Αποδεκτή για δημοσίευση

Natural Language Engineering (2022), 1-00
doi:10.1017/xxxxx

CAMBRIDGE
UNIVERSITY PRESS

A **RTICLE**

SoundexGR: Ένας αλγόριθμος φωνητικής αντιστοίχισης για την ελληνική γλώσσα

Αντρέι Κάβρος και Γιάννης Τζίτζικας

Τμήμα Πληροφορικής, Πανεπιστήμιο Κρήτης, και
Ινστιτούτο Πληροφορικής, ΙΤΕ-ΙΠ, Ελλάδα
andreaskav@outlook.com.gr, tzitzik@ics.forth.gr

(Παραλήφθηκε xx xxx xxx- αναθεωρημένο xx xxx xxx- αποδεκτό xx xxx xxx)

Περίληψη
Τα κείμενα συνήθως υποφέρουν από τυπογραφικά λάθη, τα οποία μπορούν να επηρεάσουν αρνητικά διάφορες εργασίες ανάκτησης πληροφοριών και επεξεργασίας φυσικής γλώσσας. Αν και υπάρχει μεγάλη ποικιλία επιλογών για την αντιμετώπιση αυτού του προβλήματος στην αγγλική γλώσσα, αυτό δεν ισχύει για άλλες γλώσσες. Για την ελληνική γλώσσα, οι περισσότεροι από τους υπάρχοντες φωνητικούς αλγόριθμους παρέχουν μάλλον ανεπαρκή υποστήριξη. Για το λόγο αυτό, στην παρούσα εργασία παρουσιάζουμε έναν αλγόριθμο για φωνητική αντιστοίχιση σχεδιασμένο για την ελληνική γλώσσα: ξεκινάμε από το αρχικό Soundex και το επανασχεδιάζουμε και το επεκτείνουμε για να φιλοξενήσει τους φωνητικούς κανόνες της ελληνικής γλώσσας, καταλήγοντας σε μια οικογένεια αλγορίθμων, την οποία ονομάζουμε Soundex$_{GR}$. Στη συνέχεια, αναφέρουμε διάφορα πειραματικά αποτελέσματα που δείχνουν πώς συμπεριφέρεται ο αλγόριθμος σε διάφορα σενάρια και παρέχουμε συγκριτικά αποτελέσματα για διάφορες παραμέτρους του αλγορίθμου για την αναπροσαρμογή του συμβιβασμού μεταξύ ακρίβειας και ανάκλησης σε σύνολα δεδομένων με διαφορετικά είδη σφαλμάτων. Παρέχουμε επίσης συγκριτικά αποτελέσματα με την αντιστοίχιση με χρήση στεμίσματος, πλήρους φωνημικής μεταγραφής,
και edit-distance, που αποδεικνύουν ότι το Soundex$_{GR}$ αποδίδει καλύτερα (ενδεικτικά, επιτυγχάνει
F-Score πάνω από 95% σε συλλογές λέξεων με παρόμοιο ήχο). Η απλότητα, η αποδοτικότητα και η αποτελεσματικότητα του προτεινόμενου αλγορίθμου τον καθιστά εφαρμόσιμο και προσαρμόσιμο σε ένα ευρύ φάσμα εργασιών.

1. Εισαγωγή

Οι ανορθόγραφες και λανθασμένα προφερόμενες λέξεις μπορούν να επηρεάσουν αρνητικά διάφορες εργασίες στην Ανάκτηση Πληροφορίας (IR), και εργασίες Επεξεργασίας Φυσικής Γλώσσας (NLP), όπως η ευρετηρίαση, η επανα-τρίβωση, η αυτόματη συμπλήρωση (Fafalios et al. (2012)), η αναγνώριση οντοτήτων (Yadav and Bethard (2018)), η απάντηση ερωτήσεων (Dimitrakis et al. (2019)), η ενσωμάτωση δομημένων δεδομένων (Mountantonakis and Tzitzikas (2019)), και οι φωνητικές διεπαφές γενικά (Kaur et al. (2020)). Επιπλέον, οι υπάρχουσες προσεγγίσεις για την παραγωγή ενσωμάτωσης λέξεων (όπως οι Word2Vec Mikolov et al. (2013), Glove Pennington et al.

(2014) και BERT Devlin et al. (2018)) έχουν περιορισμένη εφαρμογή σε κακοσχηματισμένα κείμενα, τα οποία περιέχουν μη αμελητέα ποσότητα λέξεων εκτός λεξιλογίου (Piktus et al. (2019)), πράγμα που σημαίνει ότι δεν μπορούν να παρέχουν ενσωμάτωση για λέξεις που δεν έχουν παρατηρηθεί κατά τη στιγμή της εκπαίδευσης.

Για την αντιμετώπιση τέτοιων περιπτώσεων χρησιμοποιούνται συνήθως αποστάσεις *που σχετίζονται με το στίγμα* και την *επεξεργασία* (π.χ. η απόσταση Levenstein Levenshtein (1966)) (π.χ. Medhat et al. (2015)). Ωστόσο, αυτές οι μέθοδοι δεν είναι πάντα επαρκείς: δεν μπορούμε να εφαρμόσουμε το stemming σε ονόματα προσώπων και τοποθεσιών, ενώ η απόσταση επεξεργασίας μεταξύ μιας λέξης και μιας ανορθόγραφης λέξης (που έχει περισσότερα από ένα ορθογραφικά λάθη), μπορεί να είναι πολύ μεγάλη (π.χ. η απόσταση επεξεργασίας μεταξύ "Schumacher" και

"Soumaher" είναι 4), περιορίζοντας έτσι την αξία της αντιστοίχισης *με βάση την* απόσταση *επεξεργασίας*. Μια άλλη οικογένεια αλγορίθμων για την αντιμετώπιση αυτού του ζητήματος είναι η οικογένεια αλγορίθμων *φωνητικής αντιστοίχισης*. Πράγματι, οι *φωνητικοί κώδικες* έχουν χρησιμοποιηθεί σε διάφορα πλαίσια, π.χ. για την ευρετηρίαση και ανάκτηση ονομάτων από ένα μεγάλο σύνολο δεδομένων (Koneru et al. (2016)), για την ανάκτηση SMS (Pinto et al. (2012)), για την ανακάλυψη συνδέσμων (Ahmed et al. (2019)), για την ανίχνευση διπλών εγγραφών (Elmagarmid et al. (2006)), για τη διατήρηση της ιδιωτικότητας (Karakasidis and Verykios (2009)) και άλλα.

Η πρώτη εφαρμογή φωνητικών αλγορίθμων χρονολογείται από το 1918, με τον αλγόριθμο Soundex (Russell (1918 1922)), ο οποίος προσπαθεί να κωδικοποιήσει τις λέξεις με βάση τον τρόπο που ακούγονται. Αν και υπάρχει πληθώρα προτεινόμενων λύσεων για την αντιμετώπιση αυτού του ζητήματος στην αγγλική γλώσσα (Soundex, Metaphone, Double Metaphone, Metaphone 3, NYSIIS και άλλες), αυτό δεν ισχύει για την ελληνική γλώσσα. Στην παρούσα εργασία προτείνουμε και αξιολογούμε έναν αλγόριθμο που ανήκει σε αυτή την οικογένεια και στοχεύει στην αντιμετώπιση τέτοιων ζητημάτων για την ελληνική γλώσσα. Ένας τέτοιος αλγόριθμος θα πρέπει να είναι σε θέση να αντιμετωπίσει μια ευρύτερη ποικιλία λαθών με υψηλή ακρίβεια. Για παράδειγμα, για τη λέξη ѕtolμος (η οποία γράφεται σωστά και ακούγεται [΄ετίμος]), θα πρέπει να είναι σε θέση να ανακτήσει (να ταιριάξει) ανορθόγραφες παραλλαγές της ίδιας λέξης και της ίδιας έννοιας της λέξης, όπως ѕtlμος ([΄ετίμος]) , ѕtlμς ([΄ετίμος]) , αίτρμος ([΄ετίμος]) , ή παρόμοιους όρους διαφορετικής έννοιας όπως ѕztlμος ([΄εντίμος]). Στο εξής θα χρησιμοποιούμε [ ] για να περικλείουμε τόσο φωνητικές όσο και φωνηματικές μεταγραφές λέξεων.

Η προσέγγισή μας για το σχεδιασμό ενός τέτοιου αλγορίθμου είναι να προσαρμόσουμε τη βασική ιδέα του Soundex στα χαρακτηριστικά της ελληνικής γλώσσας, για να έχουμε μια βασική μέθοδο, και στη συνέχεια να διευρύνουμε τους κανόνες του, όπως έκαναν οι περισσότεροι σύγχρονοι (μετά τον Soundex) φωνητικοί αλγόριθμοι, για να προσαρμόσουμε τους φωνητικούς κανόνες της ελληνικής γλώσσας. Για το σκοπό αυτό, εισάγουμε μια οικογένεια αλγορίθμων που ονομάζουμε Soundex_{GR} . Με το Soundex_{GR} επιτυγχάνουμε την απόδοση του ίδιου κωδικού σε σύνολο λέξεων που πρέπει να ταιριάζουν, όπως το σύνολο των λέξεων *{μήνυμα*, μύνημα, μίνιμα, *μοίνειμα}*, το σύνολο *{* εύδοξος, *εβδοξος}* και το σύνολο *{ευάερο*, αιβάερρο, *αιββάαιρο}*.

Στη συνέχεια αναφέρουμε συγκριτικά πειραματικά αποτελέσματα που δείχνουν ποια παραλλαγή/διαμόρφωση του αλγορίθμου συμπεριφέρεται καλύτερα στην αξιολόγηση σε σύνολα δεδομένων με διάφορα είδη σφαλμάτων. Συγκεκριμένα, ο αρχικός αλγόριθμος Soundex, τροποποιημένος για να αντιστοιχεί στο ελληνικό αλφάβητο, επιτυγχάνει ένα μέσο F-Score ίσο με 0,64 σε διάφορα είδη λαθών (προσθήκη, διαγραφή ή αντικατάσταση γραμμάτων). Η βελτιωμένη έκδοση που λαμβάνει υπόψη και τους ελληνικούς φωνητικούς κανόνες, επιτυγχάνει μέσο F-Score 0,66. Η παραλλαγή που χρησιμοποιεί και τις δύο προηγούμενες εκδόσεις για την εύρεση μιας αντιστοιχίας, επιτυγχάνει μέσο F-Score 0,70, ενώ σε ένα σύνολο δεδομένων που περιέχει λέξεις με παρόμοιο ήχο φτάνει σε F-Score ίσο με 0,91, ενώ το Soundex_{GR} επιτυγχάνει F-Score ίσο με 0,97. Επιπλέον, αναφέρουμε συγκριτικά πειραματικά αποτελέσματα με stemming και πλήρη φωνητική μεταγραφή που δείχνουν ότι ο προτεινόμενος αλγόριθμος έχει καλύτερες επιδόσεις. Αξιολογούμε επίσης πώς το μήκος του κώδικα επηρεάζει το F-Score σε σύνολα δεδομένων διαφορετικών μεγεθών, τύπων σφαλμάτων και μήκους λέξεων και μετράμε την αποδοτικότητα εφαρμόζοντας τον πάνω σε ένα ελληνικό λεξικό. Συνολικά, η αποτελεσματικότητα, η απλότητα και η αποδοτικότητα της προτεινόμενης οικογένειας αλγορίθμων την καθιστά εφαρμόσιμη σε ένα ευρύ φάσμα εργασιών.

Αν και υπάρχουν εργασίες σχετικά με τη φωνητική (και φωνημική) μεταγραφή των ελληνικών λέξεων (π.χ. Themistocleous (2011)), εξ όσων γνωρίζουμε, δεν υπάρχει καμία εργασία σχετικά με τη χρήση και την αξιολόγηση τέτοιων κωδίκων για την αντιστοίχιση ελληνικών κειμένων.

Το υπόλοιπο της παρούσας εργασίας οργανώνεται ως εξής. Η Ενότητα 2 περιγράφει το ιστορικό και αναλύει τις σχετικές εργασίες. Η ενότητα 3 περιγράφει την προτεινόμενη οικογένεια αλγορίθμων και παρέχει διάφορα παραδείγματα εφαρμογών για την αποκάλυψη των διαφορών αυτών των παραλλαγών. Η Ενότητα 4 επικεντρώνεται στην αξιολόγηση, παρουσιάζει εκτενή συγκριτικά αποτελέσματα (για διάφορα σύνολα δεδομένων, μεγέθη κωδικών και μεθόδους αντιστοίχισης, συμπεριλαμβανομένων του stemming, της πλήρους φωνητικής μεταγραφής και της απόστασης επεξεργασίας) και συζητά εφαρμογές. Τέλος, η Ενότητα 5 ολοκληρώνει την εργασία και προσδιορίζει θέματα που χρήζουν περαιτέρω εργασίας και έρευνας.

2. Ιστορικό και συναφείς εργασίες

Υπάρχει μεγάλη ποικιλία φωνητικών αλγορίθμων, πολλοί, αν όχι όλοι, είναι απόγονοι του αλγορίθμου Soundex (που περιγράφεται λεπτομερώς στην §2.1), όπως ο Philips (1990), Hood (2002). Αυτοί οι αλγόριθμοι στοχεύουν στην ανάκτηση ανορθόγραφων λέξεων και στη βελτίωση της ανάκτησης πληροφοριών, δημιουργώντας μια κωδικοποίηση του ερωτήματος με βάση φωνητικούς κανόνες προφοράς. Χρησιμοποιούνται κυρίως σε Συστήματα Βάσεων Δεδομένων για να βοηθήσουν στη διαδικασία ανάκτησης, καθώς και σε διάφορες εργασίες IR, όπως η αναζήτηση, η αυτόματη συμπλήρωση ερωτημάτων και η ανάκτηση. Είναι επίσης χρήσιμες σε εργασίες NLP, όπως η αναγνώριση και η σύνδεση ονομαστικών οντοτήτων και η αποσαφήνιση της σημασίας των λέξεων γενικά. Δυστυχώς, οι περισσότερες από αυτές παρέχουν στην καλύτερη περίπτωση ελάχιστη ή καθόλου υποστήριξη για την ελληνική γλώσσα.

Οι εργασίες που αφορούν την επεξεργασία της ελληνικής γλώσσας γενικά δεν είναι υπερβολικές (βλέπε Papantoniou and Tzitzikas (2020) για μια πρόσφατη έρευνα), σε σύγκριση με την αγγλική γλώσσα. Ωστόσο, υπάρχουν αρκετά έργα σχετικά με τη φωνητική της ελληνικής γλώσσας, τα οποία περιγράφονται συνοπτικά παρακάτω.

Το βιβλίο Newton (1972) μελετά την ελληνική φωνολογία γενικά, ενώ οι Epitropakis et al. (1993) παρουσιάζουν έναν αλγόριθμο για την παραγωγή τονισμού (περιγράμματα F0) για το ελληνικό σύστημα Text-To-Speech. Οι Fourakis et al. (1999) αναλύουν τα ακουστικά χαρακτηριστικά των ελληνικών φωνηέντων (διάρκεια, θεμελιώδης συχνότητα, πλάτος και άλλα). Στην ίδια κατεύθυνση, η Sfakianaki (2002) αναλύει τα ακουστικά χαρακτηριστικά των ελληνικών φθόγγων που παράγονται από ενήλικες και παιδιά, ενώ η Trudgill (2009) εστιάζει στα συστήματα φθόγγων της ελληνικής διαλέκτου. Η Αρβανίτη (2007) περιγράφει την κατάσταση της ελληνικής φωνητικής το 2007.

Η IPAGreek (Themistocleous (2011)), είναι μια εφαρμογή (διαθέσιμη στο Themistocleous (2017)) της "φωνολογικής γραμματικής" της Πρότυπης Νεοελληνικής και της Κυπριακής Ελληνικής. Η εφαρμογή επιτρέπει στους χρήστες να μεταγράφουν κείμενο γραμμένο στην ελληνική ορθογραφία στο Διεθνές Φωνητικό Αλφάβητο (IPA).

Ο Karanikolas (2019) προτείνει μια αυτόματη προσέγγιση μηχανικής μάθησης που μαθαίνει κανόνες για τον τρόπο μεταγραφής των ελληνικών λέξεων στο φωνητικό αλφάβητο της Διεθνούς Ένωσης Φωνητικής (IPA), ωστόσο η προτεινόμενη μέθοδος δεν έχει εφαρμοστεί, ούτε έχει αξιολογηθεί.

Τέλος, ο Themistocleous (2019) περιγράφει προσεγγίσεις ταξινόμησης που βασίζονται σε βαθιά νευρωνικά δίκτυα για τη διάκριση δύο ελληνικών διαλέκτων, συγκεκριμένα της αθηναϊκής ελληνικής, της πρωτότυπης μορφής της τυπικής νεοελληνικής και της κυπριακής ελληνικής. Η εν λόγω εργασία βασίζεται στα ακουστικά χαρακτηριστικά της προφορικής γλώσσας.

Οι περισσότερες από τις παραπάνω εργασίες επικεντρώνονται στις ακουστικές πτυχές της γλώσσας και λιγότερες στη διαχείριση του ελληνικού κειμένου, και ειδικότερα στο πρόβλημα της ανάκτησης και της αντιστοίχισης. Ένας αλγόριθμος που θα μπορούσε να χρησιμοποιηθεί για την ελληνική γλώσσα και για τις εργασίες που αναφέραμε, δηλαδή για την αντιστοίχιση πάνω σε ελληνικό κείμενο, είναι ο αλγόριθμος Beider-Morse Beider (2008), αλλάζοντας τα ελληνικά γράμματα με τα αντίστοιχα αγγλικά γράμματα, χωρίς να λαμβάνονται υπόψη οι ελληνικοί φωνητικοί κανόνες, αλλά με βάση το πώς θα ακούγονταν στην αμερικανική διάλεκτο. Μια άλλη προσέγγιση θα ήταν να πάρουμε μια μέθοδο φωνημικής μεταγραφής, όπως αυτή που περιγράφεται στο Themistocleous (2017), και να την περικόψουμε ή/και να την τροποποιήσουμε (δηλαδή να ομαδοποιήσουμε διαφορετικά γράμματα στον ίδιο κωδικό ως μέσο υποβοήθησης της αντιστοίχισης), ώστε να είναι κατάλληλη για προσεγγιστική αντιστοίχιση.

Στην παρούσα εργασία επιχειρούμε να καλύψουμε αυτό το κενό στη βιβλιογραφία, δηλαδή προτείνουμε έναν αλγόριθμο γενικής χρήσης για φωνητική αντιστοίχιση για ελληνικό κείμενο και αξιολογούμε τις δυνατότητές του για αντιστοίχιση σε διάφορα σύνολα δεδομένων και υπό διαφορετικές διαμορφώσεις.

