής WLM για ενδεικτικά σύνολα υποψήφιων οντοτήτων 44

3 Παραγωγή των συνόλων predicted positive, actual positive, predicted negative και actual negative βάσει των συνόλων TP, FP, TN και FN 47

4 Τα namespaces των σελίδων της Wikipedia 61

5 Τιμές Micro Average Precision και Macro Average Precision κατά την πειραματική αξιολόγηση με χρήση του dataset 300 μικρών κειμένων 102

6 Τιμές Micro Average Recall και Macro Average Recall κατά την πειραματική αξιολόγηση με χρήση του dataset 300 μικρών κειμένων 102

7 Τιμές Micro Average Accuracy και Macro Average Accuracy κατά την πειραματική αξιολόγηση με χρήση του dataset 300 μικρών κειμένων 103

8 Τιμές Micro Average Accuracy και Macro Average Accuracy κατά την πειραματική αξιολόγηση με χρήση του Reuters-231 dataset 108

9 Σύγκριση πειραματικών αποτελεσμάτων της δικής μας μεθόδου αποσαφήνισης με τα αποτελέσματα εναλλακτικών μεθόδων αποσαφήνισης στο Reuters-231 dataset 109

Κατάλογος Αλγορίθμων

1 Αφελής τρόπος παραγωγής λεξικού ονομάτων 35

2 Πρακτικός αλγόριθμος κατασκευής λεξικού ονομάτων 36

3 Παραγωγή του συνόλου υποψήφιων οντοτήτων για μια αναφορά με βάση λεξικό ονομάτων 37

4 Παραγωγή συνόλου υποψήφιων οντοτήτων με χρήση μηχανής αναζήτησης . . 39

5 Αξιοποίηση των εναλλακτικών ονομάτων σε συνδυασμό με τη μέθοδο λεξικού για την παραγωγή του συνόλου υποψήφιων οντοτήτων για μια αναφορά 39

6 Αναγνώριση ονοματικών οντοτήτων σε κείμενο με βάση τα models του Stanford CoreNLP 72

7 Παραγωγή βάσης συνόλου υποψήφιων οντοτήτων με χρήση του GKG API . . . 74

8 Δομικό φιλτράρισμα των αποτελεσμάτων του GKG API 74

9 Σημασιολογικό φιλτράρισμα των αποτελεσμάτων του GKG API 75

10 Παραγωγή τελικού συνόλου υποψήφιων οντοτήτων με χρήση του GKG API, καθώς και των MediaWiki APIs 76

11 Κανονικοποίηση resultScores ενός συνόλου υποψήφιων οντοτήτων 76

12 Υπολογισμός document similarity βάσει των κοινών όρων δύο κειμένων 77

13 Υπολογισμός σημασιολογικής εγγύτητας μεταξύ δύο οντοτήτων με χρήση του WLM 78

Κατάλογος Τμημάτων Κώδικα

1 Πρώτο παράδειγμα χρήσης του GKG API 53

2 Δεύτερο παράδειγμα χρήσης του GKG API 55

3 Παράδειγμα χρήσης του MediaWiki Links API 62

4 Παράδειγμα χρήσης του MediaWiki Redirects API 64

5 Παράδειγμα χρήσης του MediaWiki Linkshere API 66

6 Παράδειγμα χρήσης του Stanford CoreNLP 68

Κεφαλαιο 1

Εισαγωγή

Αυτό το κεφάλαιο περιλαμβάνει μια σύντομη εισαγωγή στα ζητήματα που πραγματεύεται η εργασία. Στην ενότητα 1.1 δίνονται κάποιοι ορισμοί που θα φανούν χρήσιμοι σε όλη την έκταση του κειμένου. Στην ενότητα 1.2 παρουσιάζονται οι προκλήσεις που αντιμετωπίζει κανείς κατά την αποσαφήνιση αναφορών σε αδόμητο κείμενο, με ένα απλό παράδειγμα. Στην ενότητα 1.3 αναφέρονται οι σημαντικές εφαρμογές της αποσαφήνισης οντοτήτων, που αποτελούν βασικά κίνητρα για την έρευνα στον τομέα αυτόν. Στις ενότητες 1.4 και 1.5 παρουσιάζεται το εύρος και οι συνεισφορές της παρούσας εργασίας, αντίστοιχα. Τέλος, στην ενότητα 1.6 περιγράφεται η δομή του υπόλοιπου κειμένου.

1.1 Χρήσιμοι ορισμοί

Σε αυτή την ενότητα καταγράφονται μερικοί ορισμοί εννοιών, οι οποίες χρησιμοποιούνται στο κείμενο και μπορεί να μην είναι οικείες στον αναγνώστη.

1. *Αδόμητο/ανεπεξέργαστο κείμενο (unstructured/unprocessed text)* είναι ένα κείμενο σε φυσική γλώσσα, που δεν έχει υποστεί καμία, αυτοματοποιημένη ή μη, κατεργασία και δεν εμπίπτει σε κάποιο μοντέλο δεδομένων. Αν δεν αναφέρεται διαφορετικά, η λέξη *κείμενο* χρησιμοποιείται για συντομία.

2. *Ονοματική οντότητα (named entity)*, στο πεδίο της εξαγωγής πληροφορίας, είναι μια οντότητα του κόσμου, όπως ένα πρόσωπο, μια τοποθεσία, ένας οργανισμός ή ένα προϊόν, στην οποία μπορούμε να αναφερθούμε με το όνομά της, ενώ μπορεί να έχει ή να μην έχει φυσική υπόσταση. Για συντομία, μπορεί να χρησιμοποιείται εναλλακτικά μόνο η λέξη *οντότητα,* χωρίς τον χαρακτηρισμό *ονοματική.* Για παράδειγμα, στην πρόταση «Trump is the president of the United States», οι «Trump» και «United States» είναι ονοματικές οντότητες, διότι αναφέρονται σε συγκεκριμένα αντικείμενα του κόσμου. Ωστόσο, δεν μπορούμε να πούμε το ίδιο για τη λέξη «president», καθώς μπορεί να αναφέρεται σε διαφορετικούς προέδρους διαφορετικών χωρών σε διαφορετικές χρονικές περιόδους. Οι συμβολοσειρές «Trump» και «United States» καλούνται και *αναφορές (mentions)* ονοματικών οντοτήτων ή *ονοματικές αναφορές*. Αυτό συμβαίνει ώστε να υπάρχει διάκριση μεταξύ της αναφοράς στο κείμενο και της οντότητας του κόσμου στην οποία αυτή αντιστοιχεί. Ωστόσο, αν δεν υπάρχει κίνδυνος σύγχυσης, μπορεί να

χρησιμοποιείται ο όρος *ονοματική οντότητα* ή *οντότητα* τόσο για την αναφορά σε ένα κείμενο όσο και για την οντότητα του κόσμου στην οποία αυτή αντιστοιχεί.

3. *Βάση γνώσης (knowledge base)*, στο πλαίσιο αυτής της εργασίας, είναι μια δομή που αποθηκεύει πληροφορίες για τις ονοματικές οντότητες του κόσμου. Περιέχει δεδομένα που περιγράφουν τις οντότητες και τις μεταξύ τους σχέσεις, με βαθμούς κάλυψης που εξαρτώνται από την εκάστοτε υλοποίηση. Οι βάσεις γνώσης είναι τα εργαλεία με τα οποία οι υπολογιστές αποκτούν μια εικόνα για τις οντότητες του κόσμου.

4. *Αναγνώριση ονοματικών οντοτήτων/αναγνώριση οντοτήτων/ταυτοποίηση οντοτήτων/ εξαγωγή οντοτήτων (named entity recognition/NER/entity identification/entity chunking/entity extraction)* είναι η διαδικασία αναγνώρισης σε αδόμητο κείμενο των συμβολοσειρών που αναφέρονται σε ονοματικές οντότητες. Αυτή η διαδικασία συνήθως περιλαμβάνει και την κατάταξη των ονοματικών οντοτήτων σε μια αφηρημένη ταξινομία τύπων. Σημειώνεται ότι η αναγνώριση ονοματικών οντοτήτων προσδιορίζει τις ονοματικές οντότητες σε αδόμητο κείμενο, αλλά δεν τις αντιστοιχίζει στις οντότητες κάποιας βάσης γνώσης.

5. *Αποσαφήνιση ονοματικών οντοτήτων/αποσαφήνιση οντοτήτων (named entity disambiguation/NED/named entity linking)* είναι η διαδικασία αντιστοίχισης των ονοματικών οντοτήτων ενός αδόμητου κειμένου σε οντότητες κάποιας βάσης γνώσης. Σε αντίθεση με την αναγνώριση ονοματικών οντοτήτων, η αποσαφήνιση προσδιορίζει τις οντότητες της βάσης γνώσης, και κατ' επέκταση του κόσμου, στις οποίες αντιστοιχούν οι ονοματικές αναφορές. Για συντομία, μπορεί να χρησιμοποιείται απλά ο όρος *αποσαφήνιση*.

6. *NERD (Named Entity Recognition & Disambiguation)* είναι ο συνδυασμός των λειτουργιών της αναγνώρισης ονοματικών οντοτήτων σε κείμενο και της αποσαφήνισης των οντοτήτων αυτών.

7. *API (Application Programming Interface)* είναι ένα σύνολο συγκεκριμένων μεθόδων που μπορούν να χρησιμοποιηθούν για επικοινωνία με ένα εκτεταμένο σύστημα. Εδώ ενδιαφέρουν, μεταξύ άλλων, τα *RESTful (representational state transfer)* APIs, τα οποία επικοινωνούν με HTTP messages.

8. *Dataset*, στο πλαίσιο αυτής της εργασίας, είναι ένα σύνολο κειμένων που χρησιμοποιείται για την αξιολόγηση ενός συστήματος αποσαφήνισης οντοτήτων. Στα κείμενα ενός dataset είναι σημειωμένη η απάντηση που αναμένεται από το σύστημα. Η χρήση του ίδιου dataset σε διαφορετικά συστήματα αποσαφήνισης οντοτήτων επιτρέπει τη σύγκρισή τους.

9. *Ground truth* είναι ένας χαρακτηρισμός που αφορά ένα dataset. Περιγράφει την ακρίβεια των δεδομένων που περιέχονται σε αυτό. Στο πλαίσιο αυτής της εργασίας, το *ground truthing* σημαίνει ότι για τα κείμενα ενός dataset συλλέχθηκαν αντικειμενικά δεδομένα για τις οντότητες που περιέχουν. Αυτή η διαδικασία κατά κανόνα περιλαμβάνει ανθρώπινη επίβλεψη.

1.2 Προκλήσεις κατά την αποσαφήνιση οντοτήτων

Οι περισσότεροι άνθρωποι έχουν την ικανότητα να επεξεργάζονται χωρίς προσπάθεια τη φυσική γλώσσα και να εξάγουν με ακρίβεια τις οντότητες που περιέχονται σε αδόμητο κείμενο. Αντιθέτως, αυτή η διαδικασία είναι αδύνατο να πραγματοποιηθεί από έναν υπολογιστή, αν δεν καταβληθεί μεγάλη προσπάθεια ώστε να προσομοιωθεί η ανθρώπινη συλλογιστική ικανότητα. Ας πάρουμε, για παράδειγμα, την πρόταση «Leo finally won an Oscar this year, for his tremendous performance in The Revenant.», η οποία αναφέρεται στον ηθοποιό Leonardo DiCaprio, που κέρδισε το Academy Award για τον ρόλο του στην ταινία The Revenant. Γίνεται αμέσως φανερό γιατί η αφαίρεση της αμφισημίας, που γίνεται από έναν άνθρωπο που παρακολουθεί κινηματογράφο σχεδόν υποσυνείδητα, είναι τόσο δύσκολη για ένα σύστημα με περιορισμένη συλλογιστική ικανότητα.

Για αρχή, δεν είναι προφανές το πώς θα εξαχθούν οι οντότητες προς αποσαφήνιση. Το πρώτο βήμα προς αυτήν την κατεύθυνση είναι η συντακτική ανάλυση της πρότασης, αλλά αυτό δεν αρκεί. Πράγματι, ακόμα και για αυτό το προπαρασκευαστικό στάδιο της αποσαφήνισης οντοτήτων απαιτείται η χρήση ισχυρών εργαλείων, που θα εξετασθούν στη συνέχεια της εργασίας. Όμως, εδώ θα επικεντρωθούμε στο κύριο πρόβλημα, αυτό της αμφισημίας των αναφορών.

Η αμφισημία στο κείμενο δεν περιορίζεται σε καμία περίπτωση στις ονοματικές οντότητες. Πράγματι, σύμφωνα με τη λεξικολογική βάση δεδομένων WordNet, μπορούμε να κάνουμε τις ακόλουθες παρατηρήσεις σχετικά με κάποιες από τις υπόλοιπες λέξεις, που αποτελούν τα συμφραζόμενα (context) των ονοματικών αναφορών:

- Η λέξη won, που έχει ως λήμμα τη λέξη win, μπορεί να χρησιμοποιηθεί όταν κάποιος κερδίζει μια διάκριση, όπως στη συγκεκριμένη περίπτωση, αλλά επίσης όταν κάποιος κερδίζει ένα χρηματικό έπαθλο.

- Η λέξη tremendous μπορεί να έχει θετική σημασία, όπως εδώ, αλλά και αρνητική.

- Η λέξη performance μπορεί να αναφέρεται σε μια ερμηνεία ηθοποιού, αλλά και στην εκτέλεση ενός τραγουδιού από κάποιον καλλιτέχνη ή στην επίδοση ενός αθλητή.

Αν και το νόημα των λέξεων που βρίσκονται έξω από τα όρια των ονοματικών αναφορών μπορεί να επηρεάσει τον τρόπο με τον οποίο ένα κείμενο γίνεται αντιληπτό, η μεγαλύτερη πρόκληση έγκειται στην αντιστοίχιση των ονοματικών αναφορών σε οντότητες του κόσμου:

- Η αναφορά Leo μπορεί να αντιστοιχεί σε διάφορες οντότητες, μεταξύ των οποίων ο ηθοποιός Leonardo DiCaprio, ο ποδοσφαιριστής Lionel Messi, ο αστερισμός του Λέοντα, ο συγγραφέας Leo Tolstoy, το Saint Leo University στη Florida των ΗΠΑ, ένας αριθμός από Πάπες της Καθολικής Εκκλησίας, και πολλές άλλες οντότητες.

- Η αναφορά Oscar μπορεί να αντιστοιχεί σε διάφορες οντότητες, μεταξύ των οποίων το χρυσό αγαλματίδιο της Αμερικάνικης Ακαδημίας Κινηματογράφου, ο συγγραφέας Oscar Wilde, ο Βραζιλιάνος ποδοσφαιριστής Oscar, ένα είδος ψαριού, ένας αριθμός από βασιλιάδες της Σουηδίας, και πολλές άλλες οντότητες.

- Η αναφορά The Revenant μπορεί με πρώτη ματιά να φαίνεται μη διφορούμενη (unambiguous), αλλά μια γρήγορη αναζήτηση διαψεύδει αυτήν την πρώτη εντύπωση. Πράγματι, εκτός από τη βραβευμένη ταινία του 2015 με τίτλο «The Revenant», υπάρχει και μια άλλη ταινία του 2009, με τον ίδιο ακριβώς τίτλο. Επίσης, υπάρχει και ένα βιβλίο και μια σειρά comics με τον ίδιο ακριβώς τίτλο.

Όπως γίνεται φανερό, υπάρχει μια σειρά από προκλήσεις που πρέπει να αντιμετωπιστούν ώστε να προσεγγιστεί η ανθρώπινη ικανότητα αποσαφήνισης ονοματικών οντοτήτων σε αδόμητα κείμενα. Το χάσμα που υπάρχει μεταξύ της ανθρώπινης αντίληψης και της ικανότητας ενός αυτόματου συστήματος να προσεγγίζει την αντίληψη αυτή, όσον αφορά την επεξεργασία φυσικής γλώσσας και την εξαγωγή οντοτήτων από ανεπεξέργαστο κείμενο, φαίνεται στην Εικόνα 1. Η προσπάθεια γεφύρωσης του συγκεκριμένου χάσματος αποτελεί το αντικείμενο της παρούσας εργασίας.

1.3 Κίνητρα για αποσαφήνιση οντοτήτων

Όπως περιγράφηκε στην ενότητα 1.2, η αποσαφήνιση οντοτήτων σε αδόμητα κείμενα οδηγεί σε αρκετές προκλήσεις ακόμα και σε μικρά κείμενα που βγάζουν νόημα για τους περισσότερους αναγνώστες. Ωστόσο, η ερευνητική εργασία σε αυτόν τον τομέα είναι έντονη τα τελευταία χρόνια. Αυτό συμβαίνει διότι τα δεδομένα στο διαδίκτυο αυξάνονται με ταχύτατους ρυθμούς, και η πλειονότητα αυτής της πληροφορίας είναι παγιδευμένη σε μορφή φυσικής γλώσσας, που είναι αναπόφευκτα αμφίσημη και διφορούμενη. Μεγάλο μέρος αυτής της αμφισημίας πηγάζει από τις αναφορές σε ονοματικές οντότητες. Η άρση της εν λόγω αμφισημίας μέσω αποσαφήνισης ονοματικών οντοτήτων μπορεί να συνεισφέρει σε πλήθος εφαρμογών. Κάποιες από τις εφαρμογές που αποτελούν τα κίνητρα για αποσαφήνιση οντοτήτων σε κείμενο απαριθμούνται σε αυτήν την ενότητα.

1.3.1 Ανάκληση πληροφορίας

Η καθιερωμένη μέθοδος ανάκλησης πληροφορίας στο διαδίκτυο έχει να κάνει με αναζήτηση βάσει λέξεων-κλειδιών (keyword-based search). Ωστόσο, μια βελτίωση αυτής της μεθόδου που έχει μελετηθεί στις εργασίες [CYC07, GXCL09, DIdV10, BSV10, BML13] αφορά την εξέλιξή της σε σημασιολογική αναζήτηση βάσει οντοτήτων (semantic entity-based search). Αυτή η εξέλιξη δεν μπορεί να γίνει χωρίς την ύπαρξη μιας αξιόπιστης μεθόδου αποσαφήνισης οντοτήτων, καθώς είναι αναγκαίος ο εμπλουτισμός του σώματος κειμένων του διαδικτύου με μη-διφορούμενες αναφορές. Αυτό, βέβαια, είναι ένα τιτάνιο έργο, αλλά θα επιτρέψει την ανάκληση πληροφορίας στο διαδίκτυο με όρους που είναι πιο κοντά στην ανθρώπινη νόηση. Για παράδειγμα, όταν ένας χρήστης αναζητά τον όρο «Jaguar», προσδιορίζοντας ότι ενδιαφέρεται για το ζώο, δε θα πρέπει να επιστρέφονται αποτελέσματα για τη μάρκα αυτοκινήτων. Αυτό είναι αδύνατο να πραγματοποιηθεί με χρήση αφελούς αναζήτησης με λέξεις-κλειδιά, και γίνεται μόνο εάν ένας σημασιολογικά εμπλουτισμένος ιστός είναι διαθέσιμος.

1.3.2 Εξαγωγή χαρακτηριστικών κειμένου

Η μακροσκοπική ανάλυση ενός κειμένου, δηλαδή η εξαγωγή των θεματικών ενοτήτων και των ιδεών που πραγματεύεται, μπορεί να ωφεληθεί σε μεγάλο βαθμό από την αποσαφή-

Εικόνα 1: Το παράδειγμα της ενότητας 1.2. Παρουσιάζεται εποπτικά η πληροφορία που αντλεί ένας άνθρωπος από τη συγκεκριμένη πρόταση σε σύγκριση με την πληροφορία που αντλεί ένας υπολογιστής, που χρησιμοποιεί μια βασική βιβλιοθήκη διαχείρισης strings.

Unstructured Text

Leo finally won an Oscar this year, for his tremendous performance in The Revenant.

What a human can make of the text at a glance

Leo *finally won an* ***Oscar*** *this year, for his tremendous performance in* ***The Revenant.***

What a computer can make of the text at a glance

```
total_number_of_characters == 83    total_number_of_words == 14

number_of_capitalized_words == 4  number_of_punctuation_marks == 2
```

• • •

νιση των οντοτήτων που περιέχει. Στη συνέχεια, αυτή η πληροφορία μπορεί να αξιοποιηθεί με διάφορους τρόπους. Ένα παράδειγμα αποτελεί η παροχή εξατομικευμένων υπηρεσιών. Αν ένα ειδησεογραφικό site διαπιστώσει ότι ένας χρήστης δημοσίευσε στον λογαριασμό του στο Twitter την πρόταση που αποτελεί το παράδειγμα της ενότητας 1.2 και καταφέρει να αποσαφηνίσει σωστά τις ονοματικές οντότητες που αυτή περιέχει, τότε μπορεί να εξάγει με ασφάλεια το συμπέρασμα ότι το κείμενο αυτό αφορά τον κινηματογράφο και τη βράβευση ενός ηθοποιού από την Αμερικάνικη Ακαδημία Κινηματογράφου. Αυτή η γνώση μπορεί να χρησιμοποιηθεί ώστε να προταθεί στον συγκεκριμένο χρήστη ένα άρθρο που αφορά τη βραδιά των βραβείων Oscar, ή τις πιο πετυχημένες ταινίες της χρονιάς, καθώς προβλέπεται ότι αυτά τα θέματα θα ανήκουν στα ενδιαφέροντά του. Με λίγα λόγια, η αποσαφήνιση κειμένων μπορεί να βοηθήσει ώστε μια πρόταση λίγων χαρακτήρων να αρκεί σε ένα recommender system για να να πάρει μια απόφαση για το τι θα προτείνει σε κάποιο άτομο με ικανοποιητικό βαθμό βεβαιότητας. Αυτή η πλευρά της αποσαφήνισης κειμένου μελετήθηκε στις εργασίες [PMS09, LDP10, WLJH10, MM10, GLG+13].

1.3.3 Απάντηση ερωτήσεων

Η αποσαφήνιση οντοτήτων μπορεί να ενισχύσει συστήματα απάντησης ερωτήσεων. Η αποσαφήνιση του ερωτήματος και η ανακάλυψη των οντοτήτων που περιέχει παίζει βασικό ρόλο στην ακρίβεια των απαντήσεων που λαμβάνονται. Ένα σύστημα μπορεί να περιέχει μια πλούσια βάση γνώσης που περιέχει την απάντηση σε κάθε ερώτηση που μπορεί να σκεφτεί κανείς, αλλά να αποτυγχάνει να επιλέξει τη σωστή απάντηση, διότι δεν κατάφερε να αποσαφηνίσει την οντότητα που ζητείται. Για παράδειγμα, είναι αρκετά συχνό οι θαυμαστές των διάσημων προσώπων να αναζητούν την ηλικία τους. Έτσι, ένας χρήστης μπορεί να ρωτήσει «How old is singer Demi?» ρωτώντας για την ηλικία της τραγουδίστριας Demi Lovato. Το σύστημα θα πρέπει να αποσαφηνίσει ότι η ερώτηση αφορά την τραγουδίστρια και όχι κάποιο άλλο άτομο (π.χ. την ηθοποιό Demi Moore), και να επιστρέψει από τη βάση γνώσης την ηλικία της συγκεκριμένης οντότητας. Διαφορετικά, θα επιστραφεί λάθος αποτέλεσμα. Η ζητούμενη πληροφορία πιθανότατα υπάρχει στην υποκείμενη βάση γνώσης, αλλά θα έχει χαθεί στη μετάφραση από την αδόμητη φυσική γλώσσα του query στη δομημένη γλώσσα της βάσης γνώσης. Η αποσαφήνιση κειμένου με στόχο την ενίσχυση συστημάτων απάντησης ερωτήσεων μελετήθηκε στο πλαίσιο του συστήματος απάντησης ερωτήσεων Watson της IBM[WMKF12].

1.3.4 Εμπλουτισμός βάσης γνώσης

Ο εμπλουτισμός βάσης γνώσης είναι ένα ακόμα πεδίο που μπορεί να επωφεληθεί από την αξιόπιστη αποσαφήνιση οντοτήτων. Αυτό το πεδίο είναι συμπληρωματικό της απάντησης ερωτήσεων, που αναλύθηκε παραπάνω. Για να διατηρείται μια βάση γνώσης ενημερωμένη, ειδικά με τους σημερινούς ρυθμούς παραγωγής δεδομένων, θα πρέπει να ενσωματώνει τακτικά μεγάλους όγκους από νέα δεδομένα. Αυτά τα δεδομένα, σε μεγάλο μέρος, προέρχονται από κείμενα γραμμένα σε φυσική γλώσσα, τα οποία πρέπει να αποσαφηνιστούν πριν μπορέσουν να εισαχθούν στη βάση. Συνεχίζοντας το παράδειγμα της παραγράφου 1.3.3, κατά την εξόρυξη δεδομένων μπορεί να βρεθεί η πρόταση «Popular singer Demi turns 24 today». Είναι αναγκαία η ορθή αποσαφήνιση της αναφοράς Demi ώστε να αντιστοιχεί στην τραγου-

δίστρια Demi Lovato, διαφορετικά μπορεί να εισαχθεί λανθασμένα στη βάση ότι κάποιο άλλο άτομο έχει αυτήν την ηλικία. Είναι φανερό ότι αν η επιστροφή λανθασμένης απάντησης σε κάποια ερώτηση λόγω αποτυχημένης αποσαφήνισης των οντοτήτων που την απαρτίζουν είναι ένα αρνητικό ενδεχόμενο, η μόλυνση της βάσης γνώσης με λανθασμένες πληροφορίες για τους ίδιους λόγους είναι κάτι που πρέπει να αποφεύγεται με κάθε τρόπο. Η αποσαφήνιση οντοτήτων σε κείμενο με στόχο τον εμπλουτισμό βάσης γνώσης μελετήθηκε στην εργασία [DMR+10].

1.4 Εύρος της εργασίας

Αυτή η εργασία επικεντρώνεται στη σχεδίαση και υλοποίηση ενός ευέλικτου συστήματος αποσαφήνισης οντοτήτων σε αδόμητο κείμενο. Μελετώνται οι τρόποι με τους οποίους μπορούν να αναγνωριστούν ονοματικές οντότητες σε κείμενο, και έπειτα να αντιστοιχιστούν σε οντότητες του κόσμου. Για τον σκοπό αυτόν, χρησιμοποιούνται ανοιχτά εργαλεία και αξιόπιστες τεχνικές. Το τελικό σύστημα παραμετροποιείται, ώστε να μπορεί να δώσει ποιοτικά αποτελέσματα σε κάθε κείμενο που χαρακτηρίζεται από έναν σφιχτό σημασιολογικό πυρήνα. Τέλος, το σύστημα ενσωματώνεται σε μια διαδικτυακή εφαρμογή αποσαφήνισης, ώστε η δοκιμή του να γίνεται με τρόπο φιλικό στο χρήστη.

1.5 Συνεισφορές της εργασίας

Όπως θα φανεί στο επόμενο κεφάλαιο, υπάρχει πληθώρα ερευνητικών εργασιών που αντιμετωπίζουν το πρόβλημα της αποσαφήνισης ονοματικών οντοτήτων σε κείμενα. Οπότε προκύπτει το ερώτημα, «Γιατί χρειάζεται μια ακόμα λύση για ένα πρόβλημα που, φαινομενικά, έχει λυθεί ήδη;». Εδώ αναφέρονται τα σημεία του ερευνητικού πεδίου στα οποία συνεισφέρει η παρούσα εργασία.

- **Σύγχρονα εργαλεία:** Η έρευνα στους τομείς που είναι σχετικοί με την αποσαφήνιση οντοτήτων προσφέρει συνεχώς στους ερευνητές νέα εργαλεία με τα οποία μπορούν να αντιμετωπίσουν τις προκλήσεις που αναφέρθηκαν στην ενότητα 1.2. Σε αυτήν την εργασία αξιοποιούνται εργαλεία και υπηρεσίες που έγιναν διαθέσιμες τα τελευταία 1-2 χρόνια, συνεπώς παρέχεται ένα μέτρο της χρησιμότητάς τους στο συγκεκριμένο πεδίο εφαρμογής.

- **Online δεδομένα:** Σε πολλές από τις προηγούμενες εργασίες η αποσαφήνιση γίνεται με χρήση δεδομένων που συλλέγονται από πολλαπλές πηγές και επεξεργάζονται σε μια χρονοβόρα και υπολογιστικά ακριβή διαδικασία. Στην παρούσα εργασία όλα τα δεδομένα μπορούν να συλλεχθούν από RESTful APIs. Έτσι, η εφαρμογή είναι πιο εύκολο να διατηρηθεί ενημερωμένη.

- **Ακρίβεια αποσαφήνισης:** Όπως θα φανεί από την αξιολόγηση του συστήματος αποσαφήνισης, αυτό υπερέχει των state-of-the-art συστημάτων και των συστημάτων της βιβλιογραφίας, ειδικά όσον αφορά την αποσαφήνιση κειμένων με υψηλή σημασιολογική συνοχή. Αυτό το καθιστά καταλληλότερο σε εφαρμογές όπου αναμένονται κείμενα που δεν απλώνονται σε πολλαπλές θεματικές περιοχές.

1.6 Διάρθρωση του κειμένου

Στο κεφάλαιο 2 χτίζεται το θεωρητικό υπόβαθρο που είναι αναγκαίο για την κατανόηση ενός συστήματος αποσαφήνισης, και γίνεται εκτεταμένη αναφορά στη σχετική βιβλιογραφία. Στο κεφάλαιο 3 γίνεται αναφορά στα κύρια APIs που χρησιμοποιούνται στο πλαίσιο της εργασίας, με τις αναγκαίες τεχνικές λεπτομέρειες. Στο κεφάλαιο 4 περιγράφεται η κατασκευή ενός νέου συστήματος αποσαφήνισης. Στο κεφάλαιο 5 αναφέρονται ζητήματα που αφορούν την προγραμματιστική υλοποίηση του συστήματος αποσαφήνισης. Στο κεφάλαιο 6 περιέχεται η πειραματική αξιολόγηση του συστήματος αποσαφήνισης και ο σχολιασμός των πειραματικών αποτελεσμάτων. Τέλος, στο κεφάλαιο 7 καταγράφονται συμπεράσματα που προέκυψαν από την εκπόνηση της παρούσας εργασίας, καθώς και δυνατές επεκτάσεις που μπορούν να πραγματοποιηθούν στο μέλλον.

ΚΕΦΑΛΑΙΟ 2

Θεωρητικό υπόβαθρο και προηγούμενες εργασίες

Στο κεφάλαιο αυτό αναλύεται το θεωρητικό υπόβαθρο της αποσαφήνισης οντοτήτων σε κείμενο, δίνοντας έμφαση στα βασικά δομικά συστατικά και χαρακτηριστικά κάθε συστήματος που καλείται να εκτελέσει τη διαδικασία αυτήν. Γίνονται αναφορές σε προηγούμενες εργασίες και στις επιλογές που έχουν κάνει οι σχεδιαστές τους. Αυτή η ανάλυση θα κάνει ευκολότερη την επιλογή των χαρακτηριστικών ενός νέου συστήματος αποσαφήνισης, δεδομένου του πεδίου εφαρμογής του.

Αρχικά, στην ενότητα 2.1 παρουσιάζεται η μοντελοποίηση του προβλήματος της αποσαφήνισης οντοτήτων σε κείμενο, με στόχο την ακριβέστερη κατανόηση των ιδιοτήτων του. Στην ενότητα 2.2 παρουσιάζονται κάποιες εναλλακτικές επιλογές για βάσεις γνώσης, που αποτελούν τον ακρογωνιαίο λίθο κάθε συστήματος αποσαφήνισης οντοτήτων. Στην ενότητα 2.3 παρατίθενται τα βασικά βήματα που κάθε σύστημα αποσαφήνισης πρέπει να λάβει υπόψιν. Τέλος, στην ενότητα 2.4 εξετάζονται οι τρόποι αξιολόγησης των συστημάτων αποσαφήνισης.

2.1 Μοντελοποίηση του προβλήματος

Δεδομένης μιας βάσης γνώσης KB που περιέχει ένα σύνολο οντοτήτων E και ενός κειμένου T, στο οποίο περιέχεται ένα σύνολο ονοματικών αναφορών M, ο στόχος της αποσαφήνισης οντοτήτων είναι η αντιστοίχιση κάθε αναφοράς $m \in M$ σε μια οντότητα $e \in E$ στη βάση γνώσης. Ένα σημείο ενδιαφέροντος είναι αν το σύνολο M θεωρείται δοσμένο με το κείμενο T ή αν η ανακάλυψή του αποτελεί ευθύνη του συστήματος αποσαφήνισης. Ένα ακόμα ενδιαφέρον σημείο είναι η διαχείριση οντοτήτων που δεν υπάρχουν στη βάση γνώσης, δηλαδή των περιπτώσεων όπου $\exists m \in M : m \mapsto e \wedge e \notin E$. Αυτά τα σημεία σχολιάζονται στην ενότητα 2.4 και στην παράγραφο 2.3.5 αντίστοιχα.

2.2 Βάσεις γνώσης

Μια βάση γνώσης (knowledge base) αποτελεί αναπόσπαστο κομμάτι κάθε συστήματος αποσαφήνισης οντοτήτων, καθώς παρέχει τις οντότητες του κόσμου στις οποίες θα προσπαθήσουμε να αντιστοιχίσουμε τις ονοματικές αναφορές στο κείμενο. Ο ρόλος της βάσης

γνώσης είναι να παρέχει πληροφορίες για τις οντότητες του κόσμου. Οι πληροφορίες αυτές μπορεί να είναι τα εναλλακτικά ονόματα των οντοτήτων, οι τύποι τους, οι σχέσεις μεταξύ τους, και άλλα γνωρίσματα που μπορούν να βοηθήσουν στην ταυτοποίησή τους όταν εμφανίζονται ως αναφορές σε αδόμητα κείμενα. Εδώ παρατίθενται κάποια παραδείγματα βάσεων γνώσης που είναι κατάλληλες ως πηγές πληροφορίας για αποσαφήνιση οντοτήτων.

- **Wikipedia:** Η Wikipedia είναι μια δωρεάν εγκυκλοπαίδεια στο διαδίκτυο, που είναι διαθέσιμη σε πολλές γλώσσες. Δημιουργήθηκε και διατηρείται ενημερωμένη από εθελοντές παντού στον κόσμο, και αποτελεί το μεγαλύτερο crowdsourcing project της ιστορίας. Σήμερα, είναι η πιο δημοφιλής και εκτεταμένη εγκυκλοπαίδεια του διαδικτύου, έχοντας τη στιγμή συγγραφής αυτού του κειμένου 5,429,948 άρθρα στην αγγλική γλώσσα. Η βασική καταχώρηση της Wikipedia είναι ένα άρθρο, που αναφέρεται σε μια συγκεκριμένη οντότητα και ξεχωρίζει από τα υπόλοιπα άρθρα μέσω μοναδικών αναγνωριστικών, ένα από τα οποία είναι ο πλήρης τίτλος του άρθρου. Η Wikipedia έχει καταχωρήσεις για έναν μεγάλο αριθμό οντοτήτων, που περιέχουν πληροφορίες χρήσιμες στη διαδικασία της αποσαφήνισης. Επίσης, όπως θα φανεί στη συνέχεια, η δομή της Wikipedia έχει ένα σύνολο χαρακτηριστικών που μπορούν να αξιοποιηθούν από συστήματα αποσαφήνισης: κατηγορίες άρθρων, redirect pages, disambiguation pages και συνδέσμους που οδηγούν από το ένα άρθρο στο άλλο.

- **DBpedia:** Η DBpedia είναι μια βάση γνώσης που στηρίζεται στην Wikipedia ώστε να κατασκευάσει ένα πιο δομημένο σύνολο δεδομένων. Πράγματι, η Wikipedia περιέχει κατά κόρον αδόμητα δεδομένα και η περισσότερη πληροφορία βρίσκεται στο κυρίως σώμα των άρθρων της, σε φυσική γλώσσα. Η DBpedia αναλαμβάνει να συγκεντρώσει δεδομένα όπως πληροφορίες των infoboxes, καθώς και στοιχεία όπως κατηγορίες των άρθρων και συνδέσμους προς εξωτερικές ιστοσελίδες. Σήμερα η DBpedia περιέχει 4.58 εκατομμύρια καταχωρήσεις οντοτήτων, από τις οποίες τα 4.22 εκατομμύρια είναι οργανωμένα σε μια δομημένη οντολογία. Στην εργασία [ABK+07] περιγράφεται το project της DBpedia και κάποιες από τις εφαρμογές του.

- **YAGO:** Η YAGO είναι μια εκτεταμένη βάση γνώσης που δημιουργήθηκε συνδυάζοντας την πληροφορία της Wikipedia αλλά και της λεξικολογικής βάσης δεδομένων WordNet. Συγκεκριμένα, δανείζεται από την Wikipedia το πλούσιο σύνολο των οντοτήτων και από την WordNet την καθαρή ταξινομία των εννοιών. Περιέχει εκατομμύρια οντότητες και δεκάδες εκατομμυρίων κατηγορήματα σχετικά με αυτές. Η βάση γνώσης YAGO χρησιμοποιήθηκε από τον Hoffart και τους συνεργάτες του κατά τον σχεδιασμό του δικού τους συστήματος αποσαφήνισης οντοτήτων [HYB+11].

- **Wikidata:** Η Wikidata είναι μια βάση γνώσης που αποτελεί το ημι-δομημένο αντίστοιχο της Wikipedia. Τα δύο projects έχουν γενικά τα ίδια χαρακτηριστικά της αποκεντρωμένης και συλλογικής διαχείρισης, και διατηρούνται από τον ίδιο οργανισμό (Wikimedia Foundation). Η Wikidata αποτελείται από μια σειρά κειμένων (documents), που ονομάζονται αντικείμενα (items). Αυτά τα αντικείμενα μπορεί να αφορούν από συγκεκριμένα πρόσωπα, όπως ένας πολιτικός, μέχρι αφηρημένες έννοιες, όπως η πολιτική. Σήμερα το Wikidata item repository περιέχει πάνω από 25 εκατομμύρια αντικείμενα.

- **Freebase:** Η Freebase είναι μια ακόμα βάση γνώσης που αναπτύχθηκε από τα μέλη της κοινότητάς της και αγοράστηκε από την Google. Ένας από τους βασικούς στόχους του project ήταν η δημιουργία ενός εύκολα προσβάσιμου συνόλου γνώσης. Η Freebase σταμάτησε να λειτουργεί τον Αύγουστο του 2016. Τα δεδομένα της ενσωματώθηκαν σε μια σειρά από άλλες βάσεις γνώσης, μεταξύ των οποίων και η Wikidata. Η Google αντικατέστησε τη Freebase με τον Γράφο Γνώσης της, που περιγράφεται στη συνέχεια.

- **Google Knowledge Graph:** Ο Google Knowledge Graph είναι η βάση γνώσης που διατηρεί η Google, και χρησιμοποιείται ως βασικό συστατικό της μηχανής αναζήτησης της εταιρείας. Ο Google Knowledge Graph αντλεί τις πληροφορίες του από πολλές πηγές, όπως το CIA World Factbook, την Wikidata και την Wikipedia. Το 2012 ο σημασιολογικός ιστός της Google περιείχε 570 εκατομμύρια αντικείμενα και πάνω από 18 δισεκατομμύρια γεγονότα και σχέσεις μεταξύ τους, που χρησιμοποιούνται για την κατανόηση των όρων αναζήτησης. Η Google ανακοίνωσε τον Οκτώβριο του 2016 ότι ο Knowledge Graph που διατηρεί πλέον περιέχει πάνω από 70 δισεκατομμύρια γεγονότα. Από τον Δεκέμβριο του 2015 υπάρχει ένα API που μπορεί να χρησιμοποιηθεί για αναζήτηση οντοτήτων στον γράφο της Google. Η συγκεκριμένη βάση γνώσης θα αναλυθεί περαιτέρω στη συνέχεια της εργασίας.

2.3 Βασικά βήματα κατά την αποσαφήνιση οντοτήτων

2.3.1 Επιλογή γλώσσας ενδιαφέροντος

Σε έναν ιδανικό κόσμο, η ανάπτυξη των μεθόδων αποσαφήνισης ονοματικών οντοτήτων σε κείμενα θα ήταν ανεξάρτητη της γλώσσας των εν λόγω κειμένων. Ωστόσο, οι διαθέσιμες βάσεις γνώσης, που αποτελούν την καρδιά κάθε συστήματος αποσαφήνισης, έχουν την πλειονότητα της πληροφορίας τους στην αγγλική γλώσσα. Χαρακτηριστικό παράδειγμα αποτελεί η ελληνική Wikipedia, που διαθέτει περίπου 130,000 άρθρα, δηλαδή περίπου το 2% των άρθρων της αντίστοιχης αγγλικής έκδοσης. Επίσης, πολλά εργαλεία που είναι απαραίτητα στη διαδικασία της αποσαφήνισης λειτουργούν καλύτερα στα Αγγλικά, με περιορισμένες δυνατότητες στις άλλες γλώσσες. Για αυτούς τους λόγους, η συντριπτική πλειονότητα των ερευνητικών εργασιών στον τομέα της αποσαφήνισης οντοτήτων αφορούν την αγγλική γλώσσα. Αξίζει να σημειωθεί ότι και αυτός είναι ένας λόγος που ωθεί τους μελλοντικούς ερευνητές στην ίδια κατεύθυνση, βάσει της αρχής ότι «οι πλούσιοι γίνονται πλουσιότεροι, και οι φτωχοί γίνονται φτωχότεροι». Η ύπαρξη πλούσιας βιβλιογραφίας για αγγλικά κείμενα αποτελεί βάση για περαιτέρω έρευνα και προσφέρει δυνατότητες σύγκρισης με πολυάριθμα συστήματα. Αντιθέτως, το φτωχό ερευνητικό υπόβαθρο που υπάρχει για λιγότερο δημοφιλείς γλώσσες αποθαρρύνει την επιστημονική δραστηριότητα σε αυτήν την κατεύθυνση.