2.1 Ο αρχικός αλγόριθμος **Soundex**

Όπως αναφέρθηκε στην εισαγωγή, η προσέγγισή μας για το σχεδιασμό ενός αλγορίθμου φωνητικής αντιστοίχισης για την ελληνική γλώσσα, είναι να προσαρμόσουμε τη βασική ιδέα του Soundex στους χαρακτήρες της ελληνικής γλώσσας, για να έχουμε μια βασική μέθοδο, και στη συνέχεια να διευρύνουμε τους κανόνες του για να προσαρμόσουμε τους φωνητικούς κανόνες της ελληνικής γλώσσας.

Ο αλγόριθμος Soundex ξεκίνησε το 1918 και αναπτύχθηκε από τους Robert C. Russell και Margaret King Odell και είχε ένα απλό σύνολο κανόνων. Δημιουργεί έναν κώδικα αγνοώντας τα φωνήεντα και το γράμμα *h*, αν δεν βρίσκεται στην αρχή της λέξης, και κωδικοποιώντας τα σύμφωνα με βάση τον τρόπο που ακούγονται, δημιουργώντας έναν κώδικα μήκους μόλις 4 χαρακτήρων. Συγκεκριμένα, τα βήματα του αρχικού αλγορίθμου Soundex είναι τα εξής:

(i) κρατήστε το πρώτο γράμμα μη κωδικοποιημένο,
(ii) αφαιρεί όλες τις εμφανίσεις των a, e, h, i, o, u, w, y, εκτός αν εμφανίζονται ως το πρώτο γράμμα της λέξης,
(iii) αντικαταστήστε τα σύμφωνα μετά το πρώτο γράμμα όπως φαίνεται στον πίνακα 1,
(iv) αφαιρείτε τα γειτονικά διπλά ψηφία,
(v) να παράγει έναν κωδικό της μορφής Γράμμα Ψηφίο Ψηφίο Ψηφίο Ψηφίο Ψηφίο αγνοώντας τα ψηφία μετά το τρίτο (αν χρειάζεται) ή προσθέτοντας μηδενικά (αν χρειάζεται).

Για παράδειγμα, το όνομα SMITH θα κωδικοποιηθεί σε S530, όπως και τα ονόματα SCHMIDT και SMYTH, ενώ τόσο το ROBERT όσο και το RUPERT θα δώσουν R163. Ωστόσο, είναι δυνατόν να προκύψουν ανακριβή αποτελέσματα, π.χ. τα BLACK και BAILS αποδίδουν τον κωδικό B420.

Ο Christian (1998) περιέγραψε τα προβλήματα του αρχικού Soundex, αγνοώντας τη διαφορετική ορθογραφία των γραμμάτων σε διαφορετικά περιβάλλοντα και συνδυασμούς γραμμάτων. Άλλα προβλήματα, περιλαμβάνουν την αγνόηση των φωνηέντων εάν δεν βρίσκονται στην αρχή της λέξης και τον σύντομο παραγόμενο κώδικα. Όλα αυτά τα ζητήματα, βλάπτουν σημαντικά τα επίπεδα ακρίβειας του Soundex.

Η πρώτη χρήση του αλγορίθμου ήταν για την ανάκτηση ονομάτων ανθρώπων από ένα μεγάλο σύνολο δεδομένων, ενώ σήμερα ο αλγόριθμος Soundex ή οι απόγονοί του συναντώνται σε διάφορα συστήματα, π.χ. για την ανάκτηση SMS (Pinto et al. (2012)), για την ευρετηρίαση ονομάτων (Raghavan and Allan (2004)), για την ανακάλυψη συνδέσμων (Ahmed et al. (2019)), για την ανίχνευση διπλών εγγραφών (Elmagarmid et al. (2006)), για τη σύνδεση εγγραφών (da Silva et al. (2020)) κ.λπ.

Επιπλέον, έχει προσαρμοστεί για διάφορες γλώσσες, συμπεριλαμβανομένης της ταϊλανδικής γλώσσας (Karoonboonyanan et al. (1997)), της αραβικής γλώσσας (Yahia et al. (2006), Shedeed and Abdel (2011), Ousidhoum and Bensaou (2013)), της βιετναμέζικης γλώσσας (Nguyen et al. (2008)), η κινεζική γλώσσα (Li and Peng (2011)), η ινδική γλώσσα (Shah (2014)- Gautam et al. (2019)), η γλώσσα Assamese (Baruah and Mahanta (2015)), η ισπανική γλώσσα (del Pilar Angeles et al. (2015)) και άλλες.

Πίνακας 1. : Αντικατάσταση συμφώνων στο Soundex

b, f, p, v →	1
c, g, j, k, q, s, x, z →	2
d, t →	3
l →	4

$$m, n \to 5$$
$$r \to 6.$$

2.2 Άλλοι σχετικοί αλγόριθμοι

Αρκετοί αλγόριθμοι μετά τον Soundex, που αναπτύχθηκαν από την κεντρική ιδέα του, ομαδοποιούν τα γράμματα με βάση την προφορά τους, με στόχο τη βελτίωση του αρχικού αλγορίθμου. Μερικοί από τους πιο γνωστούς είναι οι εξής:

Metaphone (Philips (1990)): Εφαρμόζει έναν μετασχηματισμό στην αρχική λέξη, πριν η λέξη κωδικοποιηθεί μέσω κάδων προφοράς γραμμάτων και ενός τεράστιου συνόλου φωνητικών κανόνων. Στη συνέχεια, έγιναν διάφορες βελτιώσεις σε αυτό: Ο Philips (2000) δημιουργεί μια πρωτεύουσα και μια δευτερεύουσα κωδικοποίηση για μια δεδομένη λέξη και εφαρμόζει κανόνες με βάση τη γλώσσα προέλευσης της λέξης εισόδου, ενώ ο Philips (2013) πρόσθεσε ρυθμιζόμενους κανόνες στον αλγόριθμο, καθώς και βελτίωσε περαιτέρω την ανάκτηση ξένων λέξεων.

Caverphone (Hood (2002)): Εφαρμόζει μετασχηματισμούς στη λέξη που μπορεί να είναι μεγαλύτεροι από 2gram κάθε φορά, για να παράγει μια κωδικοποίηση. Δημιουργήθηκε αρχικά με βάση τις προφορές σε μια συγκεκριμένη περιοχή της Νέας Ζηλανδίας.

BMPM (Beider (2008)): Προτού εφαρμοστεί ένα σύνολο φωνητικών κανόνων στη λέξη, πραγματοποιείται μια διαδικασία αναγνώρισης της προέλευσης της λέξης και στη συνέχεια εφαρμόζονται οι αντίστοιχοι γλωσσικοί κανόνες.

Η MRA (Match Rating Approach) που αναπτύχθηκε από τη Western Airlines το 1977, είχε ένα απλό σύνολο φωνητικών κανόνων, παρέχοντας μέσω ενός συνόλου κανόνων σύγκρισης για την κωδικοποίηση. Άλλοι φωνητικοί αλγόριθμοι παράγουν περισσότερες από μία κωδικοποιήσεις στη λέξη, προκειμένου να βελτιώσουν την ανάκτηση Soundex. Σε γενικές γραμμές, αυτοί οι αλγόριθμοι στόχευαν να αντιμετωπίσουν τις ελλείψεις του αρχικού Soundex που περιγράφηκαν στην ενότητα 2.1 και το βελτίωσαν, όπως προτείνουν οι Koneru et al. (2016), όσον αφορά την ακρίβεια, η οποία είναι η κύρια έλλειψη του Soundex αλγόριθμος.

3. Ο αλγόριθμος Soundex_{GR} (και παραλλαγές)

Η παρούσα ενότητα οργανώνεται ως εξής: Αρχικά, στην §3.1, περιγράφουμε εν συντομία τις απαιτήσεις. Στη συνέχεια, στην §3.2 περιγράφουμε τη βασική ιδέα του νέου αλγορίθμου, τον οποίο ονομάζουμε Soundex_{GR} , ενώ στην §3.3 περιγράφουμε λεπτομερώς τα ακριβή βήματα αυτού του αλγορίθμου. Για λόγους συγκριτικής αξιολόγησης, στην §3.4, ορίζουμε μια παραλλαγή που ονομάζουμε $^{Soundexnaive}_{GR}$, η οποία μοιράζεται τις ίδιες αρχές του αρχικού αλγορίθμου Soundex, αλλά χωρίς καμία προεπεξεργασία λέξης πριν από την κωδικοποίηση της λέξης. Τέλος, στην §3.5 παρουσιάζουμε μια άλλη παραλλαγή για φωνητική αντιστοίχιση ($\text{Soundex}^{comp}_{GR}$) που χρησιμοποιεί τόσο το Soundex_{GR} όσο και το $^{Soundexnaive}_{GR}$.

3.1 Απαιτήσεις για την ελληνική γλώσσα.

Η βασική ιδέα του αρχικού αλγορίθμου Soundex μπορεί εύκολα να μεταφραστεί σε μια ελληνική έκδοση. Πράγματι, μια απλή εκδοχή θα ήταν να υιοθετηθούν ακριβώς οι ίδιοι κανόνες με τον Soundex, όπως περιγράφονται στην ενότητα 2.1, με ελληνικά σύμφωνα. Ωστόσο, θέλαμε να αντιμετωπίσουμε τις ελλείψεις του αρχικού Soundex (που περιγράφεται στην ενότητα 2.1), συνεπώς να λάβουμε υπόψη μας τα συμφραζόμενα των γραμμάτων, τους συνδυασμούς γραμμάτων και γενικά τους γραμματικούς κανόνες που αφορούν ειδικά τα ελληνικά. Επιπλέον, ενώ το αρχικό Soundex υλοποιήθηκε για χρήση κυρίως σε ονόματα, θα θέλαμε έναν αλγόριθμο και για κανονικές λέξεις. Αυτό σημαίνει ότι θα θέλαμε να επιτύχουμε υψηλή ακρίβεια για τις κανονικές (συχνές) λέξεις (για να

αποφύγουμε να έχουμε πολλές λέξεις που έχουν τον ίδιο κωδικό), ενώ για τα ονόματα θα θέλαμε να επιτύχουμε υψηλή ανάκληση (δηλαδή χαμηλό ποσοστό ψευδώς θετικών αποτελεσμάτων), καθώς αυτά εμφανίζονται πιο σπάνια.

Για παράδειγμα, θα θέλαμε έναν αλγόριθμο που θα ομαδοποιούσε σωστά θάλασα με θάλασσα (και τα δύο ακούγονται [θ΄alasa]), μήzυμα με μύzρμα (και τα δύο ακούγονται [m΄ınima]), αίτρμα με ́stlμα ή ́stolμα (όλα ακούγονται [΄etima]), εύκολα με ́sφκολα (και οι δύο ήχοι

[efkola]). Ο αλγόριθμος θα πρέπει να ανακτά όλες αυτές τις περιπτώσεις με ελάχιστο θόρυβο και όσο το δυνατόν μεγαλύτερη ανάκληση.

3.2 Η βασική ιδέα του Soundex_{GR}

Εδώ περιγράφουμε τον αλγόριθμό μας που ονομάζουμε Soundex_{GR} . Όπως και στο αρχικό Soundex, διατηρούμε ένα μήκος κωδικοποίησης μόλις 4 χαρακτήρων. Όπως θα δούμε στα πειράματα που αναφέρονται στην ενότητα 4.5, αν αυξήσουμε το μήκος από 4 σε 5 έχουμε υψηλότερη ακρίβεια κατά 5-10% τοις εκατό, ωστόσο η ανάκληση μειώνεται κατά 10-15%. Ωστόσο, σε μεγαλύτερα σύνολα δεδομένων, ένα μεγαλύτερο μήκος μπορεί να είναι καταλληλότερο (λεπτομερή πειραματικά αποτελέσματα δίνονται στην ενότητα 4.10).

Όπως συζητήθηκε στην ενότητα 2.1, το Soundex έχει ένα πρόβλημα ακρίβειας, το οποίο προέρχεται από το συνδυασμό σύντομου κωδικού μόλις 4 χαρακτήρων και τη μη συνεκτίμηση οποιουδήποτε λεξιλογικού πλαισίου. Για να βελτιώσουμε τα επίπεδα ακρίβειας του αλγορίθμου Soundex, πρέπει να εστιάσουμε σε αυτά. Σε αντίθεση με το Soundex, στο Soundex_{GR} λαμβάνουμε υπόψη ένα πιο πλούσιο σύνολο κανόνων, που αντιστοιχούν στους φωνητικούς κανόνες της ελληνικής γλώσσας. Παρακάτω περιγράφουμε τα βασικά σημεία και στη συνέχεια περιγράφουμε τα ακριβή βήματα.

Πριν από την κωδικοποίηση μιας λέξης, την προεπεξεργαζόμαστε και δημιουργούμε μια διαφορετική μορφή λέξης. Οι διαδικασίες προεπεξεργασίας περιλαμβάνουν: εντοπισμό των περιπτώσεων που ένα φωνήεν ακούγεται ως σύμφωνο στα ελληνικά, ομαδοποίηση των ζευγαριών φωνηέντων με βάση τον τρόπο που ακούγονται, αφαίρεση του τονισμού και αποσυναρμολόγηση των διγραμμάτων σε μεμονωμένα γράμματα. Όταν ολοκληρωθεί αυτή η διαδικασία, η λέξη κωδικοποιείται.

Για παράδειγμα, το μπαΐζω, που ακούγεται [b΄eno], θα μετασχηματιστεί σε bεzo και τελικά θα κωδικοποιηθεί σε b*7$, ενώ το όνομα Γλαζψ (που ακούγεται [j΄anis]), θα μετασχηματιστεί σε Γλαζλ και στη συνέχεια θα κωδικοποιηθεί σε γ@97 (περισσότερα παραδείγματα θα δοθούν αργότερα).

Μια άλλη διαφορά είναι ότι το Soundex αγνοεί τα φωνήεντα, ωστόσο το Soundex_{GR} δεν αγνοεί τα
φωνήεντα, αλλά τα ομαδοποιεί σε 3 κατηγορίες με βάση τον τρόπο που ακούγονται, συγκεκριμένα σε α, ο, π, προκειμένου να βελτιώσει την ακρίβεια του αλγορίθμου.

Το τελευταίο γράμμα της λέξης, αγνοείται αν είναι σύμφωνο, συγκεκριμένα αν είναι ς ή z, καθώς δεν προσθέτει μεγάλη αξία στη λέξη.

3.3 Τα ακριβή βήματα του Soundex_{GR}

Ο αλγόριθμος Soundex_{GR} και οι διαδικασίες που χρησιμοποιούνται από τον αλγόριθμο δίνονται σε ψευδοκώδικα στο Alg. 1.

Στο πρώτο μέρος, προ-επεξεργαζόμαστε τη λέξη, εφαρμόζοντας σε αυτήν συντακτικούς και γραμματικούς κανόνες της ελληνικής γλώσσας. Συγκεκριμένα, στο `UnwrapConsonantBigrams`(*word*), αλλάζουμε τα κοινά ελληνικά *διγράμματα συμφώνων* με τα ισοδύναμα, πανομοιότυπα προφερόμενα μεμονωμένα γράμματα. Αυτό βασίζεται στις αντικαταστάσεις που παρουσιάζονται στον Πίνακα 2 (πάνω μέρος).

Στη συνέχεια, στο `TransformVowelsToConsonant`(*word*) συνεχίζουμε με τον προσδιορισμό του αν το ελληνικό γράμμα υ ενεργεί ως φωνήεν ή ως σύμφωνο. Αυτή η διάκριση χρειάζεται να γίνει μόνο αν το προηγούμενο γράμμα είναι α ή ε και το επόμενο σύμφωνο εμπίπτει στην κατηγορία του Πίνακα 4 και του Πίνακα 5. Για παράδειγμα,
αύξωz [΄afkson] (υ: σύμφωνο-φ),
αυτός [aft΄os] (υ: σύμφωνο-φ), αυλή
[avl΄ι] (υ: σύμφωνο-β),

Εύξελζος [΄efksinos] (υ: σύμφωνο-φ),
Εύδοξος [΄evdoksos] (υ: σύμφωνο-β),
ευεξία [eveks΄ια] (υ: σύμφωνο-β).

Μετά από αυτό, αφαιρούμε τα γράμματα ΄ς΄ ή ΄z΄ αν είναι το τελευταίο γράμμα της λέξης, καθώς αυτά τα γράμματα δεν προσθέτουν μεγάλη αξία στον κόσμο.