2.3.2 Αναγνώριση ονοματικών οντοτήτων

Ένα βασικό προπαρασκευαστικό στάδιο για την αποσαφήνιση οντοτήτων σε κάποιο αδόμητο κείμενο είναι η αναγνώριση των αναφορών που αντιστοιχούν σε αυτές. Αυτή η διαδικασία είναι γνωστή ως αναγνώριση ονοματικών οντοτήτων (Named Entity Recognition/NER). Αν και δε βρίσκεται στον πυρήνα ενός συστήματος αποσαφήνισης, το υποσύ-

στημα αναγνώρισης οντοτήτων μπορεί να επηρεάσει σημαντικά την απόδοση ενός συστήματος που έχει ως στόχο την επεξεργασία κειμένων στα οποία δεν έχουν σημειωθεί οι ονοματικές οντότητες εκ των προτέρων.

Γενικά, η αναγνώριση των ονοματικών οντοτήτων περιλαμβάνει δύο υποπροβλήματα, τον εντοπισμό των οντοτήτων στο κείμενο (εύρεση των ορίων στο κείμενο για τις αναφορές που αποτελούν ονοματικές οντότητες), και την ταξινόμησή τους σε κάποιες βασικές κατηγορίες, όπως πρόσωπο, τοποθεσία ή οργανισμός. Έχουν χρησιμοποιηθεί διάφορες προσεγγίσεις με στόχο την αναγνώριση οντοτήτων, από τεχνικές βασισμένες στους γραμματικούς κανόνες της εκάστοτε γλώσσας μέχρι και μηχανική μάθηση. Μηχανική μάθηση χρησιμοποίησαν οι Milne και Witten στην κλασική εργασία τους [MW08], με στόχο να ανακαλύψουν σημαντικούς όρους σε ανεπεξέργαστα κείμενα, τους οποίους στη συνέχεια προσπάθησαν να αντιστοιχίσουν με άρθρα της Wikipedia.

Όπως εξηγήθηκε παραπάνω, η αναγνώριση ονοματικών οντοτήτων μπορεί να αναπτυχθεί παράλληλα με το σύστημα αποσαφήνισης. Αυτή είναι η προσέγγιση που παρέχει τη μεγαλύτερη ευελιξία, καθώς επιτρέπει τον σχεδιασμό όλων των κομματιών με γνώμονα το πεδίο εφαρμογής του συστήματος. Ωστόσο, είναι ελκυστική και η χρήση ενός ήδη διαδεδομένου και αξιόπιστου συστήματος που εκτελεί την αναγνώριση ονοματικών οντοτήτων. Αυτό επιτρέπει στους προγραμματιστές να συγκεντρωθούν στο κύριο έργο της αποσαφήνισης και απεμπλέκει τις εργασίες της αναγνώρισης και της αποσαφήνισης οντοτήτων. Η συγκεκριμένη σχεδιαστική επιλογή καθιστά ομαλότερη τη διαδικασία ανάπτυξης και αξιολόγησης του συνολικού συστήματος. Σήμερα υπάρχουν αρκετά δοκιμασμένα συστήματα επεξεργασίας φυσικής γλώσσας που περιέχουν εργαλεία αναγνώρισης ονοματικών οντοτήτων. Παραδείγματα τέτοιων συστημάτων αποτελούν τα GATE, OpenNLP και Stanford Named Entity Recognizer. Το τελευταίο από τα συστήματα αυτά χρησιμοποιήθηκε με επιτυχία από τους Hoffart et al. [HYB+11]

2.3.3 Παραγωγή υποψήφιων οντοτήτων

Αυτό το βήμα της αποσαφήνισης οντοτήτων απαιτεί να έχει προηγηθεί η αναγνώριση ονοματικών οντοτήτων, όπως περιγράφηκε στην παράγραφο 2.3.2. Μιλώντας με τους όρους της ενότητας 2.1, για κάθε αναφορά $m \in M$, θέλουμε να δημιουργήσουμε ένα σύνολο $E_m \subset E$ από υποψήφιες οντότητες που θα μπορούσαν να αντιστοιχούν στη συγκεκριμένη αναφορά. Σε γενικές γραμμές, το σύνολο E_m δημιουργείται διαλέγοντας τις οντότητες από το σύνολο E στις οποίες θα μπορούσε να αναφερθεί κανείς λέγοντας m, ή $E_m = \{e \in E \mid e \text{ could be referred to by saying } m\}$. Ωστόσο, η επιλογή των οντοτήτων που θα μπορούσαν να αντιστοιχούν σε μια αναφορά οδηγεί σε δυσκολίες, και επιπλέον μπορεί να επηρεάσει δραστικά το αποτέλεσμα της αποσαφήνισης [HRN+13]. Στη συνέχεια περιγράφονται οι βασικές προσεγγίσεις που έχουν ακολουθηθεί για την παραγωγή υποψήφιων οντοτήτων στο πλαίσιο της αποσαφήνισης οντοτήτων σε κείμενο.

2.3.3.1 Χρήση λεξικών ονομάτων Στην Επιστήμη Υπολογιστών, ένα λεξικό (dictionary) είναι μια αφηρημένη δομή δεδομένων αποτελούμενη από μια συλλογή ζευγών κλειδιού-τιμής $(key, value)$, όπου το κάθε κλειδί μπορεί να εμφανίζεται το πολύ μία φορά. Για την επίλυση του προβλήματος εύρεσης υποψήφιων οντοτήτων, μπορεί να χρησιμοποιηθεί ένα λεξικό ονομάτων D, όπου για κλειδιά θα έχουμε τις αναφορές m που μπορούν να εμφανίζονται σε

Αλγόριθμος 1: Αφελής τρόπος παραγωγής λεξικού ονομάτων D, με στόχο την αντιστοίχιση αναφορών με τις οντότητες της βάσης γνώσης στις οποίες θα μπορούσαν να αναφέρονται.

Require: set of entities E in knowledge base KB

```
for all conceivable mentions m do
    D[m] ← ∅
    for all e ∈ E do
        if m could possibly refer to e then
            D[m] ← D[m] ∪ {e}
        end if
    end for
end for
```

ένα κείμενο και σε κάθε κλειδί θα αντιστοιχεί το σύνολο E_m των οντοτήτων στις οποίες η m θα μπορούσε να αναφέρεται. Με άλλα λόγια, ισχύει $D[m] = E_m$. Η χρήση λεξικού ονομάτων, η κατασκευή του οποίου περιγράφεται αφηρημένα από τον Αλγόριθμο 1, είναι η πιο διαδεδομένη μέθοδος παραγωγής υποψήφιων οντοτήτων σε συστήματα αποσαφήνισης οντοτήτων σε κείμενο [BP06, Cuc07, HZ09, KSRC09, VBR+10, ZSTW10, ZTSS10, CTL+10, ZLHZ10, LMN+10, ZTS+11, Cuc11, ZSST11, MLN+11, HS11, HSZ11, RRDA11, SWLW12b, GLG+13, GCK13, SWLW13].

Ο Αλγόριθμος 1 περιγράφει αφελώς τη διαδικασία κατασκευής του λεξικού D. Δεν υπάρχει, άλλωστε, τρόπος παραγωγής όλων των δυνατών strings που θα μπορούσαν να αποτελούν ονοματικές αναφορές. Στην πράξη, τα συστήματα αποσαφήνισης ξεκινάνε αντίστροφα. Δηλαδή, δεν προσπαθούν να προσδιορίσουν για όλες τις δυνατές ονοματικές αναφορές τις αντίστοιχες καταχωρήσεις στη βάση γνώσης τους. Αντιστρόφως, προσπαθούν να προσδιορίσουν για κάθε οντότητα της βάσης γνώσης τους τα διαφορετικά ονόματα με τα οποία θα μπορούσε να εμφανιστεί. Δηλαδή, ξεκινούν από τις τιμές του λεξικού, και όχι από τα κλειδιά. Αυτή η μέθοδος κατασκευής του λεξικού φαίνεται στον Αλγόριθμο 2.

Μια σοβαρή πρόκληση κατά τη διαδικασία κατασκευής του λεξικού D με βάση τον Αλγόριθμο 2 είναι η αξιόπιστη αντιστοίχιση οντοτήτων της βάσης γνώσης, σύνολα των οποίων είναι οι τιμές του λεξικού, με τα πιθανά ονόματά τους, που αποτελούν τα κλειδιά του λεξικού. Αυτή η αντιστοίχιση θα πρέπει να βασιστεί στην πληροφορία που αποθηκεύεται στη βάση γνώσης που χρησιμοποιείται. Για τον σκοπό αυτόν έχει χρησιμοποιηθεί εκτεταμένα η Wikipedia, καθώς περιέχει ένα σύνολο χρήσιμων χαρακτηριστικών, που μπορούν να αξιοποιηθούν για την κατασκευή του λεξικού ονομάτων. Μερικά από τα βασικότερα τέτοια χαρακτηριστικά ακολουθούν.

- **Τίτλοι άρθρων.** Το βασικό δομικό στοιχείο της Wikipedia είναι τα άρθρα της. Το κάθε άρθρο αναφέρεται σε κάποια συγκεκριμένη οντότητα του κόσμου, την οποία περιγράφει με λεπτομέρεια. Ο τίτλος ενός άρθρου είναι η πιο συχνή ονομασία για την οντότητα στην οποία αναφέρεται το άρθρο. Συνεπώς, βάσει των παραπάνω, αυτή η ονομασία m θα πρέπει να εισαχθεί ως κλειδί του λεξικού D και η αντίστοιχη οντότητα θα πρέπει

Αλγόριθμος 2: Πρακτικός αλγόριθμος κατασκευής λεξικού D που αντιστοιχεί ονοματικές αναφορές σε υποψήφιες οντότητες της βάσης γνώσης.

Require: set of entities E in knowledge base KB

```
for all e ∈ E do
    for all possible names m for entity e according to KB do
        if m exists in D's keys then
            D[m] ← D[m] ∪ {e}
        else
            create new entry in D for name m
            D[m] ← {e}
        end if
    end for
end for
```

να προστεθεί στο σύνολο $D[m]$. Επιστρέφοντας στο παράδειγμα της ενότητας 1.2, το άρθρο της Wikipedia που αφορά τον Leonardo DiCaprio έχει αυτόν ακριβώς τον τίτλο, και άρα θα πρέπει να γίνει καταχώρηση στο λεξικό D που να αντιστοιχεί την αναφορά «Leonardo DiCaprio», μεταξύ άλλων, στον συγκεκριμένο ηθοποιό.

- **Σελίδες ανακατεύθυνσης.** Στην Wikipedia υπάρχουν σελίδες ανακατεύθυνσης (redirect pages) για τα διαφορετικά πιθανά ονόματα που μπορούν να χρησιμοποιηθούν ώστε να αναφερθούμε σε μια οντότητα. Αυτό το χαρακτηριστικό μπορεί να χρησιμοποιηθεί για τον εμπλουτισμό του λεξικού D, προσθέτοντας ως υποψήφια μια οντότητα για κάθε τίτλο που ανακατευθύνει σε αυτήν. Για παράδειγμα, το όνομα «DiCaprio» ανακατευθύνει στον Leonardo DiCaprio. Αυτό σημαίνει ότι ο ηθοποιός πρέπει να είναι μέλος του συνόλου D[DiCaprio].

- **Σελίδες αποσαφήνισης.** Στην Wikipedia υπάρχουν σελίδες αποσαφήνισης (disambiguation pages) για τις περιπτώσεις που το ίδιο όνομα μπορεί να αναφέρεται σε περισσότερα από ένα άρθρα, δηλαδή σε περισσότερες από μία οντότητες. Αυτές οι σελίδες περιέχουν μια απαρίθμηση των άρθρων στα οποία μπορεί να αναφέρεται ένα συγκεκριμένο όνομα. Αυτό το χαρακτηριστικό μπορεί να χρησιμοποιηθεί για το χτίσιμο του D, καθώς όλες οι οντότητες στη σελίδα αποσαφήνισης μιας αναφοράς m θα πρέπει να συμπεριληφθούν στην τιμή $D[m]$. Για παράδειγμα, η σελίδα αποσαφήνισης για την αναφορά «Leo» περιέχει, μεταξύ άλλων, τον ηθοποιό Leonardo DiCaprio. Αυτό σημαίνει ότι ο Leonardo DiCaprio, μαζί με τις άλλες οντότητες της σελίδας αποσαφήνισης, θα πρέπει να ανήκει στο σύνολο D[Leo].

- **Σύνδεσμοι από άρθρο προς άρθρο.** Στην Wikipedia είναι πολύ συχνό ένα άρθρο να αναφέρεται σε άλλα άρθρα μέσω συνδέσμων (hyperlinks) που περιέχει. Αυτοί οι σύνδεσμοι έχουν περιφραστικές περιγραφές (anchor texts), οι οποίες είναι μια πηγή πληροφορίας που μπορεί να χρησιμοποιηθεί στο λεξικό D. Αν ένας σύνδεσμος έχει anchor text m και οδηγεί προς κάποια οντότητα e, τότε θα πρέπει $e \in D[m]$. Για παράδειγμα,

Αλγόριθμος 3: Παραγωγή του συνόλου υποψήφιων οντοτήτων για μια αναφορά m με βάση λεξικό ονομάτων D.

Require: dictionary D and mention m

```
1: function CandidateSetGenerationByDictionary(D, m)
2:     E_m ← ∅
3:     for each key ∈ D.keys do
4:         if m matches with key then        ▶ use some sort of string matching method
5:             E_m ← E_m ∪ D[key]
6:         end if
7:     end for
8:     return E_m
9: end function
```

αν ένας σύνδεσμος έχει anchor text «Leonardo» και οδηγεί στη σελίδα του ηθοποιού Leonardo DiCaprio, τότε το συγκεκριμένο πρόσωπο θα πρέπει να συμπεριληφθεί στο σύνολο D[Leonardo].

- **Δόμηση περιεχομένου.** Στην Wikipedia υπάρχει ένα σύνολο κανόνων για τη δόμηση του περιεχομένου. Για παράδειγμα, στην πρώτη παράγραφο ενός άρθρου οι αναφορές που γίνονται με έντονα (bold) γράμματα είναι πολύ πιθανό να αποτελούν εναλλακτικά ονόματα για την ίδια οντότητα. Επίσης, το infobox μπορεί να παρέχει ακόμα περισσότερα εναλλακτικά ονόματα για την ίδια οντότητα. Αυτές οι παρατηρήσεις μπορούν να χρησιμοποιηθούν για περαιτέρω επέκταση του D. Για παράδειγμα, στη σελίδα του διάσημου διαδικτυακού κωμικού PewDiePie η πρώτη παράγραφος περιέχει με έντονα γράμματα τις αναφορές «PewDiePie» και «Felix Arvid Ulf Kjellberg», ενώ το infobox αναφέρει επιπλέον τα ψευδώνυμα «Pewds» και «Pewdie». Αυτό σημαίνει ότι το συγκεκριμένο πρόσωπο θα πρέπει να ανήκει στο καθένα από τα σύνολα $D[m]$, για $m \in$ {PewDiePie, Felix Arvid Ulf Kjellberg, Pewds, Pewdie}.

Ένα ενδεικτικό τμήμα ενός λεξικού ονομάτων D, που αναφέρεται στο παράδειγμα της ενότητας 1.2, φαίνεται στον Πίνακα 1. Αφού έχει κατασκευαστεί το λεξικό D, μπορεί να χρησιμοποιηθεί για την παραγωγή του συνόλου E_m των υποψήφιων οντοτήτων για μια αναφορά m. Η γενική διαδικασία έγκειται στη σύγκριση της αναφοράς m με τα κλειδιά του λεξικού, μέσω τεχνικών σύγκρισης συμβολοσειρών (string matching), και περιγράφεται στον Αλγόριθμο 3. Ωστόσο, αξίζει να σημειωθεί ότι υπάρχουν διάφορες εναλλακτικές όσον αφορά το string matching, που κυμαίνονται από exact string matching μέχρι fuzzy string matching.

2.3.3.2 Χρήση μηχανών αναζήτησης Μια εναλλακτική μέθοδος παραγωγής των υποψήφιων οντοτήτων για μια αναφορά m περιλαμβάνει τη χρήση μιας μηχανής αναζήτησης, όπου το string m υποβάλλεται ως query. Από τα αποτελέσματα διατηρούνται μόνο αυτά που αναφέρονται σε οντότητες, π.χ. τα άρθρα της Wikipedia. Αυτή η μέθοδος έχει το πλεονέκτημα ότι δεν απαιτεί την κατασκευή και τακτική ενημέρωση της πολύπλοκης και εκτε-

Πίνακας 1: Ενδεικτικό τμήμα ενός λεξικού ονομάτων, που αναφέρεται στο παράδειγμα της ενότητας 1.2.

KEYS	VALUES
Leo	Leonardo DiCaprio Lionel Messi Leo (constellation) Leo Tolstoy Pope Leo I …
Oscar	Academy Awards Oscar Wilde Oscar II of Sweden Oscar (footballer, born 1991) Oscar (fish) …
The Revenant	The Revenant (2015 film) The Revenant (2009 film) The Revenant (novel) The Revenant (comics) …

ταμένης δομής του λεξικού ονομάτων, και στηρίζεται σε ένα αξιόπιστο σύστημα παραγωγής υποψήφιων οντοτήτων. Η χρήση μηχανών αναζήτησης για την παραγωγή των στοιχείων του συνόλου E_m έχει χρησιμοποιηθεί αξιοποιώντας τόσο τη μηχανή αναζήτησης της Google [HZ09, LMN+10, DMR+10, MLN+11] όσο και τη μηχανή αναζήτησης της ίδιας της Wikipedia [ZSTW10]. Η γενική διαδικασία περιγράφεται στον Αλγόριθμο 4.

2.3.3.3 Αναζήτηση εναλλακτικών ονομάτων στο κείμενο Μια συμπληρωματική μέθοδος παραγωγής του συνόλου υποψήφιων οντοτήτων, που μπορεί να χρησιμοποιηθεί συνδυαστικά με τις παραπάνω τεχνικές, είναι η αναζήτηση στο κείμενο εναλλακτικών ονομάτων για την ίδια οντότητα. Σε αρκετές περιπτώσεις, ειδικά σε μεγαλύτερα κείμενα, υπάρχει περίπτωση να γίνεται αναφορά στην ίδια οντότητα με διαφορετικά ονόματα. Η αναγνώριση αυτών των αναφορών, που μπορεί να περιγραφεί ως εκτεταμένο coreference resolution, μπορεί να βοηθήσει στην παραγωγή του συνόλου E_m για μια αναφορά m. Αν μπορούμε να πούμε με βεβαιότητα ότι μια σειρά από αναφορές μιλάνε για την ίδια οντότητα πριν καν καταφύγουμε σε ένα λεξικό ονομάτων ή σε μια μηχανή αναζήτησης για την παραγωγή του συνόλου υποψήφιων οντοτήτων, τότε όχι μόνο η παραγωγή του συνόλου αυτού θα είναι πιο πλήρης (δεδομένου ότι θα έχουμε διάφορα εναλλακτικά κλειδιά για το λεξικό και διάφορα εναλλακτικά queries για τη μηχανή αναζήτησης), αλλά και η μετέπειτα αποσαφήνιση θα είναι πιο απλή. Πράγματι, η εύρεση δύο αναφορών με εναλλακτικά ονόματα σημαίνει ότι θα πρέπει να αντιστοιχιστούν στην ίδια οντότητα της βάσης γνώσης. Ένα απλό παράδειγμα εφαρμογής αυτής της τεχνικής είναι η εύρεση συντομογραφιών στο κείμενο, όπως USA για

Αλγόριθμος 4: Παραγωγή συνόλου υποψήφιων οντοτήτων με χρήση μηχανής αναζήτησης.

Require: search engine SE and mention m

```
1: function CandidateSetGenerationBySearchEngine(SE, m)
2:     E_m ← ∅
3:     resultSet ← SE(m)              ▶ query search engine with mention string
4:     for each result ∈ resultSet do
5:         if result is an entity that matches mention m then
6:             E_m ← E_m ∪ {result}
7:         end if
8:     end for
9:     return E_m
10: end function
```

Αλγόριθμος 5: Αξιοποίηση των εναλλακτικών ονομάτων σε συνδυασμό με τη μέθοδο λεξικού για την παραγωγή του συνόλου E_m.

Require: dictionary D and alternative names set AN for mention m
Require: some way to generate candidate entities for each name

```
1: function CandidateSetGenerationByAlternativeNames(D, AN)        ▶ m ∈ AN
2:     E_m ← ∅
3:     for each name ∈ AN do      ▶ use name dictionary method ∀ alternative name
4:         E_m ← E_m ∪ CandidateSetGenerationByDictionary(D, name)
5:     end for
6:     return E_m
7: end function
```

την πλήρη αναφορά United States of America. Μια σειρά εργασιών έχουν αξιοποιήσει την εν λόγω τεχνική για την παραγωγή του συνόλου υποψήφιων οντοτήτων, με παραλλαγές που κυμαίνονται από ευριστικές μεθόδους [HZ09, CTL+10, LMN+10] μέχρι μεθόδους επιβλεπόμενης μηχανικής μάθησης [ZSST11]. Η αναζήτηση εναλλακτικών ονομάτων στο κείμενο ως προκαταρκτικό στάδιο της χρήσης λεξικού ονομάτων περιγράφεται από τον Αλγόριθμο 5.

2.3.4 Ταξινόμηση υποψήφιων οντοτήτων

Παραπάνω αναλύθηκαν οι βασικές τεχνικές με τις οποίες μπορεί να παραχθεί το σύνολο υποψήφιων οντοτήτων $E_m \subset E$ για μια αναφορά m. Εδώ ακολουθείται ο συμβολισμός e_i για τις οντότητες του συνόλου E_m, όπου $1 \leq i \leq |E_m|$. Εν γένει, αυτό το σύνολο περιέχει περισσότερα από ένα στοιχεία ($|E_m| > 1$), πράγμα που σημαίνει ότι δεν είναι τετριμμένη η αντιστοίχιση της αναφοράς m με την οντότητα της βάσης γνώσης στην οποία αναφέρε-

ται. Σε μερικές περιπτώσεις, ο πληθάριθμος $|E_m|$ μπορεί να είναι αρκετά μεγάλος. Για παράδειγμα, στην εργασία του Hoffart και των συνεργατών του [HYB+11], ο μέσος αριθμός των υποψήφιων οντοτήτων για τις αναφορές του dataset που χρησιμοποιήθηκε για την αξιολόγηση του συστήματος είναι 73. Συνεπώς, ένα σημείο-κλειδί της διαδικασίας αποσαφήνισης οντοτήτων σε κείμενο είναι η επιλογή ορθών κριτηρίων με βάση τα οποία επιλέγεται η πιο κατάλληλη οντότητα από το σύνολο E_m για την αναφορά m. Εδώ αναλύονται τα πιο βασικά χαρακτηριστικά που μπορούν να χρησιμοποιηθούν για τη λήψη αυτής της απόφασης.

2.3.4.1 Τεχνικές ανεξάρτητες των συμφραζομένων Οι τεχνικές αυτές δεν αξιοποιούν τα συμφραζόμενα μιας αναφοράς m ώστε να επιλέξουν την καταλληλότερη οντότητα από το σύνολο υποψήφιων οντοτήτων E_m (context-independent). Οι βασικότερες τεχνικές που εμπίπτουν στην κατηγορία αυτήν περιγράφονται παρακάτω.

- **Ταίριασμα ονομάτων:** Αυτή είναι η πιο βασική τεχνική ταξινόμησης υποψήφιων οντοτήτων. Για μια αναφορά m, συγκρίνουμε το string m με τα ονόματα των οντοτήτων του E_m. Το όνομα που ταιριάζει περισσότερο είναι και αυτό που καθορίζει την αποσαφήνιση της αναφοράς m. Το βασικό σημείο που ξεχωρίζει τα διάφορα συστήματα τα οποία αξιοποιούν αυτήν τη μέθοδο είναι οι αλγόριθμοι string matching που εφαρμόζουν. Γενικά, το κριτήριο που χρησιμοποιείται είναι αυτό του edit distance[1], με διαφορετικά επιτρεπόμενα σύνολα string operations να οδηγούν σε διαφορετικά αποτελέσματα.

- **Εκ των προτέρων δημοτικότητα:** Ένα από τα βασικότερα context-independent χαρακτηριστικά που χρησιμοποιούνται από συστήματα αποσαφήνισης είναι η εκ των προτέρων δημοτικότητα (popularity prior). Αυτή αποτελεί μια έκφραση της δεσμευμένης πιθανότητας να εμφανιστεί στο κείμενο μια οντότητα $e_i \in E_m$, δεδομένου ότι έχει γίνει η αναφορά m.

  $$\text{Popularity Prior}_i = \mathbb{P}[e_i \text{ εμφανίζεται στο κείμενο} \mid \text{η αναφορά είναι } m]$$

  Με βάση αυτήν την προσέγγιση, επιλέγεται από το σύνολο E_m εκείνη η οντότητα e_i που έχει τη μέγιστη εκ των προτέρων δημοτικότητα βάσει της αναφοράς m, και άρα είναι πιθανότερο να εμφανιστεί ως η συγκεκριμένη αναφορά.

  $$m \mapsto e_i, \text{ όπου } \forall e_j \in E_m \text{ ισχύει Popularity Prior}_j \leq \text{Popularity Prior}_i$$

  Αυτή η μέθοδος, αν και δεν είναι αλάνθαστη, στηρίζεται στη λογική παρατήρηση ότι μια αναφορά m αναφέρεται στην οντότητα με τη μέγιστη δημοτικότητα στις περισσότερες περιπτώσεις. Αυτή η πρόταση μπορεί να φαίνεται ως ταυτολογία, καθώς ουσιαστικά δηλώνει ότι οι δημοφιλείς οντότητες είναι δημοφιλείς, αλλά αποτελεί βασικό στοιχείο πολλών επιτυχημένων συστημάτων αποσαφήνισης [KSRC09, MLN+11, HYB+11, RRDA11, SWLW12b, SWLW12a, SWLW13, LLW+13, GCK13]. Για παράδειγμα,

[1] Edit distance είναι το πλήθος στοιχειωδών πράξεων σε χαρακτήρες που απαιτούνται για να μετατρέψουμε ένα string s_1 σε ένα string s_2. Ανάλογα με το ποιες πράξεις είναι επιτρεπτές, παίρνουμε διαφορετικά edit distances. Για παράδειγμα, αν επιτρέπουμε διαγραφή, εισαγωγή και αντικατάσταση χαρακτήρων, υπολογίζουμε την απόσταση Levenshtein, ενώ αν δεν επιτρέπεται η αντικατάσταση χαρακτήρων, υπολογίζουμε την απόσταση μέγιστης κοινής υπακολουθίας.

η αναφορά «Leo» σε ένα κείμενο είναι πολύ πιο πιθανό να αναφέρεται στον ηθοποιό Leonardo DiCaprio ή στον ποδοσφαιριστή Lionel Messi, παρά στον νομπελίστα Ιάπωνα φυσικό Leo Esaki. Αν και η προσέγγιση της εκ των προτέρων δημοτικότητας είναι υποσχόμενη, ο υπολογισμός της δεν είναι απλός. Κατά κανόνα, χρησιμοποιείται η δομή συνδέσμων της Wikipedia για την κατασκευή ενός μέτρου δημοτικότητας των οντοτήτων. Όπως είχε αναφερθεί και κατά την περιγραφή της κατασκευής λεξικού ονομάτων, τα άρθρα της Wikipedia περιέχουν συνδέσμους προς άλλα άρθρα, με διαφορετικά anchor texts. Ένα μέτρο της δημοτικότητας μιας οντότητας $e_i \in E_m$, δεδομένης της αναφοράς m, μπορεί να υπολογιστεί βάσει του συνόλου συνδέσμων από ολόκληρη την Wikipedia, που έχουν anchor text m και οδηγούν σε κάποιο στοιχείο του συνόλου E_m. Αν, λοιπόν, $L = \{\text{link in Wikipedia} \mid anchor(\text{link}) = m \wedge (\exists e_j \in E_m : \text{link} \mapsto e_j)\}$, τότε ένας τρόπος υπολογισμού της εκ των προτέρων δημοτικότητας για την οντότητα $e_i \in E_m$, δεδομένης της αναφοράς m, είναι:

$$\text{Popularity Prior}_i = \frac{|\{l \in L \mid l \mapsto e_i\}|}{|L|}$$

- **Τύποι αναφορών & οντοτήτων:** Ένα ακόμα χαρακτηριστικό που μπορεί να απομονώσει ένα υποσύνολο του E_m ως σύνολο οντοτήτων που μπορούν να αντιστοιχιστούν με την αναφορά m είναι η χρήση τύπων. Οι περισσότερες βάσεις γνώσης αποθηκεύουν κάποια ιεραρχία τύπων για τις οντότητες που περιέχουν. Επίσης, οι μέθοδοι επεξεργασίας φυσικής γλώσσας που χρησιμοποιούνται για την παραγωγή του συνόλου M των αναφορών σε ένα αδόμητο κείμενο έχουν τη δυνατότητα να παράγουν αξιόπιστες προβλέψεις για τους τύπους των αναφορών. Μια απλή ιεραρχία που χρησιμοποιείται από αρκετά συστήματα αναγνώρισης ονοματικών οντοτήτων περιλαμβάνει τους τύπους Person, Organization και Location. Επίσης, αντίστοιχοι τύποι είναι αποθηκευμένοι στις βάσεις γνώσης. Αυτό σημαίνει ότι μια απλή σύγκριση τύπων μπορεί να απομακρύνει από το σύνολο E_m τις υποψήφιες οντότητες που έχουν διαφορετικούς τύπους. Δεδομένου ότι η βάση γνώσης έχει αποθηκευμένη σωστή πληροφορία τύπων και ότι η εξαγωγή τύπων από το κείμενο είναι αξιόπιστη, μεγάλο μέρος του E_m μπορεί να αφαιρεθεί χωρίς κίνδυνο απώλειας της ορθής οντότητας. Η μέθοδος αυτή στηρίζεται στην παρατήρηση ότι μια αναφορά m που μιλάει για κάποιο πρόσωπο δεν μπορεί ποτέ να αντιστοιχιστεί σε κάποιον οργανισμό ή κάποια τοποθεσία, και όμοια για τις αναφορές που μιλάνε για οργανισμούς και τοποθεσίες. Στο παράδειγμα της ενότητας 1.2, αν ήταν διαθέσιμη η πληροφορία ότι η αναφορά «Leo» έχει τον τύπο Person, τότε θα μπορούσαμε να αφαιρέσουμε από το σύνολο E_m την οντότητα του αστερισμού του Λέοντα, την οντότητα του Πανεπιστημίου της Florida, και πολλές άλλες υποψήφιες οντότητες. Αυτή η προσέγγιση ακολουθήθηκε στις εργασίες [LMN+10, MLN+11], όπου χρησιμοποιήθηκε το έτοιμο σύστημα αναγνώρισης ονοματικών οντοτήτων CiceroLite για εύρεση των αναφορών και παραγωγή των τύπων τους, οι οποίοι στη συνέχεια συγκρίθηκαν με τις αντίστοιχες καταχωρήσεις των βάσεων γνώσης.

2.3.4.2 Τεχνικές βασιζόμενες στα συμφραζόμενα Αν και οι τεχνικές που δε χρησιμοποιούν τα συμφραζόμενα μιας αναφοράς m για να την αντιστοιχίσουν σε μια οντότητα του

συνόλου E_m είναι χρήσιμες, δύσκολα μπορεί να επιτευχθεί ποιοτικό αποτέλεσμα αποσαφήνισης χωρίς να λαμβάνονται υπόψιν τα συμφραζόμενα. Εδώ περιγράφονται οι βασιζόμενες στα συμφραζόμενα (context-dependent) τεχνικές.

- **Ομοιότητα κειμένου:** Η πιο βασική προσέγγιση χρήσης των συμφραζομένων μιας αναφοράς m έχει να κάνει με την ομοιότητα των συμφραζομένων αυτών με τα κείμενα που περιγράφουν τις οντότητες του συνόλου E_m. Η αναπαράσταση των συμφραζομένων μπορεί να γίνει με δύο τρόπους:

  1. *Αναπαράσταση συμφραζομένων με bag of words:* Σε αυτήν την προσέγγιση, τα συμφραζόμενα μιας αναφοράς m αποτελούν ένα σύνολο λέξεων (bag of words) από το υπόλοιπο κείμενο. Τα συμφραζόμενα μπορεί να προέρχονται από ολόκληρο το κείμενο [ŠM09, ZTSS10, CTL+10, ZTS+11, GCK13, LLW+13], δηλαδή να είναι προκύπτουν ως $T \setminus \{m\}$, ή να υπολογίζονται σε κάποιο παράθυρο γύρω από την αναφορά [BP06, KSRC09, HSZ11, RRDA11, LME12, SWLW13], και άρα να προκύπτουν ως $\{w \in T \mid 0 < distance(w, m) \leq W\}$ για μέγεθος παραθύρου $2W$. Αυτή η αναπαράσταση συγκρίνεται με το αντίστοιχο bag of words model για τις οντότητες του συνόλου E_m στη βάση γνώσης, που για μια οντότητα μπορεί να παραχθεί με βάση ολόκληρο το άρθρο της Wikipedia που την περιγράφει [BP06, ZTSS10, ZTS+11, HSZ11, LME12, LLW+13], τις πρώτες λίγες γραμμές του άρθρου αυτού, οι οποίες περιέχουν και τις λέξεις που είναι πιο σχετικές με την οντότητα [KSRC09], ή με άλλες μεθόδους.
  2. *Αναπαράσταση συμφραζομένων με διανύσματα εννοιών:* Σε αυτήν την προσέγγιση, τα συμφραζόμενα μιας αναφοράς σε κάποιο κείμενο ή το άρθρο που περιγράφει μια οντότητα σε μια βάση γνώσης, όπως η Wikipedia, κωδικοποιούνται με χρήση διανυσμάτων εννοιών (concept vectors). Αυτά τα διανύσματα κατασκευάζονται με βάση λέξεις κλειδιά [HYB+11], κατηγορίες της Wikipedia [Cuc07, DMR+10], ή ακόμα και συγγενικές οντότητες και σχετικά θέματα [CTL+10, LMN+10, DMR+10].

  Ανεξάρτητα από τον τρόπο που επιλέγουμε να επεξεργαστούμε τα συμφραζόμενα, πάντα καταλήγουμε στη χρήση ενός vector space model για να υπολογίσουμε ένα βαθμό ομοιότητας μεταξύ των συμφραζομένων μιας αναφοράς και του κειμένου με το οποίο περιγράφεται μια υποψήφια οντότητα. Αφού τα κείμενα έχουν απεικονισθεί ως διανύσματα (λέξεων στην πρώτη περίπτωση και εννοιών στη δεύτερη), μπορούμε να υπολογίσουμε την ομοιότητά τους με διάφορες μεθόδους. Η πιο συχνά χρησιμοποιούμενη μέθοδος είναι η ομοιότητα συνημιτόνου [BP06, KSRC09, ŠM09, ZSTW10, ZTSS10, ZLHZ10, DMR+10, ZTS+11, CJ11, RRDA11, LME12, SWLW13]. Εναλλακτικές μέθοδοι περιλαμβάνουν τη χρήση απόκλισης Kullback–Leibler [HYB+11] ή συντελεστή ομοιότητας Jaccard [KSRC09].

- **Συνοχή μεταξύ οντοτήτων στο ίδιο κείμενο:** Για την αποσαφήνιση μιας ονοματικής αναφοράς, σημαντικό ρόλο παίζουν οι άλλες ονοματικές αναφορές στο ίδιο κείμενο, που πρέπει να αποσαφηνιστούν παράλληλα. Πολλά συστήματα αποσαφήνισης [ŠM09, Cuc07, KSRC09, FS10, HYB+11, HSZ11, RRDA11, SWLW12b, SWLW12a, HS12, GCK13, SWLW13] κάνουν την υπόθεση ότι ένα κείμενο αναφέρεται σε οντότητες με υψηλή συνοχή, που ανήκουν σε κάποιο συγκεκριμένο θέμα. Αυτή η τοπικότητα των αναφορών

επιτρέπει τη συλλογική αποσαφήνιση (joint/collective disambiguation) των οντοτήτων του κειμένου, με στόχο την επιλογή εκείνων των στοιχείων από τα σύνολα E_m τα οποία παρουσιάζουν τη μέγιστη σημασιολογική εγγύτητα. Οι πιο συνηθισμένες μετρικές για τη σημασιολογική εγγύτητα μεταξύ δύο οντοτήτων e_1 και e_2 βασίζονται στη δομή συνδέσμων της Wikipedia, και στηρίζονται στην υπόθεση ότι δύο οντότητες που συγγενεύουν σημασιολογικά θα έχουν μεγάλη επικάλυψη στους εισερχόμενους συνδέσμους τους. Από αυτές χρησιμοποιείται περισσότερο [KSRC09, FS10, HYB+11, HSZ11, RRDA11, SWLW12b, SWLW13, LLW+13] το Wikipedia Link-based Measure (WLM), που προτάθηκε από τους Milne και Witten [MW08], και αποτελεί μια έκφραση του Normalized Google Distance [CV07]. Συγκεκριμένα, αν IN_1 είναι το σύνολο των εισερχόμενων συνδέσμων για το άρθρο που περιγράφει την οντότητα e_1 στην Wikipedia, IN_2 είναι το σύνολο των εισερχόμενων συνδέσμων για το άρθρο που περιγράφει την οντότητα e_2 στην Wikipedia, και WP είναι το σύνολο όλων των άρθρων της Wikipedia, τότε:

$$\mathrm{WLM}(e_1, e_2) = 1 - \frac{\log(\max(|IN_1|, |IN_2|)) - \log(|IN_1 \cap IN_2|)}{\log(|WP|) - \log(\min(|IN_1|, |IN_2|))}$$

Άλλες μετρικές συσχέτισης από τη θεωρία πληροφορίας που έχουν εφαρμοστεί περιλαμβάνουν την PMI (Pointwise Mutual Information measure) [RRDA11], και την Jaccard distance [GCK13], οι οποίες υπολογίζονται ως εξής:

$$\mathrm{PMI}(e_1, e_2) = \frac{|IN_1 \cap IN_2|/|WP|}{|IN_1|/|WP| \cdot |IN_2|/|WP|}$$

$$d_J(e_1, e_2) = \frac{|IN_1 \cap IN_2|}{|IN_1 \cup IN_2|}$$

Για καλύτερη κατανόηση της σημασιολογικής εγγύτητας, εξετάζουμε το παράδειγμα της ενότητας 1.2, με τις αναφορές $m_1 =$ Leo, $m_2 =$ Oscar και $m_3 =$ The Revenant και θεωρούμε τα σύνολα υποψήφιων οντοτήτων:

$E_1 =$ {Leonardo DiCaprio, Lionel Messi}
$E_2 =$ {Academy Awards, Oscar Wilde}
$E_3 =$ {The Revenant (2015 film), The Revenant (2009 film)}

Για κάθε υποψήφια οντότητα υπολογίζεται η μετρική WLM με κάθε άλλη υποψήφια οντότητα, που δεν ανήκει στο ίδιο σύνολο οντοτήτων E_k. Είναι φανερό ότι δεν έχει νόημα ο υπολογισμός της σημασιολογικής εγγύτητας μεταξύ οντοτήτων που ανήκουν στο ίδιο σύνολο υποψήφιων οντοτήτων. Τα αποτελέσματα αυτού του υπολογισμού φαίνονται στον Πίνακα 2. Η σημασιολογική συγγένεια μεταξύ αυτών των οντοτήτων δίνεται εποπτικά στην Εικόνα 2. Όπως φαίνεται, τα πράγματα αρχίζουν να περιπλέκονται ήδη όταν έχουμε λίγες αναφορές και υποψήφιες οντότητες. Σε μεγαλύτερα κείμενα, οι αναφορές θα ήταν περισσότερες. Επίσης, είτε σε μικρά είτε σε μεγάλα κείμενα, το πλήθος των υποψήφιων οντοτήτων δε θα ήταν τόσο μικρό. Δεν μπορούμε να περιμένουμε ικανοποιητικά αποτελέσματα αν οι υποψήφιες οντότητες δεν είναι τουλάχιστον κάποιες δεκάδες. Αν και η σημασιολογική συνοχή είναι χρήσιμο εργαλείο για την αποσαφήνιση, η χρήση της στο πλαίσιο της συλλογικής αποσαφήνισης όλων των

Πίνακας 2: Υπολογισμός μετρικής συνοχής WLM για ενδεικτικά σύνολα υποψήφιων οντοτήτων.

$\mathbf{e_1}$	$\mathbf{e_2}$	**WLM**
Leonardo DiCaprio	Academy Awards	0.574
Leonardo DiCaprio	Oscar Wilde	0.362
Leonardo DiCaprio	The Revenant (2015 film)	0.674
Leonardo DiCaprio	The Revenant (2009 film)	0
Lionel Messi	Academy Awards	0.099
Lionel Messi	Oscar Wilde	0.174
Lionel Messi	The Revenant (2015 film)	0.268
Lionel Messi	The Revenant (2009 film)	0
Academy Awards	The Revenant (2015 film)	0.594
Academy Awards	The Revenant (2009 film)	0
Oscar Wilde	The Revenant (2015 film)	0.319
Oscar Wilde	The Revenant (2009 film)	0

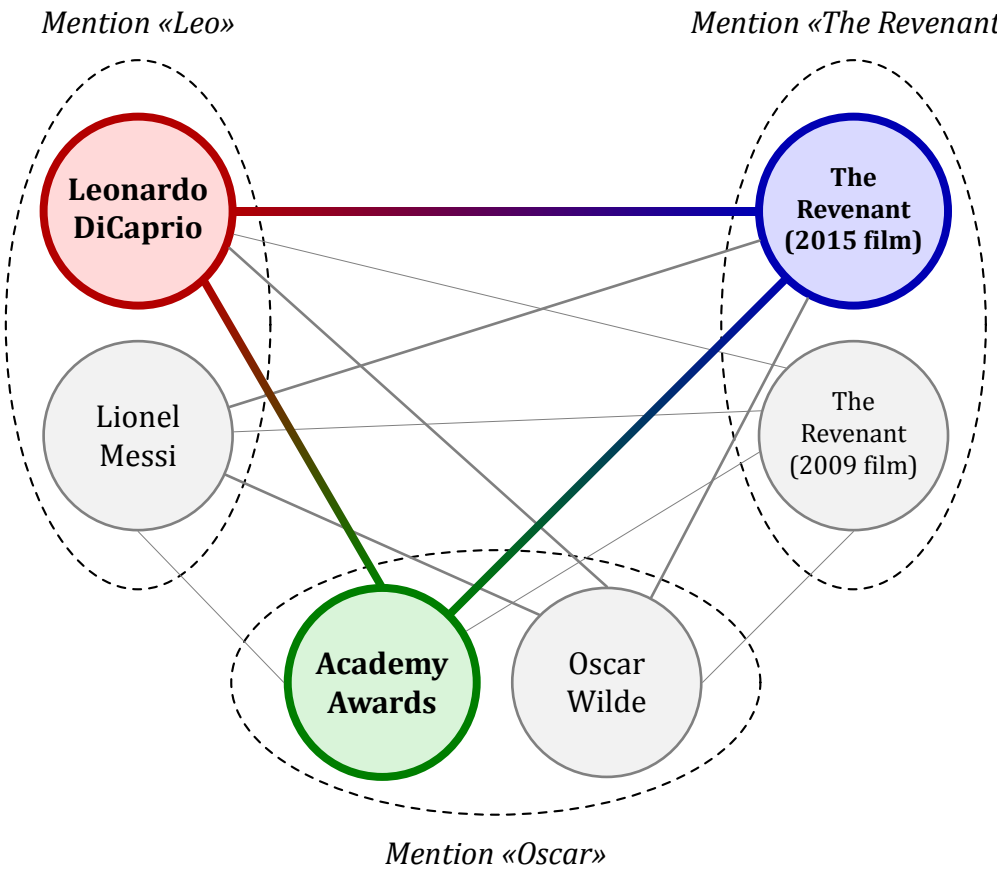
οντοτήτων ενός κειμένου οδηγεί σε δύσκολα προβλήματα συνδυαστικής βελτιστοποίησης. Μάλιστα, όπως έχει αναλυθεί στη βιβλιογραφία [KSRC09, HYB+11, SWLW12a, LLW+13], αυτά τα προβλήματα είναι NP-δύσκολα. Αυτό σημαίνει ότι η επιλογή της σημασιολογικής εγγύτητας ως εργαλείου αποσαφήνισης επιβαρύνει τους σχεδιαστές των αντίστοιχων συστημάτων με την ευθύνη εύρεσης ικανοποιητικών προσεγγιστικών μεθόδων, ώστε η υπολογιστική πολυπλοκότητα να παραμένει σε λογικά όρια. Το συγκεκριμένο θέμα αναλύεται περαιτέρω στη συνέχεια της εργασίας.

2.3.5 Αναγνώριση μη αποσαφηνίσιμων αναφορών

Στην παράγραφο 2.3.4 περιγράφηκαν οι κυριότερες τεχνικές ταξινόμησης των στοιχείων του συνόλου E_m για μια αναφορά m, ώστε να επιλεγεί η καταλληλότερη οντότητα από το σύνολο E_m. Ωστόσο, σε ορισμένες περιπτώσεις, μπορεί η οντότητα e_x του κόσμου στην οποία αντιστοιχεί η m να μην ανήκει στο σύνολο E_m. Αν υποθέσουμε ότι η μέθοδος παραγωγής υποψήφιων οντοτήτων ήταν επιτυχημένη (άρα δεν έχουμε $e_x \notin E_m \wedge e_x \in E$), αυτό σημαίνει ότι $e_x \notin E$, όπου E υπενθυμίζεται ότι αποτελεί το σύνολο των οντοτήτων που είναι καταχωρημένες στη βάση γνώσης KB. Δεδομένου ότι καμία βάση γνώσης δεν είναι πλήρης, και λαμβάνοντας υπόψιν τους ρυθμούς με τους οποίους αναδύονται νέες οντότητες στη σημερινή εποχή, αυτό το ενδεχόμενο δεν είναι καθόλου σπάνιο.

Η πιο απλή προσέγγιση, που χρησιμοποιείται σε αρκετές εργασίες [Cuc07, KSRC09, LSC10, HSZ11, SWLW12a, HS12, DDCM12], απλά θεωρεί ότι η υποκείμενη βάση γνώσης είναι πλήρης, και άρα υποθέτει ότι υπάρχει η αντίστοιχη οντότητα για κάθε αναφορά. Μια εναλλακτική προσέγγιση είναι η χρήση ενός κατάλληλου κατωφλίου [BP06, HZ09, LMN+10, FS10, GJ11, PP11, SWLW12b, SWLW13, LWH+13], όπου η επικρατέστερη οντότητα του συνόλου E_m βαθμολογείται κατάλληλα και συγκρίνεται με μια τιμή κατωφλίου. Αν η βαθμολογία ξεπερνάει αυτήν την τιμή, τότε γίνεται η αντιστοίχιση, διαφορετικά η αναφορά θεωρείται ότι

Εικόνα 2: Σημασιολογική συγγένεια μεταξύ οντοτήτων διαφόρων συνόλων E_k. Το πάχος των ακμών είναι ανάλογο της μετρικής WLM. Έντονα φαίνεται η σωστή επιλογή οντοτήτων από τα σύνολα. Αυτές οι οντότητες εμφανίζουν μεταξύ τους τη μέγιστη σημασιολογική συνοχή.



δεν μπορεί να αποσαφηνιστεί. Επιπλέον, έχουν χρησιμοποιηθεί και τεχνικές επιβλεπόμενης μηχανικής μάθησης, που περιλαμβάνουν εκπαίδευση ενός binary classifier [ZTSS10, LMN+10, ZLHZ10, ZTS+11, ZSST11, MLN+11, RRDA11] για τη λήψη της απόφασης αν μια αναφορά m μπορεί να αντιστοιχιστεί με την επικρατέστερη οντότητα του E_m. Οι παραπάνω προσεγγίσεις εντοπισμού μη αποσαφηνίσιμων αναφορών εκτελούν ένα επιπλέον βήμα στη διαδικασία της αποσαφήνισης. Όπως φάνηκε στις εργασίες [DMR+10, McN10, HS11], αυτό δεν είναι αναγκαίο. Πράγματι, αν θεωρήσουμε μια οντότητα $\bar{e}$, η οποία αντιπροσωπεύει τις οντότητες που απουσιάζουν από τη βάση γνώσης, τότε η αποσαφήνιση για μια αναφορά m γίνεται με σύνολο υποψήφιων οντοτήτων $E_m \cup \{\bar{e}\}$. Αν η επικρατέστερη οντότητα είναι η $\bar{e}$, τότε η αναφορά m είναι μη αποσαφηνίσιμη.

2.4 Αξιολόγηση συστημάτων αποσαφήνισης

Η αξιολόγηση των συστημάτων αποσαφήνισης γίνεται χρησιμοποιώντας datasets από κείμενα, για τα οποία είναι διαθέσιμη εκ των προτέρων η πληροφορία αποσαφήνισης, που έχει ελεγχθεί για την ορθότητά της (ground truth). Αυτή η πληροφορία παράγεται και επαληθεύεται με ανθρώπινη επίβλεψη. Δημοφιλείς μετρικές επίδοσης ενός συστήματος αποτελούν οι precision, recall και F_1 (F-score ή F-measure). Για να καταλήξουμε στους συγκεκριμένους τύπους που χρησιμοποιούνται στην αξιολόγηση συστημάτων αποσαφήνισης οντοτήτων, ξεκινάμε από τους ορισμούς που παρέχονται από τον τομέα ανάκτησης πληροφορίας [WPR, DG06]. Στη γενική περίπτωση, εξετάζεται ένα σύστημα S το οποίο καλείται να απομονώσει από ένα σύνολο στοιχείων E μόνο εκείνα τα οποία είναι σχετικά με τους όρους αναζήτησης T (relevant elements with respect to T). Τα στοιχεία του συνόλου E, με βάση τον τρόπο με τον οποίο τα χωρίζει το σύστημα S βάσει των όρων T, κατανέμονται στα εξής ξένα μεταξύ τους σύνολα:

- True Positive (TP), όπου ανήκουν τα στοιχεία τα οποία καλώς επιστράφηκαν από το σύστημα, βάσει του ground truth. Αυτά είναι relevant elements που το σύστημα κατάφερε να αναγνωρίσει.
- False Positive (FP), όπου ανήκουν τα στοιχεία τα οποία κακώς επιστράφηκαν από το σύστημα, βάσει του ground truth. Αυτά είναι non-relevant elements που το σύστημα, για οποιονδήποτε λόγο, αναγνώρισε ως relevant elements.
- True Negative (TN), όπου ανήκουν τα στοιχεία τα οποία καλώς δεν επιστράφηκαν από το σύστημα, βάσει του ground truth. Αυτά είναι non-relevant elements που το σύστημα κατάφερε να αναγνωρίσει.
- False Negative (FN), όπου ανήκουν τα στοιχεία τα οποία κακώς δεν επιστράφηκαν από το σύστημα, βάσει του ground truth. Αυτά είναι relevant elements που το σύστημα, για οποιονδήποτε λόγο, αναγνώρισε ως non-relevant elements.

Από τους παραπάνω ορισμούς προκύπτει άμεσα ότι $E \overset{S,T}{=} TP \cup FP \cup TN \cup FN$, δηλαδή το σύνολο E μπορεί να γραφεί ως η ένωση των TP, FP, TN και FN, όπως προκύπτουν από το σύστημα S και τους όρους αναζήτησης T. Επίσης, μπορούμε να ορίσουμε τα εξής σύνολα:

Πίνακας 3: Παραγωγή των συνόλων predicted positive, actual positive, predicted negative και actual negative βάσει των συνόλων TP, FP, TN και FN.

	actual positive	actual negative
predicted positive	TP	FP
predicted negative	FN	TN

- predicted positive $= TP \cup FP$, όπου ανήκουν τα στοιχεία που επιστράφηκαν από το σύστημα.
- actual positive $= TP \cup FN$, όπου ανήκουν τα στοιχεία που θα θέλαμε να επιστραφούν από το σύστημα.
- predicted negative $= TN \cup FN$, όπου ανήκουν τα στοιχεία που δεν επιστράφηκαν από το σύστημα.
- actual negative $= TN \cup FP$, όπου ανήκουν τα στοιχεία που θα θέλαμε να μην επιστραφούν από το σύστημα.

Οι παραπάνω σχέσεις παρουσιάζονται εποπτικά στον Πίνακα 3. Μετά από αυτήν την ανάλυση, μπορούν εύκολα να προκύψουν οι τύποι των precision, recall και F_1, όπου καταχρηστικά χρησιμοποιούνται οι ίδιοι συμβολισμοί για τα σύνολα και τους πληθικούς τους αριθμούς:

$$\text{precision} = \frac{TP}{\text{predicted positive}} = \frac{TP}{TP + FP}$$
$$\text{recall} = \frac{TP}{\text{actual positive}} = \frac{TP}{TP + FN}$$
$$F_1 = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Παρατηρούμε ότι η μετρική precision είναι το ποσοστό των σωστών αποτελεσμάτων μέσα σε όλα τα αποτελέσματα που επέστρεψε το σύστημα. Από την άλλη, η μετρική recall είναι το ποσοστό των σωστών αποτελεσμάτων που επέστρεψε το σύστημα, συγκρίνοντας με όλα τα αποτελέσματα που είναι σωστά βάσει ground truth. Τέλος, η μετρική F_1 ορίζεται απλά ως ο αρμονικός μέσος των precision και recall.

Επιστρέφοντας στο πεδίο της αποσαφήνισης οντοτήτων, η εξειδίκευση των παραπάνω τύπων δίνει άμεσα τα εξής αποτελέσματα:

$$\text{precision} = \frac{\text{πλήθος αναφορών που αποσαφηνίστηκαν σωστά στο κείμενο}}{\text{πλήθος αναφορών που βρέθηκαν στο κείμενο}}$$
$$\text{recall} = \frac{\text{πλήθος αναφορών που αποσαφηνίστηκαν σωστά στο κείμενο}}{\text{πλήθος αναφορών που υπάρχουν στο κείμενο}}$$
$$F_1 = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Για να γίνεται η αξιολόγηση των συστημάτων με μια μόνο μετρική, αλλά και για να εξαλειφθεί η επίδραση της εκάστοτε τεχνικής αναγνώρισης ονοματικών αναφορών, συχνά δίνονται στο σύστημα οι ονοματικές αναφορές προς αποσαφήνιση. Τότε, χρησιμοποιείται μόνο η μετρική accuracy[2] για την αξιολόγηση της ακρίβειας αποσαφήνισης:

$$\text{accuracy} = \text{precision} = \text{recall} = F_1$$

Εξετάζοντας το παράδειγμα της ενότητας 1.2, υποθέτουμε ότι σε ένα σύστημα αποσαφήνισης οντοτήτων δίνεται ως είσοδος το κείμενο χωρίς τις αναφορές προς αποσαφήνιση. Αν αναγνωριστούν οι ονοματικές αναφορές «Leo» και «The Revenant», και η πρώτη αντιστοιχιστεί στον Lionel Messi και η δεύτερη στην ταινία The Revenant του 2015, τότε θα έχουμε τις εξής τιμές για τις μετρικές:

$$\text{precision} = \frac{1}{2} = 0.5$$
$$\text{recall} = \frac{1}{3} \approx 0.333$$
$$F_1 = 2\frac{(1/2)\cdot(1/3)}{1/2 + 1/3} = \frac{2}{5} = 0.4$$

Ωστόσο, αν δίνονται στο σύστημα οι τρεις αναφορές που πρέπει να αποσαφηνιστούν, και θεωρώντας ότι για τις αναφορές «Leo» και «The Revenant» έχουμε τα ίδια αποτελέσματα με προηγουμένως, και επιπλέον ότι η αναφορά «Oscar» αποσαφηνίστηκε ορθά, τότε θα έχουμε:

$$\text{accuracy} = \text{precision} = \text{recall} = F_1 = \frac{2}{3} \approx 0.667$$

Ενδιαφέρον παρουσιάζει η περίπτωση όπου έχουμε ένα dataset, δηλαδή ένα σύνολο κειμένων, πάνω στο οποίο θέλουμε να υπολογίσουμε συνολικά τις μετρικές precision, recall, F_1 και accuracy, ώστε με έναν αριθμό να αποκτούμε αίσθηση της επίδοσης του συστήματος. Τότε, μπορούν να ακολουθηθούν δύο προσεγγίσεις [SL01]:

- *Micro Average.* Σε αυτήν την προσέγγιση, τα κείμενα θεωρούνται ότι συνδυάζονται για να δημιουργήσουν ένα μεγάλο κείμενο, πάνω στο οποίο το σύστημα αποσαφήνισης αξιολογείται με βάση τις μετρικές που έχουν περιγραφεί παραπάνω
- *Macro Average*. Σε αυτήν την προσέγγιση, εξάγεται ο μέσος όρος των παραπάνω μετρικών για να αξιολογηθεί το σύστημα πάνω στο σύνολο των κειμένων.

Έτσι, λαμβάνουμε τους τύπους:

$$\text{Micro Average Precision} = \frac{\text{πλήθος αναφορών που αποσαφηνίστηκαν σωστά στο dataset}}{\text{πλήθος αναφορών που βρέθηκαν στο dataset}}$$
$$\text{Macro Average Precision} = \text{μέσος όρος των επιμέρους precision}$$
$$\text{Micro Average Recall} = \frac{\text{πλήθος αναφορών που αποσαφηνίστηκαν σωστά στο dataset}}{\text{πλήθος αναφορών που υπάρχουν στο dataset}}$$
$$\text{Macro Average Recall} = \text{μέσος όρος των επιμέρους recall}$$

[2] Σημειώνεται ότι ο συγκεκριμένος ορισμός για τη μετρική accuracy μπορεί να μη συμφωνεί με τον γενικό τύπο που εφαρμόζεται στη θεωρία ανάκτησης πληροφορίας.

Η μετρική F_1 προκύπτει από τον αρμονικό μέσο των αντίστοιχων aggregated μετρικών, δηλαδή:

$$\text{Micro Average } F_1 = 2\frac{\text{Micro Average Precision} \cdot \text{Micro Average Recall}}{\text{Micro Average Precision} + \text{Micro Average Recall}}$$
$$\text{Macro Average } F_1 = 2\frac{\text{Macro Average Precision} \cdot \text{Macro Average Recall}}{\text{Macro Average Precision} + \text{Macro Average Recall}}$$

Τέλος, σε περίπτωση που οι ονοματικές αναφορές δίνονται, τότε χρησιμοποιούνται οι μετρικές:

$$\begin{aligned}\text{Micro Average Accuracy} &= \text{Micro Average Precision}\\ &= \text{Micro Average Recall}\\ &= \text{Micro Average } F_1\\ \text{Macro Average Accuracy} &= \text{Macro Average Precision}\\ &= \text{Macro Average Recall}\\ &= \text{Macro Average } F_1\end{aligned}$$

Σημειώνεται ότι, σε κάθε περίπτωση, το ποιες αναφορές αναμένεται να αποσαφηνίσει ένα σύστημα αποσαφήνισης, καθώς και οι σωστές οντότητες που αναμένεται να επιστρέψει, είναι παράγοντες που καθορίζονται πλήρως από το ground truth.

Κεφαλαιο 3

Εξέταση APIs

Στο κεφάλαιο αυτό περιγράφονται τα κύρια APIs (Application Programming Interfaces) τα οποία χρησιμοποιούνται στον πυρήνα του συστήματος αποσαφήνισης που αναλύεται και σχεδιάζεται στο κεφάλαιο 4. Στην ενότητα 3.1 περιγράφεται το Google Knowledge Graph Search API, στην ενότητα 3.2 περιγράφονται τα πιο χρήσιμα MediaWiki APIs και στην ενότητα 3.3 περιγράφεται το Named Entity Recognition module του πακέτου Stanford CoreNLP.

3.1 Google Knowledge Graph Search API

Το Google Knowledge Graph Search API είναι ένα API που επιτρέπει την αναζήτηση οντοτήτων στον Google Knowledge Graph, τον γράφο γνώσης της Google. Για συντομία, το Google Knowledge Graph Search API θα αναφέρεται ως GKG API και ο υποκείμενος γράφος γνώσης θα αναφέρεται ως GKG. Χαρακτηριστικά παραδείγματα χρήσης αυτής της υπηρεσίας περιλαμβάνουν:

- Αναζήτηση οντοτήτων στον GKG με συγκεκριμένα κριτήρια, και ταξινόμηση των αποτελεσμάτων που λαμβάνονται με βάση τον βαθμό στον οποίο αυτά τα κριτήρια ικανοποιούνται.
- Παραγωγή προτεινόμενων αποτελεσμάτων από τον GKG σε κάποιο πεδίο αναζήτησης.
- Αντιστοίχιση αναφορών ενός κειμένου σε οντότητες του GKG.

Από τα παραπάνω use cases, το τρίτο είναι αυτό που ενδιαφέρει περισσότερο σε μια εφαρμογή αποσαφήνισης οντοτήτων. Ωστόσο, αξίζει να αναφερθεί ότι το GKG API δεν επιστρέφει ως αποτέλεσμα γράφους ή συνδεδεμένες οντότητες, παρά τα όσα υποδεικνύει το όνομά του. Πράγματι, όπως θα φανεί στη συνέχεια, το συγκεκριμένο API απλά επιστρέφει λίστες από οντότητες που ταιριάζουν με τα κριτήρια αναζήτησης, χωρίς να δίνει στοιχεία για τις μεταξύ τους σχέσεις.

Τα ερωτήματα προς το GKG API γίνονται χρησιμοποιώντας HTTP GET requests της μορφής `https://kgsearch.googleapis.com/v1/entities:search`, με τις εξής παραμέτρους:

- `key`, που είναι το κλειδί που πρέπει να έχει κανείς για να χρησιμοποιήσει την υπηρεσία. Αυτό λαμβάνεται εύκολα από τους χρήστες του GKG API, με την προϋπόθεση ότι έχουν λογαριασμό Google.

- `query`, που είναι ο όρος αναζήτησης στον γράφο γνώσης.
- `ids`, που μπορεί να χρησιμοποιηθεί για αναζήτηση οντοτήτων του GKG με βάση το μοναδικό τους αναγνωριστικό, που στη συνέχεια θα αναφέρεται ως GKG ID.
- `languages`, που μπορεί να χρησιμοποιηθεί για αναζήτηση σε γλώσσες πέραν της αγγλικής, που είναι η προεπιλογή, αλλά και σε πολλές γλώσσες παράλληλα.
- `types`, που μπορεί να περιορίσει τα αποτελέσματα που λαμβάνονται σε αυτά που ανήκουν σε κάποιους τύπους, όπως αυτοί ορίζονται από το schema.org.
- `indent`, που μπορεί να χρησιμοποιηθεί για καλύτερη μορφοποίηση των αποτελεσμάτων του GKG API.
- `prefix`, που επιτρέπει το ταίριασμα του όρου αναζήτησης με τα προθέματα των ονομάτων και των συνωνύμων των οντοτήτων.
- `limit`, που περιορίζει το πλήθος οντοτήτων που θα επιστραφούν από το GKG API. Η μέγιστη τιμή που μπορεί να λάβει αυτή η παράμετρος είναι 222.

Η απάντηση σε ένα ερώτημα προς το GKG API λαμβάνεται σε JSON-LD, η οποία είναι μια ειδική κατηγορία της γλώσσας JSON που αφορά συνδεδεμένα δεδομένα (Linked Data). Η απάντηση επιστρέφει ένα σύνολο οντοτήτων του GKG, και χαρακτηρίζει κάθε οντότητα που επιστράφηκε με κάποια από τα εξής πεδία:

- `@id`, που είναι το GKG ID της οντότητας, ένα μοναδικό αναγνωριστικό που μπορεί να χρησιμοποιηθεί για να την ξεχωρίσει από τις υπόλοιπες οντότητες του GKG.
- `name`, που είναι το όνομα της οντότητας.
- `@type`, που είναι οι schema.org τύποι στους οποίους ανήκει η οντότητα.
- `description`, που είναι μια σύντομη περιγραφή της οντότητας.
- `image`, που είναι μια εικόνα της οντότητας, ώστε να αναγνωριστεί ευκολότερα από τους χρήστες.
- `detailedDescription`, που είναι μια εκτενέστερη περιγραφή της οντότητας σε σχέση με αυτήν που προσφέρεται από το πεδίο `description`. Αυτή η περιγραφή αντλείται από την Wikipedia, και είναι οι πρώτες προτάσεις της πρώτης παραγράφου του αντίστοιχου άρθρου της οντότητας, οι οποίες δίνονται μέσω του `articleBody`. Επίσης, παρατίθεται και ο σύνδεσμος του εν λόγω Wikipedia άρθρου (`url`), και μια άδεια χρήσης (`license`).
- `url`, που είναι ο σύνδεσμος της επίσημης ιστοσελίδας της οντότητας.
- `resultScore`, που είναι ένα μέτρο του κατά πόσο η οντότητα ταιριάζει με τα κριτήρια αναζήτησης.

Σημειώνεται ότι σε πολλές περιπτώσεις κάποια από τα παραπάνω πεδία λείπουν από τις απαντήσεις του GKG API. Για παράδειγμα, κάποιες οντότητες δεν έχουν αντίστοιχο άρθρο στην Wikipedia, ενώ άλλες δε διαθέτουν επίσημη ιστοσελίδα. Για να γίνει περισσότερο κατανοητός ο τρόπος χρήσης του GKG API, παρουσιάζονται παραδείγματα χρήσης στα Τμήματα Κώδικα 1 και 2.

Τμήμα Κώδικα 1: Εδώ αναζητούμε στο GKG API οντότητες με βάση το string Obama, με τύπο Person, ζητώντας το αποτέλεσμα να είναι μορφοποιημένο, και οι οντότητες που θα επιστραφούν να μην ξεπερνούν τις δύο. Παρουσιάζεται το ερώτημα που υποβάλλεται στο API, καθώς και η απάντηση που λαμβάνεται σε JSON-LD. Όπως ήταν αναμενόμενο, επιστρέφεται το πρώην προεδρικό ζεύγος των ΗΠΑ.

```
https://kgsearch.googleapis.com/v1/entities:search?key=API_KEY&
↪ query=Obama&types=Person&indent=True&limit=2
```

```
{
  "@context": {
    "@vocab": "http://schema.org/",
    "goog": "http://schema.googleapis.com/",
    "EntitySearchResult": "goog:EntitySearchResult",
    "detailedDescription": "goog:detailedDescription",
    "resultScore": "goog:resultScore",
    "kg": "http://g.co/kg"
  },
  "@type": "ItemList",
  "itemListElement": [
    {
      "@type": "EntitySearchResult",
      "result": {
        "@id": "kg:/m/02mjmr",
        "name": "Barack Obama",
        "@type": [
          "Thing",
          "Person"
        ],
        "description": "44th U.S. President",
        "image": {
          "contentUrl":
          ↪ "http://t0.gstatic.com/images?q=tbn:ANd9GcSkJEGgR2wJ0bp8
          ↪ DhOXx2QuexPLTslqt0v-G2iTiDWVp3iRhSnc",
          "url": "https://commons.wikimedia.org/wiki/
          ↪ File:BarackObama2005portrait.jpg"
        },
```

```
      "detailedDescription": {
        "articleBody": "Barack Hussein Obama II is an American
        ↪ politician who served as the 44th President of the United
        ↪ States from 2009 to 2017. He is the first African
        ↪ American to have served as president, as well as the
        ↪ first born outside the contiguous United States. ",
        "url": "https://en.wikipedia.org/wiki/Barack_Obama",
        "license":
        ↪ "https://en.wikipedia.org/wiki/Wikipedia:Text_of_Creative_
        ↪ Commons_Attribution-ShareAlike_3.0_Unported_License"
      },
      "url": "http://whitehouse.gov"
    },
    "resultScore": 183.739258
  },
  {
    "@type": "EntitySearchResult",
    "result": {
      "@id": "kg:/m/025s5v9",
      "name": "Michelle Obama",
      "@type": [
        "Person",
        "Thing"
      ],
      "description": "Former First Lady of the United States",
      "image": {
        "contentUrl":
        ↪ "http://t3.gstatic.com/images?q=tbn:ANd9GcQPO1I4Mc2rkyOz
        ↪ X7OCTXd1M3b5_sPLqnYPOSMnV3__JkwSMXl5",
        "url": "https://en.wikipedia.org/wiki/Michelle_Obama"
      },
      "detailedDescription": {
        "articleBody": "Michelle LaVaughn Robinson Obama is an
        ↪ American lawyer and writer who was First Lady of the
        ↪ United States from 2009 to 2017. She is married to the
        ↪ 44th President of the United States, Barack Obama, and
        ↪ was the first African-American First Lady. ",
        "url": "https://en.wikipedia.org/wiki/Michelle_Obama",
        "license":
        ↪ "https://en.wikipedia.org/wiki/Wikipedia:Text_of_Creative_
        ↪ Commons_Attribution-ShareAlike_3.0_Unported_License"
      },
      "url":
      ↪ "http://www.whitehouse.gov/1600/first-ladies/michelleobama"
```

```
      },
      "resultScore": 63.544678
    }
  ]
}
```


Τμήμα Κώδικα 2: Εδώ αναζητούμε στο GKG API οντότητες με βάση το string Tripoli, στην αγγλική και στην ελληνική γλώσσα, με τύπο City, ζητώντας το αποτέλεσμα να είναι μορφοποιημένο, και οι οντότητες που θα επιστραφούν να μην ξεπερνούν τις τρεις. Παρουσιάζεται το ερώτημα που υποβάλλεται στο API, καθώς και η απάντηση που λαμβάνεται σε JSON-LD. Επιστρέφεται η πρωτεύουσα της Λιβύης, η πόλη που εντοπίζεται στο βόρειο Λίβανο και η πρωτεύουσα του Νομού Αρκαδίας.

```
https://kgsearch.googleapis.com/v1/entities:search?key=API_KEY&
↪ query=Tripoli&languages=en,el&types=City&indent=True&limit=3
```

```
{
  "@context": {
    "@vocab": "http://schema.org/",
    "goog": "http://schema.googleapis.com/",
    "EntitySearchResult": "goog:EntitySearchResult",
    "detailedDescription": "goog:detailedDescription",
    "resultScore": "goog:resultScore",
    "kg": "http://g.co/kg"
  },
  "@type": "ItemList",
  "itemListElement": [
    {
      "@type": "EntitySearchResult",
      "result": {
        "@id": "kg:/m/07p7g",
        "name": [
          {
            "@value": "Tripoli",
            "@language": "en"
          },
          {
            "@value": "Τρίπολη",
            "@language": "el"
          }
```

```
],
"@type": [
  "City",
  "Place",
  "Thing"
],
"description": [
  {
    "@value": "City",
    "@language": "en"
  },
  {
    "@value": "Πόλη",
    "@language": "el"
  }
],
"image": {
  "contentUrl": "http://t2.gstatic.com/images?q=tbn:ANd9GcQ-
  ↪ _LrItTcYVAO34XpWYwvWblfZgiVHl0jwShTVe-p4apXgtgUL",
  "url":
  ↪ "https://commons.wikimedia.org/wiki/File:Marcus_Aurelius_
  ↪ Arch_Tripoli_Libya.jpg",
  "license": "http://creativecommons.org/licenses/by/2.0"
},
"detailedDescription": [
  {
    "articleBody": "Tripoli is the capital city and the largest
    ↪ city of Libya. Tripoli, with its metropolitan area, has
    ↪ a population of about 1.1 million people. ",
    "inLanguage": "en",
    "url": "https://en.wikipedia.org/wiki/Tripoli",
    "license":
    ↪ "https://en.wikipedia.org/wiki/Wikipedia:Text_of_
    ↪ Creative_Commons_Attribution-
    ↪ ShareAlike_3.0_Unported_License"
  },
  {
```

```
        "articleBody": "Η Τρίπολη είναι πρωτεύουσα της Λιβύης.
        ↪ Είναι μεγαλύτερη πόλη της χώρας με πληθυσμό 1.682.000
        ↪ κατοίκων και έκταση 400 τετ.χλμ. Βρέχεται από τη
        ↪ Μεσόγειο Θάλασσα. Είναι το κυριότερο βιομηχανικό και
        ↪ πολιτιστικό κέντρο και το πιο μεγάλο λιμάνι της
        ↪ Λιβύης.\nΙδρύθηκε το XII αιώνα π.Χ. Στην πόλη υπάρχουν
        ↪ μηχανουργία και βιομηχανία τροφίμων. Στην πόλη
        ↪ βρίσκονται 5 πανεπιστήμια και διεθνής αερολιμένας.",
        "inLanguage": "el",
        "url":
        ↪ "https://el.wikipedia.org/wiki/%CE%A4%CF%81%CE%AF%CF%80
        ↪ %CE%BF%CE%BB%CE%B7_(%CE%9B%CE%B9%CE%B2%CF%8D%CE%B7)",
        "license":
        ↪ "https://en.wikipedia.org/wiki/Wikipedia:Text_of_
        ↪ Creative_Commons_Attribution-
        ↪ ShareAlike_3.0_Unported_License"
      }
    ],
    "url": "http://www.tripoli.info/"
  },
  "resultScore": 77.876526
},
{
  "@type": "EntitySearchResult",
  "result": {
    "@id": "kg:/m/01np92",
    "name": [
      {
        "@value": "Tripoli",
        "@language": "en"
      },
      {
        "@value": "Τρίπολη",
        "@language": "el"
      }
    ],
    "@type": [
      "Thing",
      "City",
      "Place"
    ],
    "description": [
      {
        "@value": "City",
```

```
      "@language": "en"
    },
    {
      "@value": "Πόλη",
      "@language": "el"
    }
  ],
  "image": {
    "contentUrl":
    ↪ "http://t2.gstatic.com/images?q=tbn:ANd9GcS1JW1bVMovAp
    ↪ j2gkJReL6os1xBg5idp_jEZ8m3Ug8dTlwKh-S5",
    "url":
    ↪ "https://commons.wikimedia.org/wiki/File:Tripoli,_Lebanon_
    ↪ photos,_Aug_2012.jpg",
    "license": "http://www.gnu.org/copyleft/fdl.html"
  },
  "detailedDescription": [
    {
      "articleBody": "Tripoli is the largest city in northern
      ↪ Lebanon and the second-largest city in the country.
      ↪ Situated 85 kilometers north of the capital Beirut, it
      ↪ is the capital of the North Governorate and the Tripoli
      ↪ District. ",
      "inLanguage": "en",
      "url": "https://en.wikipedia.org/wiki/Tripoli,_Lebanon",
      "license":
      ↪ "https://en.wikipedia.org/wiki/Wikipedia:Text_of_
      ↪ Creative_Commons_Attribution-
      ↪ ShareAlike_3.0_Unported_License"
    },
    {
      "articleBody": "Η Τρίπολη είναι η μεγαλύτερη πόλη στο
      ↪ βόρειο Λίβανο και η δεύτερη μεγαλύτερη πόλη στη χώρα.
      ↪ Βρίσκεται 85 χλμ. βόρεια της πρωτεύουσας Βηρυτού και
      ↪ αποτελεί την πρωτεύουσα του Βόρειου Κυβερνείου και της
      ↪ επαρχίας Τριπόλεως. ",
      "inLanguage": "el",
      "url":
      ↪ "https://el.wikipedia.org/wiki/%CE%A4%CF%81%CE%AF%CF%80
      ↪ %CE%BF%CE%BB%CE%B7_(%CE%9B%CE%AF%CE%B2%CE%B1%CE
      ↪ %BD%CE%BF%CF%82)",
```

```
          "license":
          ↪ "https://en.wikipedia.org/wiki/Wikipedia:Text_of_
          ↪ Creative_Commons_Attribution-
          ↪ ShareAlike_3.0_Unported_License"
        }
      ],
      "url": "http://tripoli-city.org/"
    },
    "resultScore": 50.12508
  },
  {
    "@type": "EntitySearchResult",
    "result": {
      "@id": "kg:/m/02fyn4",
      "name": [
        {
          "@value": "Tripoli",
          "@language": "en"
        },
        {
          "@value": "Τρίπολη Αρκαδίας",
          "@language": "el"
        }
      ],
      "@type": [
        "Place",
        "Thing",
        "City"
      ],
      "description": [
        {
          "@value": "City",
          "@language": "en"
        },
        {
          "@value": "Πόλη",
          "@language": "el"
        }
      ],
      "image": {
        "contentUrl":
        ↪ "http://t3.gstatic.com/images?q=tbn:ANd9GcTDrSUGxUsnEyQ
        ↪ b2zilg4Ilaxj-3cOoDCkEMifLbaR0z267sFE8",
```

```
        "url": "https://commons.wikimedia.org/wiki/
        ↪ File:TripoliGreece1.jpg",
        "license": "http://creativecommons.org/licenses/by-sa/3.0"
      },
      "detailedDescription": [
        {
          "articleBody": "Tripoli is a city in the central part of
          ↪ the Peloponnese, in Greece. It is the capital of the
          ↪ Peloponnese region as well as of the regional unit of
          ↪ Arcadia. The homonym municipality has around 47,000
          ↪ inhabitants.",
          "inLanguage": "en",
          "url": "https://en.wikipedia.org/wiki/Tripoli,_Greece",
          "license":
          ↪ "https://en.wikipedia.org/wiki/Wikipedia:Text_of_
          ↪ Creative_Commons_Attribution-
          ↪ ShareAlike_3.0_Unported_License"
        },
        {
          "articleBody": "Η Τρίπολη είναι πόλη στην κεντρική
          ↪ Πελοπόννησο, η μεγαλύτερη πόλη και πρωτεύουσα του Νομού
          ↪ Αρκαδίας και της τέως επαρχίας Μαντινείας. Βρίσκεται σε
          ↪ υψόμετρο 660 μέτρων. Είναι επίσης έδρα της Περιφέρειας
          ↪ Πελοποννήσου. Ο πληθυσμός του ομώνυμου Δήμου είναι
          ↪ 47.457 κάτοικοι.",
          "inLanguage": "el",
          "url":
          ↪ "https://el.wikipedia.org/wiki/%CE%A4%CF%81%CE%AF%CF%80
          ↪ %CE%BF%CE%BB%CE%B7_%CE%91%CF%81%CE%BA%CE%B1%CE%B4%CE%AF
          ↪ %CE%B1%CF%82",
          "license":
          ↪ "https://en.wikipedia.org/wiki/Wikipedia:Text_of_
          ↪ Creative_Commons_Attribution-
          ↪ ShareAlike_3.0_Unported_License"
        }
      ],
      "url": "http://www.tripolis.gr/"
    },
    "resultScore": 39.179363
  }
 ]
}
```

Πίνακας 4: Τα namespaces των σελίδων της Wikipedia. Παρατίθενται οι κωδικοί των namespaces και η κατηγορία των αντίστοιχων σελίδων.

Subject namespaces		Talk namespaces	
0	Main/Article	Talk	1
2	User	User talk	3
4	Wikipedia	Wikipedia talk	5
6	File	File talk	7
8	MediaWiki	MediaWiki talk	9
10	Template	Template talk	11
12	Help	Help talk	13
14	Category	Category talk	15
100	Portal	Portal talk	101
108	Book	Book talk	109
118	Draft	Draft talk	119
446	Education Program	Education Program talk	447
710	TimedText	TimedText talk	711
828	Module	Module talk	829
2300	Gadget	Gadget talk	2301
2302	Gadget Definition	Gadget Definition talk	2303

3.2 MediaWiki API

Το MediaWiki API είναι μια διαδικτυακή υπηρεσία που προσφέρει πρόσβαση στα δεδομένα της Wikipedia. Κάποια από τα δεδομένα αυτά είναι χρήσιμα σε μια εφαρμογή αποσαφήνισης οντοτήτων. Συγκεκριμένα, ενδιαφέρουν οι σελίδες αποσαφήνισης της Wikipedia (disambiguation pages), οι σελίδες ανακατεύθυνσης της Wikipedia (redirect pages) και οι εισερχόμενοι σύνδεσμοι για ένα άρθρο της Wikipedia (what links here).

3.2.1 MediaWiki Links API

Το MediaWiki Links API επιτρέπει την εύρεση των συνδέσμων σε μια σελίδα της Wikipedia. Η εύρεση των συνδέσμων δεν περιορίζεται στα συνηθισμένα άρθρα, αλλά αφορά όλες τις σελίδες που διατηρεί η Wikipedia. Αυτό σημαίνει ότι μπορεί να χρησιμοποιηθεί για να βρεθούν οι προτάσεις αποσαφήνισης της Wikipedia, αντλώντας τους συνδέσμους από την αντίστοιχη σελίδα αποσαφήνισης. Τα ερωτήματα προς το Links API γίνονται μέσω HTTP GET requests της μορφής `https://en.wikipedia.org/w/api.php?action=query&prop=links`, με τις εξής παραμέτρους:

- `titles`, που ορίζει τον τίτλο της σελίδας για την οποία θέλουμε να ανακτήσουμε τους συνδέσμους.
- `plnamespace`, που περιορίζει το είδος των συνδέσμων που επιστρέφονται με βάση το namespace της σελίδας στην οποία δείχνουν. Στον Πίνακα 4 φαίνονται τα δυνατά

namespaces που μπορεί να έχει μια σελίδα της Wikipedia.

- `pllimit`, που ορίζει το μέγιστο πλήθος αποτελεσμάτων που θα επιστραφούν. Η μέγιστη τιμή που μπορεί να λάβει αυτή η παράμετρος είναι 500.
- `plcontinue`, που επιτρέπει τη συνέχεια λήψης αποτελεσμάτων σε περίπτωση που το `pllimit` δεν αρκεί για να λάβουμε όλα τα αποτελέσματα με την πρώτη.
- `pltitles`, που επιτρέπει το φιλτράρισμα με βάση τους δοσμένους τίτλους, ώστε, για παράδειγμα, να ελέγξουμε αν ένα άρθρο έχει σύνδεσμο προς κάποιο άλλο άρθρο.
- `pldir`, που ελέγχει την κατεύθυνση στη σελίδα με την οποία επιστρέφονται οι σύνδεσμοι, από πάνω προς τα κάτω ή από κάτω προς τα πάνω.