Αλγόριθμος 1 Soundex$_{GR}$
Είσοδος: λέξη
Έξοδος: Μια κωδικοποίηση της λέξης

```
διαδικασία Soundex_GR (λέξη)
    w ←UnwrapConsonantBigrams(word)
    w ←TransformVowelsToConsonants(w)
    w ←RemoveLast(w)
    w ←GroupVowels(w)
    w ←RemoveIntonation(w)
    Code ←SoundexEncode(w)
    Κωδικός ←Αφαίρεση των αντιγράφων(κωδικός)
    ←TrimLength(code, 4)
    κωδικός επιστροφής
τέλος της διαδικασίας
```

```
διαδικασία UnwrapConsonantBigrams(word)
    για όλα τα digram στο word do
        Αντίγραφο=΄μπ΄ τότε Replace(digram,΄b΄)
        τότε Replace(digram,΄d΄)
        τότε Replace(digram,΄g΄) 6:          else if digram ∈
{΄τσ΄΄, ΄τζ΄΄} then Replace(digram,΄c΄) 7:          else if digram
∈ {΄πσ΄, ΄πς΄} then Replace(digram,΄ψ΄) 8: else if digram∈
{΄κσ΄, ΄κς΄} then Replace(digram,΄ξ΄)
        τέλος αν
    τέλος για
    λέξη επιστροφής
τέλος της διαδικασίας
```

```
διαδικασία TransformVowelsToConsonants(word)
    για το γράμμα στη λέξη do
        και προηγούμενο =΄α΄ ή προηγούμενο =΄ε΄ τότε
            (Πίνακας 5) τότε
                Αντικατάσταση(γράμμα,΄φ΄)
            (Πίνακας 4) ή το επόμενο είναι φωνήεν τότε
                Αντικατάσταση(γράμμα,΄β΄)
            τέλος αν
        τέλος αν
    τέλος για
    λέξη επιστροφής
τέλος της διαδικασίας
```

```
διαδικασία GroupVowels(word)
Αντικατάσταση(λέξη, ΄ά΄ →
΄α΄) Αντικατάσταση(λέξη, ΄έ΄
→ ΄ε΄) Αντικατάσταση(λέξη,
΄ί΄ → ΄ι΄)
    για όλα τα digram στο word do
        τότε Replace(digram,΄ε΄)
        else if digram∈ {΄ει΄, ΄οι΄} then Replace(digram,΄ι΄)
        τότε Replace(digram,΄ο΄΄)
        τέλος αν
    τέλος για
    Αντικατάσταση(λέξη, ΄η΄,΄ή΄,΄υ΄,΄ύ΄,΄ϋ΄,΄ΰ΄,΄ϊ΄,΄ΐ΄ → ΄ι΄)
    Αντικατάσταση(λέξη, ΄ω΄,΄ώ΄ → ΄ο΄)
    επιστροφή λέξης
τέλος της διαδικασίας
```

```
διαδικασία RemoveLast(word)
    ή word.lastLetter = ΄ν΄ τότε
        Αντικατάσταση(word.lastLetter,΄΄)
    τέλος αν
τέλος της διαδικασίας
```

Πίνακας 2. : Φωνητικοί κανόνες

διγράμματα συμφώνων re- τοποθετήσεις	
2 γραμμάρια	1 γραμμάριο
μπ	b
ντ	d
γκ,γγ	g
τσ,τζ	c
πσ,πς	ψ
κς,κσ	ξ
φθόγγων bigrams αντικατάσταση- ment	
2 γραμμάρια	1 γραμμάριο
αι	ε
οι,ει	ι
ου	ο

Πίνακας 3. : Soundex_{GR} κάδοι

Κάδοι Soundex	
Ομάδα	Κωδικός
β,φ,π,β	1
γ,χ	2
δ,τ,θ,δ	3
ζ,σ,ξ,ψ,ς,γ	4
κ,g	5
λ	6
μ,ν	7
ρ	8
α	9
ε	*
ο,ω	$
ι	@

Πίνακας 4. : Δυνατά σύμφωνα στα ελληνικά

Δυνατό σύμφωνο
β
γ
δ
ζ
λ
μ
ν
ρ

Πίνακας 5. : Σιωπηλά σύμφωνα στα ελληνικά

Σιωπηλό σύμφωνο
θ
κ
ξ
π
σ
τ
φ
χ

Στο `GroupVowels` αλλάζουμε τα κοινά ελληνικά *διγράμματα φωνηέντων* με τα ισοδύναμα, ταυτόσημα προφερόμενα, μεμονωμένα γράμματα. Αυτό βασίζεται στις αντικαταστάσεις που παρουσιάζονται στον Πίνακα 2 (κάτω μέρος).

Στην `RemoveIntonation`(*word*) (γραμμή 6), αφαιρούμε τους πιθανούς εναπομείναντες τόνους (αν υπάρχουν), αυτό είναι το τελευταίο βήμα της φάσης προεπεξεργασίας της λέξης.

Στο `SoundexEncode`(*word*) (γραμμή 7) κωδικοποιούμε τη λέξη μέσω των ζευγών *γραμμάτων-ψηφίων* του πίνακα 3. Μετά τη μετάφραση της αρχικής λέξης σε κώδικα,

αφαιρούμε τα διπλά ψηφία *που βρίσκονται δίπλα* στην RemoveDuplicates(*code*) (γραμμή 8) και *περιορίζουμε το μήκος* σε 4 χαρακτήρες ή αναθέτουμε 0 στο τέλος του κώδικα αν ο κώδικας είναι μικρότερος από 4 χαρακτήρες στην trimLength(*code,4*) (γραμμή 9).

Στον Πίνακα 6 παρουσιάζονται μερικά παραδείγματα που δείχνουν το αποτέλεσμα μετά από κάθε βήμα του αλγορίθμου.

Αρχική λέξη	ΈμΠΕΙΡΟΣ	ΝΟΎΣ	ΕΥΆΕΡΟΣ	ΔΙΑΛΛΕΙμΑ	ΔΙΆΛΥμΑ	ΑΥΛΩΝ	ΑΥΓΌ	ΑΒΓΌ	ΑΥΓΟΥΛΆΚΙΑ
Ξετυλίξτε τα μεγαλογράμματα με τον φθόγγο	ΈΒΕΙΡΟΣ	ΝΟΎΣ	ΕΥΆΕΡΟΣ	ΔΙΑΛΛΕΙμΑ	ΔΙΆΛΥμΑ	ΑΥΛΩΝ	ΑΥΓΌ	ΑΒΓΌ	ΑΥΓΟΥΛΆΚΙΑ
ΜετασχηματισμόςΦωνήεντα- ΣεΚονσόνια	ΈΒΕΙΡΟΣ	ΝΟΎΣ	ΕΒΆΕΡΟΣ	ΔΙΑΛΛΕΙμΑ	ΔΙΆΛΥμΑ	ΑΒΛΩΝ	ΑΒΓΌ	ΑΒΓΌ	ΑΒΓΟΥΛΆΚΙΑ
RemoveLast	ΈΒΕΙΡΟ	ΝΟΎ	ΕΒΆΕΡΟ	ΔΙΑΛΛΕΙμΑ	ΔΙΆΛΥμΑ	ΑΒΛΩ	ΑΒΓΌ	ΑΒΓΌ	ΑΒΓΟΥΛΆΚΙΑ
GroupVowels	ΕΒΙΡΟ	ΝΟ	ΕΒΑΕΡΟ	ΔΙΑΛΛΙΜΑ	ΔΙΑΛΙμΑ	ΑΒΛΟ	ΑΒΓΟ	ΑΒΓΟ	ΑΒΓΟΛΑΚΙΑ
SoundexEncode	ε1@8&	ν$	ε19*8*	δ@966@79	δ@96@79	α16$	α12$	α12$	α12$695@9
RemoveDuplicates	ε1@8&	ν$	ε19*8*	δ@9679	δ@9679	α16$	α12$	α12$	α12$695@9
TrimLength	ε1@8	ν$00	ε19*	δ@96	δ@96	α16$	α12$	α12$	α12$

Πίνακας 6. : Παραδείγματα δημιουργίας κωδικών Soundex_{GR} , μέσω διαφόρων σταδίων

Για να συνοψίσουμε τους κανόνες που εφαρμόστηκαν, ο Πίνακας 2 παρουσιάζει τις ομάδες 2 γραμμαρίων που παράγουν *παρόμοιους ήχους* με ένα απλό γράμμα και ως αποτέλεσμα μετατρέπονται στο αντίστοιχο απλό γράμμα κατά την προεπεξεργασία της λέξης. Ο Πίνακας 3 παρουσιάζει το πλήρες σύνολο των φωνητικών κουτιών που εφαρμόζονται στη λέξη ως τελικό βήμα στην κωδικοποίηση της λέξης. Στον Πίνακα 4 παρουσιάζεται η κατηγορία *Loud* των συμφώνων στα ελληνικά, η οποία χρησιμοποιείται προκειμένου να εντοπιστεί αν το υ λειτουργεί ως σύμφωνο, συγκεκριμένα ως β , ενώ στον Πίνακα 5 παρουσιάζεται η κατηγορία Silent των συμφώνων στα ελληνικά, η οποία χρησιμοποιείται προκειμένου να εντοπιστεί αν το υ λειτουργεί ως σύμφωνο, συγκεκριμένα ως φ. Σημειώνεται ότι η διάκριση σε Loud και Silent αφορά φωνήματα συμφώνων. Τα αθόρυβα περιλαμβάνουν αυτά του Πίνακα 5 καθώς και τα γκ, μπ, ζτ, τζ, ωστόσο τα τρία τελευταία δεν είναι απαραίτητα για την κατανόηση της ερμηνείας του υ , και αυτός είναι ο λόγος που δεν περιλαμβάνονται στον Πίνακα 5.

3.4 Ο αλγόριθμος $\text{Soundex}^{naive}_{GR}$

Για λόγους συγκριτικής αξιολόγησης, εδώ ορίζουμε έναν άλλο αλγόριθμο, τον οποίο ονομάζουμε $^{Soundexnaive}_{GR}$, ο οποίος μοιράζεται τις ίδιες αρχές με τον αρχικό αλγόριθμο Soundex, αλλά χωρίς καμία προεπεξεργασία λέξης πριν από την κωδικοποίηση της λέξης. Συγκεκριμένα, ο αλγόριθμος αγνοεί τα φωνήεντα, έχει μήκος κωδικοποίησης 4 χαρακτήρες και δεν κωδικοποιεί το πρώτο γράμμα. Το μόνο κοινό στοιχείο μεταξύ αυτού του αλγορίθμου και του Soundex_{GR} είναι ότι χρησιμοποιεί τους ίδιους κάδους από τους οποίους παράγεται η τελική κωδικοποίηση, όπως φαίνεται στον πίνακα 3. Ομοίως με το αρχικό Soundex, υιοθετούμε τα ακόλουθα βήματα:

(i) κρατήστε το πρώτο γράμμα μη κωδικοποιημένο,
(ii) αφαιρεί όλες τις εμφανίσεις των α, ε, λ, π, υ, ο, ω εκτός αν εμφανίζονται ως το πρώτο γράμμα της λέξης,
(iii) αντικαταστήστε τα σύμφωνα μετά το πρώτο γράμμα όπως φαίνεται στον πίνακα 7,
(iv) αφαιρείτε τα γειτονικά διπλά ψηφία,
(v) να παράγει έναν κωδικό της μορφής Γράμμα Ψηφίο Ψηφίο Ψηφίο Ψηφίο Ψηφίο αγνοώντας τα ψηφία μετά το τρίτο (αν χρειάζεται) ή προσθέτοντας μηδενικά (αν χρειάζεται).

Για παράδειγμα, αυτός ο αλγόριθμος θα κωδικοποιούσε το αυγό σε α200 και το

αβγό σε α120, που είναι δύο πανομοιότυπες λέξεις, αλλά με διαφορετικά αποτελέσματα κωδικοποίησης. Αυτό αποδεικνύει την υπεροχή του Soundex_{GR} σε σύγκριση με τον ^{Soundexnaive} (περιλαμβάνονται περισσότερα τέτοια παραδείγματα στην ενότητα 4.1).

Πίνακας 7. : Αντικατάσταση συμφώνων στο $Soundexnaive_{GR}$

β, φ, π →	1
γ, χ →	2
τ, δ, θ →	3
ζ, σ, ς, ψ, ξ →	4
κ →	5
λ →	6
μ, ν →	7
ρ →	8.

3.5 Φωνητική αντιστοίχιση με $Soundex^{comp}_{GR}$

Με το $Soundex_{GR}$ θεωρούμε ότι δύο λέξεις w και w' *ταιριάζουν*, συμβολιζόμενες με $w \Leftrightarrow w'$, εάν έχουν τον ίδιο κωδικό, δηλαδή εάν $Soundex_{GR}(w) = Soundex_{GR}(w')$. Ανάλογα, με το $Soundexnaive_{GR}$.

Προκειμένου να διατηρηθούν όσο το δυνατόν υψηλότερα τα επίπεδα ακρίβειας και ανάκλησης, εισάγουμε εδώ μια άλλη παραλλαγή για τη φωνητική αντιστοίχιση, την οποία ονομάζουμε $Soundex^{comp}_{GR}$. Η ιδέα είναι να χρησιμοποιήσουμε τόσο το $Soundex_{GR}$ όσο και το $Soundexnaive_{GR}$ για να διατηρήσουμε τα επίπεδα ανάκλησης όσο το δυνατόν υψηλότερα, χωρίς να πέσει η ακρίβεια. Συγκεκριμένα, η μέθοδος αυτή χρησιμοποιεί τόσο το $Soundex_{GR}$ όσο και το $Soundexnaive_{GR}$ *σε συνδυασμό* κατά τη διαδικασία αντιστοίχισης, δηλαδή το ερώτημα και το κείμενο κωδικοποιούνται και με τις δύο υλοποιήσεις, και αν κάποια από αυτές ταιριάζει, τότε θεωρείται ότι υπάρχει ταύτιση, δηλαδή:

$$w \Leftrightarrow w' \text{ if } (Soundex_{GR}(w) = Soundex_{GR}(w')) \text{ OR } (Soundexnaive_{GR}(w) = Soundexnaive_{GR}(w'))$$

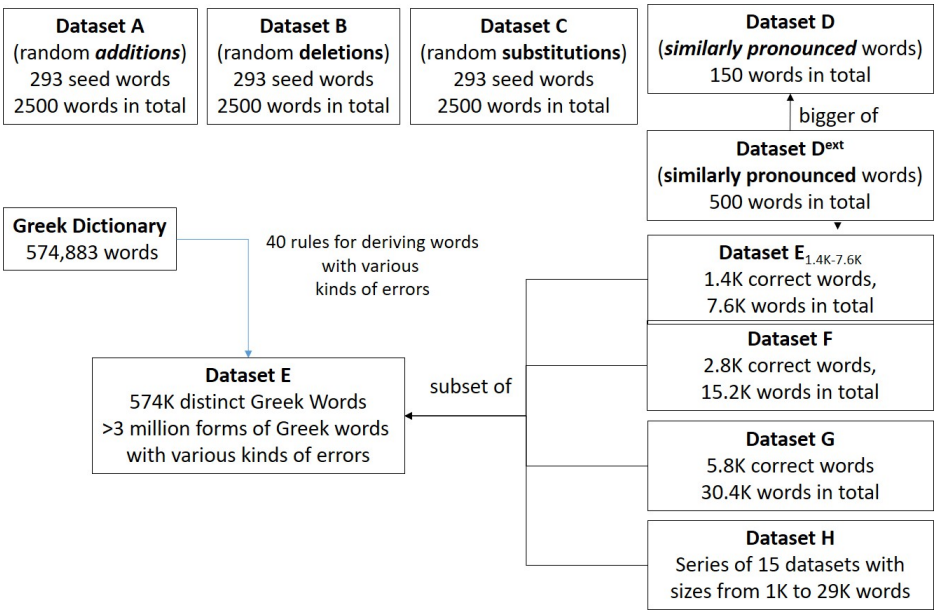
4. Αξιολόγηση

Αρχικά (στην §4.1) παραθέτουμε μερικά ενδεικτικά παραδείγματα που αναδεικνύουν τα πλεονεκτήματα των κωδικών και τις διαφορές μεταξύ $Soundexnaive_{GR}$ και $Soundex_{GR}$. Στη συνέχεια (στην §4.2) περιγράφουμε μια συλλογή αξιολόγησης που δημιουργήσαμε και περιέχει σύνολα δεδομένων (σύνολο δεδομένων A- σύνολο δεδομένων D) με διάφορους τύπους σφαλμάτων και τις μετρικές που χρησιμοποιούμε για τη σύγκριση της απόδοσης των διαφόρων επιλογών (στην §4.3). Στη συνέχεια (στην §4.4), αναφέρουμε τα αποτελέσματα της αξιολόγησης και συζητάμε τις σχετικές ανταλλαγές (στην §4.5). Για την περαιτέρω κατανόηση της απόδοσης αυτών των κωδίκων, τους συγκρίνουμε επίσης με τα λήμματα που παράγονται από τον ελληνικό stemmer (στην §4.6), και αναφέρουμε μετρήσεις πάνω σε ένα ελληνικό λεξικό (στην §4.7). Επιπλέον (στην §4.8), παρέχουμε και αξιολογούμε μια μέθοδο που αποδίδει μια πλήρη φωνητική μεταγραφή. Στην §4.9 συγκρίνουμε όλες τις μεθόδους, συμπεριλαμβανομένης της πλήρους φωνητικής μεταγραφής, καθώς και τις μεθόδους που βασίζονται στην Edit Distance, πάνω σε ένα εκτεταμένο σύνολο δεδομένων με λέξεις με παρόμοιο ήχο $Dataset^{Dext}$, ενώ στην §4.10 αναφέρουμε τα αποτελέσματα μιας σειράς πειραμάτων σε διαφορετικές κλίμακες για την κατανόηση των παραγόντων που καθορίζουν το βέλτιστο μήκος κώδικα (Dataset E-Dataset H). Στη συνέχεια (στην §4.11) συζητάμε την αποδοτικότητα και τέλος (στην §4.12) συζητάμε τη δυνατότητα εφαρμογής και περιγράφουμε μια εφαρμογή που

αναδεικνύει τα πλεονεκτήματα του Soundex$_{GR}$ για την προσεγγιστική αντιστοίχιση.

Μια επισκόπηση των συνόλων δεδομένων που χρησιμοποιούνται για σκοπούς αξιολόγησης δίνεται στο σχήμα

1.



Σχήμα **1**: Επισκόπηση των συνόλων δεδομένων που χρησιμοποιήθηκαν για σκοπούς αξιολόγησης

4.1 Ενδεικτικά παραδείγματα

Εδώ παραθέτουμε μερικά ενδεικτικά παραδείγματα για την κατανόηση της συμπεριφοράς του $Soundexnaive_{GR}$ και $Soundex_{GR}$. Συγκεκριμένα, ο πίνακας 8 παρέχει παραδείγματα όπου *τόσο* το $Soundexnaive_{GR}$ *όσο και* το $Soundex_{GR}$, αντιμετωπίζουν σωστά τις διάφορες ορθογραφικές παραλλαγές, δηλαδή αποδίδουν τον ίδιο κωδικό σε όλες τις παραλλαγές της λέξης.

Πίνακας 8. : Ενδεικτικά καλά παραδείγματα τόσο για το $Soundexnaive_{GR}$ όσο και για το $Soundex_{GR}$

λέξη		$Soundexnaive_{GR}$	$Soundex_{GR}$
Θάλασσα	→	θ740	θ969
θάλλασα	→	θ740	θ969
θάλασα	→	θ740	θ969
μήνυμα	→	μ880	μ@7@
μύνημα	→	μ880	μ@7@
μίνιμα	→	μ880	μ@7@
μοίνειμα	→	μ880	μ@7@
τζατζίκι	→	τ434	c94@
τσατζίκι	→	τ434	c94@
τσατσίκι	→	τ434	c94@
κορονοιός	→	κ!84	κ8
κοροναιός	→	κ!84	κ8
Γιάννης	→	γ840	γ@97
Γιάνης	→	γ840	γ@97
Γιάνννης	→	γ840	γ@97
αναδιατάσσω	→	α833	α793
αναδιέταξα	→	α833	α793

Τώρα ο πίνακας 9 παρέχει παραδείγματα όπου το $Soundexnaive_{GR}$ αποτυγχάνει να αποδώσει τον ίδιο κωδικό, ενώ

Το $Soundex_{GR}$ επιτυγχάνει να παρέχει τον ίδιο κωδικό σε όλες τις σχετικές παραλλαγές λέξεων.

Πίνακας 9. : Ενδεικτικά παραδείγματα όπου το $Soundexnaive_{GR}$ αποτυγχάνει ενώ το $Soundex_{GR}$ πετυχαίνει

λέξη		$Soundexnaive_{GR}$	$Soundex_{GR}$
αυγό	→	α200	α12$
αβγό	→	α120	α12$
αυγολάκια	→	α276	α12$
αβγά	→	α120	α129
αυγά	→	α200	α129
έτοιμος	→	έ384	ε3@7
αίτημος	→	α384	ε3@7
αύξων	→	α480	α14$
άφξον	→	ά148	α14$
εύδοξος	→	ε344	ε13$
εβδοξος	→	ε134	ε13$
θαύμα	→	θ800	θ917
θάβμα	→	θ180	θ917
θαυμαστικό	→	θ843	θ917
ξέρω	→	ξ!00	ξ*8$
κσαίρο	→	κ4!0	ξ*8$
οβελίας	→	ο174	ο1*6
ωβελύας	→	ω174	ο1*6
οβελίσκος	→	ο174	ο1*6
Βαγγέλης	→	β274	β95*
Βαγκέλης	→	β267	β95*
Βαγκαίλης	→	β267	β95*

4.2 Σύνολα δεδομένων αξιολόγησης (Σύνολα δεδομένων ***A-*** Σύνολα δεδομένων D***)***

Υπάρχουν διάφορα είδη σφαλμάτων, για περισσότερα δείτε την εκτενή έρευνα Kukich (1992), παρακάτω θα συνοψίσουμε τα κυριότερα. Τα ορθογραφικά λάθη που δημιουργούνται από τον άνθρωπο τείνουν μερικές φορές να αντικατοπτρίζουν τις γειτνιάσεις του πληκτρολογίου της γραφομηχανής, π.χ. η αντικατάσταση του "β" με το "ν" (στα ελληνικά β και ν). Ωστόσο, τα λάθη που εισάγονται από το OCR είναι πιο πιθανό να βασίζονται σε συγχύσεις που οφείλονται σε ομοιότητες μεταξύ των γραμμάτων (ανάλογα με τη γραμματοσειρά), π.χ. η αντικατάσταση του "Δ" από το "Ο" (στα ελληνικά μπορεί να συναντήσουμε ανάλογα προβλήματα με διάφορες ομάδες γραμμάτων όπως Ο,Θ,Χ, καθώς και Α Λ,Α, και Ε,Σ και Υ,Ψ). Μπορούμε επίσης να έχουμε τα λεγόμενα τυπογραφικά λάθη, π.χ. "spell" και "speel" (στα ελληνικά ιππότης και ιποότης), όπου θεωρείται ότι ο συγγραφέας γνωρίζει τη σωστή ορθογραφία αλλά απλώς κάνει ένα ολίσθημα κινητικού συντονισμού. Υπάρχουν επίσης γνωστικά λάθη, π.χ. "λαμβάνω" και "λαμβάνω" (στα ελληνικά φύλλο και φύλο που το καθένα έχει διαφορετική σημασία), τα οποία οφείλονται σε παρανόηση ή έλλειψη γνώσης εκ μέρους του συγγραφέα. Μπορούμε

επίσης να συναντήσουμε φωνητικά λάθη, π.χ. "άβυσσος" και "άβυσσος" (στα ελληνικά μήνυμα και μύνημα, δικλίδα και δικλείδα, καλύτερα και καλλίτερα), που αποτελούν μια ειδική κατηγορία γνωστικών λαθών, κατά την οποία ο συγγραφέας αντικαθιστά την προβλεπόμενη λέξη με μια φωνητικά σωστή αλλά ορθογραφικά λανθασμένη ακολουθία γραμμάτων.

Εκτός από τα λάθη, υπάρχουν λέξεις με περισσότερους από έναν σωστούς τύπους, π.χ. αυγό και αβγό, και το ίδιο ισχύει και για τα ονόματα οντοτήτων, για παράδειγμα η πόλη του Ηρακλείου γράφεται ως Ηράκλειο αλλά και ως Ηράκλειον, ενώ η πόλη της Αθήνας γράφεται Αθήνα αλλά και Αθήναι.

Συνολικά, σύμφωνα με τον Kukich (1992), σχεδόν το 80% των προβλημάτων των ανορθόγραφων λέξεων μπορεί να αντιμετωπιστεί είτε με την προσθήκη ενός μόνο γράμματος, είτε με την αντικατάσταση ενός μόνο γράμματος, είτε με την ανταλλαγή γραμμάτων. Όπως προτείνουν οι συγγραφείς του Koneru et al. (2016) στην αξιολόγηση διαφόρων αλγορίθμων φωνητικής αντιστοίχισης, παρέχουμε μια παρόμοια συλλογή αξιολόγησης για την ελληνική γλώσσα που αποτελείται από σύνολα δεδομένων που περιέχουν λέξεις που αντιστοιχούν σε διάφορα είδη λαθών. Συγκεκριμένα, παρακάτω περιγράφουμε καθένα από τα τέσσερα σύνολα δεδομένων αξιολόγησης που δημιουργήσαμε. Το σύνολο των λέξεων σε κάθε ένα από αυτά τα σύνολα δεδομένων περιέχει ρήματα, ουσιαστικά, επίθετα
και ονόματα. Τα 3 πρώτα σύνολα δεδομένων, σύνολο δεδομένων Α, σύνολο δεδομένων Β, σύνολο δεδομένων Γ, δημιουργήθηκαν
για τον έλεγχο της συμπεριφοράς των αλγορίθμων *σε διάφορα είδη σφαλμάτων* (προσθήκες, διαγραφές και αντικαταστάσεις) που μπορεί να προκύψουν σε μια λέξη, ενώ το τελευταίο, το σύνολο δεδομένων D, δημιουργήθηκε για την αξιολόγηση *κουτιών γραμμάτων*, δηλαδή για τον έλεγχο της συμπεριφοράς της αντιστοίχισης σε κοινά σφάλματα.

Συγκεκριμένα, το σύνολο δεδομένων A περιέχει λέξεις που παράγονται με την *προσθήκη* ενός *τυχαίου γράμματος* σε μια τυχαία θέση σε μια λέξη, π.χ. από το σύνολο των λέξεων ακρίδα, ωράριο, επιρρεπείς παράγουμε λέξεις όπως ακρπίδα, ωρλάριο, επιτρρεπείς. Λάθη αυτού του είδους μπορούν να συμβούν με την πληκτρολόγηση ενός επιπλέον πλήκτρου. Στο σύνολο δεδομένων B χρησιμοποιείται η ίδια διαδικασία για *τις διαγραφές*, δηλαδή διαγράφεται ένα γράμμα από μια τυχαία θέση, για το ίδιο σύνολο λέξεων, π.χ. αυτό το σύνολο δεδομένων περιέχει λέξεις όπως ακίδα, ωράρο, επιρρπείς. Και πάλι λάθη αυτού του είδους μπορούν να συμβούν κατά τη διάρκεια της πληκτρολόγησης, δηλαδή από ένα χαμένο πλήκτρο ή ένα τυπογραφικό λάθος (χαμένο διπλό γράμμα). Στο σύνολο δεδομένων Γ, έχουμε τυχαία *αντικατάσταση* γραμμάτων σε τυχαία θέση, π.χ. στο παράδειγμά μας, έχουμε λέξεις όπως αδρίδα, ωράριν, εψιρρεπείς. Και πάλι λάθη αυτού του είδους μπορούν να συμβούν κατά τη διάρκεια της πληκτρολόγησης με ένα λάθος πάτημα του πλήκτρου (θυμηθείτε τις ασυναρτησίες του πληκτρολογίου, τα λάθη OCR, τα τυπογραφικά και τα γνωστικά λάθη).

Κάθε ένα από τα παραπάνω σύνολα δεδομένων περιέχει 2.500 λέξεις, οι οποίες παράγονται από τις ίδιες 293 μοναδικές λέξεις, δηλαδή συνολικά 7.500 λέξεις. Η παραγωγή των λανθασμένων λέξεων, είναι τυχαία, δηλαδή δεν λαμβάνει υπόψη κανένα πλαίσιο ή αναμενόμενα λάθη ή τυπογραφικά λάθη. Τέλος, το σύνολο δεδομένων D περιέχει 150 λέξεις που περιλαμβάνουν ομάδες από λέξεις που προφέρονται με παρόμοιο τρόπο, όπως πολύ, πολλοί, πολλή, πωλεί και φύλλο, φίλο, φύλο, που δημιουργήθηκαν χειροκίνητα. Το κίνητρο για τη δημιουργία αυτού του συνόλου δεδομένων ήταν να καταγραφούν ορισμένα κοινά λάθη, δηλαδή συχνά εμφανιζόμενα ορθογραφικά λάθη.

4.3 Μετρικές αξιολόγησης

Θα χρησιμοποιήσουμε δύο βασικές μετρικές για την αξιολόγηση της αποτελεσματικότητας των αλγορίθμων, δηλαδή την Ακρίβεια και την Ανάκληση. Η ακρίβεια είναι το ποσοστό των λέξεων που ανακτήθηκαν και είναι σχετικές με το ερώτημα, ενώ η ανάκληση είναι το ποσοστό των σχετικών λέξεων που ανακτήθηκαν, τυπικά:

$Ακρίβεια = \frac{|(σχετικό) \cap (ανακτήθηκε)|}{|(ανακτήθηκε)|}$, $Ανάκληση = \frac{|(σχετικό) \cap (ανακτήθηκε)|}{|(σχετικό)|}$. Let us now εξηγήστε what "ερώτημα", "ανακτήθηκε" και "σχετικό" σημαίνουν στο πλαίσιο μας. Κάθε μία από τις 293 μοναδικές λέξεις (των τριών πρώτων συνόλων δεδομένων) θεωρείται ως ερώτημα. Για κάθε τέτοια λέξη *w*, το αντίστοιχο σύνολο λέξεων σε κάθε σύνολο δεδομένων, δηλαδή οι λέξεις που προκύπτουν με μία τροποποίηση, θεωρείται ως το σύνολο των σχετικών λέξεων.

Για παράδειγμα, για τη λέξη ακρίδα το σύνολο των σχετικών λέξεων είναι: αγκρίδα, ακτρίδα, ακρφίδα (από το σύνολο δεδομένων A), ακρίδ, ακρία, κρίδα (από το σύνολο δεδομένων B) και ακίδα, ακρίφα, εκρίδα (από το σύνολο δεδομένων Γ). Για κάθε λέξη ερώτησης, το σύνολο των ανακτημένων λέξεων θεωρείται το σύνολο *όλων των λέξεων σε όλα τα σύνολα δεδομένων* που έχουν τον ίδιο κωδικό. Στη συνέχεια, για κάθε σύνολο δεδομένων ξεχωριστά, υπολογίζουμε τη μέση ακρίβεια και τη μέση ανάκληση, με βάση την ανάκληση και την ακρίβεια των κάθε ένα από τα N ερωτήματα, δηλαδή $\overline{ακρίβεια}_{avg} = \frac{\Sigma_{i=1}^{N} Ακρίβεια_i}{N}$ και $ανάκληση_{avg} = \frac{\Sigma_{i=1}^{N} Ανάκληση_i}{N}$.

4.4 Αποτελέσματα αξιολόγησης σε σύνολο δεδομένων Α **-** σύνολο δεδομένων Δ

Αρχικά πρέπει να σημειώσουμε ότι αν αντί να εφαρμόσουμε κάποιον αλγόριθμο προσέγγισης, εφαρμόσουμε ακριβή αντιστοίχιση, τότε προφανώς έχουμε ακρίβεια ίση με 1, αλλά η ανάκληση είναι πολύ χαμηλή (περίπου 0,1), καθώς μόνο 1 από τις "σχετικές" λέξεις ανακτάται (φυσικά όσο μεγαλύτεροι είναι οι κάδοι της ομάδας λέξεων στα σύνολα δεδομένων αξιολόγησης, τόσο μικρότερη γίνεται η ανάκληση).

Στο σύνολο δεδομένων Α (η συλλογή *προσθήκης* γραμμάτων), το Soundex_{GR} πέτυχε ακρίβεια 0,83 και ανάκληση, ενώ $\text{Soundex}^{naive}_{GR}$ 0,80 και 0,45 αντίστοιχα, ενώ $\text{Soundex}^{comp}_{GR}$ πέτυχε ακρίβεια 0,74 και ανάκληση 0,75, όπως φαίνεται στο Σχήμα 2 (για την ακρίβεια) και στο Σχήμα 3 (για την ανάκληση).

Στο σύνολο δεδομένων Β (η συλλογή *διαγραφής* γραμμάτων), το $\text{Soundex}^{naive}_{GR}$ είχε μια μικρή πτώση της ακρίβειας στο 0,75 και μια αύξηση της ανάκλησης που έφτασε στο 0,57, ενώ το Soundex_{GR} παρέμεινε στα ίδια επίπεδα, με 0,82 και 0,45 αντίστοιχα. Το $\text{Soundex}^{comp}_{GR}$ διατήρησε υψηλό επίπεδο ακρίβειας 0,70 και πέτυχε την υψηλότερη ανάκληση 0,68, όπως βλέπε στο Σχήμα 2 (ακρίβεια) και στο Σχήμα 3 (ανάκληση). Η πτώση της ακρίβειας του $\text{Soundex}^{naive}_{GR}$, με την αύξηση της ανάκλησης, είναι αναμενόμενη, δεδομένου ότι το $\text{Soundex}^{naive}_{GR}$ αγνοεί ορισμένα γράμματα και επομένως μπορεί να χειριστεί καλύτερα τη διαγραφή ενός γράμματος, ενώ το Soundex_{GR} είναι πιο άκαμπτο σε τέτοια λάθη.

Στο σύνολο δεδομένων Γ (η συλλογή *υποκατάστασης* γραμμάτων), το $\text{Soundex}^{naive}_{GR}$ πέτυχε ακρίβεια 0,69 και ανάκληση 0,34. Οι χαμηλότερες βαθμολογίες οφείλονται στο στενότερο σύνολο φωνητικών κανόνων. Από την άλλη πλευρά, παρά την πτώση των βαθμολογιών, ο αλγόριθμος Soundex_{GR} διατήρησε το ίδιο επίπεδο βαθμολογίας και στα τρία σύνολα, με ακρίβεια 0,80 και ανάκληση 0,39. Στην αντικατάσταση, ο $\text{Soundex}^{comp}_{GR}$ δεν κατάφερε να κάνει τη διαφορά, καθώς συνδύασε τα καλύτερα αποτελέσματα του Soundex_{GR} με τα χειρότερα του $\text{Soundex}^{naive}_{GR}$, επιτυγχάνοντας ακρίβεια 0,67 και ανάκληση 0,49, όπως φαίνεται στο Σχήμα 2 (ακρίβεια) και στο Σχήμα 3 (ανάκληση). Γενικά οι αλγόριθμοι συμπεριφέρονται καλύτερα όταν το σφάλμα είναι συνηθισμένο στην κοινή Ελληνική Γλώσσα, δηλαδή η λέξη εξακολουθεί να ακούγεται ως η σωστή.

Στο σύνολο δεδομένων D, τη συλλογή από λέξεις *με παρόμοια προφορά*, η οποία περιλαμβάνει τις κύριες περιπτώσεις που πρέπει να μπορεί να αντιμετωπίσει ένας φωνητικός αλγόριθμος, τόσο ο $\text{Soundex}^{naive}_{GR}$, όσο και ο Soundex_{GR} έλαβαν παρόμοια υψηλά σκορ, συγκεκριμένα ο $\text{Soundex}^{naive}_{GR}$ πέτυχε ακρίβεια 0,88 και ανάκληση 0,92, ενώ ο Soundex_{GR} πέτυχε ακρίβεια 0,96 και ανάκληση 0,98, όπως φαίνεται στο Σχήμα 2 (ακρίβεια) και στο σχήμα 3 (ανάκληση). Ο συνδυασμός των ανωτέρω αλγορίθμων, δηλ. $\text{Soundex}^{comp}_{GR}$, καταφέρνει να διατηρήσει τις υψηλές βαθμολογίες, συγκεκριμένα ακρίβεια 0,86 και ανάκληση 0,98, καθώς οι βαθμολογίες i ts εξαρτώνται από τις δύο υλοποιήσεις. Αυτές οι βαθμολογίες δείχνουν ότι η κουβάδες είναι επαρκείς, με το Soundex_{GR} να έχει ελαφρώς μεγαλύτερη ακρίβεια και ανάκληση.

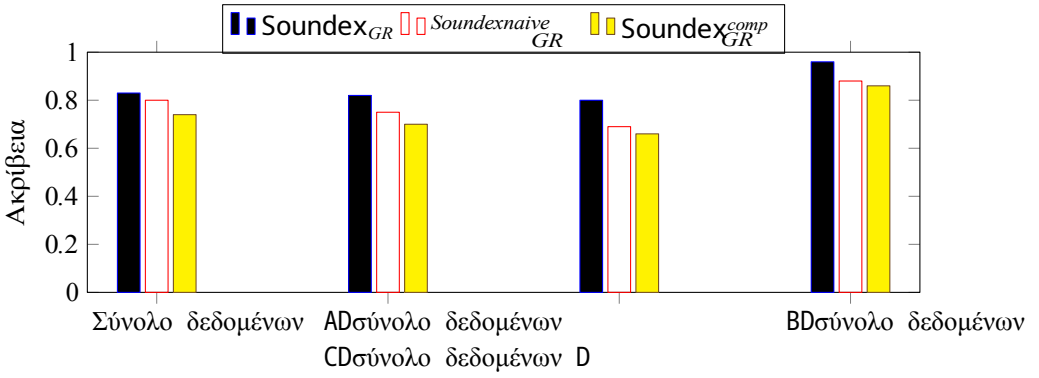
Για να συνοψίσουμε τα αποτελέσματα, μπορούμε να δούμε στο Σχήμα 4, ότι το $\text{Soundex}^{naive}_{GR}$ επιτυγχάνει F-Score (σημειώστε ότι το F-Score, ή αλλιώς F-Measure, είναι ο αρμονικός μέσος όρος της ακρίβειας και της ανάκλησης, δηλαδή $\text{F- Score} = 2\frac{Precision * Recall}{Precision + Recall}$) ίσο με 0,57, 0,65, 0,46 και 0,90 στο σύνολο δεδομένων Α, στο σύνολο δεδομένων Β, στο σύνολο δεδομένων Γ και στο σύνολο δεδομένων Δ αντίστοιχα. Το Soundex_{GR} επιτυγχάνει βαθμολογίες F-Score ίσες με 0,56, 0,58, 0,53 και 0,97 αντίστοιχα και ο συνδυασμός των δύο $\text{Soundex}^{comp}_{GR}$ επιτυγχάνει 0,64, 0,69, 0,56 και 0,91 αντίστοιχα, γεγονός που δείχνει ότι το $\text{Soundex}^{comp}_{GR}$ είναι h aves καλύτερα γενικά.