Η απάντηση επιστρέφει μια λίστα από σελίδες της Wikipedia, σύνδεσμοι προς τις οποίες υπάρχουν στη σελίδα που ορίστηκε μέσω της παραμέτρου `titles`. Όπως αναφέρθηκε ήδη, μας ενδιαφέρουν κυρίως οι σύνδεσμοι που περιέχονται σε σελίδες αποσαφήνισης. Ένα παράδειγμα ανάκτησης των συνδέσμων σε μια σελίδα αποσαφήνισης φαίνεται στο Τμήμα Κώδικα 3. Σημειώνεται η χρήση της επιπλέον παραμέτρου `redirects`, που χρησιμεύει ώστε να γίνει ανακατεύθυνση σε περίπτωση που κάποια σελίδα αποσαφήνισης είναι redirect.

Τμήμα Κώδικα 3: Εδώ αναζητούμε στο MediaWiki Links API τους συνδέσμους που περιέχονται στο Wikipedia disambiguation page «Greece (disambiguation)». Ζητούνται μόνο οι σύνδεσμοι που αντιστοιχούν σε κύριες σελίδες/άρθρα (με namespace 0), ενώ το όριο του πλήθους αποτελεσμάτων τίθεται ίσο με πέντε.

```
https://en.wikipedia.org/w/api.php?action=query&prop=links&
↪ titles=Greece%20(disambiguation)&plnamespace=0&pllimit=5&redirects
```

```
{
    "continue": {
        "plcontinue": "421372|0|Grecia_(disambiguation)",
        "continue": "||"
    },
    "query": {
        "pages": {
            "421372": {
                "pageid": 421372,
                "ns": 0,
                "title": "Greece (disambiguation)",
                "links": [
                    {
                        "ns": 0,
                        "title": "Ancient Greece"
                    },
```

```
                    {
                        "ns": 0,
                        "title": "Archaic Greece"
                    },
                    {
                        "ns": 0,
                        "title": "Byzantine Greece"
                    },
                    {
                        "ns": 0,
                        "title": "Classical Greece"
                    },
                    {
                        "ns": 0,
                        "title": "First Hellenic Republic"
                    }
                ]
            }
        }
    }
}
```


3.2.2 MediaWiki Redirects API

Το MediaWiki Redirects API επιτρέπει την εύρεση των redirects, δηλαδή των σελίδων ανακατεύθυνσης, για μια σελίδα της Wikipedia. Τα redirects μια σελίδας της Wikipedia εκφράζουν συνώνυμα για την ίδια οντότητα. Αυτή η πληροφορία είναι χρήσιμη για μια εφαρμογή αποσαφήνισης οντοτήτων, καθώς μας ενδιαφέρουν τα διαφορετικά εναλλακτικά ονόματα (aliases) με τα οποία μπορεί να εμφανιστεί η ίδια οντότητα. Τα ερωτήματα προς το Redirects API γίνονται μέσω HTTP requests της μορφής `https://en.wikipedia.org/w/api.php?action=query&prop=redirects`, με τις εξής παραμέτρους:

- `titles`, που ορίζει τον τίτλο της σελίδας για την οποία θέλουμε να ανακτήσουμε τα redirects.
- `rdprop`, που ορίζει τα χαρακτηριστικά των redirects που θα επιστραφούν, για παράδειγμα μόνο ο τίτλος των αντίστοιχων σελίδων, ή ο τίτλος μαζί με το αντίστοιχο pageid.
- `rdnamespace`, που περιορίζει το είδος των αποτελεσμάτων που επιστρέφονται με βάση τα επιθυμητά namespaces.
- `rdshow`, που ορίζει αν εμφανίζονται τα fragment ή τα non-fragment redirects, δηλαδή redirects που αναφέρονται σε κάποια συγκεκριμένο τμήμα μιας σελίδας ή σε μια σελίδα συνολικά.

- `rdlimit`, που ορίζει το μέγιστο πλήθος αποτελεσμάτων που θα επιστραφούν. Η μέγιστη τιμή που μπορεί να λάβει αυτή τη παράμετρος είναι 500.
- `rdcontinue`, που επιτρέπει τη συνέχεια λήψης αποτελεσμάτων σε περίπτωση που το `rdlimit` δεν αρκεί για να λάβουμε όλα τα αποτελέσματα με την πρώτη.

Η απάντηση επιστρέφει μια λίστα από σελίδες της Wikipedia, οι οποίες ανακατευθύνουν προς τη σελίδα που ορίστηκε μέσω της παραμέτρου `titles`. Στο Τμήμα Κώδικα 4 παρουσιάζεται ένα παράδειγμα χρήσης του Redirects API.

Τμήμα Κώδικα 4: Εδώ αναζητούμε στο MediaWiki Redirects API τα redirects για τη σελίδα «Michael Jackson», με το όριο του πλήθους αποτελεσμάτων να τίθεται ίσο με πέντε.

```
https://en.wikipedia.org/w/api.php?action=query&prop=redirects&
↪ titles=Michael%20Jackson&rdlimit=5
```

```
{
    "continue": {
        "rdcontinue": "3238513",
        "continue": "||"
    },
    "query": {
        "pages": {
            "14995351": {
                "pageid": 14995351,
                "ns": 0,
                "title": "Michael Jackson",
                "redirects": [
                    {
                        "pageid": 409661,
                        "ns": 0,
                        "title": "Michael Joseph Jackson"
                    },
                    {
                        "pageid": 945100,
                        "ns": 0,
                        "title": "Wacko Jacko"
                    },
                    {
                        "pageid": 1328480,
                        "ns": 0,
                        "title": "Shamone"
                    },
                    {
```

```
                    "pageid": 1331427,
                    "ns": 0,
                    "title": "Chamone"
                },
                {
                    "pageid": 1335629,
                    "ns": 0,
                    "title": "Cha'mone"
                }
            ]
        }
    }
}
```


3.2.3 MediaWiki Linkshere API

Το MediaWiki Linkshere API επιτρέπει την εύρεση των σελίδων της Wikipedia που περιέχουν συνδέσμους προς μια δεδομένη σελίδα. Προς αποφυγή σύγχυσης, σημειώνεται ότι το Links API, που περιγράφηκε στην παράγραφο 3.2.1, επιστρέφει όλους του συνδέσμους που περιέχονται σε κάποια σελίδα p της Wikipedia. Αντιθέτως, το Linkshere API επιστρέφει όλες τις σελίδες της Wikipedia που περιέχουν έναν ή περισσότερους συνδέσμους προς κάποια σελίδα p. Αυτό είναι χρήσιμο σε μια εφαρμογή αποσαφήνισης οντοτήτων, καθώς, όπως αναφέρθηκε στην υποπαράγραφο 2.3.4.2, οι μετρικές υπολογισμού σημασιολογικής συνοχής μεταξύ οντοτήτων σε ένα κείμενο απαιτούν γνώση των εισερχόμενων συνδέσμων των άρθρων της Wikipedia. Τα ερωτήματα προς το Linkshere API γίνονται μέσω HTTP requests της μορφής `https://en.wikipedia.org/w/api.php?action=query&prop=linkshere`, με τις εξής παραμέτρους:

- `titles`, που ορίζει τον τίτλο της σελίδας για την οποία θέλουμε να ανακτήσουμε τους εισερχόμενους συνδέσμους.
- `lhprop`, που ορίζει τα χαρακτηριστικά των σελίδων που θα επιστραφούν.
- `lhnamespace`, που περιορίζει τα αποτελέσματα που θα επιστραφούν βάσει namespaces.
- `lhshow`, που ορίζει αν εμφανίζονται αποτελέσματα που είναι redirects ή non-redirects.
- `lhlimit`, που ορίζει το μέγιστο πλήθος αποτελεσμάτων που θα επιστραφούν. Η μέγιστη τιμή που μπορεί να λάβει αυτή η παράμετρος είναι 500.
- `lhcontinue`, που επιτρέπει τη συνέχεια λήψης αποτελεσμάτων σε περίπτωση που το `lhlimit` δεν αρκεί για να λάβουμε όλα τα αποτελέσματα με την πρώτη.

Η απάντηση επιστρέφει μια λίστα από σελίδες της Wikipedia, που δείχνουν προς τη σελίδα που ορίστηκε μέσω της παραμέτρου `titles`. Στο Τμήμα Κώδικα 5 παρουσιάζεται ένα παράδειγμα χρήσης του Linkshere API.

Τμήμα Κώδικα 5: Εδώ αναζητούμε στο MediaWiki Linkshere API τους εισερχόμενους συνδέσμους προς το άρθρο «Barack Obama», με το όριο του πλήθους αποτελεσμάτων να τίθεται ίσο με πέντε.

```
https://en.wikipedia.org/w/api.php?action=query&prop=linkshere&
↪ titles=Barack%20Obama&lhlimit=5
```

```
{
    "continue": {
        "lhcontinue": "1098",
        "continue": "||"
    },
    "query": {
        "pages": {
            "534366": {
                "pageid": 534366,
                "ns": 0,
                "title": "Barack Obama",
                "linkshere": [
                    {
                        "pageid": 307,
                        "ns": 0,
                        "title": "Abraham Lincoln"
                    },
                    {
                        "pageid": 624,
                        "ns": 0,
                        "title": "Alaska"
                    },
                    {
                        "pageid": 662,
                        "ns": 0,
                        "title": "Apollo 11"
                    },
                    {
                        "pageid": 931,
                        "ns": 0,
                        "title": "The Amazing Spider-Man"
                    },
```

```
                    {
                        "pageid": 1029,
                        "ns": 0,
                        "title": "Albert Schweitzer"
                    }
                ]
            }
        }
    }
}
```


3.3 Stanford CoreNLP

Το Stanford CoreNLP [MSB+14] είναι ένα σύνολο εργαλείων επεξεργασίας φυσικής γλώσσας. Προσφέρει διάφορες δυνατότητες, όπως lemmatizing, part of speech tagging και sentiment analysis. Στο πλαίσιο ενός συστήματος αποσαφήνισης οντοτήτων σε κείμενο, ωστόσο, το Stanford CoreNLP χρησιμεύει περισσότερο ως εργαλείο αναγνώρισης ονοματικών οντοτήτων. Όπως αναφέρθηκε στην παράγραφο 2.3.2, η εύρεση των ονοματικών οντοτήτων σε ένα κείμενο είναι ένα βασικό στάδιο που πρέπει να προηγηθεί της κυρίως διαδικασίας αποσαφήνισης. Αυτή η διαδικασία μπορεί να εκτελεσθεί από τον EntityMentionsAnnotator του Stanford CoreNLP με ικανοποιητική ακρίβεια. Αυτός ο annotator εντοπίζει σε αδόμητο κείμενο τα όρια οντοτήτων, και μπορεί να χρησιμοποιηθεί με τα εξής μοντέλα ανάλογα με τους τύπους οντοτήτων που θέλουμε να εξάγουμε από το κείμενο:

- 3 class model, που βρίσκει οντότητες των τύπων Location, Person και Organization.
- 4 class model, που βρίσκει οντότητες των τύπων Location, Person, Organization και Misc, έτσι ώστε να υπάρχει μια κατηγορία για τις οντότητες που δεν ανήκουν στις τρεις πρώτες κατηγορίες.
- 7 class model, που βρίσκει οντότητες των τύπων Location, Person, Organization, Money, Percent, Date και Time.

Η πληροφορία τύπων που αντλείται από το Stanford CoreNLP μπορεί να χρησιμοποιηθεί στη συνέχεια, ώστε να γίνει πιο εύκολα η αποσαφήνιση. Σαφώς η γνώση ότι μια οντότητα είναι τοποθεσία ή πρόσωπο ή οργανισμός μπορεί να διευκολύνει τη διαδικασία αποσαφήνισής της. Ένα παράδειγμα χρήσης του Stanford CoreNLP, με τη βοήθεια του Python pycorenlp API, φαίνεται στο Τμήμα Κώδικα 6.

Τμήμα Κώδικα 6: Εδώ χρησιμοποιούμε τον EntityMentionsAnnotator του Stanford CoreNLP για να αναλύσουμε ένα κείμενο παρμένο από τον τίτλο ενός ειδησεογραφικού άρθρου, με στόχο την εξαγωγή των ονοματικών οντοτήτων που περιέχει. Χρησιμοποιείται το 3 class model. Παρατηρούμε ότι επιστρέφονται τα όρια των ονοματικών οντοτήτων, δηλαδή ο χαρακτήρας εκκίνησης και ο χαρακτήρας τερματισμού, καθώς και ο προτεινόμενος τύπος της εκάστοτε οντότητας. Εδώ βλέπουμε ότι το σύστημα αποφάνθηκε σωστά για το πλήθος των οντοτήτων, για τα όριά τους, καθώς και για τους τύπους τους.

```
# import Stanford CoreNLP wrapper
from pycorenlp import StanfordCoreNLP
# import package for pretty table printing
from prettytable import PrettyTable

# Stanford CoreNLP is running on localhost
nlp = StanfordCoreNLP('http://localhost:9000')

# text to annotate
text = 'Trump fires FBI director Comey, raising questions over
↪ Russia investigation.'

# analyze text with the entitymentions annotator, using the 3 class
↪ model
# and get the result in JSON
output = nlp.annotate(text, properties={
    'annotators': 'entitymentions',
    'ner.model': 'edu/stanford/nlp/models/ner/'
                 'english.all.3class.distsim.crf.ser.gz',
    'outputFormat': 'json'
})

# list of extracted entities
extracted_entities = []

# populate list of extracted entities using Stanford CoreNLP's
↪ output
for sentence in output['sentences']:
    for mention in sentence['entitymentions']:
        extracted_entities.append({
            'matched_string': mention['text'],
            'suggested_type': mention['ner'],
            'start_offset': mention['characterOffsetBegin'],
            'end_offset': mention['characterOffsetEnd']
        })
```

```python

# format results
t = PrettyTable(['Matched String', 'Start Offset', 'End Offset', 'Suggested Type'])

for entity in extracted_entities:
    t.add_row([entity['matched_string'], entity['start_offset'], entity['end_offset'], entity['suggested_type']])

# print results

print(t)
```

```
+----------------+--------------+------------+----------------+
| Matched String | Start Offset | End Offset | Suggested Type |
+----------------+--------------+------------+----------------+
|     Trump      |      0       |     5      |     PERSON     |
|      FBI       |      12      |     15     |  ORGANIZATION  |
|     Comey      |      25      |     30     |     PERSON     |
|     Russia     |      55      |     61     |    LOCATION    |
+----------------+--------------+------------+----------------+
```

Κεφαλαιο 4

Ανάλυση και σχεδίαση συστήματος αποσαφήνισης

Στο κεφάλαιο αυτό αναλύονται οι βασικές σχεδιαστικές αποφάσεις που λήφθηκαν κατά τον σχεδιασμό του συστήματος αποσαφήνισης οντοτήτων σε κείμενο. Ακολουθείται, σε γενικές γραμμές, η συλλογιστική πορεία του κεφαλαίου 2, όσον αφορά τα κύρια σημεία του συστήματος. Σημειώνεται ότι δίνεται μεγαλύτερη έμφαση στην κατανόηση της βασικής διαδικασίας αποσαφήνισης οντοτήτων σε κείμενο, και για αυτό χρησιμοποιείται ψευδοκώδικας ή φυσική γλώσσα για την περιγραφή των επιμέρους λειτουργιών.

4.1 Αποσαφήνιση στην αγγλική γλώσσα

Το σύστημα αποσαφήνισης που σχεδιάζουμε επεξεργάζεται κείμενα στην αγγλική γλώσσα. Για τους λόγους που αναφέρθηκαν και στην παράγραφο 2.3.1, υπάρχει πολύ μεγαλύτερο υπόβαθρο στην επεξεργασία της συγκεκριμένης γλώσσας, από καλύτερα εργαλεία αναγνώρισης ονοματικών αναφορών, μέχρι πλουσιότερες βάσεις γνώσης με τις οποίες μπορεί να γίνει η αποσαφήνιση. Έτσι, ξεκινάμε με την αγγλική γλώσσα, ώστε να δημιουργήσουμε το βασικό σύστημα αποσαφήνισης.

4.2 Αναγνώριση ονοματικών οντοτήτων με το Stanford CoreNLP

Ένα βήμα που πρέπει να προηγηθεί της αποσαφήνισης ονοματικών οντοτήτων σε κείμενο T είναι η αναγνώρισή τους μέσα στο κείμενο, δηλαδή η εύρεση των ορίων τους. Αυτό σημαίνει ότι θα πρέπει να προσδιορισθεί το σύνολο $M = \{m_1, m_2, \dots, m_k\}$ των ονοματικών αναφορών στο κείμενο, στις οποίες θα πρέπει να αντιστοιχιστούν οντότητες του κόσμου. Αν αυτές οι αναφορές δε δίνονται σαν κομμάτι της εισόδου του συστήματος, τότε χρησιμοποιείται το εργαλείο Stanford CoreNLP, που έχει αναλυθεί ήδη στην ενότητα 3.3. Συγκεκριμένα, ακολουθούνται τα εξής βήματα:

1. Γίνεται εξαγωγή ονοματικών οντοτήτων με βάση το 3 class model (Location, Person και Organization).

2. Γίνεται εξαγωγή ονοματικών οντοτήτων με βάση το 4 class model (Location, Person, Organization και Misc), και κρατιούνται οι οντότητες που δεν επικαλύπτονται με αυτές

Αλγόριθμος 6: Αναγνώριση ονοματικών οντοτήτων σε κείμενο T με βάση τα 3 class, 4 class και 7 class models του Stanford CoreNLP.

```
1:  function NamedEntityRecognition(T)
2:      M ← ∅
3:      for each m ∈ T according to 3 class model do
4:          M ← M ∪ {m}
5:      end for
6:      for each m ∈ T according to 4 class model do
7:          if m doesn't overlap with any mention in M then
8:              M ← M ∪ {m}
9:          end if
10:     end for
11:     for each m ∈ T according to 7 class model do
12:         if m doesn't overlap with any mention in M then
13:             if m has type Location or Person or Organization then
14:                 M ← M ∪ {m}
15:             end if
16:         end if
17:     end for
18:     return M
19: end function
```

που προέκυψαν από το βήμα 1.

3. Γίνεται εξαγωγή ονοματικών οντοτήτων με βάση το 7 class model (Location, Person, Organization, Money, Percent, Date και Time), και κρατιούνται οι οντότητες με τύπους Location, Person ή Organization, που δεν επικαλύπτονται με αυτές που προέκυψαν από τα βήματα 1 και 2. Θεωρήθηκε ότι οι αναφορές σε οντότητες τύπου Money, Percent, Date και Time δεν ενδιαφέρουν για αποσαφήνιση ονοματικών οντοτήτων.

Η μέθοδος αναγνώρισης ονοματικών οντοτήτων σε κείμενο T περιγράφεται πιο τυπικά στον Αλγόριθμο 6. Σημειώνεται ότι οι τύποι των οντοτήτων δε χρησιμοποιούνται στη συνέχεια της διαδικασίας αποσαφήνισης, καθώς διαπιστώθηκε ότι το σύστημα τύπων του Stanford CoreNLP είναι επιρρεπές σε λάθη, και αυτό μπορεί να αλλοιώσει το αποτέλεσμα της αποσαφήνισης. Βέβαια, όπως έχει αναφερθεί ήδη, ένα σύστημα αποσαφήνισης δεν είναι ανάγκη να εκτελεί και αναγνώριση ονοματικών οντοτήτων, καθώς οι ονοματικές αναφορές προς αποσαφήνιση μπορούν να δίνονται ως τμήμα της εισόδου.

4.3 Παραγωγή υποψήφιων οντοτήτων με χρήση APIs

Αφού έχουν προσδιορισθεί οι αναφορές προς αποσαφήνιση, θα πρέπει να παραχθούν τα αντίστοιχα σύνολα υποψήφιων οντοτήτων. Όπως αναφέρθηκε στην παράγραφο 2.3.3, θα θέλαμε για κάθε αναφορά $m \in M$ να προσδιορίσουμε ένα σύνολο υποψήφιων οντοτήτων E_m, που απαρτίζεται από τις οντότητες του κόσμου στις οποίες κανείς θα μπορούσε να

αναφερθεί λέγοντας m. Σε πρώτη φάση, αυτό μπορεί να γίνει με χρήση του GKG API, που αναλύθηκε στην ενότητα 3.1. Συγκεκριμένα, το GKG API ερωτάται με το string της αναφοράς m και με κάποιο λογικό όριο, και η λίστα των οντοτήτων που επιστρέφεται αποτελεί τη βάση για να κατασκευάσουμε το σύνολο υποψήφιων οντοτήτων E_m. Ο τρόπος με τον οποίο ερωτάται το GKG API φαίνεται στον Αλγόριθμο 7. Ωστόσο, αποδεικνύεται ότι δεν είναι όλα τα αποτελέσματα που επιστρέφει το GKG API κατάλληλα για να εισαχθούν στο σύνολο E_m. Αυτό μπορεί να συμβαίνει για δύο κύριους λόγους:

- *Δομικοί λόγοι*. Σε κάποιες περιπτώσεις, κάποια από τα πεδία που χαρακτηρίζουν μια οντότητα στο JSON response του GKG API λείπουν. Για παράδειγμα, μια οντότητα μπορεί να μην έχει άρθρο στην Wikipedia. Τέτοιες οντότητες απορρίπτονται άμεσα, καθώς δεν μπορούν να υποστούν περαιτέρω επεξεργασία από το συγκεκριμένο σύστημα αποσαφήνισης.

- *Σημασιολογικοί λόγοι*. Δυστυχώς, το GKG API δεν είναι κατασκευασμένο με αποκλειστικό στόχο να έχει ρόλο βάσης γνώσης που να αποτελεί τον πυρήνα ενός συστήματος αποσαφήνισης. Σε πολλές περιπτώσεις, παρατηρείται ότι επιστρέφει οντότητες που είναι μεν σχετικές με την αναφορά m, αλλά δεν αποτελούν κατάλληλες υποψήφιες οντότητες για την αναφορά αυτήν. Για παράδειγμα, η αναζήτηση της συμβολοσειράς «UK» επιστρέφει στα πρώτα 100 αποτελέσματα την οντότητα Elizabeth II, δηλαδή τη βασίλισσα του Ηνωμένου Βασιλείου. Αν και είναι προφανής η σχέση των οντοτήτων United Kingdom και Elizabeth II, ποτέ δε χρησιμοποιεί κανείς την αναφορά «UK» ώστε να αναφερθεί στη βασίλισσα του Ηνωμένου Βασιλείου. Συνεπώς, αν και επιστρέφεται η οντότητα Elizabeth II από το GKG API για την αναφορά $m = \text{UK}$, δεν μπορεί να ανήκει στο σύνολο E_m, καθώς δε θα μπορούσαμε να αναφερθούμε στην οντότητα αυτή λέγοντας «UK».

Από τα παραπάνω προκύπτει η ανάγκη φιλτραρίσματος των αποτελεσμάτων του GKG API, ώστε να λάβουμε ένα σύνολο E_m από οντότητες που μπορούμε να επεξεργαστούμε, και που ικανοποιούν τη βασική απαίτηση του συνόλου αυτού. Η απόρριψη των οντοτήτων που δεν είναι δομικά κατάλληλες προς επεξεργασία είναι εύκολη, και έχει να κάνει με το parsing του JSON response που παράγει το GKG API. Το δομικό φιλτράρισμα που χρησιμοποιείται φαίνεται στον Αλγόριθμο 8. Από την άλλη, το σημασιολογικό φιλτράρισμα είναι πιο απαιτητικό. Είναι δύσκολο να προσδιορισθεί, με αυτοματοποιημένο τρόπο, μια μέθοδος με την οποία να απορρίπτονται οι οντότητες στις οποίες δεν μπορούμε να αναφερθούμε λέγοντας m, αλλά όχι περισσότερες. Ωστόσο, μπορούμε να λάβουμε ικανοποιητικά αποτελέσματα χρησιμοποιώντας τα MediaWiki Links και Redirects APIs, που περιγράφηκαν στις παραγράφους 3.2.1 και 3.2.2 αντίστοιχα. Συγκεκριμένα, χρησιμοποιούμε την πληροφορία που παρέχεται από τα Wikipedia disambiguation pages και από τα Wikipedia redirects, ώστε να αποφασίσουμε αν μπορούμε να αναφερθούμε σε μια οντότητα e χρησιμοποιώντας την αναφορά m. Η χρήση των APIs αυτών οδηγεί σε καλύτερα σύνολα υποψήφιων οντοτήτων, καθώς τα Wikipedia disambiguation pages είναι αρκετά πιο συντηρητικά όσον αφορά τις προτάσεις αποσαφήνισης που παρέχουν, ενώ τα Wikipedia redirects αποτελούν μια καλή πηγή aliases για τις οντότητες. Αυτή η διαδικασία φαίνεται στον Αλγόριθμο 9. Συνδυάζοντας τις επιμέρους μεθόδους που αναλύθηκαν παραπάνω, μπορούμε να περιγράψουμε τη συνολική διαδικασία παραγωγής του συνόλου υποψήφιων οντοτήτων E_m για μια αναφορά m στον Αλγό-

Αλγόριθμος 7: Παραγωγή βάσης συνόλου υποψήφιων οντοτήτων με χρήση του GKG API, όπου οι βασικές παράμετροι αναζήτησης είναι η συμβολοσειρά της αναφοράς m και το όριο του πλήθους οντοτήτων που θα επιστραφούν $limit$.

function QUERYGKGAPI($m, limit$)
    query GKG API for mention m with at most $limit$ results
    parse JSON response
    return candidate entities
end function

Αλγόριθμος 8: Δομικό φιλτράρισμα των αποτελεσμάτων του GKG API, όπου απαιτείται να υπάρχουν σε κάθε οντότητα που επιστρέφεται τα βασικά πεδία που χρειάζεται το σύστημα για να λειτουργήσει.

function GKGENTITYISSTRUCTURALLYSOUND($entity$)
    if @id ∉ entity **then**
        return `False`
    end if
    if name ∉ entity **then**
        return `False`
    end if
    if @type ∉ entity **then**
        return `False`
    end if
    if detailedDescription ∉ entity **then**
        return `False`
    end if
    if articleBody ∉ entity **then**
        return `False`
    end if
    if url ∉ entity **then** ▶ corresponding Wikipedia article url
        return `False`
    end if
    if resultScore ∉ entity **then**
        return `False`
    end if
    return `True`
end function

Αλγόριθμος 9: Σημασιολογικό φιλτράρισμα των αποτελεσμάτων του GKG API, όπου για να δεχτούμε ότι η $entity$ μπορεί να αποτελεί υποψήφια οντότητα για μια αναφορά m, θα πρέπει αυτό να υποδεικνύεται είτε από τα Wikipedia disambiguation pages είτε από τα Wikipedia redirects.

```
1:  function entityCouldBeReferredToAsm(entity, m)
2:                                                  ▶ using MediaWiki Links API
3:      if entity can be found in Wikipedia's disambiguation page for m then
4:          return True
5:      end if
6:                                                  ▶ using MediaWiki Redirects API
7:      if m matches with some Wikipedia legal name for entity then
8:          return True
9:      end if
10:     return False
11: end function
```

ριθμο 10. Αυτή η μέθοδος φιλτραρίσματος σαφώς δεν είναι αλάνθαστη, καθώς σε κάποιες περιπτώσεις επιτρέπει να περάσουν στο τελικό σύνολο υποψήφιων οντοτήτων οντότητες που είναι ακατάλληλες, ενώ σε άλλες απορρίπτει οντότητες που δε θα έπρεπε. Ωστόσο, σε γενικές γραμμές, αυτό το αρχικό στάδιο φιλτραρίσματος λειτουργεί ικανοποιητικά, και οδηγεί σε βελτίωση του συστήματος αποσαφήνισης από πλευράς ακρίβειας και από πλευράς επίδοσης.

4.4 Κριτήρια ταξινόμησης οντοτήτων

Σε αυτήν την ενότητα αναλύονται τα βασικά κριτήρια ταξινόμησης οντοτήτων μέσα σε ένα σύνολο υποψήφιων οντοτήτων E_m. Αυτή η ταξινόμηση είναι αναγκαία, ώστε να αποφασίσουμε ποια οντότητα του συνόλου E_m θα πρέπει να αντιστοιχιστεί στην αναφορά m. Η επιλογή των κριτηρίων ταξινόμησης είναι καθοριστικής σημασίας για το αποτέλεσμα της αποσαφήνισης.

4.4.1 GKG resultScore

Όταν κάνουμε μια αναζήτηση στο GKG API, ψάχνοντας υποψήφιες οντότητες για μια αναφορά m, τότε οι οντότητες που επιστρέφονται συνοδεύονται από το πεδίο resultScore. Αυτό μας πληροφορεί για το πόσο καλά ταιριάζει η εκάστοτε οντότητα στις παραμέτρους αναζήτησης. Στο πλαίσιο αυτής της εργασίας, το resultScore χρησιμοποιείται ως έκφραση της εκ των προτέρων δημοτικότητας (popularity prior). Συγκεκριμένα, για κάθε υποψήφια οντότητα του συνόλου E_m το resultScore διαιρείται με το μέγιστο resultScore στο σύνολο E_m, και έτσι λαμβάνουμε μια κανονικοποιημένη κλίμακα όπου η επικρατέστερη οντότητα χαρακτηρίζεται από μοναδιαίο resultScore, και οι υπόλοιπες οντότητες χαρακτηρίζονται από $0 <$ resultScore < 1. Αυτή η διαδικασία φαίνεται στον Αλγόριθμο 11.

Αλγόριθμος 10: Συνδυάζοντας τις προηγούμενες μεθόδους, λαμβάνουμε ένα ικανοποιητικό σύνολο υποψήφιων οντοτήτων, φιλτράροντας πρώτα με βάση τη δομή των οντοτήτων, και έπειτα με βάση τη σημασιολογία που υποδεικνύεται από τα Wikipedia APIs. Παρατηρούμε ότι αν το σημασιολογικό φιλτράρισμα δεν καταφέρει να απομονώσει κάποιες οντότητες από το σύνολο E_b, και τις απορρίψει όλες, τότε επιστρέφεται το ίδιο το σύνολο E_b.

```
1:  function CANDIDATESETGENERATION(m, limit)
2:      E_a ← QUERYGKGAPI(m, limit)
3:      E_b ← ∅                                          ▶ structural filtering
4:      for each entity ∈ E_a do
5:          if GKGENTITYISSTRUCTURALLYSOUND(entity) then
6:              E_b ← E_b ∪ {entity}
7:          end if
8:      end for
9:      E_c ← ∅                                          ▶ semantic filtering
10:     for each entity ∈ E_b do
11:         if ENTITYCOULDBEREFERREDTOASM(entity, m) then
12:             E_c ← E_c ∪ {entity}
13:         end if
14:     end for
15:     if E_c = ∅ then
16:         return E_b
17:     else
18:         return E_c
19:     end if
20: end function
```

Αλγόριθμος 11: Κανονικοποίηση resultScores ενός συνόλου υποψήφιων οντοτήτων E_m.

```
1:  procedure NORMALIZEGKGRESULTSCORES(E_m)
2:      maxResultScore ← 0
3:      for all e ∈ E_m do
4:          if maxResultScore < e.resultScore then
5:              maxResultScore ← e.resultScore
6:          end if
7:      end for
8:      for all e ∈ E_m do
9:          e.resultScore ← e.resultScore/maxResultScore
10:     end for
11: end procedure
```

Αλγόριθμος 12: Υπολογισμός document similarity μεταξύ του context T και του κειμένου που χαρακτηρίζει μια οντότητα T_e, βάσει των κοινών όρων που περιέχουν.

```
1: function BinaryDocumentSimilarity(T, T_e)
2:     T̃ ←stemmed tokens of T\stopwords
3:     T̃_e ←stemmed tokens of T_e\stopwords
4:     if T̃ = ∅ then
5:         return 0
6:     end if
7:     T̃_common ← T̃ ∩ T̃_e                    ▶ Using fuzzy string matching
8:     return log(1 + |T̃_common|)/ log(1 + |T̃|)
9: end function
```

4.4.2 Binary document similarity

Το GKG API επιστρέφει για κάθε οντότητα και το πρώτο τμήμα του αντίστοιχου Wikipedia άρθρου για την οντότητα αυτήν, μέσω του πεδίου articleBody. Αυτό συχνά περιέχει χαρακτηριστικές λέξεις που μπορούν να διευκολύνουν την αποσαφήνιση. Για τον λόγο αυτόν, θεωρώντας όλο το κείμενο T ως context, αναζητούμε τις λέξεις του context στις λέξεις του πρώτου τμήματος του άρθρου της Wikipedia. Η ύπαρξη κοινών λέξεων, όπως actor ή president, έχει τη δυνατότητα να καθοδηγήσει την αποσαφήνιση. Μας ενδιαφέρει μόνο αν εμφανίζεται κάποιος όρος ή όχι, και όχι το πλήθος των εμφανίσεων, και για αυτό χρησιμοποιούμε binary similarity. Αν συμβολίσουμε το κείμενο που περιγράφει την οντότητα e ως T_e, τότε ακολουθούνται τα εξής βήματα:

1. Tokenizing, stop-word removal και stemming των T και T_e.
2. Αναζήτηση των λέξεων του T στο T_e, χρησιμοποιώντας fuzzy string matching.
3. Επιστροφή ως μέτρου ομοιότητας του $\log(1 + |T \cap T_e|)/\log(1 + |T|)$. Επιλέγεται λογαριθμική κλίμακα, έτσι ώστε να μην απαιτείται πλήρης επικάλυψη των δύο κειμένων, ώστε να επιτευχθεί υψηλό document similarity score. Επίσης, οι τιμές που προκύπτουν είναι εξορισμού κανονικοποιημένες.

Η παραπάνω μέθοδος περιγράφεται και στον Αλγόριθμο 12.

4.4.3 Entity relatedness

Ξεκινώντας από τη βασική υπόθεση ότι οι οντότητες που εμφανίζονται σε ένα κείμενο χαρακτηρίζονται από σημασιολογική εγγύτητα, χρησιμοποιείται ο τύπος του Wikipedia Link-based Measure (WLM), που αναλύεται στην υποπαράγραφο 2.3.4.2. Για να υπολογιστεί αυτή η μετρική σημασιολογικής εγγύτητας μεταξύ δύο οντοτήτων e_1 και e_2 θα πρέπει να γνωρίζουμε τους εισερχόμενους συνδέσμους για τα άρθρα των οντοτήτων αυτών στην Wikipedia. Αυτή η πληροφορία παρέχεται από το MediaWiki Linkshere API, που περιγράφηκε στην παράγραφο 3.2.3. Γενικά, ο τύπος WLM δίνει υψηλές τιμές για οντότητες που εμφανίζουν σημασιολογική συγγένεια, και χαμηλές τιμές για οντότητες που είναι σημασιολογικά ασυσχέτιστες. Έτσι, μας επιτρέπει να προσδιορίσουμε τις οντότητες που ανήκουν σε διαφορετικά

Αλγόριθμος 13: Υπολογισμός σημασιολογικής εγγύτητας μεταξύ δύο οντοτήτων e_1 και e_2 με χρήση του WLM.

function WLM(e_1, e_2)
    $|WP| \leftarrow$ number of articles in Wikipedia
    $IN_1 \leftarrow$ inlinks for e_1's Wikipedia article
    $IN_2 \leftarrow$ inlinks for e_2's Wikipedia article
    if $IN_1 = \emptyset$ **or** $IN_2 = \emptyset$ **then**
        return 0
    end if
    $IN_{\text{overlap}} \leftarrow IN_1 \cap IN_2$
    if $IN_{\text{overlap}} = \emptyset$ **then**
        return 0
    end if
    return $1 - \frac{\log(\max(|IN_1|,|IN_2|)) - \log(|IN_{\text{overlap}}|)}{\log(|WP|) - \log(\min(|IN_1|,|IN_2|))}$
end function

σύνολα υποψήφιων οντοτήτων και εμφανίζουν σημασιολογική συνάφεια. Σημειώνεται ότι ο τύπος WLM δίνει κανονικοποιημένες τιμές. Το WLM υπολογίζεται όπως στον Αλγόριθμο 13.

4.5 Κατασκευή γράφου

Για να αποσαφηνίσουμε τις αναφορές $M = \{m_1, m_2, \dots, m_k\}$ σε ένα κείμενο T, πρώτα υπολογίζουμε τα σύνολα υποψήφιων οντοτήτων $E_1, E_2, \dots, E_k$, όπως περιγράφηκε στην ενότητα 4.3. Έπειτα, για κάθε σύνολο $E_i = \{e_{ij} : 1 \leq j \leq |E_i|\}$, $1 \leq i \leq k$, κατασκευάζουμε ένα σύνολο κόμβων $\{e_{ij}\}$, όπου κάθε κόμβος e_{ij} αντιστοιχεί στην j-οστή υποψήφια οντότητα για την αναφορά i. Στη συνέχεια, κατασκευάζουμε τις ακμές του γράφου. Κάθε κόμβος e_{ij} συνδέεται με κάθε κόμβο e_{uv}, όπου $i \neq u$, με μη-κατευθυνόμενη ακμή βάρους:

$$weight(e_{ij}, e_{uv}) = a \cdot \left(\frac{c \cdot \text{rs}(e_{ij}) + d \cdot \text{sim}(T, T_{e_{ij}}) + c \cdot \text{rs}(e_{uv}) + d \cdot \text{sim}(T, T_{e_{uv}})}{2} \right) + b \cdot \text{WLM}(e_{ij}, e_{uv})$$

όπου $rs \equiv$ normalized GKG resultScore

$sim \equiv$ binary document similarity

$0 \leq a, b, c, d \leq 1$

$a + b = c + d = 1$

Ο γράφος G που προκύπτει από την παραπάνω διαδικασία είναι ένας πλήρης k-partite γράφος, δηλαδή ένας γράφος του οποίου οι κόμβοι χωρίζονται σε k ανεξάρτητα σύνολα (k independent sets), ενώ μεταξύ των διαφορετικών συνόλων υπάρχουν όλες οι δυνατές ακμές. Αυτό συμβαίνει διότι δεν υπάρχει λόγος να συνδέσουμε τους κόμβους που ανήκουν στο ίδιο σύνολο E_i, καθώς οι κόμβοι αυτοί είναι αμοιβαία αποκλειόμενοι, όσον αφορά το τελικό αποτέλεσμα της αποσαφήνισης. Το νόημα που έχουν οι παράμετροι a, b, c και d είναι το εξής:

- Η παράμετρος a είναι το βάρος που δίνουμε στο κατά πόσο ταιριάζει η συμβολοσειρά της αναφοράς (που μετράται από το resultScore) και η συμβολοσειρά του κειμένου (που μετράται από το document similarity) με τα αντίστοιχα χαρακτηριστικά της υποψήφιας οντότητας.

- Η παράμετρος c είναι το βάρος που δίνουμε στο resultScore.

- Η παράμετρος d είναι το βάρος που δίνουμε στο document similarity.

- Η παράμετρος b είναι το βάρος που δίνουμε στο WLM.

Παρατηρούμε ότι πριν πολλαπλασιάσουμε με την παράμετρο a, εξάγουμε τον μέσο όρο των resultScore και document similarity με τα αντίστοιχα βάρη.

4.6 Επίλυση γράφου

Στην ενότητα 4.5 περιγράφηκε η κατασκευή του γράφου G, που έχει ως κόμβους τις υποψήφιες οντότητες, οι οποίες συνδέονται με ακμές κατάλληλου βάρους. Θεωρούμε ότι το σωστό αποτέλεσμα αποσαφήνισης προκύπτει από τον υπογράφο μεγίστου βάρους $G^* \subset G$, όπου ο G^* έχει έναν μόνο κόμβο σε κάθε σύνολο κόμβων E_i. Με άλλα λόγια, θέλουμε να διαλέξουμε από κάθε σύνολο υποψήφιων οντοτήτων E_i έναν κόμβο e_{ij}, έτσι ώστε ο προκύπτων (πλήρης) γράφος να έχει το μέγιστο δυνατό συνολικό βάρος στις ακμές του. Αυτό είναι το πρόβλημα εύρεσης k-clique μεγίστου βάρους σε πλήρη k-partite γράφο, και είναι NP-δύσκολο. Συνεπώς, για να λύσουμε τον γράφο θα πρέπει να κατασκευάσουμε μια προσεγγιστική μέθοδο.