Τόσο το Soundex_{GR} όσο και το $\text{Soundex}^{naive}_{GR}$ πέτυχαν παρόμοια αποτελέσματα.

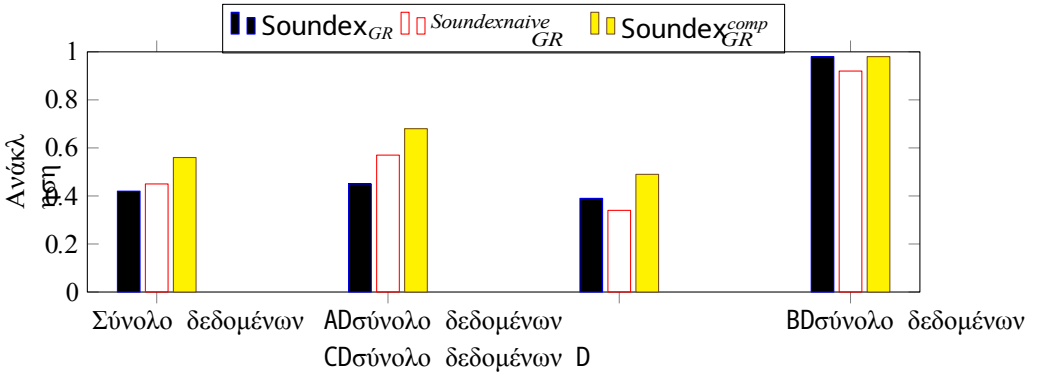
Λειτουργούν καλά όταν το σφάλμα δεν μεταβάλλει τον παραγόμενο κώδικα σε ένα κρίσιμο σημείο για τον κώδικα. Και τα δύο κρίσιμα σημεία θα ήταν κάτω από 4 χαρακτήρες και το σφάλμα που αφορά ένα σύμφωνο για το $Soundex^{naive}$ και ένα τυχαίο, απροσδόκητο σύμφωνο ή φωνήεν που δεν αντιμετωπίζεται κατά την προεπεξεργασία της λέξης για το Soundex_{GR} . Δεδομένου ότι το Soundex^{comp} περιλαμβάνει και τις δύο υλοποιήσεις στο

διαδικασία ανάκτησης, μοιράζεται τα ίδια προβλήματα, αλλά το ^{GR} καταφέρνει να έχει υψηλότερες τιμές ανάκλησης ενώ

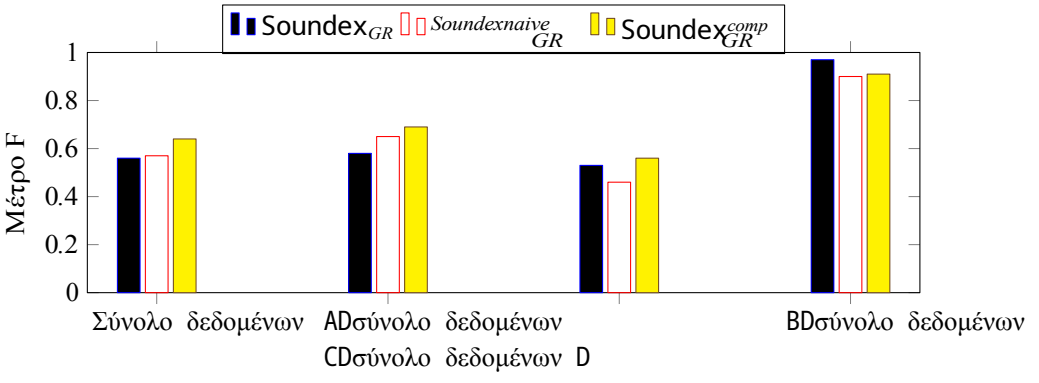
χωρίς μεγάλες απώλειες στην ακρίβεια. Η χρήση και των δύο κωδικών μπορεί να αυξήσει τα επίπεδα ανάκλησης κατά 0,05 έως 0,20, ενώ η ακρίβεια υφίσταται πτώση από 0,10 έως 0,20, σε σύγκριση με το Soundex_{GR} . Οι αλγόριθμοι λειτουργούν καλά στην ανάκτηση λέξεων, εάν το λάθος σε μια λέξη βασίζεται στους ίδιους φωνητικούς κανόνες (του πίνακα 3) ή συλλαμβάνονται στο στάδιο της προεπεξεργασίας, όταν κάνουμε τόσο το ερώτημα όσο και το κείμενο όσο το δυνατόν πιο λανθασμένα προφερόμενα, ειδικά το Soundex_{GR} . Για παράδειγμα, για



Σχήμα **2**: Επίπεδα ακρίβειας για κάθε συλλογή.



Σχήμα **3**: Επίπεδα ανάκλησης για κάθε συλλογή.



Σχήμα **4**: Επίπεδα μέτρησης F για κάθε συλλογή.

ένα ερώτημα όπως κατεβαίzω, θα ανακτούσε σωστά κατεβάzω, καταlβαίzω, κατεβαίzο, κατεμπάzο, κατεβεzο, κατεββαίzω, αλλά όχι κατβαίzω, κτεβαίzω, ρατεβαίzω. Αυτό συμβαίνει επειδή, η προσθήκη/διαγραφή/αντικατάσταση ενός και μόνο γράμματος θα αλλάξει τον κωδικό Soundex και ο Soundex δεν έχει μετρική ομοιότητας στη διαδικασία σύγκρισης.

4.5 Συζήτηση για το αποκαλυπτόμενο εμπόριο ως προς το μήκος των κωδίκων (πάνω από Σύνολο δεδομένων Α **-** Σύνολο δεδομένων Δ*)*

Κατά τη δοκιμή του αλγορίθμου παρατηρήσαμε ότι απλές αλλαγές επηρεάζουν την ακρίβεια και την ανάκληση που επιτυγχάνονται. Για παράδειγμα, η αλλαγή του *μήκους* της κωδικοποίησης του Soundex$_{GR}$, από 4 σε 6 θα βελτίωνε σημαντικά την ακρίβεια από 0,80 σε πάνω από 0,90, ενώ η ανάκληση θα μειωνόταν από 0,40-
0,45 έως 0,25-0,30. Παρόλο που οι αλγόριθμοι Soundex χρησιμοποιούνται κυρίως σε περιβάλλον όπου η ανάκληση έχει μεγαλύτερη σημασία, είναι συνετό να επιλέγεται ο αλγόριθμος που ταιριάζει καλύτερα στις απαιτήσεις του πλαισίου εφαρμογής, δηλαδή αν θα πρέπει να δοθεί έμφαση στην ακρίβεια ή στην ανάκληση. Παρατηρήσαμε επίσης ότι αφήνοντας το πρώτο γράμμα χωρίς κωδικοποίηση, όπως το αρχικό Soundex, έχουμε μια μικρή αύξηση της ακρίβειας (κατά 0,05-0,10) και μια μείωση της ανάκλησης κατά
0.05. Τέλος, ο διαχωρισμός όλων των γραμμάτων σε περισσότερες κατηγορίες θα αύξανε επίσης την ακρίβεια και θα μείωνε την ανάκληση.

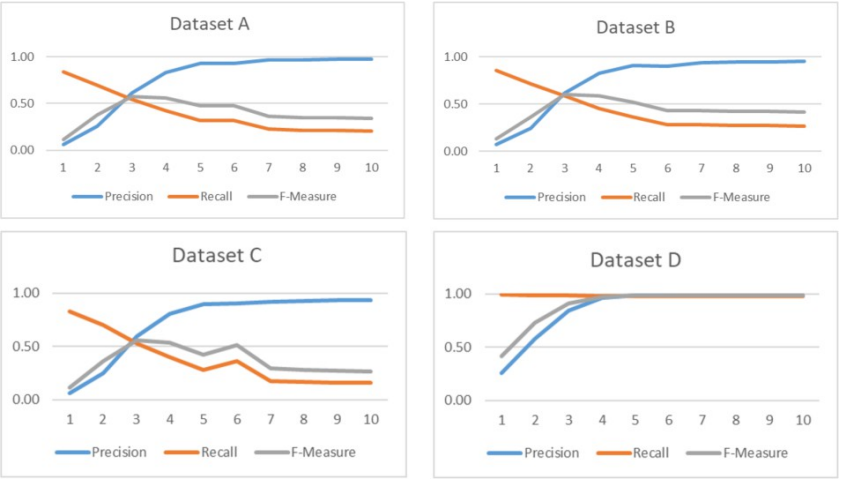
Για να γίνει καλύτερα κατανοητό πώς το μήκος των κωδικών Soundex$_{GR}$ επηρεάζει το λαμβανόμενο F-
Score, υπολογίσαμε το F-Score σε όλα τα σύνολα δεδομένων για το μήκος του κώδικα από 1 έως 10 και το μήκος 15. Τα αποτελέσματα παρουσιάζονται στον πίνακα 10. Στη δεξιά στήλη παρουσιάζεται ο μέσος όρος του F-Score για κάθε ένα από τα τέσσερα σύνολα δεδομένων. Βλέπουμε ότι το μήκος 4 αποδίδει το καλύτερο μέσο F-Score.

Πίνακας 10. : Μέσο F-Score (για τα σύνολα δεδομένων Α, Β, Γ και Δ) για διαφορετικά μήκη Soundex$_{GR}$

Soundex$_{GR}$	F-Score				
Μήκος	Σύνολο δεδομένων Α	Σύνολο δεδομένων Β	Σύνολο δεδομένων C	Σύνολο δεδομένων D	Μέσο F-Score
1	0.11	0.12	0.11	0.41	0.18
2	0.37	0.36	0.36	0.72	0.45
3	**0.57**	**0.60**	**0.56**	0.90	0.65
4	0.56	0.58	0.53	0.97	**0.66**
5	0.47	0.51	0.42	**0.98**	0.59
6	0.40	0.45	0.35	**0.98**	0.54
7	0.36	0.43	0.29	**0.98**	0.51
8	0.35	0.42	0.28	**0.98**	0.50
9	0.34	0.41	0.27	**0.98**	0.50
10	0.34	0.41	0.26	**0.98**	0.49
15	0.34	0.41	0.26	**0.98**	0.49

Για να γίνει καλύτερα κατανοητό πώς η ακρίβεια και η ανάκληση επηρεάζονται από το μήκος του κώδικα, το Σχήμα 5 δείχνει για κάθε σύνολο δεδομένων την ακρίβεια, την ανάκληση και το F-Score για κάθε μήκος από 1 έως
10. Στα σύνολα δεδομένων που αντιστοιχούν σε διάφορα είδη σφαλμάτων, δηλαδή στο σύνολο δεδομένων Α (η συλλογή *προσθήκης* γραμμάτων), στο σύνολο δεδομένων Β (η συλλογή *διαγραφής* γραμμάτων) και στο σύνολο δεδομένων Γ (η συλλογή *αντικατάστασης* γραμμάτων), βλέπουμε καθαρά ότι καθώς αυξάνεται το μήκος του κώδικα, η ακρίβεια
αυξάνεται αλλά η ανάκληση μειώνεται. Το μήκος κωδικού όπου το F-Score μεγιστοποιείται σε αυτά τα τρία σύνολα δεδομένων είναι 3. Στο σύνολο δεδομένων D (η συλλογή από λέξεις *με παρόμοια προφορά*) βλέπουμε ότι καθώς αυξάνεται το μήκος, αυξάνεται και η ακρίβεια, φτάνοντας το μέγιστο στο μήκος 5. Το επίπεδο ανάκλησης δεν μειώνεται καθώς αυξάνεται το

μήκος του κωδικού (όπως συμβαίνει στα προηγούμενα τρία σύνολα δεδομένων), επειδή, ακόμη και με μεγάλο μήκος κωδικού, το σύνολο όλων των σχετικών λέξεων είναι αυτές που ακούγονται το ίδιο και όλες τους ανακτώνται επειδή το Soundex_{GR} καταφέρνει να τους αποδώσει τον ίδιο κωδικό. Σε αυτό το σύνολο δεδομένων το μήκος που μεγιστοποιεί το F-Score είναι 5 και οποιοδήποτε μεγαλύτερο μήκος.



Σχήμα **5**: Μετρικές αξιολόγησης Precision, Recall και F-Score στο σύνολο δεδομένων Α (πάνω αριστερά), στο σύνολο δεδομένων Β (πάνω δεξιά), στο σύνολο δεδομένων Γ (κάτω αριστερά) και στο σύνολο δεδομένων Δ (κάτω δεξιά) για τους κωδικούς Soundex_{GR} μήκους 1 έως 10.

Περισσότερα πειράματα σχετικά με την επιλογή του μήκους των κωδίκων, δίνονται και αναλύονται στην ενότητα 4.10.

4.6 Σύγκριση με το Stemming

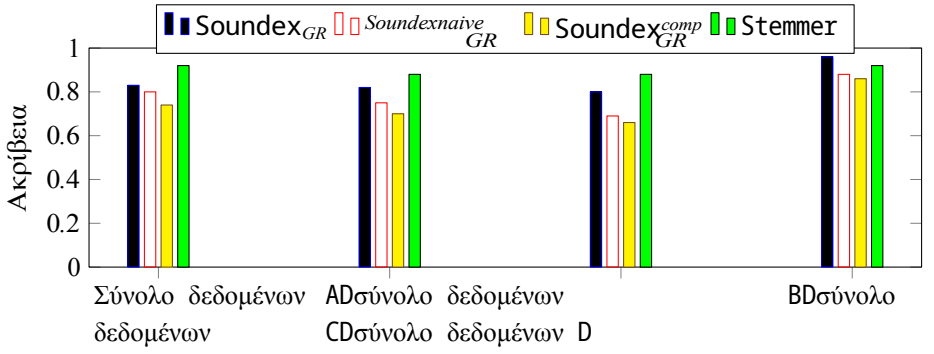
Εκτός από τη σύγκριση των διαφόρων παραλλαγών του Soundex_{GR} , αποφασίσαμε να συγκρίνουμε την ομαδοποίηση των λέξεων που προκύπτει μέσω του Soundex_{GR} , με την ομαδοποίηση που προκύπτει από ένα Stemmer για την ελληνική γλώσσα. Σε γενικές γραμμές, *το stemming* αναφέρεται στη διαδικασία αναγωγής των κλιτών (ή παράγωγων) λέξεων στη βασική ή ριζική τους μορφή. Σημειώστε ότι το στέλεχος δεν είναι απαραίτητα η μορφολογική ρίζα της λέξης, με την έννοια ότι αν δύο συγγενείς λέξεις αντιστοιχούν στο ίδιο στέλεχος, τότε ακόμη και αυτό το στέλεχος δεν είναι έγκυρη $^{\text{ρότα}}$, αρκεί για το έργο της αντιστοίχισης και της ανάκτησης. Κατά συνέπεια, το δυνατό σημείο της χρήσης ενός stemmer για το πρόβλημα της αντιστοίχισης είναι ότι μπορεί να εντοπίσει με επιτυχία μορφολογικές παραλλαγές της ίδιας λέξης, και επομένως μπορεί να αντιστοιχίσει μορφές λέξεων που είναι ορθογραφικά και φωνητικά αρκετά διαφορετικές, ωστόσο το αδύνατο σημείο της χρήσης ενός stemmer για την αντιστοίχιση είναι ότι δεν μπορεί να αντιμετωπίσει τυπογραφικά λάθη (οι stemmer δεν έχουν σχεδιαστεί για την αντιμετώπιση τυπογραφικών λαθών) και δεν μπορεί να εφαρμοστεί σε ονομαστικές οντότητες (πρόσωπα, διευθύνσεις, τόπους, εταιρείες κ.λπ.).

Χρησιμοποιήσαμε έναν stemmer της ελληνικής γλώσσας, συγκεκριμένα τον Mitos Greek Stemmer (Karamaroudis and Markidakis (2006)) που περιγράφεται στους Papadakos et al. (2008), και τον εφαρμόσαμε στα ίδια σύνολα δεδομένων. Τα αποτελέσματα για την ακρίβεια, την ανάκληση και το F-Score παρουσιάζονται στα Σχήματα 6, 7 και 8 αντίστοιχα.

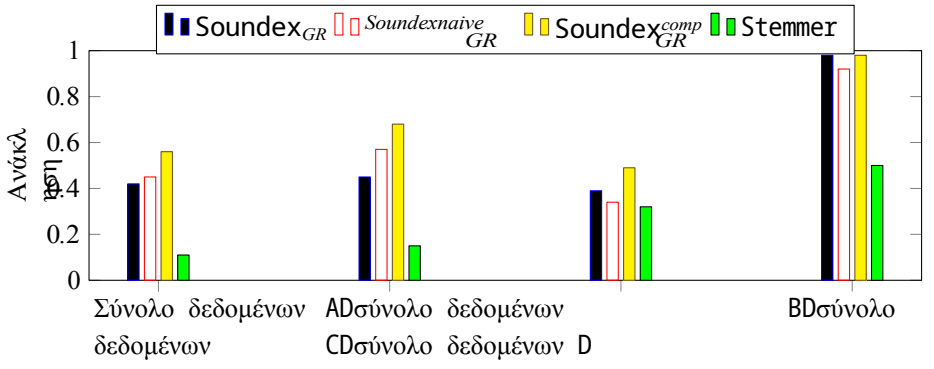
Μπορούμε να δούμε ότι το stemming έχει υψηλότερη ακρίβεια (όπως αναμενόταν), δηλαδή αν δύο λέξεις έχουν το ίδιο στέλεχος, τότε με μεγάλη πιθανότητα ανήκουν στην ίδια κατηγορία λέξεων, ωστόσο η ανάκληση είναι πολύ χαμηλή (όπως αναμενόταν), καθώς δεν μπορεί να αντιμετωπίσει ορθογραφικά λάθη που ακούγονται το ίδιο. Κατά συνέπεια, το stemming έχει φτωχό F-Score σε σύγκριση με το Soundex_{GR} - μόνο στο σύνολο δεδομένων C το stemming έχει συγκρίσιμες επιδόσεις (με επιδόσεις παρόμοιες

με αυτές των

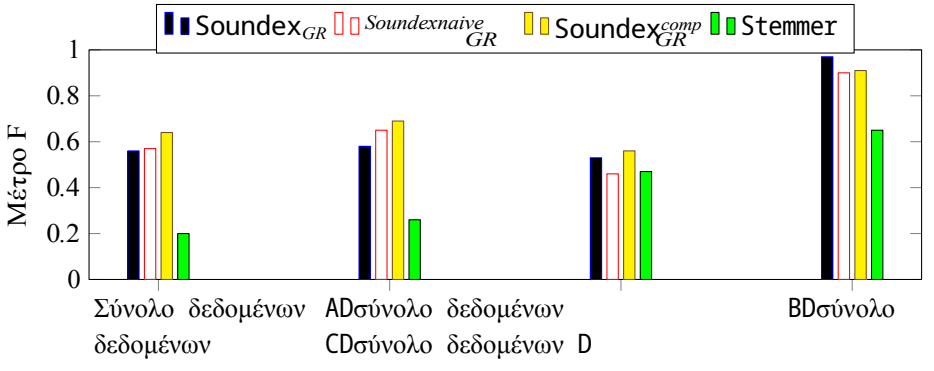
[a]όπως συμβαίνει για την αγγλική γλώσσα με το Porter stemmer `https://tartarus.org/martin/PorterStemmer/` για την αγγλική γλώσσα



Σχήμα **6**: Επίπεδα ακρίβειας για κάθε συλλογή (επίσης για το stemming)



Σχήμα **7**: Επίπεδα ανάκλησης για κάθε συλλογή (επίσης για το stemming)



Σχήμα **8**: Επίπεδα F-measure για κάθε συλλογή (επίσης για stemming)

του $^{Soundexnaive}_{GR}$). Συνολικά, το Soundex$_{GR}$ είναι σημαντικά καλύτερο για το συγκεκριμένο πρόβλημα, σε σύγκριση με τη χρήση ενός συνηθισμένου stemmer.

Τέλος, θα πρέπει να σημειώσουμε ότι δοκιμάσαμε επίσης το σενάριο όπου πρώτα εφαρμόζουμε το stemming και στη συνέχεια εφαρμόζουμε το soundex (πάνω στις λέξεις με το stemming), ωστόσο τα αποτελέσματα ήταν χειρότερα.

Περισσότερα συγκριτικά πειράματα με την αντιστοίχιση με βάση το stemmer παρατίθενται στην ενότητα 4.9, καθώς και στη σειρά πειραμάτων που περιγράφονται στην ενότητα 4.10.

Ένα λεξικό δεν είναι ένα είδος συνόλου δεδομένων για την αξιολόγηση φωνητικών αλγορίθμων, δεδομένου ότι δεν περιέχει ούτε ανορθόγραφες λέξεις, ούτε επώνυμα προσώπων, ονόματα τοποθεσιών κ.λπ. Ωστόσο, αποφασίσαμε να πραγματοποιήσουμε κάποιες μετρήσεις για να πάρουμε μια ιδέα για την κατανομή των κωδικών

(και για τη μέτρηση της αποδοτικότητας). Για το σκοπό αυτό, χρησιμοποιήσαμε το λεξικό Unicode WinEdt για τα ελληνικάb. Το εν λόγω λεξικό περιέχει ελληνικές λέξεις και τις μορφολογικές παραλλαγές τους, καθώς και ονόματα γροσίων και ακρωνύμια, π.χ. περιέχει τα Γιάννης, Γιάννη, ΑΕΙ. Στην πραγματικότητα είναι ένας κατάλογος λέξεων και συνολικά περιέχει περισσότερες από μισό εκατομμύριο ελληνικές λέξεις (συγκεκριμένα 574.883). Ο συνολικός αριθμός των χαρακτήρων αυτών των λέξεων είναι 6.279.813, επομένως το μέσο μέγεθος της λέξης είναι 10,92 χαρακτήρες και η μικρότερη λέξη (ή οι μικρότερες λέξεις) έχει μήκος 3, ενώ η μεγαλύτερη έχει μήκος 27 (στρογγυλοκουλουριαζόντουσαν).

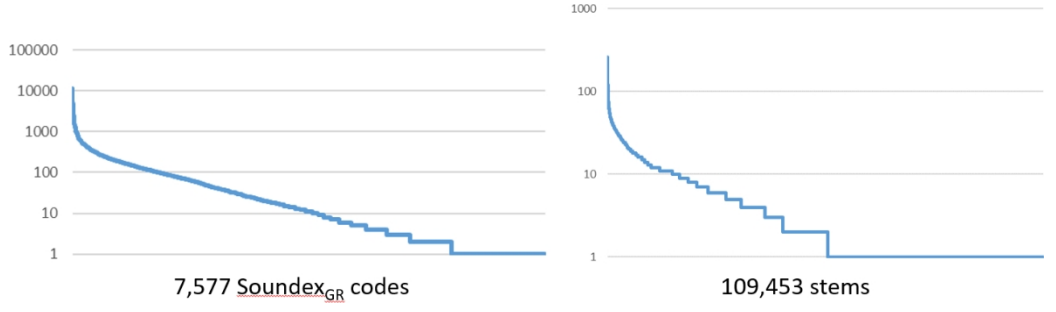
Δεδομένου ότι ο μέσος αριθμός χαρακτήρων ανά λέξη είναι 10,92, ενώ κάθε κωδικός Soundex_{GR} αποτελείται από 4 χαρακτήρες, το μέγεθος αυτών των κωδικών αντιστοιχεί στο 36% του μεγέθους του αρχικού λεξικού (ή θα έχουμε 36% αύξηση του μεγέθους του λεξικού αν αποφασίσουμε να αποθηκεύσουμε επίσης τον κωδικό Soundex_{GR} για κάθε λέξη). Χρησιμοποιώντας το stemmer που αναφέραμε στην §4.6, το μέσο μέγεθος του στελέχους είναι 7,46. Αυτό σημαίνει ότι το μέγεθος αυτών των στελεχών αντιστοιχεί στο 68% του μεγέθους του αρχικού λεξικού (ή θα έχουμε 68% αύξηση του μεγέθους του λεξικού αν αποφασίσουμε να αποθηκεύσουμε και το στέλεχος για κάθε λέξη).

Ο αριθμός των διακριτών κωδικών Soundex_{GR} είναι 7.577, δηλαδή κατά μέσο όρο κάθε κωδικός αντιστοιχεί σε 574.883 / 7.577 = 75,87 λέξεις. Ο αριθμός των διακριτών στελεχών είναι 109.453, δηλαδή κάθε στέλεχος αντιστοιχεί σε 574.883 / 109.453 = 5,25 λέξεις. Σε σύγκριση με το Soundex_{GR} , ο αριθμός των λημμάτων είναι 109.453 / 7.577 = 14,44 φορές μεγαλύτερος από τον αριθμό των κωδικών Soundex_{GR} .

Η κατανομή ούτε των κωδικών Soundex_{GR} ούτε των στελεχών είναι ομοιόμορφη, όπως αναμενόταν.

Υπάρχουν κωδικοί με μία μόνο λέξη, ενώ ο πιο "πυκνοκατοικημένος" κωδικός αντιστοιχεί σε 11.681 λέξεις (που αντιστοιχούν σε λέξεις που αρχίζουν από κατα, ένα συχνό πρόθεμα στα ελληνικά). Αντίστοιχα, ο ελάχιστος αριθμός λέξεων ανά στέλεχος είναι 1, ενώ ο μέγιστος αριθμός λέξεων ανά στέλεχος είναι 257 (που αντιστοιχεί στο λήμμα ταξ). Οι κατανομές των συχνοτήτων των κωδικών Soundex_{GR} και των λημμάτων του στελέχους, κατά τη διάρκεια του διαλόγου, παρουσιάζονται στο Σχήμα 9, όπου και οι δύο άξονες *Y* (του αριστερού και του δεξιού διαγράμματος) είναι σε κλίμακα log. Οι 10 συχνότεροι κωδικοί παρουσιάζονται στον Πίνακα 11, ενώ τα 10 συχνότερα στελέχη στον Πίνακα 12.

Φυσικά, και με βάση την εργασία που πρέπει να γίνει, μπορεί κανείς να αποφασίσει να χρησιμοποιήσει μεγαλύτερα Soundex_{GR} κώδικες, αν θέλει να βελτιώσει την ακρίβεια έναντι της ανάκλησης, όπως αναφέρεται στην ενότητα 4.5.



Σχήμα **9**: Συχνότητα των κωδικών Soundex_{GR} (αριστερά) και των λημμάτων του stemmer (δεξιά) στο λεξικό

[b]Ελληνικό λεξικό WinEdt Unicode, έκδοση 2008-10-03, λήφθηκε στις 26 Απριλίου 2020, μέγεθος 2.089 KB, `http://`www.winedt.org/dict.html.

Πίνακας 11. : Συχνότερα Soundex$_{GR}$ κωδικοί

κωδικός	συχνότητα
κ939	11681
π989	8207
π8$4	5396
π*8@	4756
π$6@	4399
ξ979	3953
α1$3	3949
μ*39	3933
α1$4	3857
π8$3	3595

Πίνακας 12. : Συχνότερα στελέχη

κωδικός	συχνότητα
ταξ	257
μασ	235
αρξ	233
κοψ	212
πασ	209
βασ	208
ποσ	194
γραψ	190
παξ	176
ορξ	159

λέξη		Soundexnaive GR	Soundex$_{GR}$	Φωνητική μεταγραφή
αυγό	→	α200	α12$	ανγο
αβγό	→	α120	α12$	ανγο
εύδοξος	→	ε344	ε13$	ενδοξος
εβδοξος	→	ε134	ε13$	ενδοξος
λιανοτράγουδα	→	λ83!	λ@97	lianotraγuδa
στρογγυλοκουλουριαζόντουσαν	→	σ3!2	σ38$	strogilokuluriaζodusan

Σχήμα **10**: Ενδεικτικά παραδείγματα πλήρους φωνημικής μεταγραφής

4.8 Άλλες παραλλαγές: Φωνητική μεταγραφή

Δεν είναι δύσκολο να δούμε ότι οι ίδιοι κανόνες, με μικρές αλλαγές, μπορούν να χρησιμοποιηθούν για την εξαγωγή της πλήρους *φωνητικής* μεταγραφής μιας ελληνικής λέξης. Με τον όρο "φωνήματα" αναφερόμαστε στις νοητικές κατηγορίες' που χρησιμοποιεί ένας ομιλητής και όχι στις πραγματικές προφορικές παραλλαγές αυτών των φωνημάτων που παράγονται στο πλαίσιο μιας συγκεκριμένης λέξης (σημειώστε ότι η φωνητική μεταγραφή καθορίζει τις λεπτότερες λεπτομέρειες του πώς παράγονται στην πραγματικότητα οι ήχοι).

Συγκεκριμένα, μπορούμε να χρησιμοποιήσουμε μόνο τα ακόλουθα τρία βήματα του Alg. 1: *w* ←*UnwrapConsonantBigrams*(*word*) *w* ←*TransformVowelsToConsonants*(*w*) *w* ←*GroupVowels*(*w*)

Τα υπόλοιπα βήματα της Alg. 1 δεν χρειάζονται, δηλαδή παραλείπουμε το βήμα της αφαίρεσης των τελευταίων χαρακτήρων (*RemoveLast*), το βήμα της κωδικοποίησης (*SoundexEncode*) και το βήμα της εξάλειψης των διπλοτύπων (*RemoveDuplicates*).

Με τα παραπάνω τρία βήματα, οι αλλαγές που απαιτούνται για την παραγωγή μιας πλήρους φωνητικής μεταγραφής των ελληνικών λέξεων είναι ελάχιστες. Η πρώτη αλλαγή είναι ότι στο *GroupVowels(w)* η ομαδοποίηση είναι λίγο διαφορετική, συγκεκριμένα ομαδοποιούμε το "ου" στο "u" (αντί για το "ο"). Η δεύτερη αλλαγή είναι ότι αντί να αντιστοιχίσουμε τόσο το ¨τσ' όσο και το ¨τζ' στο "c" αντιστοιχίζουμε το πρώτο στο "ts" και το δεύτερο στο "dz". Τέλος, αντί να χρησιμοποιούμε ελληνικά γράμματα για τη

φωνητική μεταγραφή μπορούμε να χρησιμοποιούμε λατινικά γράμματα. Ποτέ όμως δεν είναι δυνατόν, σε κάθε περίπτωση η επιλογή των χαρακτήρων στη φωνητική μεταγραφή δεν επηρεάζει τη διαδικασία αντιστοίχισης. Μερικά παραδείγματα δίνονται στην Εικόνα 10:

Έχουμε υλοποιήσει την παραπάνω έκδοση και περιλαμβάνεται στη δημόσια έκδοση της οικογένειας αλγορίθμων soundexGR (περιγράφεται στην ενότητα 4.11). Ένα άλλο σημαντικό ερώτημα είναι πώς θα συμπεριφερόταν η ακριβής φωνητική (φωνημική) μεταγραφή στα σύνολα δεδομένων αξιολόγησης (που περιγράφονται στην Ενότητα 4.2). Τα αποτελέσματα δεν είναι τόσο καλά, συγκεκριμένα:
στο σύνολο δεδομένων A (η συλλογή *προσθήκης* γραμμάτων)
πήραμε F-Score = 0,17, στο σύνολο δεδομένων B (η συλλογή
διαγραφής γραμμάτων) πήραμε F-Score = 0,31,
στο σύνολο δεδομένων C (η συλλογή *υποκατάστασης* γραμμάτων) πήραμε F-Score = 0,23, και
στο σύνολο δεδομένων D (η συλλογή λέξεων *με παρόμοια προφορά*) έχουμε F-Score = 0,93. Παρατηρούμε ότι η πλήρης φωνητική μεταγραφή συμπεριφέρεται καλά μόνο στο σύνολο δεδομένων D, επιτυγχάνοντας F-Score 0,93, ωστόσο αυτό το $_{GR}$σκορ είναι χαμηλότερο από το 0,97 που επιτυγχάνει το ^{Soundexnaive}. Όπως φαίνεται, στα υπόλοιπα σύνολα δεδομένων αξιολόγησης, η ακριβής φωνητική μεταγραφή συμπεριφέρεται πολύ χειρότερα, καθώς δεν μπορεί να αντιμετωπίσει τις περιπτώσεις προσθηκών, διαγραφών και αντικαταστάσεων γραμμάτων.

Συνολικά, ο μέσος όρος του F-Score σε όλα τα σύνολα δεδομένων αξιολόγησης του Soundex_{GR} για το μήκος 4 είναι ίσος με 0,66 (όπως φαίνεται στον πίνακα 5), ενώ ο μέσος όρος του F-Score σε όλα τα σύνολα δεδομένων αξιολόγησης της πλήρους φωνητικής μεταγραφής είναι 0,41 (=(0,17+0,31+0,23+0,93)/4).

Πρόσθετα πειράματα με την αντιστοίχιση χρησιμοποιώντας πλήρη φωνημική μεταγραφή, δίνονται στην ενότητα 4.9 και στη σειρά πειραμάτων που περιγράφονται στην ενότητα 4.10.

4.ff Σύγκριση όλων των παραλλαγών στο σύνολο δεδομένων D^{ext}

Για να δώσουμε μια γενική εικόνα της αποτελεσματικότητας των προαναφερθέντων μεθόδων, αποφασίσαμε να ετοιμάσουμε μια εκτεταμένη έκδοση του συνόλου δεδομένων D που περιέχει περισσότερες παραλλαγές για κάθε λέξη. Το παραγόμενο σύνολο δεδομένων, που συμβολίζεται με Dataset^{Dext}, περιέχει συνολικά 500 λέξεις, και συγκεκριμένα περιέχει 125 λέξεις στην ορθογραφικά σωστή τους μορφή συν 3 ορθογραφικά λάθη για κάθε μία από αυτές. Όλες οι ανορθογραφίες ακούγονται το ίδιο με τη σωστή. Προσπαθήσαμε να συμπεριλάβουμε λέξεις που γράφονται συχνά λανθασμένα καθώς και τυπογραφικά λάθη που όμως δεν αλλάζουν τον τρόπο με τον οποίο θα ακούγονταν. Ένα απόσπασμα αυτού του συνόλου δεδομένων παρουσιάζεται στην Εικόνα 11.

βύσσινο, βύσινο, βύσυνο, βύσιννο
διάλλειμα, διάλυμα, διάλοιμα, διάλειμα
παλίρροια, παλίροια, παλίρια, παλείρεια
παράλειψη, παράληψη, παράλιψη, παράλειψψη
πλημμύρα, πλημύρα, πλημίρρα, πλοιμοιρα
ωράριο, οράριο, ωράρειο, οράροιο

Σχήμα **11**: Απόσπασμα από το σύνολο δεδομένων ^{Dext}

Σε αυτό το σύνολο δεδομένων, αξιολογήσαμε όλες τις προαναφερθείσες μεθόδους, καθώς και μερικές ακόμη, συνολικά 10 μεθόδους, και συγκεκριμένα τις μεθόδους exact match, ^{Soundexnaive}, Soundex_{GR} , Soundex^{comp} , stemming
(όπως περιγράφεται στην Ενότητα 4.6), το Soundex_{GR} πάνω από τα αποτελέσματα του stemming, της πλήρους φωνημικής

μεταγραφή (όπως περιγράφεται στην ενότητα 4.8), και αντιστοίχιση με βάση την απόσταση Edit Levenshtein (1966) με ανοχή K που κυμαίνεται από 1 έως 3. Για παράδειγμα, EditDistance με K=2 σημαίνει ότι δύο λέξεις ταιριάζουν αν η απόσταση Edit Distance τους είναι μικρότερη ή ίση με 2. Το μήκος του κώδικα για τα $Soundex^{naive}$, $\mathsf{Soundex}_{GR}$ και $\mathsf{Soundex}^{comp}$ ήταν ίσο με 4. Τα αποτελέσματα είναι τα εξής φαίνεται στον πίνακα 13, όπου e^{GR} γράφονται οι υψηλότερες τιμές των $Precisio\ n^{GR}$, Recall και F-Score με έντονη γραφή. Εξετάζοντας τις τιμές μπορούμε να κατανοήσουμε τη συμπεριφορά αυτών των μεθόδων και βλέπουμε ότι το $\mathsf{Soundex}_{GR}$ επιτυγχάνει το υψηλότερο F-Score (0,97).