Η προσεγγιστική μέθοδος που προτείνεται στην παρούσα εργασία λαμβάνει υπόψιν το πεδίο εφαρμογής, και το τι αναπαριστά ο γράφος G με τους κόμβους και τις ακμές του. Από τον γράφο G προσπαθούμε να εξάγουμε ένα αποτέλεσμα αποσαφήνισης που συμφωνεί με ένα σημασιολογικά συναφές κείμενο. Έτσι, προσπαθούμε να καταλήξουμε στη λύση του γράφου αφαιρώντας διαδοχικά κόμβους που δε φαίνεται να ταιριάζουν στο κείμενο, με βάση τα κριτήρια ταξινόμησης που αναλύθηκαν παραπάνω. Ο κόμβος που αφαιρείται σε κάθε βήμα επιλέγεται βάσει μιας ευριστικής τεχνικής, που διαλέγει από τους κόμβους του γράφου G αυτόν που φαίνεται να είναι λιγότερο κατάλληλος για το τελικό αποτέλεσμα αποσαφήνισης (worst out heuristic). Συγκεκριμένα, για κάθε κόμβο $e_{ij} \in G$ υπολογίζεται το αποτέλεσμα αποσαφήνισης στο οποίο ο κόμβος e_{ij} έχει τη μέγιστη συνεισφορά βάρους με τις ακμές που προσπίπτουν σε αυτόν. Ο γράφος μέγιστης συνεισφοράς $G_{e_{ij}}$ μπορεί να βρεθεί απλά επιλέγοντας για κάθε σύνολο κόμβων E_u, με $u \neq i$, την ακμή μεγίστου βάρους που συνδέει τον e_{ij} με κάποιον κόμβο στο E_u. Η επιλογή αυτών των ακμών οδηγεί άμεσα σε μια υποψήφια συνολική αποσαφήνιση, στην οποία το συνολικό βάρος του γράφου, καθώς και η συνεισφορά του e_{ij}, μπορούν να υπολογισθούν εύκολα. Το κριτήριο που χρησιμοποιείται για να επιλεγεί ο κόμβος που θα αφαιρεθεί είναι αυτό του βάρους του γράφου μέγιστης συνεισφοράς. Σε περίπτωση ισοπαλίας, χρησιμοποιείται η μέγιστη συνεισφορά, και όχι το συνολικό βάρος του γράφου, ώστε να γίνει η επιλογή. Η παραπάνω μέθοδος αποτελείται από τα εξής βήματα:

1. Για κάθε κόμβο $e_{ij} \in G$, υπολογίζεται ο γράφος μέγιστης συνεισφοράς $G_{e_{ij}}$. Όλοι οι κόμβοι ταξινομούνται κατά αύξουσα σειρά βάρους του γράφου μέγιστης συνεισφοράς

$G_{e_{ij}}$, και δευτερευόντως κατά αύξουσα σειρά συνεισφοράς του κόμβου e_{ij} στον γράφο $G_{e_{ij}}$.

2. Από την ταξινόμηση του βήματος 1, επιλέγεται το μικρότερο στοιχείο, έστω e_{ij}. Αν $|E_i| > 1$, δηλαδή αν ο e_{ij} δεν είναι ο τελευταίος υποψήφιος κόμβος στο σύνολο E_i, τότε αφαιρείται. Διαφορετικά, προχωράμε στο επόμενο στοιχείο, και συνεχίζουμε μέχρι να αφαιρεθεί κάποιος κόμβος.

3. Επαναλαμβάνονται τα βήματα 1 και 2 μέχρι κάθε σύνολο E_i να έχει μόνο ένα στοιχείο. Αυτό σημαίνει ότι έχουμε φτάσει στη λύση του γράφου, και στην προτεινόμενη αποσαφήνιση.

Αξίζει να σημειωθεί ότι, μετά από κάθε αφαίρεση κόμβου, είναι αναγκαίος ο εκ νέου υπολογισμός των γράφων μέγιστης συνεισφοράς, καθώς η αφαίρεση ενός κόμβου περιορίζει τις επιλογές των άλλων κόμβων όσον αφορά την εύρεση ακμής μεγίστου βάρους για σύνδεση με το αντίστοιχο σύνολο κόμβων. Για να γίνει περισσότερο κατανοητή η παραπάνω διαδικασία, στην Εικόνα 3 δίνεται ένα απλό παράδειγμα κατασκευής του γράφου μέγιστης συνεισφοράς για έναν κόμβο του γράφου G. Στην Εικόνα 4 παρουσιάζεται ο αναγκαίος επανυπολογισμός του γράφου μέγιστης συνεισφοράς, μετά από την αφαίρεση ενός κόμβου.

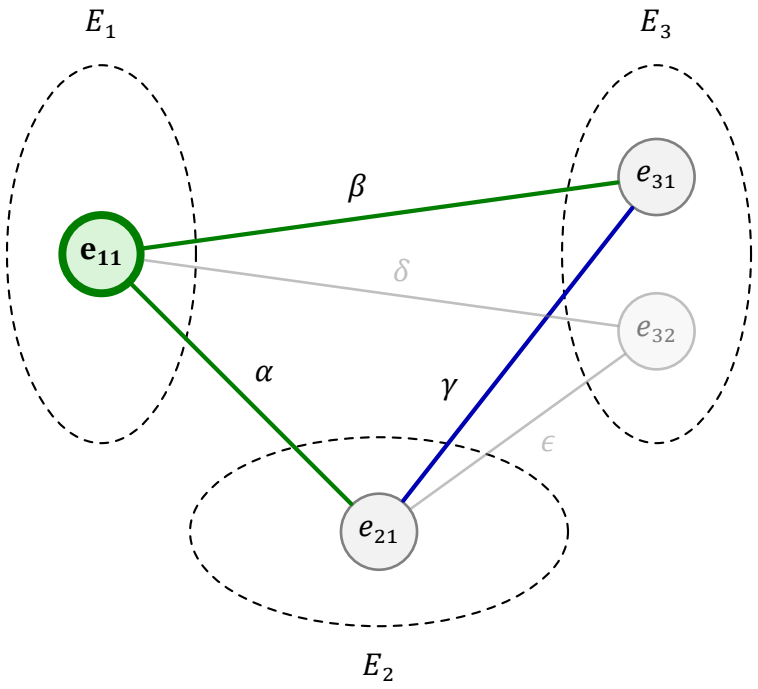
Προτού επιχειρήσουμε να υπολογίσουμε την πολυπλοκότητα χρόνου και μνήμης της παραπάνω μεθόδου, αναλύουμε περισσότερο τον γράφο G. Όπως αναφέρθηκε παραπάνω, σκοπός είναι η βέλτιστη επιλογή ενός κόμβου από κάθε σύνολο $E_i, 1 \leq i \leq k$. Αυτό σημαίνει ότι οι δυνατές επιλογές είναι:

$$\prod_{i=1}^{k} |E_i|$$

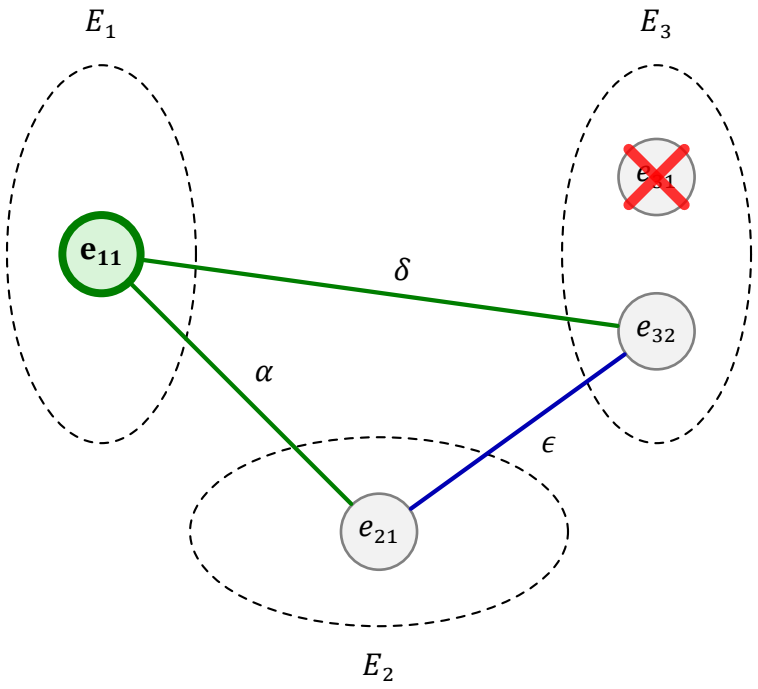
Αν, για παράδειγμα, είχαμε ένα κείμενο με 10 αναφορές, και κάθε αναφορά είχε 50 υποψήφιες οντότητες, τότε τα δυνατά αποτελέσματα αποσαφήνισης θα ήταν $50^{10} \approx 9.77 \times 10^{16}$. Αυτό μας δίνει μια αίσθηση γιατί το πρόβλημα είναι αναγκαίο να λυθεί με προσεγγιστικές μεθόδους. Στρέφοντας την προσοχή μας στις ακμές του G, υποθέτουμε ότι $|E_1| = |E_2| = \ldots = |E_k| = n$. Τότε, δεδομένου ότι κάθε κόμβος του G συνδέεται με κάθε άλλο κόμβο σε διαφορετικό σύνολο, το πλήθος των ακμών του G είναι:

$$\begin{aligned}
\text{number of edges in } G &= \sum_{i=1}^{k-1} n \sum_{u=i+1}^{k} n = n \sum_{i=1}^{k-1} n(k-i) \\
&= n^2 \sum_{i=1}^{k-1} (k-i) = n^2 \left(k(k-1) - \sum_{i=1}^{k-1} i \right) \\
&= n^2 \left(k(k-1) - \frac{k(k-1)}{2} \right) \\
&= n^2 \frac{k(k-1)}{2}
\end{aligned}$$

Εικόνα 3: Παρουσιάζεται η κατασκευή του γράφου μέγιστης συνεισφοράς για τον κόμβο e_{11} του συνόλου E_1. Ο κόμβος e_{11} επιλέγει άπληστα τους κόμβους e_{21} και e_{31} για τον γράφο $G_{e_{11}}$. Αυτό σημαίνει ότι $\beta \geq \delta$. Σημειώνεται ότι τα σύνολα E_1, E_2 και E_3 θα μπορούσαν να έχουν περισσότερους κόμβους από όσους φαίνονται εδώ, απλά σημειώνονται μόνο οι κόμβοι που ενδιαφέρουν. Οι πράσινες ακμές είναι αυτές που επιλέγονται άπληστα από τον κόμβο e_{11} επειδή είναι οι ακμές μεγίστου βάρους προς τα αντίστοιχα σύνολα E_2 και E_3. Η μπλε ακμή είναι αυτή που συμπληρώνει τον γράφο μέγιστης συνεισφοράς $G_{e_{11}}$. Είναι $weight(G_{e_{11}}) = \alpha + \beta + \gamma$, και $contribution(G_{e_{11}}, e_{11}) = \alpha + \beta$. Αυτά τα δύο κριτήρια, με αυτήν τη σειρά, είναι αυτά που χρησιμοποιούνται ώστε να αποφασιστεί ποιος κόμβος πρέπει να αποχωρήσει από τον γράφο G.



Εικόνα 4: Συνεχίζοντας το παράδειγμα της Εικόνας 3, παρουσιάζεται η ενημέρωση του γράφου μέγιστης συνεισφοράς για τον κόμβο e_{11} του συνόλου E_1, μετά την αφαίρεση του κόμβου e_{31}. Ο κόμβος e_{11} επιλέγει τώρα τον κόμβο e_{32} για τον γράφο $G_{e_{11}}$. Αυτό σημαίνει ότι ο κόμβος αυτός παρέχει την αμέσως επόμενη ακμή μεγίστου βάρους. Οι πράσινες ακμές είναι αυτές που επιλέγονται άπληστα από τον κόμβο e_{11} επειδή είναι οι ακμές μεγίστου βάρους προς τα αντίστοιχα σύνολα E_2 και E_3. Η μπλε ακμή είναι αυτή που συμπληρώνει τον γράφο μέγιστης συνεισφοράς $G_{e_{11}}$. Πλέον, είναι $weight(G_{e_{11}}) = \alpha + \delta + \epsilon$, και $contribution(G_{e_{11}}, e_{11}) = \alpha + \delta$. Παρατηρούμε ότι και τα δύο κριτήρια ταξινόμησης έχουν λάβει διαφορετικές τιμές, πράγμα που φανερώνει την αναγκαιότητα της ενημέρωσής τους.



Επιστρέφοντας στα ζητήματα πολυπλοκότητας, θεωρούμε ότι $|E_i| = \mathcal{O}(n)$ για κάθε i, και συμβολίζουμε ως $N = \mathcal{O}(nk)$ το πλήθος των κόμβων του γράφου G. Είναι φανερό ότι στην πολυπλοκότητα μνήμης κυριαρχεί η ανάγκη αποθήκευσης των ακμών του γράφου. Άρα η πολυπλοκότητα μνήμης θα είναι $\mathcal{O}(N^2)$. Όσον αφορά τη χρονική πολυπλοκότητα, το πρώτο από τα βήματα που περιγράφηκαν παραπάνω απαιτεί για καθέναν από τους N κόμβους του γράφου να ελέγξουμε περίπου N κόμβους, ώστε να αποφανθούμε ποιος είναι ο γράφος μέγιστης συνεισφοράς. Επίσης, θα πρέπει να κάνουμε $\mathcal{O}(k^2)$ βήματα ώστε να υπολογίσουμε το βάρος του γράφου μέγιστης συνεισφοράς, που έχει k κόμβους. Έπειτα, θα πρέπει τα ταξινομήσουμε όλους τους κόμβους του G βάσει του βάρους του γράφου μέγιστης συνεισφοράς, καθώς και της συνεισφοράς του κάθε κόμβου στον αντίστοιχο γράφο μέγιστης συνεισφοράς, κάτι που απαιτεί $\mathcal{O}(N \log N)$ βήματα με χρήση κάποιου συγκριτικού αλγορίθμου ταξινόμησης. Συνεπώς, το βήμα 1 απαιτεί $\mathcal{O}(N(N+k^2)+N \log N) = \mathcal{O}(N^2+Nk^2+N \log N)$ βήματα. Το βήμα 2 απαιτεί $\mathcal{O}(k)$ βήματα, καθώς αν μετά από k στοιχεία δεν μπορούμε να βρούμε κόμβο προς αφαίρεση, αυτό σημαίνει ότι η αποσαφήνιση έχει ήδη επιτευχθεί. Τα βήματα 1 και 2 επαναλαμβάνονται $\mathcal{O}(N)$ φορές, καθώς έχουμε να αφαιρέσουμε $N - k = \mathcal{O}(N)$ κόμβους. Άρα, η συνολική χρονική πολυπλοκότητα του παραπάνω αλγορίθμου είναι $\mathcal{O}(N(N^2 + Nk^2 + N \log N + k)) = \mathcal{O}(N^3 + N^2k^2 + N^2 \log N + Nk) = \mathcal{O}(N^3 + N^2k^2) = \mathcal{O}(n^3k^3 + n^2k^4)$, διότι $N = \mathcal{O}(nk)$.

Για να επιτευχθεί ασυμπτωτικά καλύτερη χρονική πολυπλοκότητα, διατηρώντας την πολυπλοκότητα μνήμης στα ίδια επίπεδα, ορίζονται οι παρακάτω δομές:

- $node\_pick\_order$, που είναι ένα dictionary το οποίο για κάθε $e_{ij} \in G$ αποθηκεύει για κάθε E_u με $u \neq i$ τους κόμβους $e_{uv} \in E_u$ κατά φθίνουσα σειρά βάρους ακμής (e_{ij}, e_{uv}). Στο παράδειγμα της Εικόνας 3, θεωρώντας ότι υπάρχουν μόνο οι κόμβοι που εμφανίζονται, είναι $node\_pick\_order[e_{11}][E_3] = [(e_{31}, \beta), (e_{32}, \delta)]$. Αυτή η δομή απαιτεί χώρο $\mathcal{O}(N^2)$, καθώς για κάθε κόμβο αποθηκεύουμε μια ταξινόμηση πάνω σε κάθε άλλο σύνολο κόμβων. Η δομή $node\_pick\_order$ μας επιτρέπει να βρίσκουμε γρήγορα τις άπληστες επιλογές που κάνει ο κάθε κόμβος του γράφου, χωρίς να διασχίζουμε τον γράφο κάθε φορά, ακόμα και μετά από αφαίρεση κόμβου.

- $node\_graph\_contribution$, που είναι ένα dictionary το οποίο για κάθε $e_{ij} \in G$ αποθηκεύει για κάθε E_u τους κόμβους $e_{uv} \in E_u$ που έχουν επιλεγεί για να συμμετέχουν στον γράφο $G_{e_{ij}}$, καθώς και το βάρος προσπιπτόντων ακμών του κάθε e_{uv} εντός του $G_{e_{ij}}$. Δηλαδή, για κάθε κόμβο e που ανήκει στον γράφο μέγιστης συνεισφοράς του e_{ij}, αποθηκεύεται η συνεισφορά του e στο $weight(G_{e_{ij}})$. Στο παράδειγμα της Εικόνας 3 είναι $node\_graph\_contribution[e_{11}][E_2] = (e_{21}, \alpha + \gamma)$. Αυτή η δομή απαιτεί χώρο $\mathcal{O}(Nk)$, καθώς για κάθε κόμβο αποθηκεύουμε πληροφορία για τις προσπίπτουσες ακμές καθενός από τους k κόμβους που ανήκουν στον γράφο μέγιστης συνεισφοράς του. Η δομή $node\_graph\_contribution$ επιτρέπει να υπολογίζουμε γρήγορα τις αλλαγές στα βάρη προσπιπτόντων ακμών για τους κόμβους ενός γράφου μέγιστης συνεισφοράς, καθώς και στο συνολικό βάρος του γράφου αυτού.

- $important\_for\_who$, που είναι ένα dictionary το οποίο για κάθε $e_{ij} \in G$ διατηρεί το σύνολο των κόμβων $e \in G$ που έχουν επιλέξει τον e_{ij} για τον γράφο μέγιστης συνεισφοράς τους G_e. Στο παράδειγμα της Εικόνας 3 ισχύει $e_{11} \in important\_for\_who[e_{31}]$. Αυτή η δομή απαιτεί χώρο $\mathcal{O}(N^2)$, αφού για καθέναν από τους N κόμβους μπορεί να

ισχύει ότι έχει επιλεγεί από περίπου N άλλους κόμβους για τους γράφους μέγιστης συνεισφοράς τους. Η δομή $important\_for\_who$ επιτρέπει τον γρήγορο υπολογισμό των κόμβων που θα πρέπει να υπολογίσουν ξανά τους γράφους μέγιστης συνεισφοράς τους, καθώς ένας από τους κόμβους που είχαν επιλέξει έχει αφαιρεθεί.

Μετά τον ορισμό των παραπάνω δομών, η νέα μέθοδος επίλυσης γράφου έχει ως εξής:

1. Για κάθε κόμβο $e_{ij} \in G$ κατασκευάζεται η αντίστοιχη καταχώρηση στο dictionary $node\_pick\_order$. Για να γίνει αυτό, διατρέχουμε όλους τους άλλους N το πλήθος κόμβους και ταξινομούμε τους κόμβους σε κάθε σύνολο E_u με $u \neq i$ κατά φθίνουσα σειρά βάρους ακμής που τους συνδέει με τον e_{ij}. Παράλληλα, η ταξινόμηση αυτή μας δίνει την απαραίτητη πληροφορία ώστε να ενημερώσουμε τη δομή $important\_for\_who$. Τέλος, κατασκευάζεται η αντίστοιχη καταχώρηση στη δομή $node\_graph\_contribution$, χρησιμοποιώντας τα δεδομένα που είναι διαθέσιμα από τη δομή $node\_pick\_order$. Αυτή η διαδικασία παίρνει χρόνο $\mathcal{O}(N)$ για να διατρέξουμε τον γράφο, $\mathcal{O}(knl \log n)$ ώστε να ταξινομήσουμε το καθένα από k σύνολα κόμβων, $\mathcal{O}(k)$ για να ενημερώσουμε τη δομή $important\_for\_who$ για καθέναν από τους k κόμβους του γράφου μέγιστης συνεισφοράς $G_{e_{ij}}$ και $\mathcal{O}(k^2)$ ώστε να ενημερωθεί η δομή $node\_graph\_contribution$ για τη συνεισφορά κάθε κόμβου στον γράφο $G_{e_{ij}}$. Συνολικά, η διαδικασία παίρνει χρόνο $\mathcal{O}(N(N + kn \log n + k + k^2)) = \mathcal{O}(N^2 + N^2 \log n + Nk^2) = \mathcal{O}(N^2 \log n + Nk^2)$, καθώς $N = \mathcal{O}(nk)$.

2. Οι N κόμβοι ταξινομούνται κατά αύξουσα σειρά βάρους γράφου μέγιστης συνεισφοράς (πρωτεύον κριτήριο) και συνεισφοράς στον εν λόγω γράφο (δευτερεύον κριτήριο)[3]. Αυτό, για κάθε κόμβο e_{ij}, απαιτεί χρόνο $\mathcal{O}(k)$ για να επισκεφθούμε όλους τους κόμβους του γράφου $G_{e_{ij}}$ μέσω των δομών $node\_pick\_order$ και $node\_graph\_contribution$. Άρα, συνολικά χρειαζόμαστε χρόνο $\mathcal{O}(Nk + N \log N)$.

3. Μέχρι να επιτευχθεί αποσαφήνιση, δηλαδή $\mathcal{O}(N)$ φορές, εκτελούνται τα εξής βήματα:

   (α′) Επιλέγεται ο κόμβος e_{evict} που θα αφαιρεθεί από τον γράφο, χρησιμοποιώντας την ταξινόμηση των κόμβων. Αυτό θα πάρει το πολύ $\mathcal{O}(k)$ βήματα, καθώς αν μετά από k κόμβους δεν μπορούμε να βρούμε έναν κόμβο που να μπορεί αφαιρεθεί (διότι όλοι είναι μοναδικοί στα σύνολά τους), τότε η αποσαφήνιση έχει ήδη επιτευχθεί.

   (β′) Ενημερώνεται σε χρόνο $\mathcal{O}(k)$ η δομή $important\_for\_who$, καθώς οι κόμβοι που είχε επιλέξει ο κόμβος e_{evict} για τον γράφο μέγιστης συνεισφοράς του θα πρέπει να ενημερωθούν ότι ο κόμβος δεν υπάρχει πια.

   (γ′) Ενημερώνονται οι κόμβοι που επηρεάζονται από την αποχώρηση του e_{evict}. Για κάθε κόμβο e_{affected} στο σύνολο $important\_for\_who[e_{\text{evict}}]$, δηλαδή για $\mathcal{O}(N)$ κόμβους, εκτελούνται τα βήματα:

      i. Ενημερώνεται η καταχώρηση $node\_pick\_order[e_{\text{affected}}]$, βρίσκοντας για το σύνολο του e_{evict} την αμέσως επόμενη ακμή μεγίστου βάρους. Αυτό γίνεται σε $\mathcal{O}(n)$ βήματα, καθώς στη χειρότερη περίπτωση όλες οι επόμενες επιλογές

[3] Αυτό μπορεί να επιτευχθεί χρησιμοποιώντας έναν stable αλγόριθμο ταξινόμησης πρώτα στο δευτερεύον κριτήριο, και έπειτα στο πρωτεύον. Η mergesort είναι κατάλληλη σε αυτήν την περίπτωση.

του κόμβου e_{affected} έχουν ήδη αφαιρεθεί από τον G, πλην της τελευταίας, και άρα θα πρέπει να διατρέξουμε ολόκληρη την ταξινομημένη λίστα μήκους περίπου n. Επίσης, ο νέος κόμβος που επιλέγεται τίθεται ως σημαντικός για την οντότητα e_{affected} στη δομή $important\_for\_who$, σε χρόνο $\mathcal{O}(1)$.

ii. Ενημερώνεται η καταχώρηση $node\_graph\_contribution[e_{\text{affected}}]$, υπολογίζοντας το βάρος των προσπιπτόντων ακμών του νέου κόμβου στον νέο γράφο $G_{e_{\text{affected}}}$. Αυτό γίνεται σε $\mathcal{O}(k)$ βήματα, καθώς πρέπει να ενημερώσουμε όλους τους κόμβους στον γράφο $G_{e_{\text{affected}}}$. Μάλιστα, δε χρειάζεται καν να διατρέξουμε ξανά τις ακμές του, καθώς το νέο βάρος του γράφου $G_{e_{\text{affected}}}$ μπορεί να υπολογισθεί εύκολα από το παλιό βάρος του γράφου, αν αφαιρέσουμε το βάρος προσπιπτόντων ακμών του κόμβου e_{evict} και προσθέσουμε το βάρος των προσπιπτόντων ακμών του νέου κόμβου. Επίσης, το δευτερεύον κριτήριο ταξινόμησης, η συνεισφορά των ακμών που προσπίπτουν στον κόμβο e_{affected}, εξάγεται άμεσα από την καταχώρηση $node\_graph\_contribution[e_{\text{affected}}]$. Έτσι, μετά τα παραπάνω, οι νέες τιμές των κριτηρίων ταξινόμησης υπολογίζονται σε χρόνο $\mathcal{O}(1)$.

(δ΄) Ταξινομούνται ξανά οι $\mathcal{O}(N)$ κόμβοι, με τις αλλαγές στα κριτήρια ταξινόμησης που προέκυψαν παραπάνω, σε χρόνο $\mathcal{O}(N \log N)$.

Το στάδιο 3 παίρνει χρόνο $\mathcal{O}(N(k + k + N(n + k) + N \log N)) = \mathcal{O}(Nk + N^2n + N^2k + N^2 \log N) = \mathcal{O}(N^2n + N^2k + N^2 \log N)$.

Αθροίζοντας τα επιμέρους στοιχεία, η χρονική πολυπλοκότητα είναι $\mathcal{O}(N^2 \log n + Nk^2 + Nk + N \log N + N^2n + N^2k + N^2 \log N) = \mathcal{O}(N^2 \log n + Nk^2 + N^2n + N^2k + N^2 \log N)$. Λαμβάνοντας υπόψιν ότι $N = \mathcal{O}(nk)$ είναι:

$$\begin{aligned}\text{χρονική πολυπλοκότητα} &= \mathcal{O}(n^2k^2 \log n + nk^3 + n^3k^2 + n^2k^3 + n^2k^2 \log(nk)) \\ &= \mathcal{O}(n^3k^2 + n^2k^3 + n^2k^2 \log(nk)) \\ &= \mathcal{O}(n^3k^2 + n^2k^3)\end{aligned}$$

Συνολικά, για έναν γράφο G που έχει k το πλήθος ανεξάρτητα σύνολα κόμβων E_i, το καθένα από τα οποία περιέχει το πολύ n κόμβους, ή, ισοδύναμα, για ένα κείμενο T που έχει k το πλήθος ονοματικές αναφορές, η καθεμία από τις οποίες αντιστοιχεί σε το πολύ n υποψήφιες οντότητες, η εύρεση προσεγγιστικής λύσης με την παραπάνω μέθοδο χαρακτηρίζεται από:

$$\begin{aligned}\text{χρονική πολυπλοκότητα} &= \mathcal{O}(n^3k^2 + n^2k^3) \\ \text{χωρική πολυπλοκότητα} &= \mathcal{O}(n^2k^2)\end{aligned}$$

Μέχρι αυτό το σημείο έχουμε υποθέσει ότι ισχύει $k \geq 2$, οπότε ο γράφος G αποτελείται από k ανεξάρτητα σύνολα κόμβων που συνδέονται μεταξύ τους με ακμές. Αυτή η προσέγγιση αφορά τη συνολική μοντελοποίηση του προβλήματος, και είναι αυτή που πρέπει να ακολουθηθεί ώστε να υπολογιστεί η παραπάνω πολυπλοκότητα. Ωστόσο, υπάρχουν και οι εξής περιπτώσεις:

- $k = 0$, οπότε το κείμενο T δεν περιέχει ονοματικές αναφορές. Σε αυτήν την περίπτωση δεν υφίσταται πρόβλημα προς επίλυση.

- $k = 1$, οπότε το κείμενο T περιέχει μόνο μία ονοματική αναφορά. Σε αυτήν την περίπτωση ο γράφος G είναι απλά ένα σύνολο κόμβων e_{1j}. Εδώ το κριτήριο WLM δεν παρέχει καμία πληροφορία, αφού χρειαζόμαστε τουλάχιστον δύο οντότητες ώστε να υπολογίσουμε σημασιολογική εγγύτητα. Έτσι, επιστρέφεται ως προτεινόμενη αποσαφήνιση ο κόμβος με τον μέγιστο συνδυασμό normalized resultScore και document similarity, όπως αυτός υπολογίζεται βάσει των παραμέτρων c και d.

Επίσης, ενδιαφέρον παρουσιάζει το ενδεχόμενο στο οποίο κάποια σύνολα υποψήφιων οντοτήτων έχουν από τη στιγμή δημιουργίας του G λιγότερες από δύο υποψήφιες οντότητες, δηλαδή:

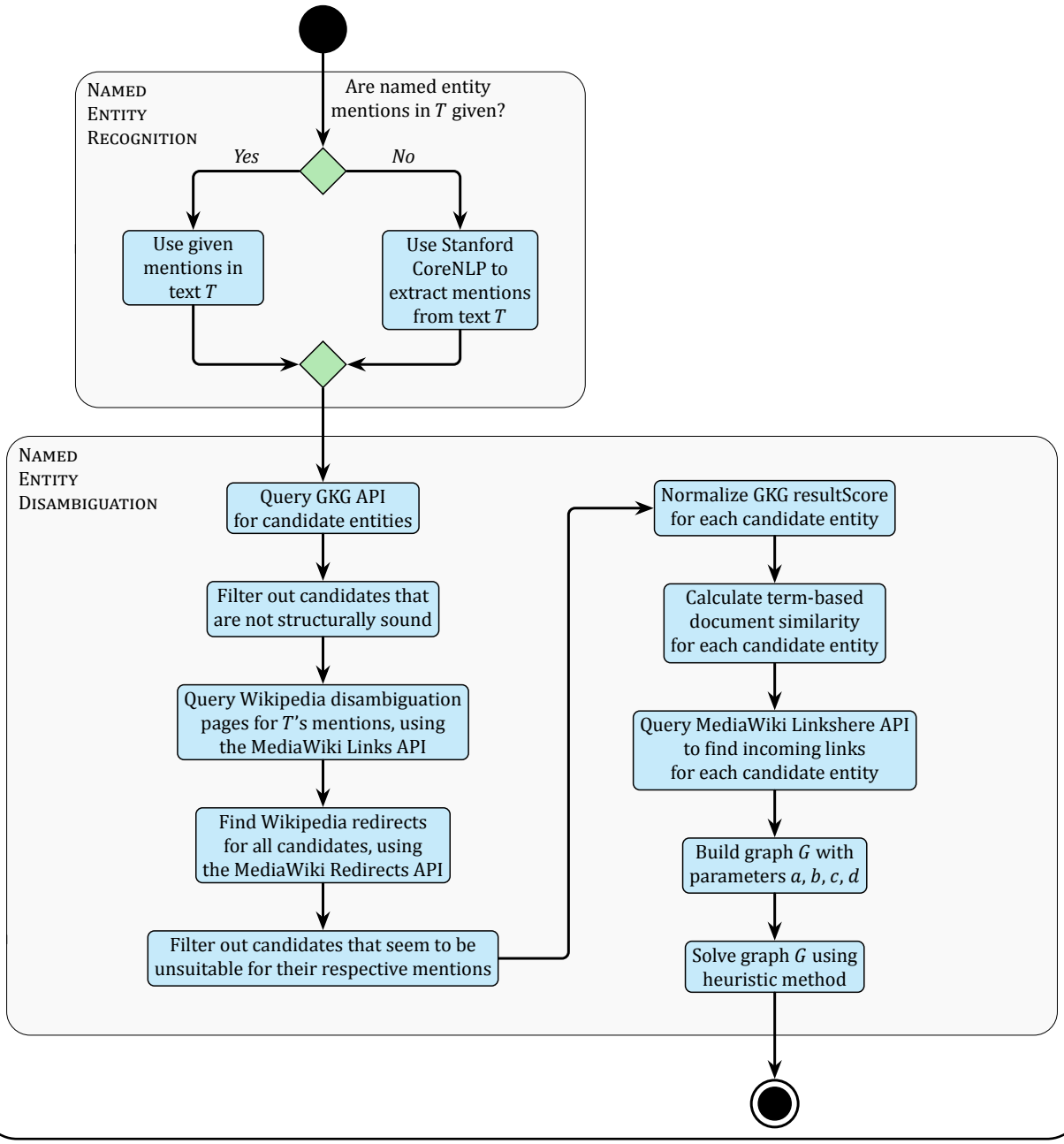
- $|E_i| = 0$, που σημαίνει ότι δεν καταφέραμε να βρούμε καμία υποψήφια οντότητα για την αναφορά m_i. Σε αυτήν την περίπτωση εκτελούμε αποσαφήνιση των υπόλοιπων αναφορών, που έχουν τουλάχιστον μία υποψήφια οντότητα.

- $|E_i| = 1$, που σημαίνει ότι η αναφορά m_i αντιστοιχεί μόνο σε μία υποψήφια οντότητα, άρα θεωρείται μη διφορούμενη. Σε αυτήν την περίπτωση γνωρίζουμε ποιο θα είναι το αποτέλεσμα αποσαφήνισης για την αναφορά m_i πριν καν εκτελεσθεί ο αλγόριθμος επίλυσης του γράφου G.

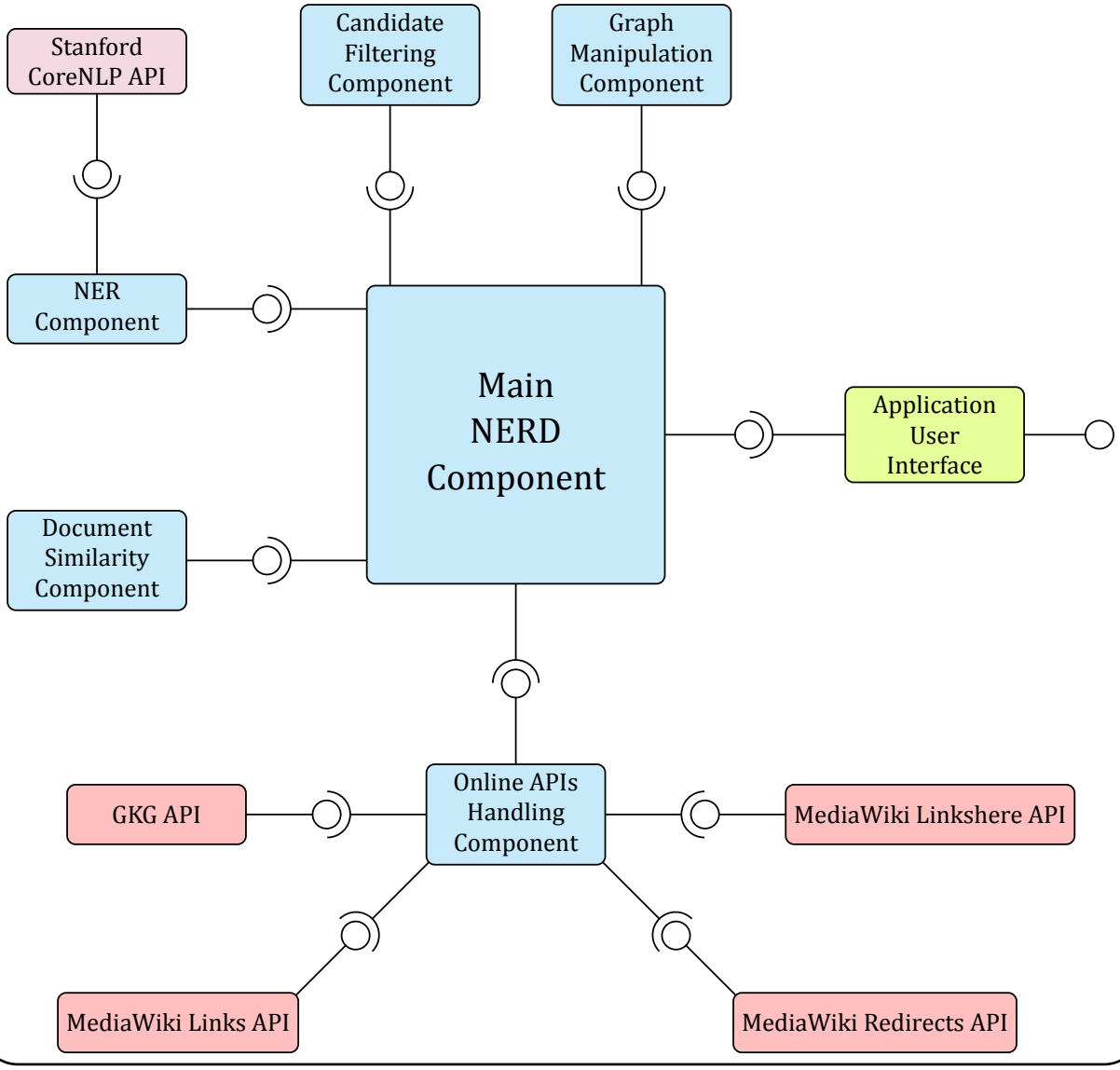
4.7 Διαγραμματική παρουσίαση συστήματος αποσαφήνισης

Σε αυτήν την ενότητα παρουσιάζεται συνολικά η λειτουργία του συστήματος αποσαφήνισης χρησιμοποιώντας διαγράμματα. Στην Εικόνα 5 φαίνεται μέσω ενός activity diagram η τυπική ακολουθία βημάτων κατά την αποσαφήνιση ενός κειμένου, από την αναγνώριση ονοματικών οντοτήτων μέχρι και την επίλυση του γράφου G με χρήση της ευριστικής μεθόδου που περιγράφηκε παραπάνω. Στην Εικόνα 6 χρησιμοποιείται ένα component diagram για να περιγραφεί η αρχιτεκτονική του συστήματος, με τα επιμέρους δομικά στοιχεία που συνδέονται ώστε να σχηματίσουν το συνολικό σύστημα αποσαφήνισης.

Εικόνα 5: Activity diagram συστήματος αποσαφήνισης. Σε αυτό το διάγραμμα φαίνονται τα διαδοχικά βήματα που πρέπει να γίνουν ώστε να αποσαφηνιστούν οι ονοματικές αναφορές ενός κειμένου T από το σύστημα αποσαφήνισης. Σημειώνεται η διαδοχή των λειτουργιών Named Entity Recognition και Named Entity Disambiguation.



Εικόνα 6: Component diagram συστήματος αποσαφήνισης. Σε αυτό το διάγραμμα φαίνονται τα επιμέρους αρχιτεκτονικά στοιχεία του συστήματος αποσαφήνισης. Παρατηρούμε ότι τα components παρέχουν συγκεκριμένα interfaces και απαιτούν για να λειτουργήσουν τη διαθεσιμότητα άλλων interfaces. Αυτό σημαίνει ότι τα components του συστήματος μπορούν να τροποποιηθούν χωρίς να επηρεαστεί το υπόλοιπο σύστημα. Για παράδειγμα, μπορεί να χρησιμοποιηθεί μια πιο σύνθετη τεχνική υπολογισμού ομοιότητας κειμένων, και το νέο Document Similarity Component θα λειτουργεί αρμονικά με το υπόλοιπο σύστημα, αρκεί να παρέχει το ίδιο interface.



Κεφαλαιο 5

Ζητήματα υλοποίησης

Στο κεφάλαιο 4 περιγράφηκε η κατασκευή ενός συστήματος αποσαφήνισης, δίνοντας έμφαση στις πηγές πληροφορίας και στην αλγοριθμική φύση του προβλήματος. Στο κεφάλαιο αυτό γίνονται σύντομες αναφορές σε πιο τεχνικά ζητήματα. Συγκεκριμένα, η ενότητα 5.1 αναφέρεται στην επιλογή γλώσσας προγραμματισμού για την εφαρμογή, η ενότητα 5.2 αφορά την ανάγκη χρήσης τοπικά αποθηκευμένων δεδομένων για την ανάπτυξη της εφαρμογής και η ενότητα 5.3 μιλά για μια βασική διαδικτυακή εφαρμογή που κάνει χρήση του συστήματος αποσαφήνισης.

5.1 Γλώσσα προγραμματισμού

Η εφαρμογή αποσαφήνισης που περιγράφηκε στο κεφάλαιο 4 θα μπορούσε να αναπτυχθεί σε οποιαδήποτε σύγχρονη γλώσσα προγραμματισμού. Για τον λόγο αυτόν, έγινε μέχρι αυτό το σημείο η συνειδητή επιλογή να ελαχιστοποιηθούν οι αναφορές σε πραγματικές γλώσσες προγραμματισμού και η παράθεση κώδικα εφαρμογής. Αντί αυτού, επιλέχθηκε η χρήση ψευδοκώδικα και φυσικής γλώσσας για την περιγραφή των μεθόδων. Ωστόσο, για να γράψει κανείς την εφαρμογή δεν αρκεί η κατανόηση των αλγορίθμων, αλλά πρέπει να χρησιμοποιηθεί μια συγκεκριμένη γλώσσα προγραμματισμού. Η γλώσσα που επιλέχθηκε στο πλαίσιο της παρούσας εργασίας είναι η Python. Μερικοί από τους λόγους που οδήγησαν σε αυτήν την επιλογή είναι οι εξής:

- *Εξαιρετικό library repository*. Για τις περισσότερες βασικές λειτουργίες που χρειάζεται ένας προγραμματιστής, υπάρχουν well-documented βιβλιοθήκες σε Python που να τις αναλαμβάνουν. Έτσι, μειώνεται ο φόρτος που αναλαμβάνει ο προγραμματιστής. Ως παραδείγματα που ενδιαφέρουν το συγκεκριμένο πεδίο αναφέρονται οι βιβλιοθήκες επεξεργασίας φυσικής γλώσσας και οι βιβλιοθήκες διαχείρισης γράφων.

- *Εύκολη επικοινωνία με APIs*. Η Python έχει τη δυνατότητα να επικοινωνεί σε λίγες γραμμές κώδικα με όλα τα APIs που χρειάζεται το σύστημα αποσαφήνισης ώστε να λειτουργήσει, επιτρέποντας ακόμα και προχωρημένες λειτουργίες όπως παράλληλη εκτέλεση HTTP requests.

- *End-to-end support*. Η Python χρησιμοποιήθηκε σε όλα τα στάδια ανάπτυξης, από τα αρχικά στάδια coding, debugging και profiling, μέχρι και την ανάπτυξη της διαδικτυα-

κής εφαρμογής που περιγράφεται στην ενότητα 5.3 και την παραγωγή των πινάκων και διαγραμμάτων που παρουσιάζονται στο κεφάλαιο 6. Η δυνατότητα χρήσης μόνο μιας γλώσσας για όλα τα στάδια της ανάπτυξης είναι πολύτιμη για τον προγραμματιστή, καθώς δε χρειάζεται να ασχολείται με πολλαπλά προγραμματιστικά περιβάλλοντα ταυτόχρονα.

5.2 Τοπικά αποθηκευμένα δεδομένα

Μέχρι αυτό το σημείο, ως μοναδικές πηγές πληροφορίας για το σύστημα αποσαφήνισης χρησιμοποιήθηκαν τα APIs του GKG και της Wikipedia. Πράγματι, η εφαρμογή μπορεί να λειτουργήσει μόνο με χρήση των APIs, αλλά αυτό οδηγεί σε σοβαρά προβλήματα απόδοσης. Θεωρώντας ότι έχουμε ένα κείμενο T που περιέχει k αναφορές προς αποσαφήνιση, καθεμιά από τις οποίες επιστρέφει n' δομικά αποδεκτές οντότητες από το GKG API, και τελικά το κάθε σύνολο υποψήφιων οντοτήτων περιέχει n οντότητες ($n \leq n'$), τότε θα πρέπει να γίνουν οι εξής κλήσεις προς τα APIs:

- k κλήσεις προς το GKG API, για να λάβουμε τις υποψήφιες οντότητες, και k κλήσεις προς το MediaWiki Links API, ώστε να λάβουμε τις προτάσεις αποσαφήνισης από τα Wikipedia disambiguation pages.
- kn' κλήσεις προς το MediaWiki Redirects API για να λάβουμε τα Wikipedia aliases για κάθε οντότητα, ώστε να ολοκληρωθεί η διαδικασία του φιλτραρίσματος οντοτήτων.
- kn κλήσεις προς το MediaWiki Linkshere API για να βρεθούν οι εισερχόμενοι σύνδεσμοι για κάθε υποψήφια οντότητα του τελικού γράφου G, ώστε να μπορεί να υπολογισθεί το WLM.

Παρατηρούμε ότι οι κλήσεις προς τα MediaWiki Redirects και Linkshere APIs μπορούν να είναι πολλές, και κάθε κλήση, ειδικά στην περίπτωση του Linkshere API, μπορεί να επιστρέφει από μερικές εκατοντάδες μέχρι και εκατοντάδες χιλιάδες αποτελέσματα. Συνδυάζοντας το παραπάνω γεγονός με τον περιορισμό του πλήθους αποτελεσμάτων που χαρακτηρίζει το MediaWiki API, βλέπουμε ότι δεν είναι βιώσιμη η αποκλειστική χρήση των APIs, ειδικά στην περίπτωση εκτέλεσης της πειραματικής αξιολόγησης του συστήματος, που απαιτεί επεξεργασία χιλιάδων κειμένων. Αξίζει να σημειωθεί ότι, παρά τα όσα υποδεικνύει η ασυμπτωτική πολυπλοκότητα που υπολογίσθηκε στην ενότητα 4.6, το πραγματικό bottleneck για μικρά και μεσαίου μεγέθους κείμενα (μέχρι 100 οντοτήτων) είναι ο χρόνος που απαιτείται για να συλλεχθούν οι αναγκαίες πληροφορίες από τα APIs.

Με τα παραπάνω κατά νου, ελήφθη η απόφαση να χρησιμοποιηθούν τα Wikipedia dumps ώστε να αποθηκευθούν τα δεδομένα τοπικά, σε μια σχεσιακή βάση δεδομένων. Συγκεκριμένα, χρησιμοποιήθηκαν τα dumps της 20ης Απριλίου 2017, και ειδικότερα τα dumps που αφορούν τα redirects και τα page-to-page link records. Αυτή η μέθοδος έχει το πλεονέκτημα ότι καθιστά περιττή τη χρήση των MediaWiki Redirects και MediaWiki Linkshere APIs, με αποτέλεσμα να επιταχύνεται η διαδικασία αποσαφήνισης. Πλέον, μόνο τα ερωτήματα προς το GKG API και το MediaWiki Links API είναι αναγκαία. Ωστόσο, υπάρχει το μειονέκτημα ότι δε στηριζόμαστε αποκλειστικά σε online δεδομένα. Αυτό σημαίνει ότι κατά καιρούς θα

πρέπει η τοπική βάση δεδομένων να ενημερώνεται, ώστε τα δεδομένα που περιέχει να είναι συμβατά με τα δεδομένα που λαμβάνονται από τα online APIs. Με τη χρήση της βάσης δεδομένων, έχουμε τις εξής κλήσεις προς πηγές πληροφορίας:

- k κλήσεις προς το GKG API, για να λάβουμε τις υποψήφιες οντότητες, και k κλήσεις προς το MediaWiki Links API, ώστε να λάβουμε τις προτάσεις αποσαφήνισης από τα Wikipedia disambiguation pages.
- kn' κλήσεις προς τη βάση δεδομένων για να λάβουμε τα Wikipedia aliases για κάθε οντότητα, ώστε να ολοκληρωθεί η διαδικασία του φιλτραρίσματος οντοτήτων.
- kn κλήσεις προς τη βάση δεδομένων για να βρεθούν οι εισερχόμενοι σύνδεσμοι για κάθε υποψήφια οντότητα του τελικού γράφου G, ώστε να μπορεί να υπολογισθεί το WLM.

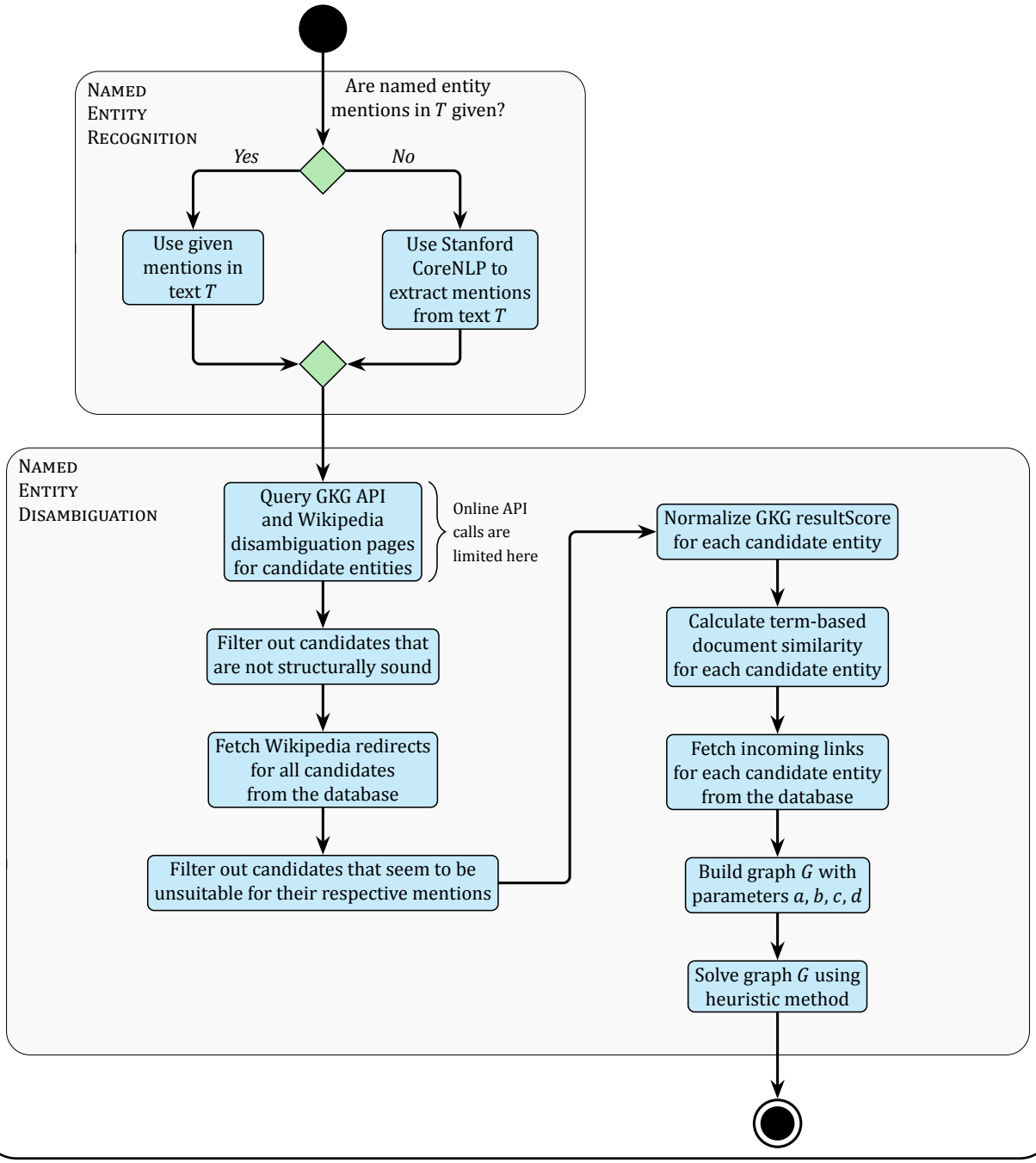
Έτσι, οι κλήσεις προς τα APIs μπορούν να γίνουν αμέσως μετά την αναγνώριση των ονοματικών αναφορών, και είναι όλες ανεξάρτητες μεταξύ τους. Απλά αναζητούμε στο GKG API και στα Wikipedia disambiguation pages τις προτάσεις αποσαφήνισης για το σύνολο των αναφορών $\{m_1, m_2, \dots, m_k\}$. Το γεγονός ότι αυτά τα requests είναι ανεξάρτητα μεταξύ τους επιτρέπει την παράλληλη εκτέλεσή τους. Έτσι, ελαχιστοποιείται ο χρόνος που καταναλώνεται από την εφαρμογή για το online API querying.

Δεδομένης της παραπάνω τροποποίησης, επηρεάζεται σε κάποιο βαθμό η ροή λειτουργιών και η αρχιτεκτονική σχεδίαση του συστήματος αποσαφήνισης. Για λόγους πληρότητας, το νέο activity diagram παρουσιάζεται στην Εικόνα 7, ενώ το νέο component diagram παρουσιάζεται στην Εικόνα 8.

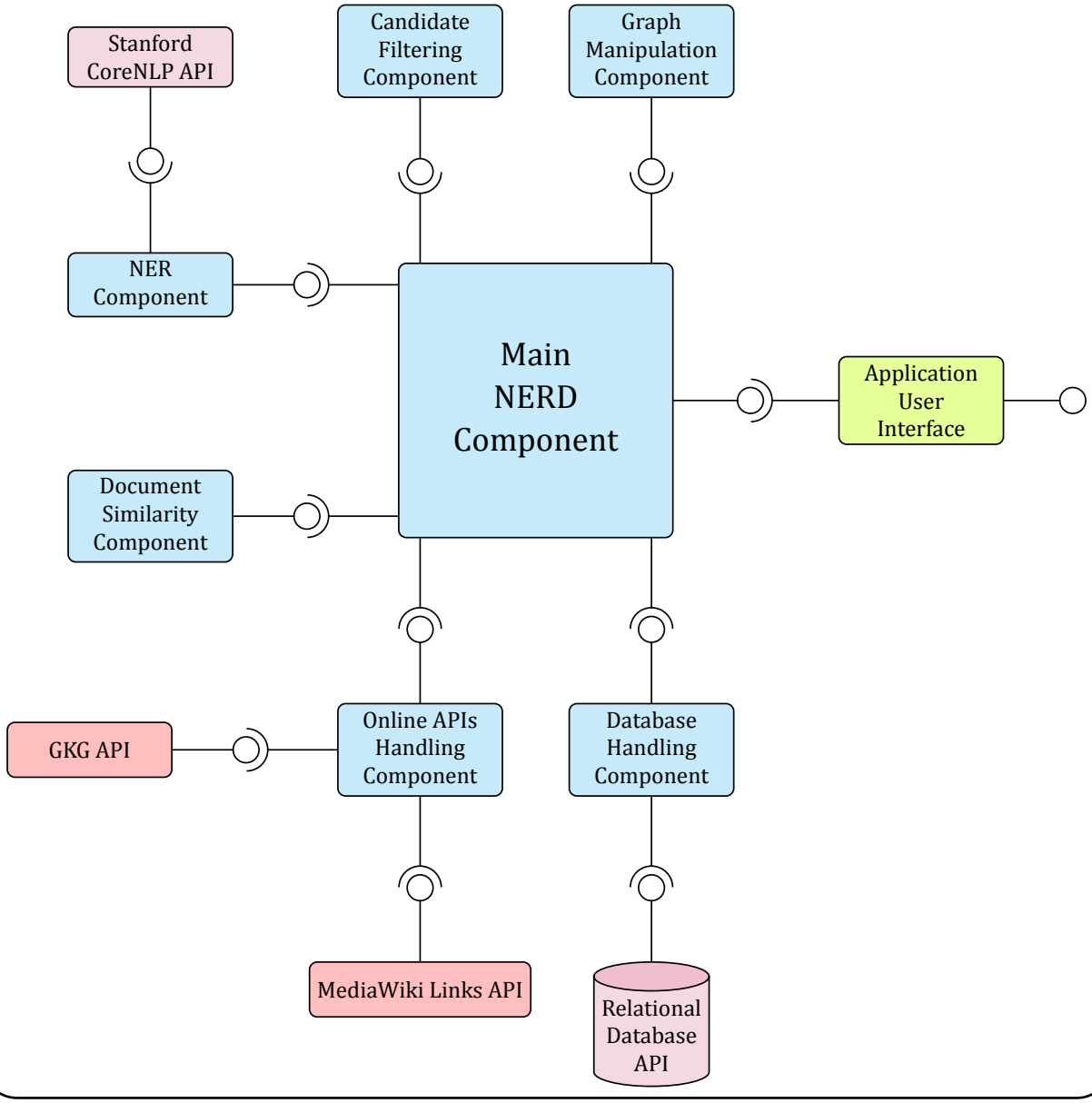
5.3 Διαδικτυακή εφαρμογή

Στο πλαίσιο της εργασίας, αναπτύχθηκε μια βασική διαδικτυακή εφαρμογή με χρήση του Django framework της Python, που επιτρέπει στον χρήστη να αποσαφηνίσει κείμενα, βλέποντας τα πιο ουσιαστικά στάδια της αποσαφήνισης καθώς συμβαίνουν. Συγκεκριμένα, ο χρήστης εισάγει το κείμενο προς επεξεργασία, καθώς και τις παραμέτρους αποσαφήνισης GKG API entity limit, a (και κατ' επέκταση b), c (και κατ' επέκταση d). Επίσης, ο χρήστης μπορεί να επιλέξει τη μέθοδο NER που προτιμά. Στον χρήστη παρουσιάζονται τα στάδια από τα οποία περνά η αποσαφήνιση, και τελικά του επιστρέφεται το αποτέλεσμα της αποσαφήνισης, μαζί με κάποια χρήσιμα στατιστικά στοιχεία. Στην Εικόνα 9 παρουσιάζεται μέσω ενός sequence diagram η ροή μηνυμάτων μεταξύ του Client και του Server σε ένα τυπικό σενάριο χρήσης της διαδικτυακής εφαρμογής αποσαφήνισης, ενώ στις Εικόνες 10, 11 και 12 παρουσιάζονται screenshots που επιδεικνύουν τα βασικά στοιχεία της εφαρμογής.

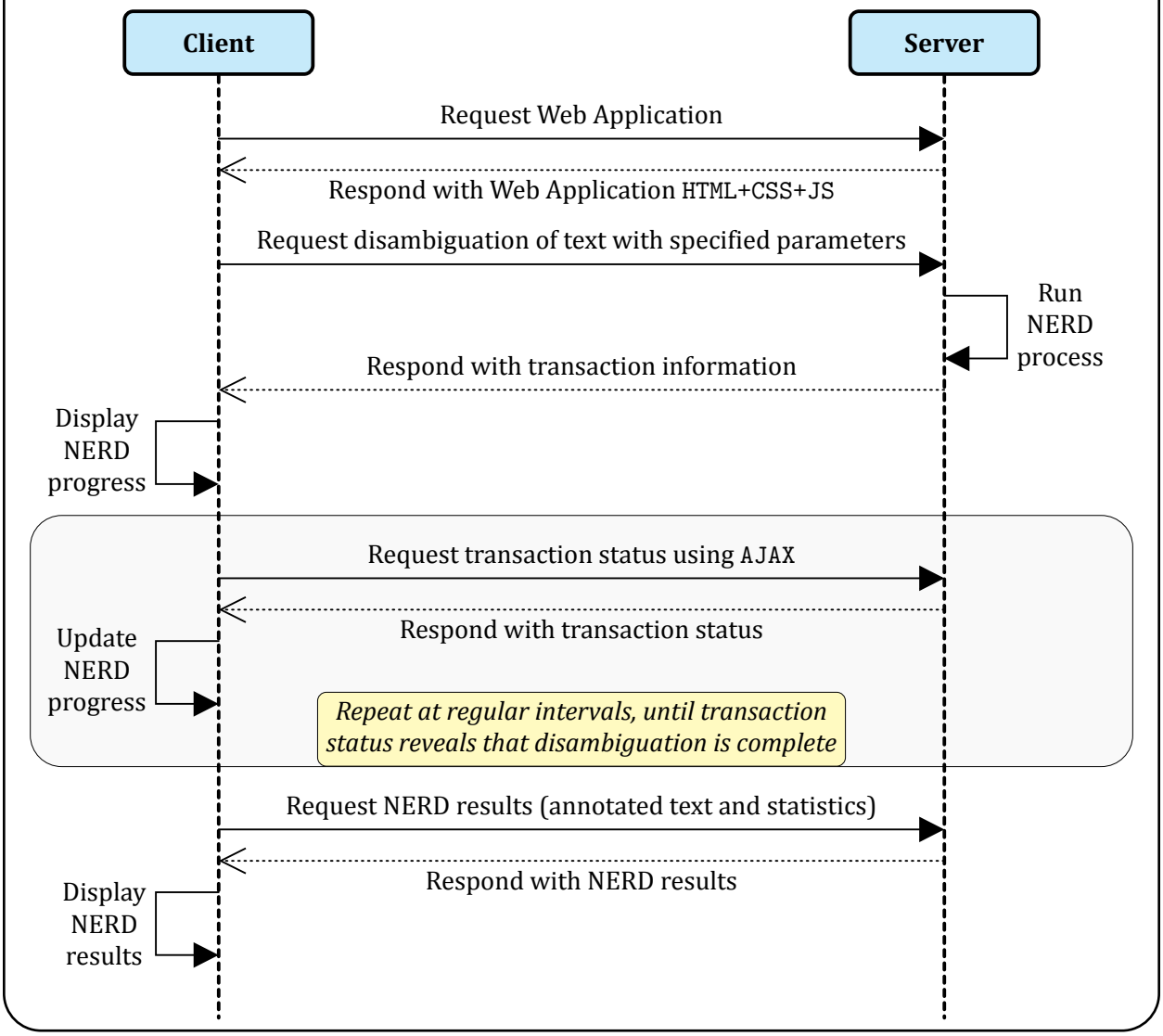
Εικόνα 7: Activity diagram συστήματος αποσαφήνισης, με τα Wikipedia redirects και τη δομή συνδέσμων της Wikipedia να αποθηκεύονται τοπικά, για λόγους απόδοσης. Πλέον, οι μόνες κλήσεις προς online APIs γίνονται προς το GKG API και προς το MediaWiki Links API, ώστε να βρεθούν οι προτάσεις αποσαφήνισης του GKG και της Wikipedia αντίστοιχα. Αυτές οι κλήσεις εκτελούνται μαζί αμέσως μετά το στάδιο του Named Entity Recognition.



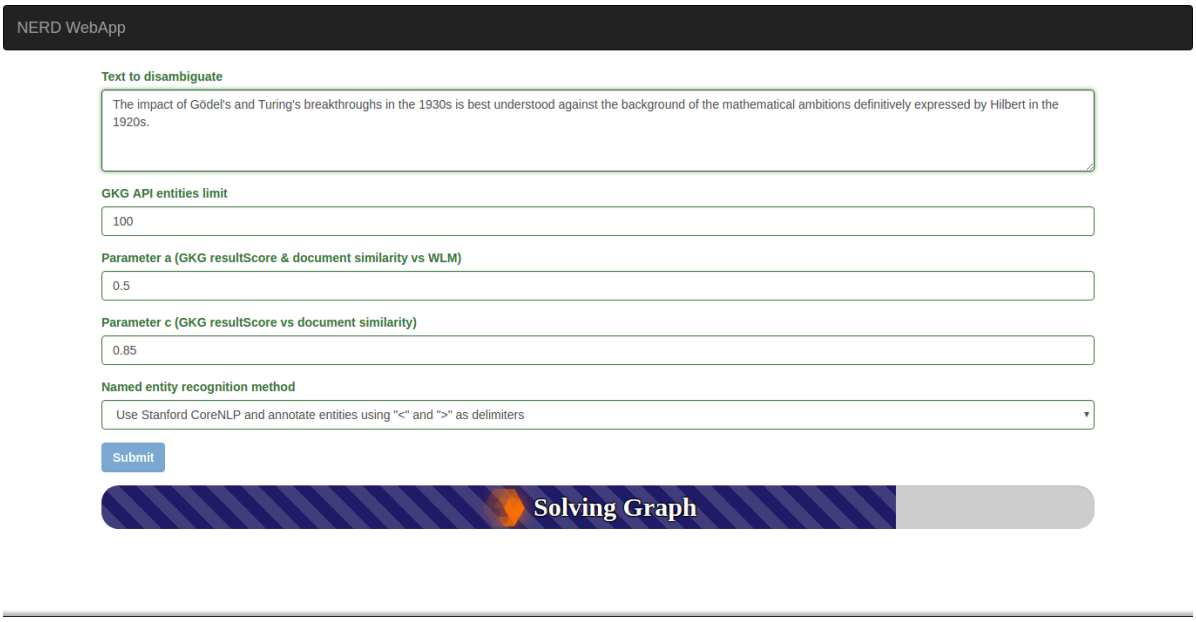
Εικόνα 8: Component diagram συστήματος αποσαφήνισης, με τα Wikipedia redirects και τη δομή συνδέσμων της Wikipedia να αποθηκεύονται τοπικά, για λόγους απόδοσης. Παρατηρούμε ότι το Online APIs Handling Component πλέον αλληλεπιδρά μόνο με το GKG API και με το MediaWiki Links API. Επίσης, προστέθηκε ένα δομικό στοιχείο που αναλαμβάνει την αλληλεπίδραση με τη βάση δεδομένων, το Database Handling Component.



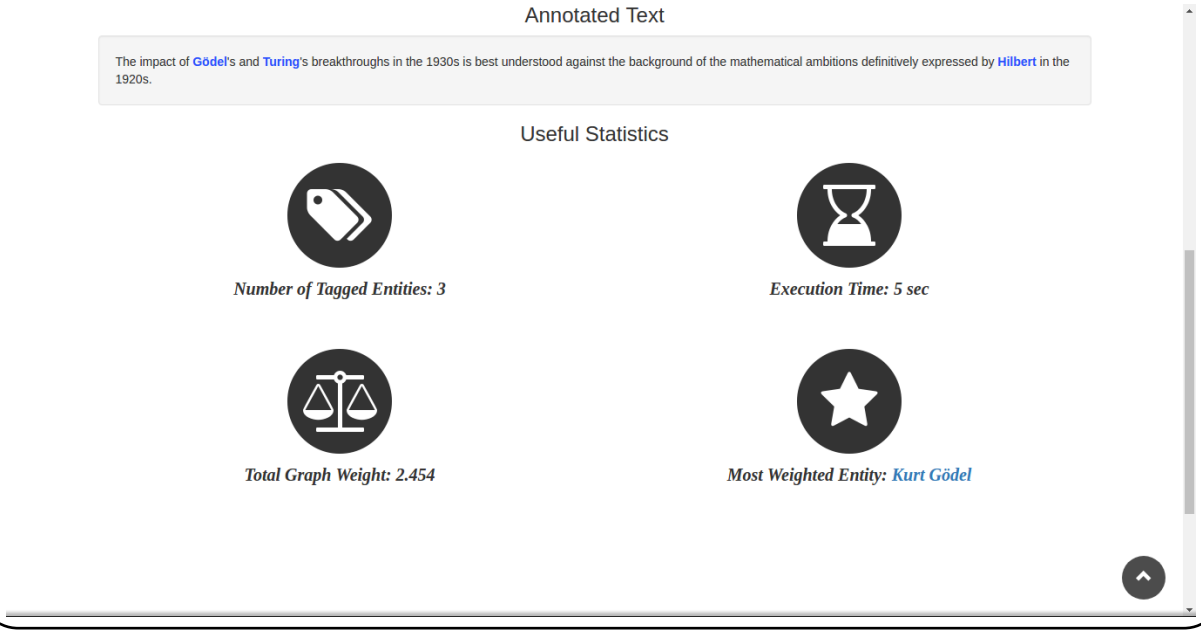
Εικόνα 9: Sequence diagram διαδικτυακής εφαρμογής αποσαφήνισης. Παρουσιάζονται οι ανταλλαγές μηνυμάτων μεταξύ ενός Client που επιθυμεί να αποσαφηνίσει ένα κείμενο και του Server στον οποίο τρέχει το σύστημα αποσαφήνισης.



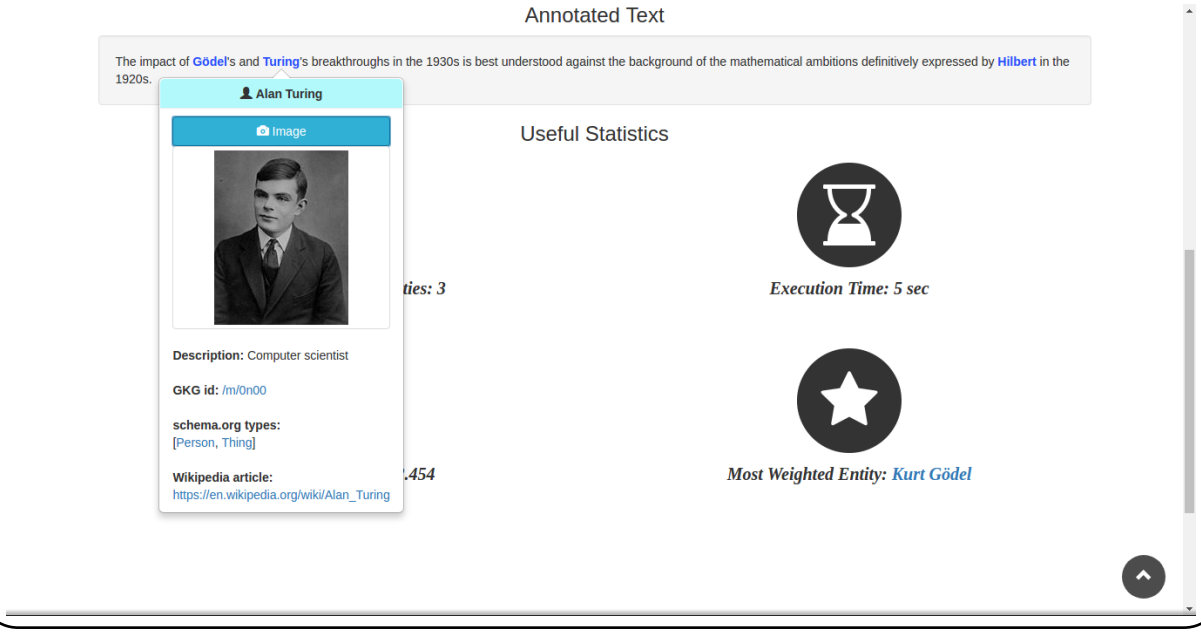
Εικόνα 10: Παράδειγμα εισόδου στη διαδικτυακή εφαρμογή αποσαφήνισης. Συγκεκριμένα, παρατηρούμε ότι δίνεται το κείμενο 18 από τα κείμενα του Παραρτήματος προς αποσαφήνιση. Επίσης, δίνονται οι παράμετροι GKG API entities limit $= 100$, $a = 0.5$, $b = 1 - a = 0.5$, $c = 0.85$, $d = 1 - c = 0.15$. Τέλος, για την αναγνώριση οντοτήτων επιλέγεται η χρήση του Stanford CoreNLP, ενώ ο χρήστης μπορεί να σημειώσει τις δικές του οντότητες στο κείμενο με χρήση των delimiters < και >. Βλέπουμε ότι ο αλγόριθμος είναι στο τελικό του στάδιο, αυτό της επίλυσης του γράφου G.



Εικόνα 11: Η έξοδος της εφαρμογής αποσαφήνισης που παράγεται από την είσοδο της Εικόνας 10. Συγκεκριμένα, σημειώνονται πάνω στο κείμενο οι οντότητες που επιστράφηκαν, με τις πιο σημαντικές πληροφορίες για την καθεμία. Επίσης, επιστρέφονται συνολικά στατιστικά που αφορούν την εκτέλεση: αριθμός οντοτήτων που αποσαφηνίστηκαν, χρόνος εκτέλεσης της μεθόδου αποσαφήνισης, συνολικό βάρος του τελικού γράφου και ο κόμβος του τελικού γράφου που έχει το μέγιστο βάρος ακμών.



Εικόνα 12: Η έξοδος της διαδικτυακής εφαρμογής αποσαφήνισης περιλαμβάνει πληροφορίες για κάθε οντότητα που επιστρέφεται. Συγκεκριμένα, παρέχεται η εικόνα της οντότητας, μια σύντομη περιγραφή αυτής, το μοναδικό GKG αναγνωριστικό της, οι schema.org τύποι της και το αντίστοιχο Wikipedia άρθρο της.



Κεφαλαιο 6

Πειραματική αξιολόγηση συστήματος αποσαφήνισης

Σε αυτό το κεφάλαιο αξιολογείται πειραματικά το σύστημα αποσαφήνισης που υλοποιήθηκε, χρησιμοποιώντας μετρικές αξιολόγησης οι οποίες περιγράφηκαν στην ενότητα 2.4. Στην ενότητα 6.1 περιγράφεται η διαδικασία παραγωγής ενός νέου dataset μικρών κειμένων που είναι κατάλληλο για την αξιολόγηση του συστήματος αποσαφήνισης, καθώς και ο τρόπος αξιολόγησης του συστήματος βάσει του dataset αυτού. Στην ενότητα 6.2 αξιοποιείται ένα dataset από ειδησεογραφικά άρθρα που έχει ήδη χρησιμοποιηθεί στη βιβλιογραφία.

6.1 Πειραματική αξιολόγηση με μικρά κείμενα

6.1.1 Dataset μικρών κειμένων

Για τους σκοπούς της πειραματικής αξιολόγησης του συστήματος αποσαφήνισης, δε βρέθηκε στη βιβλιογραφία ένα dataset με μικρά κείμενα που να κρίθηκε κατάλληλο. Το πιο ποιοτικό τέτοιο dataset είναι το KORE 50 NIF NER Corpus, το οποίο δεν μπορεί να χρησιμοποιηθεί για την αξιόπιστη πειραματική αξιολόγηση του συστήματος, για τους εξής λόγους:

- Κάποιες από τις ονοματικές οντότητες στο dataset δεν αποτελούν ονοματικές οντότητες με την αυστηρή έννοια. Για παράδειγμα, αναφέρονται είδη μουσικής ως ονοματικές οντότητες.
- Κάποιες από τις ονοματικές οντότητες στο dataset δεν έχουν αντίστοιχη καταχώρηση στην Wikipedia, διότι δεν είναι αρκετά γνωστές.
- Κάποια από τα κείμενα έχουν αμφισβητούμενο ground truth, δηλαδή θα μπορούσαν τα ταιριάξουν περισσότερες από μία οντότητες σε κάθε αναφορά.
- Το dataset, όπως δηλώνει και το όνομά του (**K**eyphrase **O**verlap **Re**latedness), είναι κατασκευασμένο ώστε να αξιολογήσει τη χρήση εναλλακτικών μεθόδων υπολογισμού entity relatedness, που είναι απαλλαγμένες από την ανάγκη χρήσης της δομής συνδέσμων της Wikipedia. Αυτό, όμως, ξεφεύγει από τα όρια αυτής της εργασίας.

Έτσι, κατασκευάστηκε ένα dataset που συγκεντρώνει τα εξής χαρακτηριστικά:

- Μικρά κείμενα, με 98.92 χαρακτήρες ανά κείμενο κατά μέσο όρο.
- Υψηλή πυκνότητα ονοματικών οντοτήτων, με 3.21 οντότητες ανά κείμενο κατά μέσο όρο.
- Οντότητες που να είναι αμφίσημες για ένα αυτόματο σύστημα αποσαφήνισης.
- Ground truth που να είναι σχετικά προφανές για τον αναγνώστη που έχει γνώση του αντίστοιχου πεδίου, ώστε να μπορεί να γίνει αξιόπιστα η αξιολόγηση ενός αυτόματου συστήματος αποσαφήνισης οντοτήτων βάσει αυτού.
- Οντότητες που να έχουν αντίστοιχες καταχωρήσεις στην Wikipedia και στον GKG.

Το dataset αποτελείται από 300 μικρά κείμενα που έχουν συλλεχθεί από ποικίλες πηγές, όπως άρθρα της Wikipedia, ειδησεογραφικά άρθρα, blog posts, άλλα datasets (μεταξύ των οποίων και το KORE 50 NIF NER Corpus), και υπέστησαν επεξεργασία ως εξής:

- Αναγνωρίστηκαν οι ονοματικές οντότητες στα κείμενα, δηλαδή βρέθηκαν τα όριά τους.
- Βρέθηκαν στην Wikipedia και στον GKG οι αντίστοιχες οντότητες που ταιριάζουν καλύτερα στις εκάστοτε αναφορές και στα εκάστοτε contexts.

Το dataset μπορεί να βρεθεί στο Παράρτημα αυτής της εργασίας. Παρατηρούμε ότι δόθηκε ιδιαίτερη έμφαση στις ονοματικές αναφορές που μπορούν να έχουν διαφορετική σημασία σε διαφορετικά contexts, καθώς αυτή θεωρήθηκε ότι είναι μια από τις βασικές προκλήσεις που καλείται να αντιμετωπίσει ένα σύστημα αποσαφήνισης οντοτήτων.

6.1.2 Παραμετροποίηση και πειραματικά αποτελέσματα

Με χρήση του dataset που περιγράφηκε στην παράγραφο 6.1.1, μπορούμε να αξιολογήσουμε το σύστημα αποσαφήνισης. Υπάρχουν, ωστόσο, αρκετές παράμετροι (βλέπε κεφάλαιο 4) που μπορούν να επιλεγούν για την αξιολόγηση του συστήματος αποσαφήνισης:

- $limit$, που είναι το μέγιστο πλήθος οντοτήτων που επιστρέφει το GKG API. Μεγαλύτερες τιμές του $limit$ επιτρέπουν την εύρεση περισσότερων οντοτήτων κατά την αποσαφήνιση, αλλά αυξάνουν τον θόρυβο εντός των συνόλων υποψήφιων οντοτήτων.
- c, που δείχνει πόσο βάρος δίνεται στο GKG resultScore έναντι του document similarity.
- a, που δείχνει πόσο βάρος δίνεται στο GKG resultScore και στο document similarity έναντι του WLM.

Πέραν των παραπάνω αριθμητικών παραμέτρων, ενδιαφέρον παρουσιάζει και η περίπτωση όπου δίνουμε στο σύστημα αποσαφήνισης το κείμενο ως έχει, σε σύγκριση με την περίπτωση που δίνουμε και τις αναφορές προς αποσαφήνιση. Στο πρώτο σενάριο, το σύστημα αναλαμβάνει, εκτός από την αποσαφήνιση, και την ευθύνη της αναγνώρισης ονοματικών οντοτήτων.

Για την πειραματική αξιολόγηση με χρήση του dataset 300 μικρών κειμένων γίνονται οι εξής επιλογές:

- $limit = 100$. Αυτό το όριο επιτρέπει, στις περισσότερες περιπτώσεις, να εντοπίζονται όλες οι λογικές υποψήφιες οντότητες για μια αναφορά. Η χρήση μεγαλύτερου ορίου όχι μόνο οδηγεί σε ογκώδη σύνολα υποψήφιων οντοτήτων χωρίς να υπάρχει λόγος, αλλά οδηγεί και σε προβλήματα απόδοσης.

- $c = 0.85$. Αυτό σημαίνει ότι δίνουμε μεγαλύτερο βάρος στο GKG resultScore, παρά στο απλό μέτρο document similarity που περιγράφηκε στην παράγραφο 4.4.2. Αυτό συμβαίνει επειδή το document similarity είναι απλά ένα βοηθητικό κριτήριο ταξινόμησης οντοτήτων, το οποίο σε αρκετές περιπτώσεις δε δίνει αρκετή πληροφορία.

- $a \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. Αυτή είναι η βασική παράμετρος που θέλουμε να εξετάσουμε κατά την πειραματική αξιολόγηση. Δείχνει πόσο βάρος δίνουμε στον συνδυασμό των GKG resultScore και document similarity έναντι του WLM. Με άλλα λόγια, αφορά το αν επιλέγουμε να δώσουμε μεγαλύτερο βάρος στο ταίριασμα οντοτήτων από μόνες τους (με βάση το GKG resultScore και το κείμενο στο οποίο ανήκουν, ως σύνολο όρων) ή στη συνολική σημασιολογική συνάφεια του κειμένου βάσει του WLM.

- Τα κείμενα δίνονται στο σύστημα με δύο τρόπους:

    1. Ως απλά αδόμητα κείμενα, δηλαδή ως ακολουθίες χαρακτήρων, από τις οποίες το σύστημα καλείται να εξάγει τις ονοματικές οντότητες και έπειτα να τις αποσαφηνίσει. Εδώ χρησιμοποιούνται ως μετρικές αξιολόγησης οι Micro Average Precision, Macro Average Precision, Micro Average Recall και Macro Average Recall.
    2. Ως κείμενα, με δοσμένες τις οντότητες προς αποσαφήνιση. Σε αυτήν την περίπτωση μόνη ευθύνη του συστήματος είναι η αποσαφήνιση των δοθέντων οντοτήτων. Εδώ χρησιμοποιούνται ως μετρικές αξιολόγησης οι Micro Average Accuracy και Macro Average Accuracy.

Τα αποτελέσματα της πειραματικής αξιολόγησης με την παραπάνω παραμετροποίηση φαίνονται στους Πίνακες 5, 6 και 7, ενώ παρουσιάζονται γραφικά στις Εικόνες 13, 14 και 15. Μπορούμε να κάνουμε τα εξής σχόλια:

- Το σύστημα αποδίδει πολύ ικανοποιητικά. Συγκεκριμένα, παρατηρούμε ότι η καλύτερη παραμετροποίηση είναι αυτή με $a = 0.5$, και σε αυτήν την περίπτωση η μετρική accuracy, που αφορά το σύστημα χωρίς την ανάγκη αναγνώρισης ονοματικών οντοτήτων, ξεπερνά το 97 %. Αυτό σημαίνει ότι όταν υπάρχει ισορροπία ανάμεσα στο βάρος που δίνουμε στην εκ των προτέρων δημοτικότητα (σε συνδυασμό με την επικάλυψη όρων στο κείμενο) και στο βάρος που δίνουμε στη σημασιολογική συγγένεια, παίρνουμε τα καλύτερα αποτελέσματα.

- Από τις τιμές του precision παρατηρούμε ότι το σύστημα έχει τη δυνατότητα να αποσαφηνίζει σωστά τις οντότητες που ανακαλύπτει στο κείμενο. Το γεγονός ότι το precision έχει χαμηλότερες τιμές από το accuracy οφείλεται στο γεγονός ότι το σύστημα δεν καταφέρνει να αποσαφηνίσει τις οντότητες που εντοπίζει στο κείμενο με την ίδια ακρίβεια, καθώς οι οντότητες που δεν κατάφερε να ανακαλύψει δεν του είναι διαθέσιμες, και αυτό επηρεάζει τη συλλογική αποσαφήνιση.