Πίνακας 13. : Αξιολόγηση 10 μεθόδων αντιστοίχισης σε σύνολο δεδομένων ^{Dext}

	Μέθοδος	Ακρίβεια	Ανάκληση	F-Score
1	exactMatch	**1.0**	0.25	0.40
2	$Soundex_{GR}$	0.95	**0.99**	**0.97**
3	$Soundexnaive_{GR}$	0.92	0.91	0.91
4	$Soundex_{GR}^{comp}$	0.88	**0.99**	0.93
5	Stemmer	0.94	0.30	0.46
6	$Soundex_{GR}$ over Stemmer	0.85	0.79	0.82
7	Πλήρης φωνημική μεταβίβαση	**1.0**	0.66	0.80
8	Απόσταση επεξεργασίας ≤ 1	0.97	0.58	0.73
9	Απόσταση επεξεργασίας ≤ 2	0.78	0.84	0.81
10	Απόσταση επεξεργασίας ≤ 3	0.52	0.93	0.67

4.10 Πειράματα σε διάφορες κλίμακες - Σχετικά με την επιλογή του μήκους των κωδίκων (σε σύνολο δεδομένων E *-* σύνολο δεδομένων H*)*

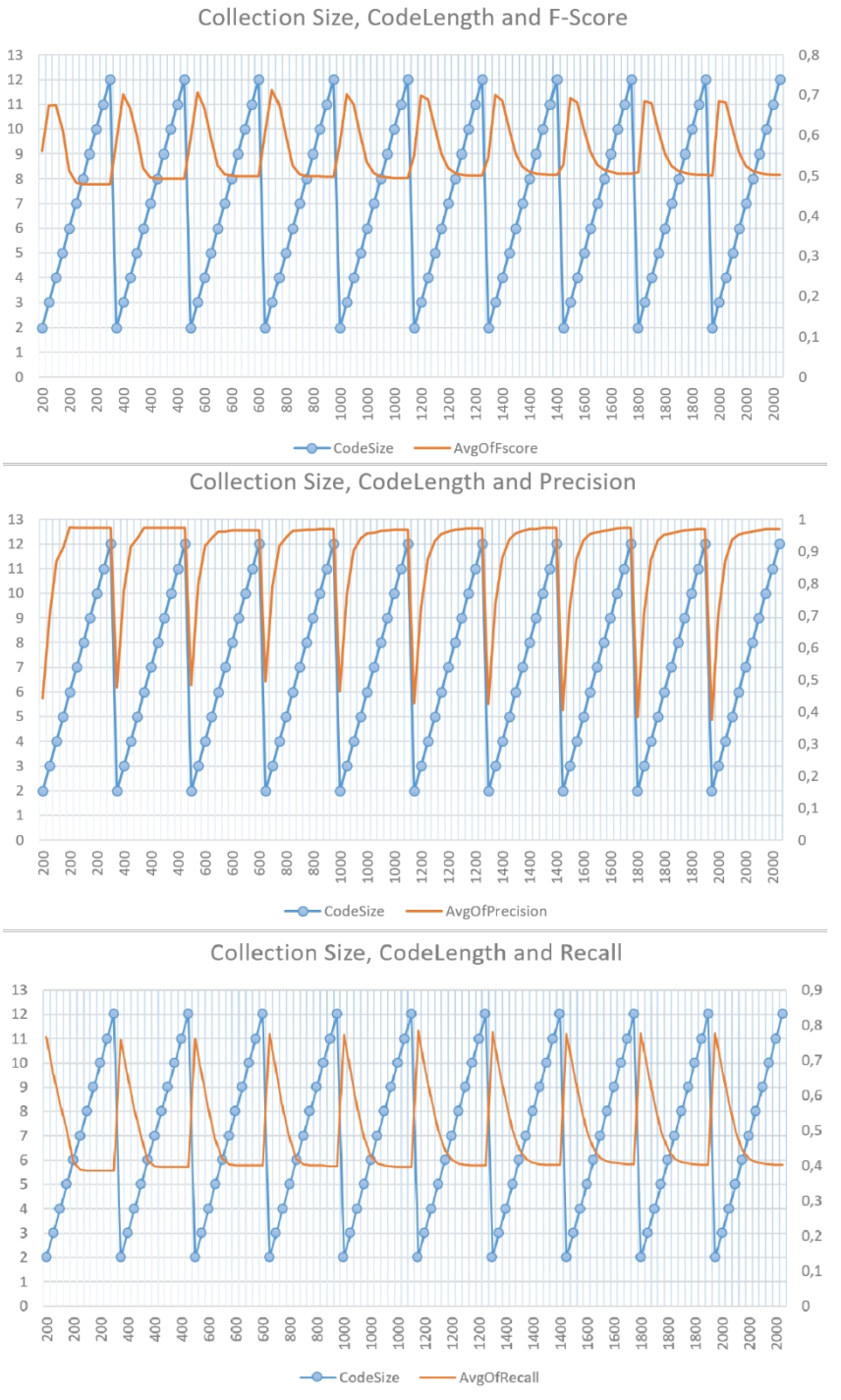
Στην ενότητα 4.5 είδαμε ότι το μήκος 4 αποδίδει το καλύτερο μέσο F-Score στα τέσσερα σύνολα δεδομένων αξιολόγησης. Τα ερωτήματα που προκύπτουν είναι τα εξής: Εξαρτάται το βέλτιστο μήκος από το μέγεθος του συνόλου δεδομένων; Θα πρέπει να χρησιμοποιούμε μικρότερους κώδικες σε μικρότερα σύνολα δεδομένων και μεγαλύτερους κώδικες σε μεγαλύτερες συλλογές; Μια προσέγγιση για την αντιμετώπιση αυτών των ερωτημάτων είναι να γίνουν τα πειράματα (όπως αυτά που αναφέρονται στον Πίνακα 10), αλλά αντί να εξεταστούν ολόκληρα τα σύνολα δεδομένων αξιολόγησης, να εξεταστούν μόνο τμήματα αυτών των συνόλων δεδομένων ξεκινώντας από πολύ μικρά τμήματα και φτάνοντας σε ολόκληρα τα σύνολα δεδομένων αξιολόγησης. Για το σκοπό αυτό πραγματοποιήσαμε πειράματα αφού περιορίσαμε τον αριθμό των λέξεων που θα εξεταστούν από κάθε σύνολο δεδομένων, ξεκινώντας από 200 λέξεις έως 2000 λέξεις με βήμα αύξησης ίσο με 200.

Για κάθε τέτοιο μέγεθος συνόλου δεδομένων, αξιολογήσαμε τα μήκη των κωδικών $Soundex_{GR}$ ξεκινώντας από
2 έως 12. Τα πειραματικά αποτελέσματα, όσον αφορά το μέσο F-Score, παρουσιάζονται στο Σχήμα 12 (πάνω διάγραμμα). Ο αριστερός άξονας *Y* αντιστοιχεί στο μήκος του κώδικα (από 2 έως 12), ενώ ο δεξιός άξονας *Y* αντιστοιχεί στο μέσο F-Score (στα 4 μέρη του συνόλου δεδομένων αξιολόγησης). Ο *άξονας X* δείχνει τα μεγέθη του συνόλου δεδομένων (από 200 έως 2000 λέξεις με βήμα ίσο με 200), και για κάθε τέτοιο μέγεθος ο *άξονας X* έχει 11 στίγματα που το καθένα αντιστοιχεί σε ένα μήκος κώδικα (από 2 έως 12).

Το σχήμα 12 (πάνω διάγραμμα) αποκαλύπτει το ακόλουθο γενικό μοτίβο: Καθώς το μήκος του κώδικα αυξάνεται, το F-Score αυξάνεται φτάνοντας σε μια κορυφή γύρω στο 0,7 (συνήθως για μήκος κώδικα 3 ή 4) και στη συνέχεια μειώνεται και καταλήγει στο 0,5. Το Σχήμα 12 (μεσαίο και κάτω διάγραμμα) δείχνει τη μέση ακρίβεια και τη μέση ανάκληση, που μας βοηθά να εξηγήσουμε την κατανομή του μέσου F- Score. Από αυτές τις μετρήσεις, θα μπορούσαμε να πούμε ότι το μέγεθος του συνόλου δεδομένων δεν είναι πολύ καθοριστικό (τουλάχιστον για τα εξεταζόμενα μεγέθη σε αυτό το πείραμα, δηλαδή για 200 έως 2000), αφού βλέπουμε ότι το μέγεθος του συνόλου δεδομένων δεν επηρεάζει σημαντικά το F-Score. Δεν είναι δύσκολο να δούμε ότι δεν είναι μόνο το μέγεθος του συνόλου δεδομένων που έχει σημασία, αλλά και το μήκος των λέξεων, μέγεθος που δεν εξαρτάται από το μέγεθος του συνόλου δεδομένων: Ακόμη και σε μικρά

σύνολα δεδομένων οι πολύ σύντομοι κώδικες ή οι πολύ μεγάλοι κώδικες βλάπτουν το F-Score που επιτυγχάνουμε και αυτό αποδεικνύεται από τις μετρήσεις, δηλαδή από τις χαμηλές τιμές F-Score που παίρνουμε για πολύ σύντομους και πολύ μεγάλους κώδικες στο Σχήμα 12(πάνω διάγραμμα). Για να ελέγξουμε αυτή την υπόθεση και να κατανοήσουμε περαιτέρω τι επηρεάζει την απόδοση, σχεδιάσαμε το πείραμα που ακολουθεί.

Σύνολα δεδομένων με μεγαλύτερες διακυμάνσεις στο μέγεθος των λέξεων. Αξιοποιώντας την εμπειρία από τη δημιουργία (με το χέρι) του συνόλου δεδομένων ^{Dext}, αποφασίσαμε να χρησιμοποιήσουμε το λεξικό των ελληνικών λέξεων



Σχήμα **12**: Μέσο F-Score (πάνω), Precision (μέση) και Recall (κάτω) ως συνάρτηση του μήκους του κώδικα (αριστερός άξονας Y, μπλε κουκκίδες) και του μεγέθους του συνόλου δεδομένων (άξονας X) του Soundex_{GR} στο σύνολο δεδομένων A, στο σύνολο δεδομένων B, στο σύνολο δεδομένων C και στο σύνολο δεδομένων D.

(που αναφέρεται στην ενότητα 4.7 και περιέχει 574.883 διαφορετικές λέξεις) για την παραγωγή μεγαλύτερων συνόλων δεδομένων για περαιτέρω αξιολόγηση και πειραματισμό σχετικά με το μέγεθος των κωδίκων. Για κάθε λέξη του εν λόγω λεξικού παράγουμε έναν κάδο που περιέχει παραλλαγές της λέξης με διάφορα είδη σφαλμάτων. Αποφασίσαμε να συμπεριλάβουμε λέξεις που περιέχουν περισσότερα από ένα λάθη, όχι μόνο επειδή υπάρχουν πολλά συχνά ορθογραφικά λάθη που περιέχουν περισσότερα από ένα λάθη,
π.χ. μύzρμα αντί για μήzυμα, θάλλασα αντί για θάλασσα, αλλά και για την αξιολόγηση περιπτώσεων που δεν μπορούν να καταγραφούν εύκολα από την απόσταση επεξεργασίας. Γι' αυτό το λόγο συμπεριλάβαμε διάφορα λάθη που δεν επηρεάζουν τον τρόπο που ακούγεται η λέξη, οπότε δίνεται έμφαση στα ορθογραφικά λάθη.

Συγκεκριμένα, για την παραγωγή τέτοιων σφαλμάτων δημιουργήσαμε περίπου 40 κανόνες για την καταγραφή διαφόρων περιπτώσεων. Οι περισσότεροι από αυτούς είναι κανόνες αντικατάστασης, με όρους για τους χαρακτήρες που δεν πρέπει να εμφανίζονται πριν ή μετά τον χαρακτήρα που πρέπει να αντικατασταθεί. Για παράδειγμα, ο κανόνας Rule(*{ο*,α,*ε}*,ι,υ,-) αντικαθιστά το ι με το υ μόνο αν το γράμμα πριν από το ι δεν είναι ένα από τα ο,α,ε, αφού σε αυτή την περίπτωση έχουμε δίφθογγο και ένα τέτοιο λάθος δεν θα ήταν συνηθισμένο. Αντίστοιχα ο κανόνας Rule(-,ο,ω,*{υ*,ύ,ι,ι,*ί}*) αντικαθιστά το ο με το ω μόνο αν ο χαρακτήρας μετά το ο δεν είναι ένας από τη λίστα, αφού και σε αυτή την περίπτωση έχουμε δίφθογγο. Το σύνολο των κανόνων δεν υποτίθεται ότι παράγει όλα τα πιθανά λάθη, αλλά μπορούν να συλλάβουν αρκετά καλά διάφορα είδη κοινών λαθών, επομένως οι παραλλαγές που παράγουν μπορούν να χρησιμοποιηθούν για την αξιολόγηση της προσεγγιστικής αντιστοίχισης. Για να διασφαλίσουμε ότι για κάθε λέξη (και για τις πολύ μικρές) έχουμε τουλάχιστον ένα ορθογραφικό λάθος, έχουμε συμπεριλάβει έναν κανόνα που διπλασιάζει ένα μεσαίο σύμφωνο. Ας ονομάσουμε αυτό το σύνολο δεδομένων
Σύνολο δεδομένων Ε.

Οι λέξεις στο αρχικό λεξικό είναι ταξινομημένες με βάση το μέγεθός τους. Για να δημιουργήσουμε ένα σύνολο δεδομένων που να καλύπτει όλα τα μεγέθη λέξεων χρησιμοποιήσαμε το βήμα 400, δηλαδή ξεχωρίσαμε μία λέξη κάθε 400 λέξεις του λεξικού. Το σύνολο δεδομένων που προέκυψε, το οποίο θα συμβολίσουμε με Dataset $E_{1.4K-7.6K}$, έχει 1.438 διακριτές σωστές λέξεις 7.608 λέξεις συνολικά, το μέσο μέγεθος των μπλοκ είναι 5,29, δηλαδή κατά μέσο όρο το σύνολο δεδομένων περιέχει περισσότερα από 4 ορθογραφικά λάθη ανά λέξη. Ένα μικρό απόσπασμα από το παραγόμενο σύνολο δεδομένων παρουσιάζεται στην Εικόνα 13.

- θολά, θωλά, θολλά
- Χάρης, Χάρυς, Χάρρνης
- αποπήρε, αποπύρε, αποπήραι, απωπήρε, απποπήρε
- επιρρεπώς, επυρρεπώς, αιπυρρεπώς, επυρρεπός, επυρεπός
- λιπόψυχης, λιπόψυχυς, λιπόψηχυς, λυπόψηχυς, λυπώψηχυς, λυππόπσηχυς
- προσυπογραφόμουν, προσηπογραφόμουν, πρωσηπογραυόμουν, πρωσηπογραυώμουν, προσηπογραυώμουν, πρροσηπογραυόμουν
- ετεροδημότη, ετεροδυμότη, ετεροδημότη, ετεροδημότει, αιτεροδημότοι, ετερωδημότοι, ετερωδημώτοι, ετεροδημώτοι, ετεροδημότοι, εττεροδημότοι
- Δεντροφυτευόταν, δαιντροφητευόταν, δεντροφητεβόταν, δεντροφητεφόταν, δεντρωφητευόταν, δεντρωφητευώταν, δεντροφητευώταν, δεντροφητευόταν, δενντροφητευόταν

Σχήμα 13: Απόσπασμα από το σύνολο δεδομένων $E_{1.4K-7.6K}$

Σε αυτό το σύνολο δεδομένων, το οποίο συμβολίζεται ως σύνολο δεδομένων

$E_{1.4K-7.6K}$, εκτελούμε τα πειράματα και τα αποτελέσματα παρουσιάζονται στον πίνακα 14. Αρχικά παρατηρούμε ότι η ακριβής αντιστοίχιση επιτυγχάνει F-Score 0,37, το Stemming 0,40, ενώ η πλήρης φωνημική μεταγραφή 0,86. Η απόσταση επεξεργασίας επιτυγχάνει το μέγιστο F-Score, δηλαδή 0,9, με $K \leq 3$. Σημειώστε ότι *το* Soundex_{GR} είναι καλύτερο από όλες τις παραπάνω επιλογές για κάθε μήκος κωδικού ίσο ή μεγαλύτερο από 6. Το βέλτιστο F-Score, δηλαδή 0,98, επιτυγχάνεται με το Soundex^{comp} και μήκος κώδικα ίσο με 10. Αυτό το μήκος είναι μεγαλύτερο από

Πίνακας 14. : Αξιολόγηση 10 μεθόδων σε σύνολο δεδομένων $E_{1.4K-7.6K}$

	Μέθοδος	Ακρίβεια	Ανάκληση	F-Score
1	exactMatch	**1.0**	0.23	0.37
2-i	$Soundex_{GR}$ (4)	0.63	0.96	0.76
3-i	$Soundex^{naive}_{GR}$ (4)	0.74	0.89	0.80
4-i	$Soundex^{comp}_{GR}$ (4)	0.55	0.98	0.70
2-ii	$Soundex_{GR}$ (5)	0.82	0.96	0.89
3-ii	$Soundex^{naive}_{GR}$ (5)	0.91	0.88	0.90
4-ii	$Soundex^{comp}_{GR}$ (5)	0.78	0.98	0.87
2-iii	$Soundex_{GR}$ (6)	0.93	0.96	0.94
3-iii	$Soundex^{naive}_{GR}$ (6)	0.96	0.87	0.91
4-iii	$Soundex^{comp}_{GR}$ (6)	0.90	0.98	0.94
2-iv	$Soundex_{GR}$ (7)	0.97	0.95	0.96
3-iv	$Soundex^{naive}_{GR}$ (7)	0.98	0.88	0.92
4-iv	$Soundex^{comp}_{GR}$ (7)	0.95	0.98	**0.97**
2-v	$Soundex_{GR}$ (8)	0.98	0.95	**0.97**
3-v	$Soundex^{naive}_{GR}$ (8)	0.98	0.87	0.92
4-v	$Soundex^{comp}_{GR}$ (8)	0.97	0.98	**0.97**
2-vi	$Soundex_{GR}$ (9)	0.99	0.95	**0.97**
3-vi	$Soundex^{naive}_{GR}$ (9)	0.98	0.87	0.92
4-vi	$Soundex^{comp}_{GR}$ (9)	0.97	0.98	**0.97**
2-vii	$Soundex_{GR}$ (10)	0.99	0.95	**0.97**
3-vii	$Soundex^{naive}_{GR}$ (10)	0.98	0.87	0.92
4-vii	$Soundex^{comp}_{GR}$ (10)	0.97	0.98	**0.98**
2-viii	$Soundex_{GR}$ (11)	0.99	0.95	**0.97**
3-viii	$Soundex^{naive}_{GR}$ (11)	0.98	0.87	0.92
4-viii	$Soundex^{comp}_{GR}$ (11)	0.97	0.98	**0.98**
5	Stemmer	0.98	0.25	0.40
6	Πλήρης φωνημική μεταβίβαση	0.99	0.75	0.86
7	Απόσταση επεξεργασίας ≤ 1	0.99	0.44	0.60
8	Απόσταση επεξεργασίας ≤ 2	0.99	0.69	0.82
9	Απόσταση επεξεργασίας ≤ 3	0.95	0.87	0.90
10	Απόσταση επεξεργασίας ≤ 4	0.80	0.95	0.87

αυτό που περιμέναμε, ωστόσο, αυτό μπορεί να εξηγηθεί από το γεγονός ότι το λεξικό περιέχει πολλές μεγάλες λέξεις.