Πίνακας 5: Τιμές Micro Average Precision και Macro Average Precision κατά την πειραματική αξιολόγηση με χρήση του dataset 300 μικρών κειμένων.

a	Micro Average Precision (%)	Macro Average Precision (%)
0.0	73.68	72.93
0.1	78.95	77.70
0.2	84.64	83.63
0.3	89.47	88.72
0.4	93.66	93.48
0.5	94.41	94.85
0.6	90.12	90.80
0.7	83.78	84.18
0.8	75.73	75.96
0.9	69.71	69.26
1.0	66.17	65.51

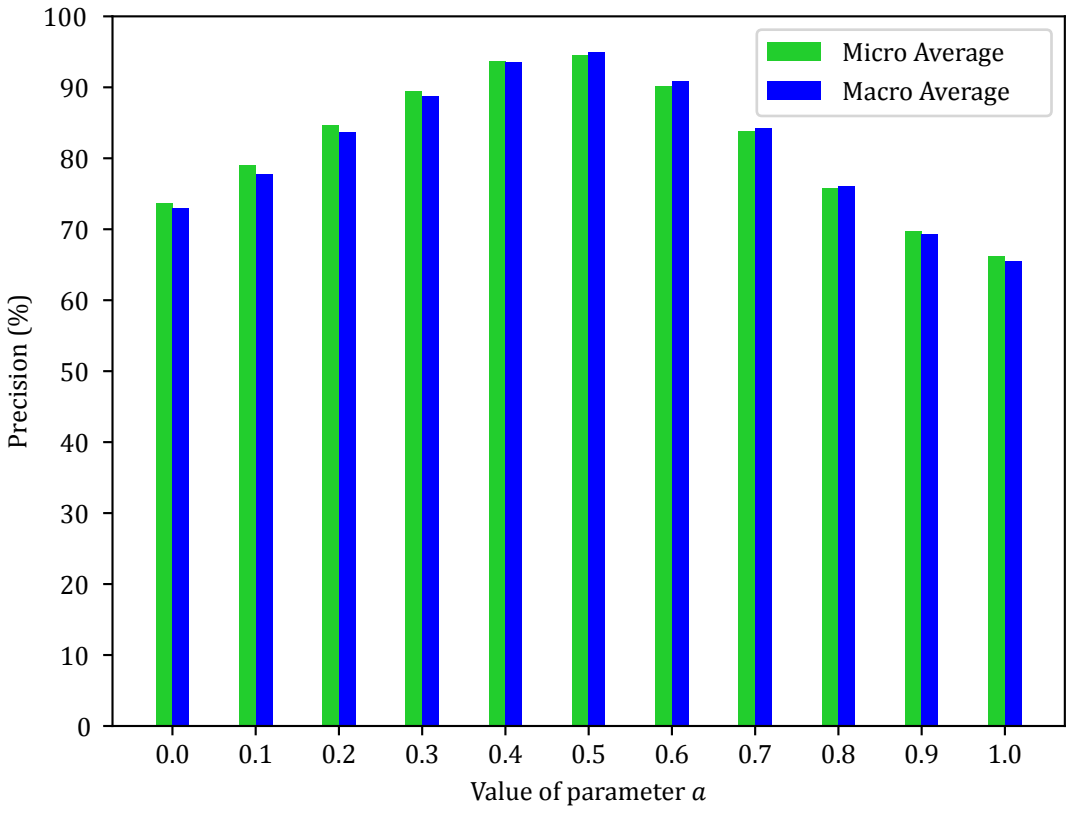
Πίνακας 6: Τιμές Micro Average Recall και Macro Average Recall κατά την πειραματική αξιολόγηση με χρήση του dataset 300 μικρών κειμένων.

a	Micro Average Recall (%)	Macro Average Recall (%)
0.0	71.31	70.92
0.1	76.40	75.53
0.2	81.91	81.49
0.3	86.59	86.27
0.4	90.64	90.98
0.5	91.37	92.35
0.6	87.21	88.32
0.7	81.08	81.83
0.8	73.28	73.87
0.9	67.46	67.34
1.0	64.03	63.72

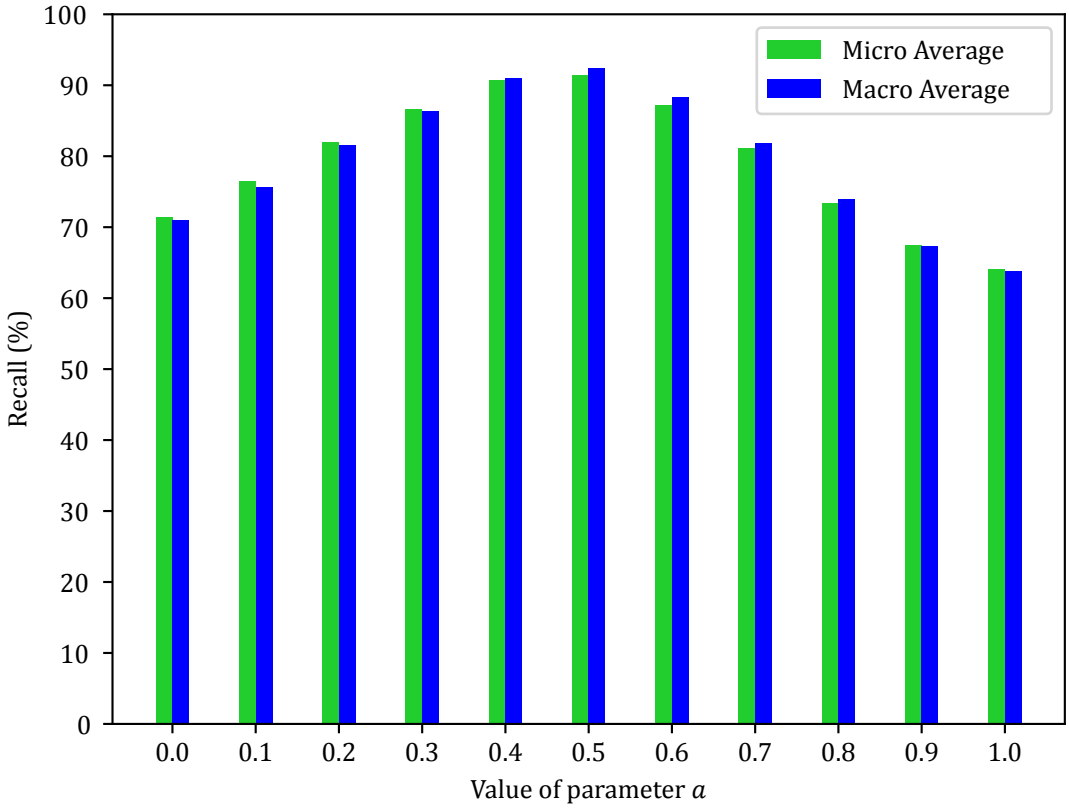
Πίνακας 7: Τιμές Micro Average Accuracy και Macro Average Accuracy κατά την πειραματική αξιολόγηση με χρήση του dataset 300 μικρών κειμένων.

a	Micro Average Accuracy (%)	Macro Average Accuracy (%)
0.0	76.61	75.79
0.1	82.22	81.15
0.2	87.63	86.99
0.3	92.31	91.59
0.4	96.99	96.67
0.5	97.30	97.74
0.6	93.66	93.78
0.7	87.01	86.91
0.8	78.38	78.35
0.9	72.45	71.81
1.0	68.81	67.94

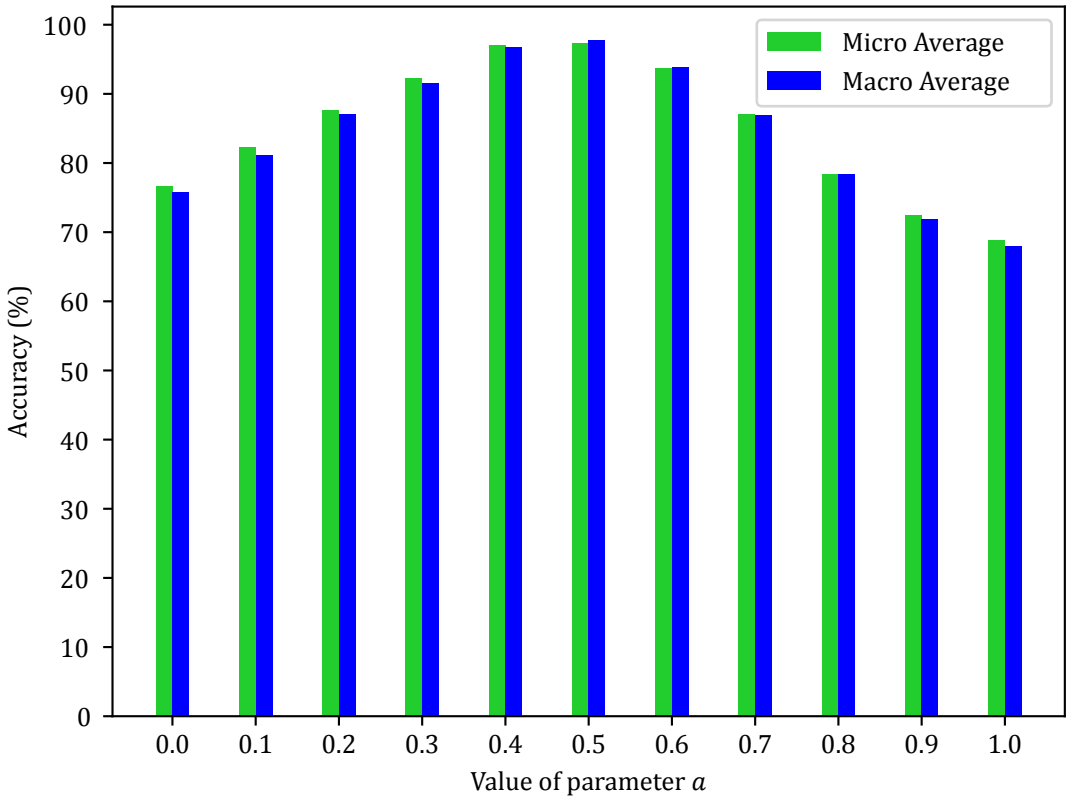
Εικόνα 13: Γραφική απεικόνιση Micro Average Precision και Macro Average Precision κατά την πειραματική αξιολόγηση με χρήση του dataset 300 μικρών κειμένων.



Εικόνα 14: Γραφική απεικόνιση Micro Average Recall και Macro Average Recall κατά την πειραματική αξιολόγηση με χρήση του dataset 300 μικρών κειμένων.



Εικόνα 15: Γραφική απεικόνιση Micro Average Accuracy και Macro Average Accuracy κατά την πειραματική αξιολόγηση με χρήση του dataset 300 μικρών κειμένων.



- Από τις τιμές του recall, που είναι χαμηλότερες από τις τιμές του precision, βλέπουμε ότι η μεγαλύτερη αδυναμία του συστήματος δεν είναι η αποσαφήνιση των οντοτήτων που έχουν ανακαλυφθεί, αλλά η εύρεσή τους στο κείμενο. Δηλαδή, το εργαλείο Stanford CoreNLP είναι αρκετά συντηρητικό ως προς την εύρεση οντοτήτων στο κείμενο. Αυτό όχι μόνο επηρεάζει τη δυνατότητα συλλογικής αποσαφήνισης, καθώς είναι πιθανό να παραλείπονται οντότητες-κλειδιά για την αποσαφήνιση, αλλά μειώνει και εξορισμού το recall.

- Όπως αναφέρθηκε παραπάνω, η βέλτιστη απόδοση του συστήματος όσον αφορά τις μετρικές precision-recall-accuracy επιτυγχάνεται όταν $a = 0.5$. Κοιτώντας στις Εικόνες 13, 14 και 15 τη μεταβολή των μετρικών αυτών καθώς αλλάζει η παράμετρος a, παρατηρούμε ότι ξεκινάμε από σχετικά χαμηλές τιμές για $a = 0$, έπειτα βλέπουμε αύξηση καθώς πλησιάζουμε το $a = 0.5$ και μετά ξανά μείωση μέχρι το $a = 1$. Αξίζει να σημειωθεί ότι για το συγκεκριμένο dataset παίρνουμε καλύτερα αποτελέσματα για $a = 0$ παρά για $a = 1$. Αυτό δεν προκαλεί έκπληξη, καθώς το dataset περιέχει μικρά, σημασιολογικά συναφή κείμενα. Άρα, μεγαλύτερη αξία φαίνεται να έχει το κριτήριο σημασιολογικής συνάφειας.

- Σε όλες τις μετρικές (precision, recall, accuracy) παρατηρείται ότι οι προσεγγίσεις Micro Average και Macro Average δίνουν περίπου τα ίδια αποτελέσματα. Αυτό σημαίνει ότι τα συνολικά ποσοστά μετρικών για όλο το dataset (Micro Average) εκφράζουν και τον μέσο όρο των μετρικών για κάθε κείμενο ξεχωριστά (Macro Average).

6.2 Πειραματική αξιολόγηση με το CoNLL dataset

6.2.1 Επεξεργασία του CoNLL dataset

Ο Hoffart και οι συνεργάτες του [Hof15], έχοντας την ανάγκη δημιουργίας ενός dataset για την αξιολόγηση του δικού τους συστήματος αποσαφήνισης, ονόματι AIDA, και για τη σύγκρισή του με άλλα συστήματα αποσαφήνισης οντοτήτων, βασίστηκαν στα δεδομένα του CoNLL-2003 ώστε να δημιουργήσουν το δικό τους dataset, που το ονόμασαν CoNLL. Αυτό αποτελείται από 1393 ειδησεογραφικά άρθρα από το πρακτορείο Reuters, στα οποία έχουν σημειωθεί τα όρια των ονοματικών οντοτήτων, καθώς και οι αντίστοιχες καταχωρήσεις στην Wikipedia και στη Freebase. Από αυτά τα κείμενα χρησιμοποιήθηκαν 231 κείμενα για τη σύγκριση με άλλα συστήματα αποσαφήνισης. Αυτό το σύνολο κειμένων, που στο εξής θα καλείται Reuters-231, χρησιμοποιείται για την ολοκλήρωση της αξιολόγησης του δικού μας συστήματος αποσαφήνισης. Για να χρησιμοποιηθεί το Reuters-231, έγιναν κάποια προπαρασκευαστικά βήματα:

- *Φιλτράρισμα οντοτήτων*. Το Reuters-231 είναι παλιό dataset, καθώς έχει να ενημερωθεί ουσιαστικά περίπου 7 χρόνια. Αυτό σημαίνει ότι αρκετά από τα Wikipedia urls που περιέχει δεν είναι ορθά σήμερα. Επίσης, κάποια Freebase ids δε μεταφράζονται σε GKG ids, και κάποιες οντότητες μπορεί ακόμα και να λείπουν από τον GKG, καθώς δε μεταφέρθηκαν μετά την κατάργηση της Freebase. Για τον λόγο αυτόν, έγινε φιλτράρισμα των οντοτήτων ώστε να διατηρηθούν μόνο αυτές που είναι κατάλληλες προς αποσαφήνιση με χρήση του GKG API. Μετά το φιλτράρισμα, έμεινε προς αποσαφήνιση το

95.81 % των οντοτήτων που υπήρχαν στο Reuters-231 dataset. Αξίζει να σημειωθεί, επιπλέον, ότι το Reuters-231, όπως και όλο το CoNLL dataset, έχει επισημειωθεί με βάση τα περιεχόμενα της βάσης γνώσης YAGO2. Αυτό σημαίνει ότι ονοματικές οντότητες που εμφανίζονται στο κείμενο και μπορούν να υποστούν επεξεργασία από το δικό μας σύστημα αποσαφήνισης μπορεί να μην έχουν σημειωθεί διότι απουσιάζουν από τη συγκεκριμένη βάση γνώσης.

- *Απομόνωση ονοματικών αναφορών*. Κατά την αξιολόγηση του συστήματος AIDA, οι ονοματικές αναφορές δίνονται στο σύστημα. Δηλαδή, δίνονται οι συμβολοσειρές προς αποσαφήνιση. Έτσι, για κάθε κείμενο του Reuters-231, απομονώνονται οι ονοματικές αναφορές, και δίνονται ως είσοδος στο δικό μας σύστημα αποσαφήνισης μαζί με το κείμενο. Αυτό, βέβαια, σημαίνει ότι χρησιμοποιούνται μόνο οι μετρικές Micro Average Accuracy και Macro Average Accuracy για την αξιολόγηση του συστήματος αποσαφήνισης.

Το CoNLL, και συνεπώς το Reuters-231, έχει βασιστεί σε κείμενα που δεν είναι δημόσια διαθέσιμα (copyrighted material), για αυτό και δε γίνονται αναφορές σε συγκεκριμένα κείμενα στην παρούσα εργασία.

6.2.2 Παραμετροποίηση και πειραματικά αποτελέσματα

Για την αξιολόγηση του συστήματος αποσαφήνισης με χρήση του Reuters-231 dataset χρησιμοποιήθηκε η παραμετροποίηση που περιγράφηκε στην παράγραφο 6.1.2. Τα αποτελέσματα της πειραματικής αξιολόγησης παρουσιάζονται στον Πίνακα 8 και στην Εικόνα 16. Για λόγους ευχέρειας, αναγράφονται τα τελικά πειραματικά αποτελέσματα από την εργασία του Hoffart [Hof15] στον Πίνακα 9, μαζί με τα πειραματικά αποτελέσματα που προέκυψαν με χρήση της δική μας μεθόδου αποσαφήνισης στο ίδιο dataset. Τονίζεται ότι αυτά τα αποτελέσματα δεν έχουν αναπαραχθεί στο πλαίσιο αυτής της εργασίας, και η σύγκριση με άλλες μεθόδους αποσαφήνισης έγινε μόνο βάσει βιβλιογραφίας. Μπορούμε να κάνουμε τα εξής σχόλια:

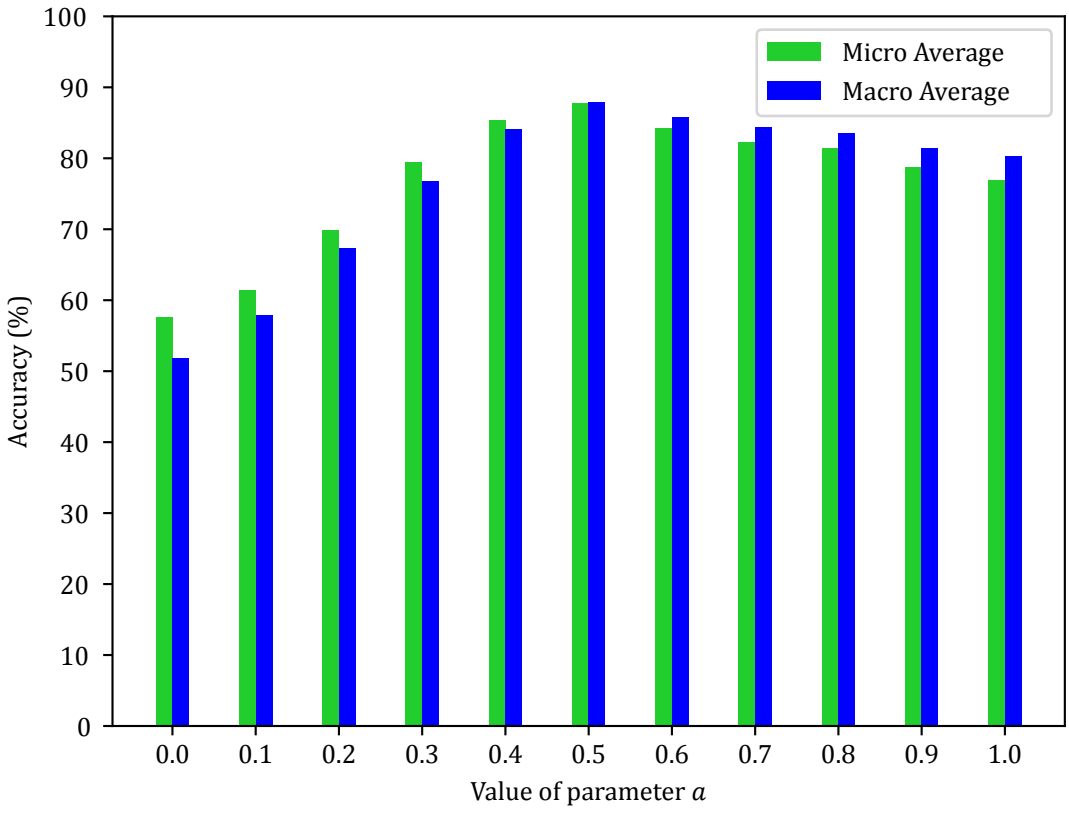
- Το σύστημα αποδίδει ικανοποιητικά. Συγκεκριμένα, παρατηρούμε ότι η καλύτερη παραμετροποίηση είναι αυτή με $a = 0.5$, όπως και στην περίπτωση του dataset με τα 300 μικρά κείμενα. Τότε, η ακρίβεια ξεπερνά το 87 %. Σημειώνεται ότι το σύστημα AIDA πετυχαίνει Micro Average Accuracy $= 82.03$ % και Macro Average Accuracy $= 82.63$ %. Το δικό μας σύστημα αποσαφήνισης πετυχαίνει Micro Average Accuracy $= 87.66$ % και Macro Average Accuracy $= 87.84$ %, που αποτελεί βελτίωση πάνω από 5 %. Ακόμα και αν ληφθούν υπόψιν οι οντότητες που αφαιρέθηκαν στο στάδιο του φιλτραρίσματος της παραγράφου 6.2.1 ως λανθασμένες, το σύστημα διατηρεί προβάδισμα στα ποσοστά ακρίβειας.

- Από την Εικόνα 16 φαίνεται ότι η μετρική accuracy ξεκινάει από πολύ χαμηλές τιμές για $a = 0$, αυξάνεται μέχρι να κορυφωθεί για $a = 0.5$ και μετά φθίνει αργά μέχρι να φτάσουμε την τιμή $a = 1$. Σε αντίθεση με όσα διαπιστώθηκαν στην παράγραφο 6.1.2, λαμβάνουμε καλύτερα αποτελέσματα για $a = 1$ παρά για $a = 0$. Δηλαδή, εδώ είναι πιο πολύτιμη η πληροφορία που παρέχει το GKG resultScore σε συνδυασμό με το document

Πίνακας 8: Τιμές Micro Average Accuracy και Macro Average Accuracy κατά την πειραματική αξιολόγηση με χρήση του Reuters-231 dataset.

a	Micro Average Accuracy (%)	Macro Average Accuracy (%)
0.0	57.53	51.83
0.1	61.42	57.78
0.2	69.83	67.35
0.3	79.37	76.67
0.4	85.29	84.03
0.5	87.66	87.84
0.6	84.21	85.70
0.7	82.26	84.31
0.8	81.35	83.51
0.9	78.65	81.36
1.0	76.86	80.27

Εικόνα 16: Γραφική απεικόνιση Micro Average Accuracy και Macro Average Accuracy κατά την πειραματική αξιολόγηση με χρήση του Reuters-231 dataset.



Πίνακας 9: Σύγκριση πειραματικών αποτελεσμάτων της δικής μας μεθόδου αποσαφήνισης με τα αποτελέσματα εναλλακτικών μεθόδων αποσαφήνισης στο Reuters-231 dataset. Αυτά τα δεδομένα, πλην βεβαίως των δεδομένων που αφορούν το δικό μας σύστημα, είναι παρμένα από την εργασία του Hoffart [Hof15].

	Δική μας μέθοδος	*Εναλλακτικές μέθοδοι*					
	$a = 0.50$ $c = 0.85$ GKG limit $= 100$	AIDA [Hof15]	prior	Cucerzan [Cuc07]	Kulkarni [KSRC09]	Kulkarni + prior	Kulkarni + coherence
Macro Average Accuracy (%)	**87.84**	82.63	70.57	43.74	58.06	76.74	76.74
Micro Average Accuracy (%)	**87.66**	82.03	75.10	51.03	63.42	72.31	72.87

similarity, παρά η πληροφορία που παρέχει η μετρική WLM. Αυτό οφείλεται στη φύση του dataset, καθώς πολλές από τις αναφορές που περιέχει μιλούν για συγκεκριμένες οντότητες, οι οποίες μπορούν να προσδιορισθούν με αρκετή βεβαιότητα από το GKG resultScore, και δευτερευόντως με χρήση του document similarity. Για παράδειγμα, υπάρχουν κείμενα που αναφέρονται σε άτομα με το ονοματεπώνυμό τους, και η αποσαφήνιση αυτών των αναφορών μπορεί να γίνει χωρίς να χρειάζεται να εμπλακεί η μετρική WLM.

- Από την Εικόνα 16 παρατηρούμε ότι οι τιμές των Micro Average Accuracy και Macro Average Accuracy είναι μεν σχετικά κοντά (ιδίως στην περίπτωση $a = 0.5$), αλλά υπάρχουν αποκλίσεις. Συγκεκριμένα, για $a < 0.5$ βλέπουμε ότι Micro Average Accuracy $>$ Macro Average Accuracy, με την απόκλιση να αυξάνεται για μικρότερες τιμές της παραμέτρου a. Αυτό υποδεικνύει ότι τα λάθη τείνουν να συμβαίνουν σε κείμενα με λιγότερες οντότητες, πράγμα που ρίχνει τον μέσο όρο του Macro Average Accuracy. Από την άλλη, για $a > 0.5$ βλέπουμε ότι Macro Average Accuracy $>$ Micro Average Accuracy, με την απόκλιση να αυξάνεται για μεγαλύτερες τιμές της παραμέτρου a. Αυτό υποδεικνύει ότι τα λάθη τείνουν να συμβαίνουν σε κείμενα με περισσότερες οντότητες, πράγμα που δεν επηρεάζει ιδιαίτερα τον μέσο όρο του Macro Average Accuracy. Αυτή η συμπεριφορά είναι αναμενόμενη καθώς:

  - για μικρά a δίνουμε βάρος στη σημασιολογική συνάφεια, που είναι πιο χρήσιμη όσο περισσότερες οντότητες υπάρχουν σε ένα κείμενο, καθώς ορίζουν καλύτερα έναν σημασιολογικό πυρήνα. Άρα, το σύστημα αποδίδει καλύτερα και κάνει λιγότερα λάθη σε κείμενα με μεγάλο πλήθος οντοτήτων, ενώ αποδίδει χειρότερα και κάνει περισσότερα λάθη σε κείμενα με λίγες και ασυσχέτιστες οντότητες.
  - για μεγάλα a δίνουμε βάρος στον συνδυασμό GKG resultScore και document similarity, που χρησιμεύει σε κείμενα όπου οι οντότητες είναι λίγες και ασυσχέτιστες βάσει WLM. Άρα, το σύστημα αποδίδει καλύτερα και κάνει λιγότερα λάθη σε κεί-

μενα με λίγες οντότητες, ενώ αποδίδει χειρότερα και κάνει περισσότερα λάθη σε κείμενα με πολλές και σημασιολογικά συναφείς οντότητες.

Κεφαλαιο 7

Επίλογος

Στο κεφάλαιο αυτό συγκεντρώνονται τα συμπεράσματα τα οποία εξάγονται από τα κεφάλαια που έχουν προηγηθεί και αριθμούνται στην ενότητα 7.1. Έπειτα, στην ενότητα 7.2, προτείνονται δυνατές επεκτάσεις που θα μπορούσαν να βελτιώσουν το σύστημα αποσαφήνισης, ή να το καταστήσουν καταλληλότερο για διαφορετικά πεδία εφαρμογής.

7.1 Συμπεράσματα

Από τη διαδικασία σχεδίασης, υλοποίησης και πειραματικής αξιολόγησης του συστήματος αποσαφήνισης, μπορούν να προκύψουν τα εξής:

- Το GKG API, ένα σχετικά νέο API τη στιγμή της συγγραφής αυτής της εργασίας, αποδίδει πολύ καλά ως ένα entity repository με σκοπό την αποσαφήνιση αμφίσημων αναφορών σε κείμενο, αν συνδυαστεί κατάλληλα με τα δεδομένα που είναι διαθέσιμα από την Wikipedia.

- Δεν υπάρχει η ανάγκη παραγωγής, διαχείρισης και χρήσης πολύπλοκης σημασιολογικής πληροφορίας, για να επιτευχθεί ένα κατά μέσο όρο ικανοποιητικό αποτέλεσμα αποσαφήνισης. Από τα πειραματικά αποτελέσματα του κεφαλαίου 6 φαίνεται ότι το σύστημα αποσαφήνισης που αναπτύχθηκε αποδίδει καλύτερα από το state-of-the-art του πεδίου, και αυτό χρησιμοποιώντας μόνο δεδομένα που μπορούν να ληφθούν από APIs. Ακόμα και τα τοπικά αποθηκευμένα δεδομένα, που χρησιμοποιήθηκαν αντί ορισμένων MediaWiki APIs, προέκυψαν μετά από μηδαμινή επεξεργασία.

- Το υπολογιστικά σύνθετο πρόβλημα της συλλογικής αποσαφήνισης μπορεί να λυθεί ικανοποιητικά από μια intuitive προσέγγιση, όπου αφαιρείται σε κάθε βήμα η οντότητα που φαίνεται να δίνει το χειρότερο αποτέλεσμα αποσαφήνισης. Αν και αυτή είναι μια προσέγγιση της οποίας η χρησιμότητα ενδεχομένως περιορίζεται στο συγκεκριμένο πεδίο εφαρμογής, μας επέτρεψε να λύσουμε ένα πολύπλοκο πρόβλημα με κατανάλωση λογικών υπολογιστικών πόρων.

7.2 Δυνατές επεκτάσεις

Το σύστημα αποσαφήνισης που αναπτύχθηκε αποδίδει ικανοποιητικά, αλλά είναι βέβαιο ότι μπορεί να βελτιωθεί και να επεκταθεί. Κάποιες από τις τροποποιήσεις που θα μπορούσαν να πραγματοποιηθούν είναι:

- *Υποστήριξη άλλων γλωσσών.* Το σύστημα στην παρούσα μορφή του υποστηρίζει μόνο την αγγλική γλώσσα, για λόγους που αναφέρθηκαν στην παράγραφο 2.3.1 και στην ενότητα 4.1. Ωστόσο, δεν υπάρχει κάποιος περιορισμός που να εμποδίζει τη λειτουργία του συστήματος σε άλλες γλώσσες. Μια γρήγορη εξέταση των πηγών πληροφορίας του συστήματος αποσαφήνισης φανερώνει ότι όλες υποστηρίζουν διάφορες γλώσσες. Συγκεκριμένα, το GKG API και τα MediaWiki APIs υποστηρίζουν πολλαπλές γλώσσες, μεταξύ των οποίων και την ελληνική γλώσσα. Ένα πρόβλημα ίσως να προέκυπτε από το σύστημα αναγνώρισης ονοματικών οντοτήτων σε κείμενο, γιατί τέτοια συστήματα έχουν υλοποιηθεί μόνο για τις πιο δημοφιλείς γλώσσες του κόσμου. Όμως, όπως έχει αναφερθεί σε αρκετές περιπτώσεις, η αναγνώριση οντοτήτων δεν είναι το πιο βασικό στάδιο κατά την αποσαφήνιση οντοτήτων.

- *Αξιοποίηση δομής κειμένου.* Το σύστημα αποσαφήνισης που υλοποιήθηκε αξιοποιεί τη δομή του κειμένου μόνο κατά το στάδιο αναγνώρισης ονοματικών οντοτήτων. Από αυτό το σημείο και μετά μας ενδιαφέρουν μόνο οι συμβολοσειρές των ονοματικών αναφορών, ενώ σε κάποιο βαθμό λαμβάνεται υπόψιν το κείμενο ως bag of words για τον υπολογισμό ενός βασικού document similarity measure. Εναλλακτικά, θα μπορούσε να αναλύεται συντακτικά το κείμενο με στόχο την εξαγωγή σχέσεων μεταξύ των οντοτήτων, και να χρησιμοποιούνται αυτές οι σχέσεις για να παραχθεί ένα πιο ποιοτικό αποτέλεσμα αποσαφήνισης.

- *Υποστήριξη μη αποσαφηνίσιμων αναφορών.* Το σύστημα, ως έχει, δεν κάνει καμία προσπάθεια για να αναγνωρίσει αναφορές που είναι μη αποσαφηνίσιμες, δηλαδή αναφορές για τις οποίες δεν υπάρχει αντίστοιχη οντότητα στη βάση γνώσης. Με άλλα λόγια, θεωρείται ότι η βάση γνώσης είναι πλήρης και ότι υπάρχει η αντίστοιχη οντότητα για κάθε αναφορά. Αυτή είναι μια απλή και αρκετά κοινή προσέγγιση, όπως αναφέρθηκε στην παράγραφο 2.3.5. Το σύστημα θα μπορούσε να επεκταθεί, ώστε να αναγνωρίζονται στο τελικό αποτέλεσμα αποσαφήνισης οι οντότητες που επιλέχθηκαν μόνο και μόνο επειδή κάποια οντότητα έπρεπε να επιλεγεί. Για παράδειγμα, μια λύση είναι να τεθεί ένα κάτω όριο μετρικής WLM μεταξύ των οντοτήτων του τελικού αποτελέσματος αποσαφήνισης. Έτσι, αν μια οντότητα δεν είχε σημασιολογική συγγένεια με καμία άλλη οντότητα στο κείμενο, τότε θα μπορούσε να σημειωθεί ως ενδεχομένως μη αποσαφηνίσιμη.

- *Διαδραστική αποσαφήνιση.* Το σύστημα αποσαφήνισης, στην παρούσα μορφή του, εκτελεί συλλογική αποσαφήνιση, χωρίς να αλληλεπιδρά με τον χρήστη μετά το στάδιο της παραμετροποίησης του συστήματος. Εναλλακτικά, θα μπορούσε να γίνεται αποσαφήνιση σε στάδια. Δηλαδή, καθώς ο χρήστης γράφει το κείμενο, θα μπορούσε να επισημαίνει τις ονοματικές αναφορές και το σύστημα να προτείνει υποψήφιες οντότητες on the go. Με αυτόν τον τρόπο, ο γράφος θα παραγόταν σταδιακά, και η αποσαφήνιση θα

έδινε καλύτερα αποτελέσματα, καθώς μετά από λίγες οντότητες θα είχε οριστεί πολύ καλά ο σημασιολογικός πυρήνας του κειμένου.

ΑΝΑΦΟΡΕΣ

[ABK+07] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. *DBpedia: A Nucleus for a Web of Open Data*, pages 722–735. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. URL: http://dx.doi.org/10.1007/978-3-540-76298-0_52, doi:10.1007/978-3-540-76298-0_52.

[Ben12] Benjamin Bengfort. A survey of stochastic and gazetteer based approaches for named entity recognition. 2012.

[BML13] Ilaria Bordino, Yelena Mejova, and Mounia Lalmas. Penguins in sweaters, or serendipitous entity search on user-generated content. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 109–118, New York, NY, USA, 2013. ACM. URL: http://doi.acm.org/10.1145/2505515.2505680, doi:10.1145/2505515.2505680.

[BP06] Razvan C Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Eacl*, volume 6, pages 9–16, 2006.

[BSV10] Krisztian Balog, Pavel Serdyukov, and Arjen P de Vries. Overview of the trec 2010 entity track. Technical report, DTIC Document, 2010.

[CJ11] Zheng Chen and Heng Ji. Collaborative ranking: A case study on entity linking. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 771–781, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. URL: http://dl.acm.org/citation.cfm?id=2145432.2145520.

[CTL+10] Zheng Chen, Suzanne Tamang, Adam Lee, Xiang Li, Wen-Pin Lin, Matthew Snover, Javier Artiles, Marissa Passantino, and Heng Ji. Cuny-blender tac-kbp2010. 2010.

[Cuc07] Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of EMNLP-CoNLL 2007*, page 708–716, June 2007. URL: https://www.microsoft.com/en-us/research/publication/large-scale-named-entity-disambiguation-based-on-wikipedia-data/.

[Cuc11] Silviu Cucerzan. Tac entity linking by performing full-document entity extraction and disambiguation. In *TAC*, 2011.

[CV07] R. L. Cilibrasi and P. M. B. Vitanyi. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):370–383, March 2007. doi:10.1109/TKDE.2007.48.

[CYC07] Tao Cheng, Xifeng Yan, and Kevin Chen-Chuan Chang. Entityrank: Searching entities directly and holistically. In *Proceedings of the 33rd International Conference on Very Large Data Bases*, VLDB '07, pages 387–398. VLDB

Endowment, 2007. URL: http://dl.acm.org/citation.cfm?id=1325851.1325898.

[DDCM12] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. Zencrowd: Leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 469–478, New York, NY, USA, 2012. ACM. URL: http://doi.acm.org/10.1145/2187836.2187900, doi:10.1145/2187836.2187900.

[DG06] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 233–240, New York, NY, USA, 2006. ACM. URL: http://doi.acm.org/10.1145/1143844.1143874, doi:10.1145/1143844.1143874.

[DIdV10] Gianluca Demartini, Tereza Iofciu, and Arjen P. de Vries. *Overview of the INEX 2009 Entity Ranking Track*, pages 254–264. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. URL: http://dx.doi.org/10.1007/978-3-642-14556-8_26, doi:10.1007/978-3-642-14556-8_26.

[DMR+10] Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 277–285, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL: http://dl.acm.org/citation.cfm?id=1873781.1873813.

[FS10] Paolo Ferragina and Ugo Scaiella. Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1625–1628, New York, NY, USA, 2010. ACM. URL: http://doi.acm.org/10.1145/1871437.1871689, doi:10.1145/1871437.1871689.

[GCK13] Stephen Guo, Ming-Wei Chang, and Emre Kiciman. To link or not to link? a study on end-to-end tweet entity linking. In *HLT-NAACL*, pages 1020–1030, 2013.

[GJ11] Swapna Gottipati and Jing Jiang. Linking entities to a knowledge base with query expansion. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 804–813, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. URL: http://dl.acm.org/citation.cfm?id=2145432.2145523.

[GLG+13] Abhishek Gattani, Digvijay S. Lamba, Nikesh Garera, Mitul Tiwari, Xiaoyong Chai, Sanjib Das, Sri Subramaniam, Anand Rajaraman, Venky Harinarayan, and AnHai Doan. Entity extraction, linking, classification, and tagging for social media: A wikipedia-based approach. *Proc. VLDB Endow.*, 6(11):1126–1137, August 2013. URL: http://dx.doi.org/10.14778/2536222.2536237, doi:10.14778/2536222.2536237.

[GXCL09] Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. Named entity recognition in query. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 267–274, New York, NY, USA, 2009. ACM. URL: http://doi.acm.org/10.1145/1571941.1571989, doi:10.1145/1571941.1571989.

[Hof15] Johannes Hoffart. *Discovering and disambiguating named entities in text*. PhD thesis, Universität des Saarlandes, Postfach 151141, 66041 Saarbrücken, 2015. URL: http://scidok.sulb.uni-saarland.de/volltexte/2015/6022.

[HRN+13] Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R. Curran. Evaluating entity linking with wikipedia. *Artificial Intelligence*, 194:130 – 150, 2013. URL: http://www.sciencedirect.com/science/article/pii/S0004370212000446, doi:http://dx.doi.org/10.1016/j.artint.2012.04.005.

[HS11] Xianpei Han and Le Sun. A generative entity-mention model for linking entities with knowledge base. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 945–954, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. URL: http://dl.acm.org/citation.cfm?id=2002472.2002592.

[HS12] Xianpei Han and Le Sun. An entity-topic model for entity linking. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 105–115, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL: http://dl.acm.org/citation.cfm?id=2390948.2390962.

[HSZ11] Xianpei Han, Le Sun, and Jun Zhao. Collective entity linking in web text: A graph-based method. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 765–774, New York, NY, USA, 2011. ACM. URL: http://doi.acm.org/10.1145/2009916.2010019, doi:10.1145/2009916.2010019.

[HYB+11] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 782–792, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. URL: http://dl.acm.org/citation.cfm?id=2145432.2145521.

[HZ09] Xianpei Han and Jun Zhao. Nlpr_kbp in tac 2009 kbp track: A two-stage method to entity linking. In *TAC*. Citeseer, 2009.

[KSRC09] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th*

ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09, pages 457–466, New York, NY, USA, 2009. ACM. URL: http://doi.acm.org/10.1145/1557019.1557073, doi:10.1145/1557019.1557073.

[LDP10] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. Personalized news recommendation based on click behavior. In *Proceedings of the 15th International Conference on Intelligent User Interfaces*, IUI '10, pages 31–40, New York, NY, USA, 2010. ACM. URL: http://doi.acm.org/10.1145/1719970.1719976, doi:10.1145/1719970.1719976.

[LLW+13] Xiaohua Liu, Yitong Li, Haocheng Wu, Ming Zhou, Furu Wei, and Yi Lu. Entity linking for tweets. In *ACL (1)*, pages 1304–1311, 2013.

[LME12] Thomas Lin, Mausam, and Oren Etzioni. Entity linking at web scale. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, AKBC-WEKEX '12, pages 84–88, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL: http://dl.acm.org/citation.cfm?id=2391200.2391216.

[LMN+10] John Lehmann, Sean Monahan, Luke Nezda, Arnold Jung, and Ying Shi. Lcc approaches to knowledge base population at tac 2010. In *TAC*, 2010.

[LSC10] Girija Limaye, Sunita Sarawagi, and Soumen Chakrabarti. Annotating and searching web tables using entities, types and relationships. *Proc. VLDB Endow.*, 3(1-2):1338–1347, September 2010. URL: http://dx.doi.org/10.14778/1920841.1921005, doi:10.14778/1920841.1921005.

[LWH+13] Yang Li, Chi Wang, Fangqiu Han, Jiawei Han, Dan Roth, and Xifeng Yan. Mining evidences for named entity disambiguation. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 1070–1078, New York, NY, USA, 2013. ACM. URL: http://doi.acm.org/10.1145/2487575.2487681, doi:10.1145/2487575.2487681.

[McN10] Paul McNamee. Hltcoe efforts in entity linking at tac kbp 2010. In *TAC*, 2010.

[MLN+11] Sean Monahan, John Lehmann, Timothy Nyberg, Jesse Plymale, and Arnold Jung. Cross-lingual cross-document coreference with entity linking. In *TAC*, 2011.

[MM10] Matthew Michelson and Sofus A. Macskassy. Discovering users' topics of interest on twitter: A first look. In *Proceedings of the Fourth Workshop on Analytics for Noisy Unstructured Text Data*, AND '10, pages 73–80, New York, NY, USA, 2010. ACM. URL: http://doi.acm.org/10.1145/1871840.1871852, doi:10.1145/1871840.1871852.

[MSB+14] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014. URL: http://www.aclweb.org/anthology/P/P14/P14-5010.

[MW08] David Milne and Ian H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 509–518, New York, NY, USA, 2008. ACM. URL: http://doi.acm.org/10.1145/1458082.1458150, doi:10.1145/1458082.1458150.

[NS07] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[PMS09] Owen Phelan, Kevin McCarthy, and Barry Smyth. Using twitter to recommend real-time topical news. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 385–388, New York, NY, USA, 2009. ACM. URL: http://doi.acm.org/10.1145/1639714.1639794, doi:10.1145/1639714.1639794.

[PP11] Anja Pilz and Gerhard Paaß. From names to entities using thematic context distance. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 857–866, New York, NY, USA, 2011. ACM. URL: http://doi.acm.org/10.1145/2063576.2063700, doi:10.1145/2063576.2063700.

[Rad14] William Edward John Radford. Linking named entities to wikipedia. 2014.

[RJCC14] Dan Roth, Heng Ji, Ming-Wei Chang, and Taylor Cassidy. Wikification and beyond: The challenges of entity and concept grounding. In *ACL (Tutorial Abstracts)*, page 7, 2014.

[RRDA11] Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1375–1384, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. URL: http://dl.acm.org/citation.cfm?id=2002472.2002642.

[SL01] Aixin Sun and Ee-Peng Lim. Hierarchical text classification and evaluation. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 521–528, 2001. doi:10.1109/ICDM.2001.989560.

[ŠM09] Tadej Štajner and Dunja Mladenić. *Entity Resolution in Texts Using Statistical Learning and Ontologies*, pages 91–104. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. URL: http://dx.doi.org/10.1007/978-3-642-10871-6_7, doi:10.1007/978-3-642-10871-6_7.

[SWH15] W. Shen, J. Wang, and J. Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460, Feb 2015. doi:10.1109/TKDE.2014.2327028.

[SWLW12a] Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. Liege:: Link entities in web lists with knowledge base. In *Proceedings of the 18th ACM SIGKDD International*

Conference on Knowledge Discovery and Data Mining, KDD '12, pages 1424–1432, New York, NY, USA, 2012. ACM. URL: http://doi.acm.org/10.1145/2339530.2339753, doi:10.1145/2339530.2339753.

[SWLW12b] Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. Linden: Linking named entities with knowledge base via semantic knowledge. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 449–458, New York, NY, USA, 2012. ACM. URL: http://doi.acm.org/10.1145/2187836.2187898, doi:10.1145/2187836.2187898.

[SWLW13] Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. Linking named entities in tweets with knowledge base via user interest modeling. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 68–76, New York, NY, USA, 2013. ACM. URL: http://doi.acm.org/10.1145/2487575.2487686, doi:10.1145/2487575.2487686.

[VBR+10] Vasudeva Varma, Praveen Bysani, Kranthi Reddy, Vijay Bharath Reddy, Sudheer Kovelamudi, Srikanth Reddy Vaddepally, Radheshyam Nanduri, N Kiran Kumar, Santhosh Gsk, and Prasad Pingali. Iiit hyderabad in guided summarization and knowledge base population. In *TAC*, 2010.

[WLJH10] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. Twitterrank: Finding topic-sensitive influential twitterers. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 261–270, New York, NY, USA, 2010. ACM. URL: http://doi.acm.org/10.1145/1718487.1718520, doi:10.1145/1718487.1718520.

[WMKF12] Chris Welty, J. William Murdock, Aditya Kalyanpur, and James Fan. *A Comparison of Hard Filters and Soft Evidence for Answer Typing in Watson*, pages 243–256. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. URL: http://dx.doi.org/10.1007/978-3-642-35173-0_16, doi:10.1007/978-3-642-35173-0_16.

[WPR] Precision and recall. https://en.wikipedia.org/wiki/Precision_and_recall. Accessed: 21-06-2017.

[ZLHZ10] Zhicheng Zheng, Fangtao Li, Minlie Huang, and Xiaoyan Zhu. Learning to link entities with knowledge base. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 483–491, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL: http://dl.acm.org/citation.cfm?id=1857999.1858071.

[ZSST11] Wei Zhang, Yan Chuan Sim, Jian Su, and Chew Lim Tan. Entity linking with effective acronym expansion, instance selection, and topic modeling. In *IJCAI*, volume 2011, pages 1909–1914, 2011.

[ZSTW10] Wei Zhang, Jian Su, Chew Lim Tan, and Wen Ting Wang. Entity linking leveraging: Automatically generated annotation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1290–1298, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL: `http://dl.acm.org/citation.cfm?id=1873781.1873926`.

[ZTS+11] Wei Zhang, Chew Lim Tan, Jian Su, Bin Chen, Wenting Wang, Zhiqiang Toh, Yanchuan Sim, Yunbo Cao, and Chin-Yew Lin. I2r-nus-msra at tac 2011: Entity linking. In *TAC*, 2011.

[ZTSS10] Wei Zhang, Chew Lim Tan, Yan Chuan Sim, and Jian Su. Nus-i2r: Learning a combined system for entity linking. In *TAC*, 2010.

Παραρτημα

Σε αυτό το παράρτημα παρατίθεται το dataset με τα 300 μικρά κείμενα που χρησιμοποιήθηκε για μέρος της αξιολόγησης του συστήματος αποσαφήνισης. Πάνω στα κείμενα σημειώνονται τα ground truth entities που προέκυψαν με ανθρώπινη επίβλεψη. Αξίζει να σημειωθεί ότι σε κάποιες περιπτώσεις μπορεί να θεωρηθεί ορθή η αντιστοίχιση μιας ονοματικής αναφοράς με περισσότερες από μία οντότητες, αλλά η φύση του dataset επιβάλλει την επιλογή της καταλληλότερης οντότητας. Για παράδειγμα, στο κείμενο 34, που αποτελεί το παράδειγμα της ενότητας 1.2, αντιστοιχούμε στην αναφορά «Oscar» τη γενική οντότητα των βραβείων Όσκαρ, αλλά θα μπορούσαμε να αντιστοιχίσουμε και το Όσκαρ Α΄ Ανδρικού Ρόλου. Επίσης, στο κείμενο 13 υπάρχει η αναφορά «Euro 2004», που αποσαφηνίζεται ως η διοργάνωση της συγκεκριμένης χρονιάς. Ωστόσο, δύσκολα κανείς θα μπορούσε να ισχυριστεί ότι η αντιστοίχιση της εν λόγω αναφοράς στη γενική διοργάνωση του Euro είναι λανθασμένη. Αυτά τα σημεία απόφασης πηγάζουν από το γεγονός ότι ο GKG και η Wikipedia διατηρούν καταχωρήσεις τόσο για τις γενικές οντότητες, για παράδειγμα UEFA European Championship, όσο και για ειδικές οντότητες, για παράδειγμα UEFA Euro 2004. Αυτό το χαρακτηριστικό του fine granularity κάνει πιο δύσκολη την παραγωγή του ground truth, αλλά και τη μετέπειτα αξιολόγηση του συστήματος αποσαφήνισης. Εδώ ακολουθήθηκε η γενική αρχή ότι η εκάστοτε οντότητα προκύπτει από την αίσθηση που δίνει το ίδιο το κείμενο. Στην περίπτωση του κειμένου 34, δεν αναφέρεται το είδος του Όσκαρ, άρα κάνουμε αντιστοίχιση στη γενική οντότητα. Αντιθέτως, στην περίπτωση του κειμένου 18 αναφέρεται η χρονιά του θεσμού ως μέρος της ονοματικής αναφοράς, άρα κάνουμε αντιστοίχιση με το Euro του 2004.

1. **Lincoln** and **Reagan** are former **U.S.A.** presidents.
2. **Lincoln** and **Chevrolet** are **American** car brands.
3. **Angelina**, her father **Jon**, and her partner **Brad** never played together in the same movie.
4. **Jon** was the host of **The Daily Show** from 1999 until 2015, but then **Comedy Central** replaced him with **Noah**.
5. **The Red Devils** were defeated by **The Gunners** in a packed **Emirates**.
6. **Abu Dhabi** is the capital of the **Emirates**. However, the most iconic building of the country, the **Burj khalifa**, is located in **Dubai**.
7. **Adidas** and **Emirates** are **Madrid**'s main sponsors.
8. **Madrid** is the most populated city and capital of **Spain**, with **Barcelona** having the second largest population.
9. **Rubio** played for **Barcelona** before getting signed with the **Minnesota Wolves**.
10. **Felix** and **Marzia** are one of the hottest couples on **YouTube**.
11. **Felix**'s second marriage was with the famous **Mexican** composer **Lara**.

12. **Alexis**, **Merkel** and **Schauble** discussed about the future of the **Greek** economy.
13. **Greece** defeats **Portugal** to claim **Euro 2004** trophy.
14. **Sandra** played in the movie **Speed** with **Keanu**.
15. The popular comedy duo of **Key** and **Peele** stars in **Keanu**.
16. After the death of **Steve**, the former CEO of **Apple**, his commencement speech at **Stanford** was watched thousands of times.
17. **Family Feud** saw significantly improved ratings with **Steve** hosting the show.
18. The impact of **Gödel**'s and **Turing**'s breakthroughs in the 1930s is best understood against the background of the mathematical ambitions definitively expressed by **Hilbert** in the 1920s.
19. In 2006, the recipient of the **Turing Prize** was **IBM**'s **Frances Allen**, and that was the first time **ACM** bestowed the highest distinction in computer science upon a woman.
20. **The Tonight Show** will not be the same if **Jimmy** leaves.
21. Prior to his **Senate** re-election run, **Brock** was among those considered to replace **Rockefeller** as **Ford**'s running mate in the 1976 election.
22. **Brock** defeated **Hunt** on his **UFC** comeback.
23. **Kurt** and **Krist** are the founders of **Nirvana**.
24. In **Forrest Gump**, **Kurt** provided the voice over for **Elvis**.
25. **UMM** vehicles became known for their durability, especially when in the **Paris-Dakar** rally the team was able to finish with all the cars that started.
26. **Umm** was awarded honors by **King Farouk**.
27. **Tiger** was lost in the woods when he got divorced from **Elin**.
28. **Tiger** was drafted in the second round by the **Maple Leafs** in the **1974 NHL Amateur Draft**.
29. **Armstrong** was the first man to walk on the **Moon**.
30. **Armstrong** was a great performer and influenced jazz like no other.
31. **Blaugrana** defeated **Los Blancos** last night in **Camp Nou**. **Leo** and **Cristiano** stole the show.
32. The **Brazilians** won the **2002 World Cup**, with **Ronaldo**, **Rivaldo** and **Ronaldinho** playing forward.
33. On February 14, 2013, **Oscar** fatally shot **Reeva** in his **Pretoria** home.

34. **Leo** finally won an **Oscar** this year, for his tremendous performance in **The Revenant**.
35. **Oscar** wrote **De Profundis** while he was in prison.
36. **Hunter** and **Stephanie** are **WWE**'s resident power-couple.
37. In 1983, after her physical recovery from the accident which killed her mother, **Stephanie** started an apprentice programme at **Dior** under the direction of head designer **Marc Bohan**.
38. **JFK** and **Nixon** took part in the first televised presidential debate in 1960.
39. **Obama** welcomed **Merkel** upon her arrival at **JFK**.
40. **JFK** was the most successful of three films **Stone** made about **American** presidents.
41. Despite featuring some of the most prominent musicians of their decade — like **Sinatra**, **Dylan**, **Joel**, and **Santana** — **Columbia** was acquired by **Sony** in the 1980s.
42. Three of the greatest guitarists started their career in a single band: **Clapton**, **Beck**, and **Page**.
43. **Brin** and **Page** founded **Google**.
44. **Page** plays in **Nolan**'s **Inception**, alongside actors like **DiCaprio**, **Hardy** and **Gordon-Levitt**.
45. **Hardy** is best known for his essay, **A Mathematician's Apology**, in which he explains the essence of mathematics in layman's terms.
46. **The Hardy Boys** were created by **American** writer **Edward Stratemeyer**.
47. **Becks** and **Posh** got engaged in 1998.
48. **Joanne** met **Daniel**, **Emma** and **Rupert** on the set of **Harry Potter**.
49. **DNA**'s molecular structure was identified by **Watson** and **Crick** in 1953.
50. **Mars**, **Galaxy**, and **Bounty** are all chocolate.
51. **Mars**, **Migos**, **Future** & more to perform at **2017 BET Awards**.
52. **Mars** has co-written songs for **Crashdïet** and **Machina**.
53. **Trump** signed a law that makes manned missions to **Mars NASA**'s main goal in the decades to come.
54. **May** invited **Trump** to **Britain** seven days after his inauguration when she became the first foreign leader to visit him in the **White House**.
55. **Trump** grew his channel into one of the most popular **Hearthstone** streams on the game streaming website **Twitch**.

56. **Milky Way** is a candy bar manufactured and distributed by **Mars Confectionery**.

57. The **Milky Way** is expected to collide with **Andromeda** in approximately 4.5 billion years.

58. It was quite a surprise for a lot of people that **Donald** won over **Hillary**. Many think that if **Bernie** had been in her place, things might have been different.

59. The **Amazon**, the **Nile** and the **Yangtze** are the longest rivers in the world.

60. **Bezos** is the chairman of **Amazon**.

61. **Bezos** was discovered by **Thymios Karakatsanis**. In his early days, he worked on stage with **Vougiouklaki**, and later on he played on successful TV series on **Antenna** and **Mega**.

62. According to calculations based on data from **Credit Suisse** and **Forbes**, **Oxfam** reported that only eight individuals (**Gates**, **Ortega**, **Buffett**, **Slim**, **Bezos**, **Zuckerberg**, **Ellison** and **Bloomberg**) own as much wealth as the poorest half of the world's population.

63. **Slim** signed with **Dre**'s **Aftermath Entertainment** in the late 90s, and together they changed the landscape of rap music.

64. With various air forces, the **Jaguar** was used in numerous conflicts and military operations in **Mauritania**, **Chad**, **Iraq**, **Bosnia**, and **Pakistan**, as well as providing a ready nuclear delivery platform for **Britain**, **France**, and **India** throughout the latter half of the **Cold War** and beyond.

65. **Jaguars** can be found in the **Americas**, their range extending from the **Southwestern United States** and **Mexico** across much of **Central America** and south to **Paraguay** and northern **Argentina**.

66. **Jaguar** has been owned by the **Indian** company **Tata Motors** since 2008.

67. **Margot** and **Will** had worked together before **Suicide Squad**, so the chemistry was already there.

68. On August 23, 2015, **Adult Swim** offered **Smith** the opportunity to perform a song to commemorate the series finale of **Aqua Teen Hunger Force**.

69. **Smith** studied social philosophy at the **University of Glasgow** and at **Balliol College**, **Oxford**.

70. **Grenoble** has been twinned with **Oxford** since 1977.

71. **Wiles** proved the last of many conjectures written by **Fermat** in the margins of **Arithmetica**.

72. **Bosch** and **Sharp** are both home appliances producing companies.

73. **Sebastian** reigns supreme in **Suzuka**.
74. **Floyd** defeated **Manny** at the **MGM Arena** in **Las Vegas**, **Nevada**.
75. **Jennifer**, **Courteney**, **Lisa**, **Matt**, **Matthew** and **David** followed separate career paths after the finale of **Friends**.
76. The list of **Victoria's Secret** angels includes **Adriana**, **Alessandra**, **Candice**, **Lily** and **Jasmine**.
77. **Zayn** split from **1D** in March 2015 and signed a solo recording contract with **RCA**.
78. Last night's **NXT** featured a match between **Zayn** and **Corey**.
79. **Frank** and **Pepys** are famous diarists.
80. **Hearthstone** poses a serious threat to **Magic Online**, mainly because of its polished interface.
81. **Nike** and **Adidas** are sportswear manufacturers.
82. Since 1884, **Nike** has been prominently displayed at the **Louvre**.
83. **Elon** is the CEO of **Tesla Motors**.
84. A common argument is about who contributed more to science: **Tesla** or **Edison**.
85. **Socrates** was a philosopher who lived in **Athens**.
86. **Sokratis** has played for **AEK**, **Milan**, **Werder** and **Dortmund**.
87. **Cameron**'s first big screen appearance was on **The Mask**.
88. **Cameron** served as prime minister of the **UK** from 2010 to 2016.
89. **Cameron** is the director of **Titanic**.
90. **Titanic** started her maiden voyage from **Southampton** on the 10th of April, 1912.
91. **Moore** co-founded **Intel**.
92. **Moore** played the role of **Simon Templar** in **The Saint** between 1962 and 1969.
93. **Moore** starred in **Striptease**.
94. Can she be **Moore** perfect? **Julianne**, 56, looks ravishing as she shows off her svelte legs in chic mini-dress at **Cannes Festival** yacht party.
95. On May 20, 2013, parts of **Moore** and neighboring **Newcastle** and southern **Oklahoma City** were affected by a violent tornado.
96. **Newcastle** have won six **FA Cups**.

97. **Valentino** is racing for **Yamaha**.
98. **Valentino**, **Gucci** and **Armani** are **Italian** fashion labels.
99. **An Idiot Abroad** is a **British** travel documentary, with a comedic twist, starring **Ricky**, **Karl** and **Stephen**. It was broadcast on **Sky 1**.
100. When **Jackson**'s **Invincible** was released on October 30, 2001, it was already being touted as the most expensive album ever made, with reports of **Sony**'s price tag reaching $30 million.
101. **Pulp Fiction** was the first of several films in which **Jackson** performed under the direction of **Tarantino**.
102. Before being elected to the presidency, **Jackson** served in **Congress** and gained fame as a general in the **United States Army**.
103. **Dallas** is a mid-carder on **Raw**, while **Wyatt**, his brother, is more successful on **Smackdown**.
104. **TI**'s headquarters are located in **Dallas**, **Texas**.
105. **TI** has collaborated with high-profile artists, like **Rihanna** and **Timberlake**.
106. **L.B. Johnson** occupied the **White House** from 1963 to 1969.
107. **Johnson** started from the **WWE**, but then chose to focus on his acting career.
108. **Johnson** played point guard for the **Lakers**.
109. **Neutrogena** and **Aveeno** are both subsidiaries of **Johnson**.
110. **Fearless** won **Taylor** multiple album of the year awards.
111. **Taylor** had the lead role in **Mankiewicz**'s **Cleopatra**.
112. **Taylor** had a decorated career in the **United States Army**, before he became a president.
113. **APC**, a division of **Schneider**, produces quality power strips.
114. **Schneider** appears in **Grown Ups**. **Sandler** and **Rock** play in that movie as well.
115. **Robben** has been a key element of **Oranje**'s offensive in the past few years.
116. **Robben** is located 6.9 km west of the coast of **Bloubergstrand**.
117. Even though he is mostly associated with **Fender**'s products, **Hendrix** played several **Gibson** guitars throughout his career.
118. **Gibson** not only starred in **Braveheart**, but he also produced it and directed it.

119. In his professional boxing career debut, **Tyson** defeated his opponent in less than two minutes.

120. **Tyson** and his friend **Nye**, the science guy, are known for intriguing the public about scientific matters.

121. **Brown** is best known for his 2003 bestselling novel **The Da Vinci Code**.

122. On a single night—October 24, 1962—**Brown** recorded a live concert album at the **Apollo Theater** in **Harlem**.

123. In 2009, **Brown** received significant media attention after pleading guilty to felony assault of his then girlfriend, singer **Rihanna**.

124. **Madagascar** is a computer-animated franchise produced by **DreamWorks Animation**.

125. **Madagascar** is located in the **Indian Ocean**, off the coast of **Southeast Africa**.

126. **Mercury** is the only metallic element that is liquid at standard conditions for temperature and pressure.

127. Starting from the **Sun** and moving outwards, one sees **Mercury**, then **Venus**, then **Earth**.

128. **Ford** tried to bridge the gap between their main vehicle line and their luxurious **Lincolns** by launching **Mercury**, an entry-level premium car brand.

129. **Mercury**, who was the main vocalist in **Queen**, became known for his flamboyant personality and impressive vocal range.

130. **Sun** was acquired by **Oracle** on January 27, 2010 for 7.4 billion dollars.

131. **Arizona**'s biggest attraction is the **Grand Canyon**, while **Oklahoma** is famous for the longest drivable stretch of **Route 66**.

132. **Arizona** and **Oklahoma** are two of the ships that sank in **Pearl Harbor** during the events of **World War II**.

133. **Sanders** is the longest serving independent politician in the history of the **United States Congress**.

134. Even after selling the **KFC** corporation, **Sanders** remained the company's symbol and brand ambassador.

135. **Martin** is a **Puerto Rican** pop singer, who is also known for his humanitarian work.

136. **Martin** is the creative genius behind **Game of Thrones**.

137. On October 14, 1964, **Martin** received the **Nobel Peace Prize** for combating racial inequality through nonviolent resistance.

138. **James Bond** always drives a **Martin** in his movies.

139. **Martin** and **Leonardo** have worked together in many films, like **The Departed** and **Shutter Island**.

140. **Qatar** CEO **Akbar al Baker** was attending a major **IATA** meeting in **Cancun**, when he was informed about the imminent crisis.

141. **Qatar**: **UAE** and **Saudi Arabia** step up pressure in diplomatic crisis.

142. **Fotakis** spent his early years playing for **PAOK**.

143. **Fotakis**, who received his diploma and PhD from the **University of Patra**, is currently an assistant professor at **NTUA**.

144. **Dorociński** won the **Zbigniew Cybulski Award** for his role in **Pitbull**.

145. On 4 February 2007, **Pitbull** played his first **Primeira Liga** match with **Académica**, against **Naval**.

146. In 2013, **Pitbull** released a diss track towards **Lil Wayne** called "Welcome 2 **Dade County**" shortly after **Lil Wayne**'s rant on the **Miami Heat**.

147. **Casablanca** was part of the film colorization controversy of the 1980s, when a colorized version aired on the television network **WTBS**.

148. **Casablanca** is home to the **Hassan II Mosque**, designed by the **French** architect **Michel Pinseau**.

149. On 27 June 2007, **Valencia** netted **Ecuador**'s first goal of their **2007 Copa América** campaign, however they eventually lost the match 3–2 to **Chile** and finished the tournament bottom of their group.

150. **Valencia** is a successful club, considering the competitive level of **La Liga**.

151. **Valencia** is situated on the banks of the **Turia**, on the east coast of **Iberia**.

152. On November 9, 2007 an **Iberia Airbus A340-600**, registration EC-JOH, was badly damaged at **Quito**, **Ecuador** after sliding off the runway at **Old Mariscal Sucre International Airport**.

153. **Carter** plays for **Racing 92** and played for **New Zealand**'s national team, the **All Blacks**.

154. **Carter** discovered **Tut**'s grave in the **Kings Valley**.

155. **Collision Course** is the fifth installment of the **Ice Age** film series.

156. In 2004, **Carter** and **Linkin Park** released **Collision Course**, a collaborative remix album.

157. In 1952, **Carter** began an association with the **US Navy**'s fledgling nuclear submarine program, then led by Captain **Hyman G. Rickover**.

158. **Venus** and **Serena** are considered to be two of the top female tennis players.

159. **Venus** was directed by **Michell** and written by **Kureishi**.

160. **Venus** was discovered on the island of **Milos** in the **Aegean**.

161. Working alongside **Roger Ailes**, **Dan Cooper** became one of the co-founders of **Fox News Channel** in 1996.

162. The **FBI** maintained an active investigation for 45 years following the hijacking of a **Northwest Orient Airlines Boeing 727** by a man called **Dan Cooper**.

163. **Cooper** received the **Academy Award for Best Actor** for his roles in **Sergeant York** and **High Noon**.

164. **Cooper** was one of the seven original astronauts in **Project Mercury**, the first manned **USA** space program.

165. In 2014, **Cooper** co-produced and starred as **United States Navy SEAL** sniper **Chris Kyle** in **American Sniper**—a biographical war drama directed by **Clint Eastwood**.

166. The **Hollywood Vampires**, consisting of **Cooper**, **Depp** and **Perry**, performed at the **58th Grammy Awards** on February 15, 2016.

167. **Perry** got married to **Brand** in **Rajasthan**.

168. **Washington** is the capital of **America**.

169. **Mount Rushmore** features 60-foot sculptures of the heads of **Washington**, **Jefferson**, **Roosevelt** and **Lincoln**.

170. **Washington** played the role of a slave in **Django**.

171. For his role in **Malcolm X**, **Washington** won the **New York Film Critics Circle Award for Best Actor** and was nominated for an **Academy Award for Best Actor**.

172. On March 26, 1964, **Malcolm X** met **Martin Luther King Jr.** for the first and only time—and only long enough for photographs to be taken—in **Washington D.C.**, as both men attended the **Senate**'s debate on the **Civil Rights Bill**.

173. **Lee**, headquartered in **Merriam**, **Kansas**, is owned by **VF Corp.**, the largest apparel company in the world.

174. **Inside Man** was one of **Lee**'s most commercially successful pieces of work.

175. **Fist of Fury** is the first of only three movies in which **Lee** and **Chan** played together.

176. Thirty years after playing **Francisco Scaramanga** in **The Man with the Golden Gun**, **Lee** provided the voice of **Scaramanga** in the video game **GoldenEye: Rogue Agent**.

177. **Lee** is the director of **W3C**.

178. When **Virginia** declared its secession from the **Union** in April 1861, **Lee** chose to follow his home state, despite his personal desire for the country to remain intact and an offer of a senior **Union** command.

179. The **Hulk** was created by **Lee** and **Kirby** in the early sixties.

180. **Hulk** is currently signed with **SIPG**, and also plays for **Brazil**.

181. **Hulk** was removed from the **WWE Network** after his racist remarks were published.

182. In 2004 and 2005, **Jane Seymour** made six guest appearances in the **WB Network** series **Smallville**.

183. **Jane Seymour** was not educated as highly as King **Henry**'s previous wives, **Catherine of Aragon** and **Anne Boleyn**.

184. **One Hundred Years of Solitude**, considered **Márquez**'s magnum opus, remains widely acclaimed and is recognized as one of the most significant works in **Latin American** literature.

185. While **Gomez**'s first three albums had been self-produced, the band decided to work with **Tchad Blake** as producer for their fourth record.

186. On 26 May 2009, **Gomez** was eventually transferred to **Bayern** for a **Bundesliga** record transfer fee, signing a four-year contract.

187. At 16 years of age, **Gomez** was signed to a recording contract with the **Hollywood Records** label, which had already signed both **Cyrus** and **Lovato**.

188. On September 20, 2009, **Obama** appeared on all major news programs except **Fox**, a snub partially in response to remarks about the president by commentators **Glenn Beck** and **Sean Hannity**, and **Fox** coverage of **Obama**'s health-care proposal.

189. **Fox** began dating actor **Austin Green** in 2004, after meeting on the set of **Hope & Faith**.

190. Prior to **Foreigner**, **Mick Jones** was in the band **Spooky Tooth**.

191. **Mick Jones**' lack of punctuality played a major role in his dismissal from **The Clash**.

192. **Brian Wilson** has played in **MLB** for the **Giants** and the **Dodgers**.

193. After signing with **Capitol Records** in 1962, **Brian Wilson** wrote or co-wrote more than two dozen hits for the **Beach Boys**.

194. The **Rio bridge**, that crosses the **Gulf of Corinth**, was inaugurated on 7 August 2004, a week before the opening of the **2004 Summer Olympics** in **Athens**.

195. Many tourists come to **Rio** to visit **Christ the Redeemer** and the world famous **Copacabana** beach.

196. **Rio**'s theme song was performed by **Taio Cruz**.

197. In March 2007, **Edge** appeared alongside **Randy Orton**, **John Cena**, and **Bobby Lashley** on **Deal or No Deal**.

198. In 1976, at **Mount Temple Comprehensive School Edge** formed a band with his fellow students and elder brother **Dik** that would evolve into **U2**.

199. **Chrome**, **Firefox** and **Edge** are popular internet browsers.

200. In 2010, **James Stewart** received a **Logie Award** nomination for his role in **Packed to the Rafters**.

201. **James Stewart** had a noted military career and was a **World War II** and **Vietnam War** veteran.

202. After the much-publicized turmoil, **Michelle Williams**, alongside backup dancer **Farrah Franklin**, officially joined **Destiny's Child** in early 2000, replacing **LeToya Luckett** and **LaTavia Roberson** without notice.

203. **Michelle Williams** began dating **Australian** actor **Heath Ledger**, her **Brokeback Mountain** co-star, in 2004 after meeting on the set of the film.

204. For his artwork, **Steve McQueen** has received the **Turner Prize**, the highest award given to a **British** visual artist.

205. After **Charles Manson** incited the murder of five people, including **Steve McQueen**'s friends **Sharon Tate** and **Jay Sebring** at **Tate**'s home on August 9, 1969, it was reported **McQueen** was a potential target of the killers.

206. After **Columbia** dropped **Kate Hudson**, **Angelica Cob-Baehler**, then a publicity executive at the label, brought her demos to **Virgin** chairman **Jason Flom**.

207. In 2013, **Kate Hudson** began a partnership with online fashion retailer **JustFab** to launch her own line of workout clothes and active wear called **Fabletics**.

208. In August 2016, **Woolworths Holdings** announced that the **David Jones** headquarters will be moved from **Sydney** to **Richmond**, **Victoria**, owing to a combination of factors including rental lease fees and the market position for retail/fashion, as well as a lure from the **Victorian Government**.

209. In October 1990, a decade after his divorce from **Angie**, **David Jones** and **Somali** supermodel **Iman** were introduced by a mutual friend.

210. **David Jones** is best known for being a member of **The Monkees**.

211. **Beethoven** is a light-hearted family movie about a **St. Bernard**, in which **Joseph Gordon-Levitt** made his big screen debut.

212. In March 1787 **Beethoven** travelled to **Vienna** for the first time, in the hope of studying with **Mozart**.

213. After **Cleopatra**'s reign, **Ptolemaic Egypt** became a province of the recently established **Roman Empire**.

214. **Williams** and **Hugo** make up the record production duo **The Neptunes**, producing soul, hip hop and R&B music.

215. After his family moved to **Marin County**, **Williams** began his career doing stand-up comedy shows in the **San Francisco Bay Area** in the mid-1970s. His first performance took place at the **Holy City Zoo**, a comedy club in **San Francisco**, where he worked his way up from tending bar to getting on stage.

216. **Santa Clara** coach **Dick Davey** realized the potential of **Nash** and recruited him for the college team.

217. **Daskalakis**, now a professor at **MIT**, gained international recognition for his work on **Nash** equilibriums.

218. Considering factors such as **Bolt**'s position, acceleration and velocity in comparison with second-place-finisher **Richard Thompson**, the **University of Oslo** team estimated that **Bolt** could have finished in 9.55±0.04 s had he not slowed to celebrate before the finishing line.

219. **Travolta** and **Cyrus** are the main voice actors in **Bolt**.

220. **Hamilton** got his first **Formula One** win at **Circuit Gilles Villeneuve** in **Montreal**.

221. Born and raised in **Detroit**, **Ross** rose to fame as the lead singer of the vocal group **The Supremes**, which, during the 1960s, became **Motown**'s most successful act, and is to this day the **United States**' most successful vocal group, as well as one of the world's best-selling girl groups of all time. In 1994, **The Supremes** were recognized with a star on the **Hollywood Walk of Fame** at 7060 **Hollywood Blvd**.

222. **Ross** dedicated the first episode of the second season of **The Joy of Painting** to **Bill Alexander**.

223. **Ross** previously says that if **Birdman** could burn good dudes like **Khaled** and **Wayne**, then he already knew how he felt about him.

224. The **National Road** reached **Columbus** from **Baltimore** in 1831, which complemented the city's new link to the **Ohio and Erie Canal** and facilitated a population boom.

225. **Columbus** was not the first **European** explorer to reach the **Americas**, having been preceded by the **Viking** expedition led by **Leif Erikson** in the 11th century, but his voyages led to the first lasting **European** contact with the **Americas**, inaugurating a period of **European** exploration, conquest, and colonization that lasted several centuries.

226. **Aspirin** was **Bayer**'s first major product.

227. **Bender** plays as a midfielder for **Bayer**.

228. **Bender** was first mentioned as an important customs post in a commerce grant issued by the **Moldavian** voivode **Alexander the Good** to the merchants of **Lviv** on October 8, 1408.

229. In **Futurama**, **Bender** is **Fry**'s best friend and roommate.

230. **Avatar** makes the short list of **Nickelodeon** series that were acclaimed by audiences of all ages and critics alike.

231. Following the film's success, **Cameron** signed with **20th Century Fox** to produce three sequels, making **Avatar** the first of a planned tetralogy.

232. On 7 August 2008, **Georgian** forces began shelling the **South Ossetian** capital, **Tskhinvali**.

233. **Georgia** is the largest state entirely east of the **Mississippi River** in land area.

234. In an episode of the cartoon series, **Michelangelo**'s pizza cravings annoyed the others so much that **Master Splinter** hypnotized him into refusing and denouncing pizza whenever the very word was mentioned.

235. Despite holding a low opinion of painting, **Michelangelo** created two of the most influential frescoes in the history of art when he decorated the ceiling and altar wall of the **Sistine Chapel** in **Rome** with artwork inspired from the **Bible**.

236. **Olympus** is located in the **Troodos Mountains** of **Cyprus**.

237. **Olympus**, the highest mountain in **Greece**, is located between the regional units of **Pieria** and **Larissa**, about 80 km (50 mi) southwest from **Thessaloniki**.

238. **Woodford**, a 30-year **Olympus** veteran and **Olympus**' president and chief operating officer since April 2011, had sought to probe financial irregularities and unexplained payments of hundreds of millions of dollars following his appointment as CEO.

239. Shortly after **Hugo**'s election, ratings for freedom in **Venezuela** dropped according to political and human rights group **Freedom House** and **Venezuela** was rated "partly free".

240. **Hugo** was a **Reichszeugmeisterei** licensed supplier of uniforms to the **SS** and **SA**.

241. **Hugo** is buried in the **Panthéon** in **Paris**.

242. Based on **Brian Selznick**'s book **The Invention of Hugo Cabret**, **Hugo** is about a boy who lives alone in the **Gare Montparnasse** railway station in **Paris** in the 1930s.

243. **Hugo** formed over the eastern **Atlantic** near the **Cape Verde Islands** on September 9, 1989. It moved thousands of miles across the **Atlantic**, rapidly strengthening to briefly attain category 5 hurricane strength on its journey. It later crossed over **Guadeloupe** and **St. Croix** on September 17 and 18 as a category 4 hurricane. Weakening slightly more, it passed over **Puerto Rico** as a strong category 3 hurricane. Further weakening

occurred several hours after re-emerging into the **Atlantic**, becoming downgraded to a category 2 hurricane.

244. **Sheeran** made a great comeback and **Atlantic** will be releasing his new album this spring.

245. **Monarch** populations east of the **Rocky Mountains** migrate to the sanctuaries of the **Mariposa Monarca Biosphere Reserve** in **Mexico** and parts of **Florida**.

246. In the **UK**, 14 carriers including **Easyjet**, **British Airways**, **Jet2**, **Monarch**, **Thomas Cook** and **Thomson** flights will be affected by the new air travel regulations.

247. **Goldberg** appeared alongside **Jackson** and **Bassett** in the **HBO** documentary **Unchained Memories**.

248. **Goldberg** was one of the most dominant forces during his career in **WCW**.

249. His restaurant was awarded its third **Michelin star** in 2001, making **Ramsay** the first **Scottish** chef to achieve that feat.

250. **Ramsay** received the **Chemistry Nobel Prize** for his work on noble gases.

251. The concept of **Ghostbusters** was inspired by **Dan Aykroyd**'s fascination with the paranormal. **Aykroyd** conceived it as a project for himself and his friend and fellow **Saturday Night Live** alumnus **John Belushi**.

252. **Feig**'s **Ghostbusters** is considered a box office flop, having made an estimated $70 million loss.

253. At the age of 18, **William Shakespeare** married **Anne Hathaway**, with whom he had three children: **Susanna**, and twins **Hamnet** and **Judith**.

254. Her role as **Fantine** in **Hooper**'s rendition of **Les Miz** earned **Anne Hathaway** widespread acclaim.

255. **Olympiacos** got a draw against **Besiktas** in **Karaiskaki**.

256. During the early stages of the **Greek Uprising**, **Karaiskakis** served in the militia in the **Morea**.

257. **Monaco** striker **Radamel Falcao** admits **UCL** game was 'difficult' for **Dortmund**.

258. **Monaco** is a principality governed under a form of constitutional monarchy, with **Prince Albert II** as head of state.

259. In 2017, **Ferguson** played a lead in **Daniel Espinosa**'s sci-fi thriller **Life**.

260. **Ferguson** won 38 trophies with **United**.

261. Letter from **United** CEO **Oscar Munoz** calls flier 'Belligerent' and ignites anger from public.

262. **Philadelphia** is the **Quaker State**'s largest city.

263. **AEK** is a **Greek** football club based in **New Philadelphia**.

264. **Philadelphia** was one of the first mainstream films to acknowledge **AIDS**, homosexuality, and homophobia. **Tom Hanks** was awarded the **Best Actor Oscar** for his performance in the film.

265. **Diesel** and **Walker** worked together in many **Fast and Furious** movies.

266. **OTB** is the parent company of multiple fashion brands, including **Diesel** and **Marni**.

267. **Jordan** became a member of the **FIBA Hall of Fame** in 2015.

268. **John** used to baptize people in the waters of **Jordan**.

269. **Jordan** is strategically located at the crossroads of **Asia**, **Africa** and **Europe**.

270. Former **Top Gear** host **Richard Hammond** has been flown to hospital after a crash while filming in **Switzerland**.

271. **Adam West**: **TV Batman** actor dies at 88.

272. **Burton**'s **Batman** stars **Nicholson** and **Keaton**.

273. **Ethiopia** will run out of emergency food aid for 7.8 million people affected by drought at the end of this month, the **UN** has warned.

274. **French** voters are casting their ballots to pick their new MPs, a month after electing political outsider **Emmanuel Macron** as president.

275. **Jose Mourinho** has convinced **Alvaro Morata** to join **Man Utd** by 'telling him he would be important' after season as second string at **Real Madrid**.

276. **Triantafyllopoulos** is playing for **Tripoli** in **Superleague**.

277. **Karyotakis** was a **Greek** poet from **Tripoli**.

278. **Tripoli**'s **Bab al-Azizia** served as the main base for the **Libyan** leader **Gaddafi**.

279. **Beirut – Lebanese** Prime Minister **Saad Hariri**'s visit to **Tripoli** on Thursday was significant seeing as it was the first of its kind to the northern city since his appointment to his position.

280. **Hathaway** and **Blunt** don flowing frocks to attend pal **Chastain**'s wedding in **Italy**.

281. **Blunt**, who has been notoriously quiet about his private life said it was also **Sheeran** who really pushed him to open up about his family.

282. Beyond **Pluto**: **NASA**'s next **New Horizons** target is its biggest mystery yet.

283. **Pluto** is **Goofy**'s dog, who himself is an anthropomorphic dog, which has led to some confusion.

284. **Pluto** was distributed by **Sidus Pictures**.

285. **Monroe** stars in the film noir **Niagara**.

286. **Niagara Falls** are a massive tourist attraction on the border of the **US** and **Canada**.

287. As the fledgling army valued literacy in its officers, **Monroe** was commissioned with the rank of lieutenant, serving under Captain **William Washington**.

288. **Cohen** returned to music in 2001 with the release of **Ten New Songs**, which was a major hit in **Canada** and **Europe**.

289. **Cohen** tends to play in edgy and controversial movies like **Borat** and **The Dictator**.

290. For many, a fat **Montblanc** or a silver-plated **Parker** is a treasured item.

291. **Parker** is known for her leading role as **Carrie Bradshaw** on the **HBO** television series **Sex and the City**.

292. **Parker** plays for **San Antonio** and **France**.

293. In 2010 **Clarke** was cast as **Daenerys Targaryen** in the **HBO** medieval fantasy series **Game of Thrones**.

294. **Clarke** was a lifelong proponent of space travel. In 1934, while still a teenager, he joined the **British Interplanetary Society**.

295. The damage to the spacecraft made safe return from a lunar landing impossible, so Lead Flight Director **Gene Kranz** ordered an abort of the **Apollo 13** mission.

296. In **Howard**'s **Apollo 13**, **Hanks** plays the role of **Lovell**.

297. In 1995, **Grant** was arrested in **Los Angeles** with a sex worker, **Divine Brown**, for their 'lewd conduct' in a public place.

298. **Uber** chief **Travis Kalanick** may face bumpy ride.

299. **EU** nurse applicants drop by 96% since **Brexit** vote.

300. **Putin** critic **Alexei Navalny** held as thousands attend **Russia** protests.