Για να δημιουργήσουμε ένα μεγαλύτερο σύνολο δεδομένων, μειώσαμε το βήμα σε 200 και δημιουργήσαμε το σύνολο δεδομένων $F_{2.8K-15.2K}$ που περιέχει 2.875 σωστές λέξεις και 15.297 συνολικές λέξεις (μέσο μέγεθος κάδου 5,32). Τα αποτελέσματα των πειραμάτων παρουσιάζονται στον Πίνακα 15. Παρατηρούμε μια μικρή πτώση της ακρίβειας και του F-Score για μήκος 4, ωστόσο το $Soundex_{GR}$ με μήκος κώδικα ίσο με 12 διατηρεί το πολύ υψηλό F-Score (0,97).

Για να δημιουργήσουμε ένα ακόμη μεγαλύτερο σύνολο δεδομένων μειώσαμε

περαιτέρω το βήμα σε 100 και δημιουργήσαμε το σύνολο δεδομένων $G_{5.7K-30.4K}$ που περιέχει 5.749 σωστές λέξεις και 30.824 λέξεις συνολικά (μέσο μέγεθος κάδου 5,36). Τα αποτελέσματα των πειραμάτων και τα αποτελέσματα παρουσιάζονται στον Πίνακα 16.

Πίνακας 15. : Αξιολόγηση 10 μεθόδων σε σύνολο δεδομένων $F_{2.8K-15.2K}$

	Μέθοδος	Ακρίβεια	Ανάκληση	F-Score
1	exactMatch	**1.0**	0.22	0.37
2-i	$Soundex_{GR}$ (4)	0.5	0.96	0.65
3-i	$Soundex^{naive}_{GR}$ (4)	0.61	0.89	0.72
4-i	$Soundex^{comp}_{GR}$ (4)	0.40	0.98	0.57
2-ii	$Soundex_{GR}$ (12)	0.99	0.95	**0.97**
3-ii	$Soundex^{naive}_{GR}$ (12)	0.96	0.87	0.91
4-ii	$Soundex^{comp}_{GR}$ (12)	0.96	0.98	**0.97**
5	Stemmer	0.96	0.25	0.40
6	Πλήρης φωνημική μεταβίβαση	0.99	0.75	0.85
7	Απόσταση επεξεργασίας ≤ 1	0.99	0.43	0.60
8	Απόσταση επεξεργασίας ≤ 2	0.98	0.69	0.81
9	Απόσταση επεξεργασίας ≤ 3	0.91	0.86	0.89
10	Απόσταση επεξεργασίας ≤ 4	0.71	0.95	0.81

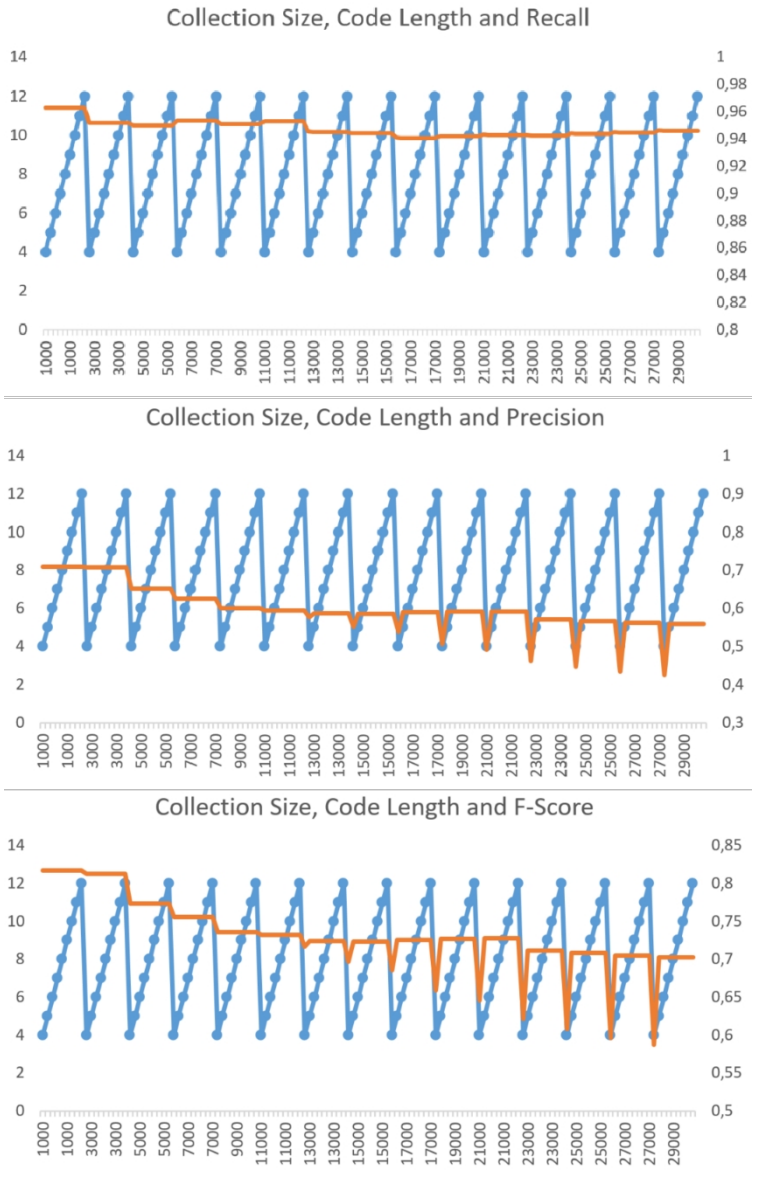
Παρατηρούμε περαιτέρω πτώση της ακρίβειας και του F-Score για το μήκος 4, ωστόσο για μήκος κώδικα ίσο με 12, *το* $Soundex_{GR}$ διατηρεί το πολύ υψηλό F-Score (0,97).

Πίνακας 16. : Αξιολόγηση 10 μεθόδων για το σύνολο δεδομένων $G_{5.7K-30.4K}$

	Μέθοδος	Ακρίβεια	Ανάκληση	F-Score
1	exactMatch	**1.0**	0.22	0.36
2-i	$Soundex_{GR}$ (4)	0.36	0.96	0.52
3-i	$Soundex^{naive}_{GR}$ (4)	0.46	0.88	0.61
4-i	$Soundex^{comp}_{GR}$ (4)	0.26	0.98	0.41
2-ii	$Soundex_{GR}$ (12)	0.99	0.95	**0.97**
3-ii	$Soundex^{naive}_{GR}$ (12)	0.93	0.87	0.90
4-ii	$Soundex^{comp}_{GR}$ (12)	0.92	0.98	0.95
5	Stemmer	0.92	0.24	0.38
6	Πλήρης φωνημική μεταβίβαση	0.99	0.75	0.85
7	Απόσταση επεξεργασίας ≤ 1	0.96	0.43	0.60
8	Απόσταση επεξεργασίας ≤ 2	0.97	0.68	0.80
9	Απόσταση επεξεργασίας ≤ 3	0.86	0.86	0.86
10	Απόσταση επεξεργασίας ≤ 4	0.61	0.95	0.74

Τα προηγούμενα σύνολα δεδομένων (Σύνολο δεδομένων $E_{1.4K-7.6K}$ -Σύνολο δεδομένων $G_{5.7K-30.4K}$), τα οποία προέκυψαν από την επιλογή λέξεων από την αρχή έως το τέλος του λεξικού, κάλυπταν όλο το φάσμα του μήκους των λέξεων. Ωστόσο, οι μεγαλύτερες λέξεις είναι λιγότερο συχνές, επομένως είναι λογικό να γίνονται πειράματα

ξεκινώντας από την αρχή και χωρίς κενά, για την εξέταση όλων των μικρών και μεσαίων λέξεων, οι οποίες αναμένεται να περιέχουν τις συχνές. Το σύνολο δεδομένων που προκύπτει είναι πιθανώς δυσκολότερο για την αντιστοίχιση, όχι μόνο επειδή υπάρχουν πολλές μικρές λέξεις που καθιστούν δύσκολη την επίτευξη ακρίβειας, αλλά και επειδή θα συμπεριληφθούν πολλές μορφολογικές παραλλαγές των περιεχόμενων λέξεων (δεδομένου ότι χρησιμοποιήθηκε το βήμα 1), οπότε είναι πιο δύσκολο να επιτευχθεί υψηλή ακρίβεια. Για το λόγο αυτό πραγματοποιήσαμε πειράματα του Soundex$_{GR}$ για όλες τις

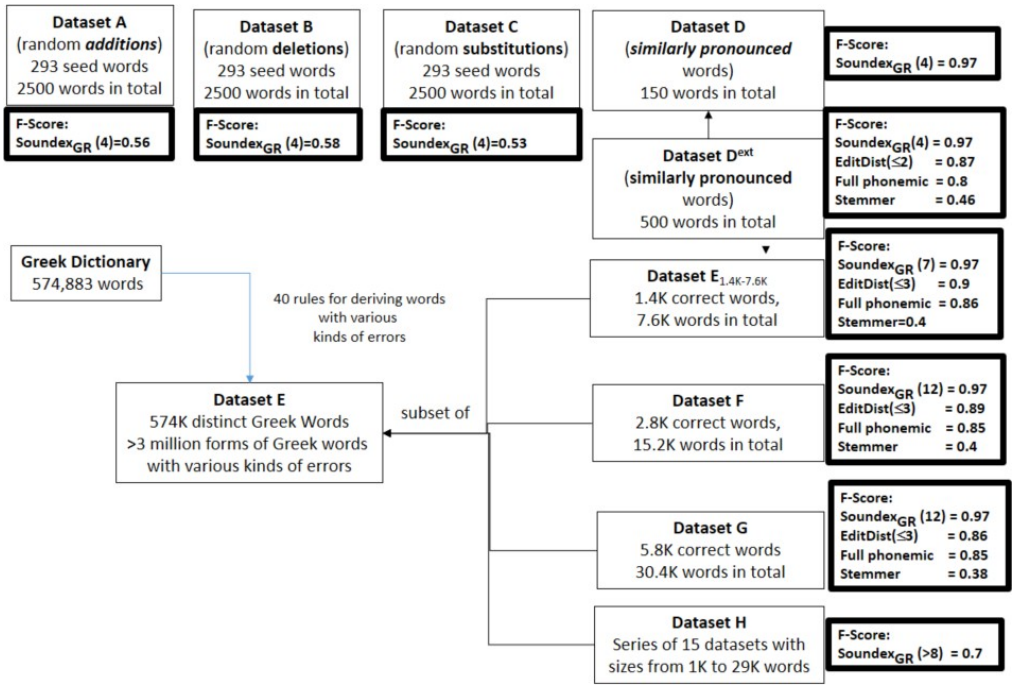


Σχήμα **14**: Ανάκληση (επάνω), ακρίβεια (μέση) και F-Score (κάτω) ως συνάρτηση του μήκους του κώδικα (αριστερός άξονας Y, μπλε κουκκίδες) και του μεγέθους του συνόλου δεδομένων (άξονας X) του Soundex$_{GR}$ στο σύνολο δεδομένων H.

μήκη κωδικών από 2 έως 12 για μεγέθη συνόλου δεδομένων που ξεκινούν από 1.000 λέξεις έως 29.000 λέξεις με βήμα αύξησης συνόλου δεδομένων 2.000 (λέξεις, όχι γραμμές). Η προκύπτουσα σειρά 15 συνόλων δεδομένων περιέχει γράμματα με λέξεις έως 6 γράμματα.

Τα αποτελέσματα δίνονται στο Σχήμα 14. Παρατηρήστε ότι οι δεξιοί κάθετοι άξονες ξεκινούν από 0,5 για το F-Score, 0,3 για την Precision, 0,8 για την Recall, για να γίνουν πιο εμφανείς οι διαφορές. Στο Σχήμα 14(πάνω διάγραμμα) παρατηρούμε ότι η ανάκληση δεν επηρεάζεται ουσιαστικά ούτε από το μέγεθος του συνόλου δεδομένων ούτε από το μήκος του κώδικα. Στο Σχήμα 14(μεσαίο διάγραμμα) παρατηρούμε, ότι (όπως αναμενόταν) η ακρίβεια είναι χαμηλότερη και επηρεάζεται από το μέγεθος της συλλογής.

Στο Σχήμα 14(κάτω διάγραμμα) παρατηρούμε ότι το F-Score επηρεάζεται από το μέγεθος της συλλογής (δηλαδή μειώνεται όσο το μέγεθος του συνόλου δεδομένων



Σχήμα **15**: Σύνοψη των κύριων αποτελεσμάτων της αξιολόγησης

αυξάνεται), αλλά επιτυγχάνει 0,7 για μήκος κώδικα ≥ 8. Γενικά, παρατηρούμε (όπως αναμενόταν) ότι σε αυτή τη σειρά συνόλων δεδομένων που περιέχουν μικρές λέξεις το F-Score είναι χαμηλότερο από αυτό στο σύνολο δεδομένων $G_{5.7K-30.4K}$. Αυτό αποδεικνύει ότι όχι μόνο το μέγεθος του λεξιλογίου και το είδος των σφαλμάτων, αλλά και το μέγεθος των λέξεων επηρεάζουν την αποτελεσματικότητα της αντιστοίχισης.

Σύνοψη και γενικές παρατηρήσεις. Το Σχήμα 15 απεικονίζει τα κύρια αποτελέσματα, δηλαδή δείχνει κάθε σύνολο δεδομένων και τα χαρακτηριστικά του, καθώς και τα καλύτερα F-Scores που λαμβάνονται από το $Soundex_{GR}$ και άλλες μεθόδους αντιστοίχισης.

Ακολουθούν μερικές γενικές παρατηρήσεις:

- Όσο μεγαλύτερη είναι η συλλογή και όσο περισσότερες λέξεις περιέχει, τόσο μεγαλύτεροι πρέπει να είναι οι κωδικοί (για να διατηρηθεί η ακρίβεια). Το ίδιο ισχύει και για την ανοχή της αντιστοίχισης με βάση την απόσταση επεξεργασίας. Σε ένα πλαίσιο όπου απαιτείται ανάκτηση υψηλής ακρίβειας (π.χ. στην ανάκτηση σχολίων χρηστών στο πλαίσιο μιας φωνητικής αλληλεπίδρασης συνομιλίας, όπως στο Dimitrakis et al. (2018)), μπορούν να επιλεγούν μεγαλύτεροι κωδικοί, ενώ σε ένα πλαίσιο εφαρμογής όπου η ανάκληση είναι πιο σημαντική (π.χ. στην αναζήτηση πατεντών), οι μικρότεροι κωδικοί θα μπορούσαν να είναι πιο κατάλληλοι. Η απόδοση εξαρτάται επίσης από το είδος των λαθών που αναμένουμε και το σχετικό ποσοστό τους (π.χ. οι μεγάλοι κώδικες είναι καλοί αν έχουμε αρκετά ορθογραφικά λάθη, όχι τυχαία λάθη).
- Αν κάποιος θέλει να επιλέξει την καλύτερη επιλογή σε ένα συγκεκριμένο περιβάλλον εφαρμογής, εκτός από την παραπάνω ανάλυση, μπορεί να εκτελέσει ad hoc πειράματα και για το λόγο αυτό ο κώδικας για την εκτέλεση των

προαναφερθέντων πειραμάτων με διάφορα μεγέθη κωδικών, έχει δημοσιοποιηθεί. Επιπλέον, και για τη διευκόλυνση των συγκριτικών αποτελεσμάτων, έχουμε αναρτήσει το πλήρες σύνολο δεδομένων που περιέχει 574.883 διακριτές ελληνικές λέξεις και 4,32

ορθογραφικά λάθη ανά λέξη κατά μέσο όρο, συνολικά πάνω από 3 εκατομμύρια μορφές ελληνικών λέξεων (3.063.143) στον Τζίτζικα (2021).

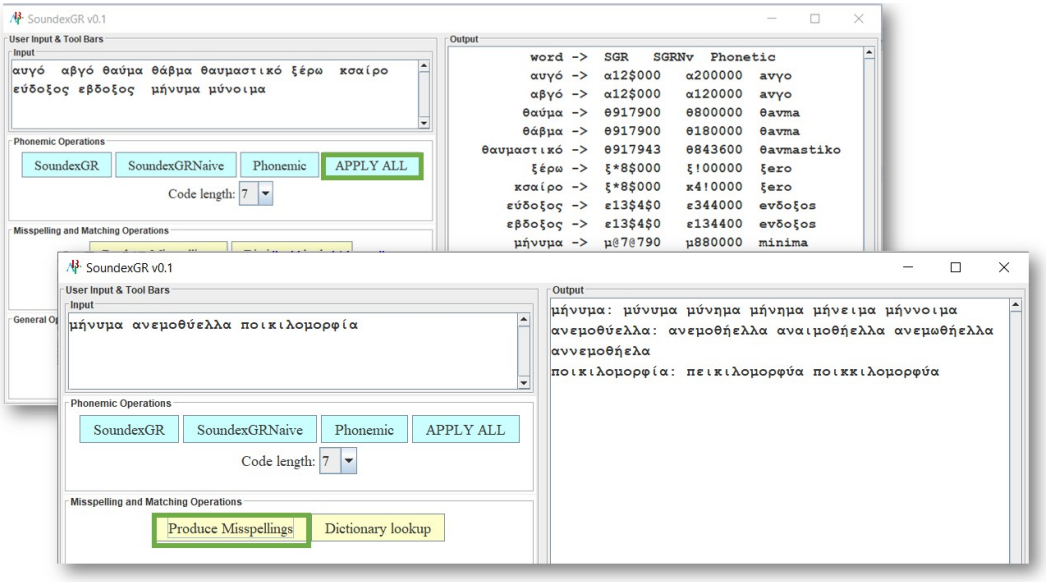
4.11 Εφαρμογή και αποτελεσματικότητα

Όσον αφορά την αποδοτικότητα, χρησιμοποιώντας ένα μηχάνημα με 1,8 GHz i7, 4MB cache και 16 GB RAM, το Soundex_{GR} κωδικοποιεί τις λέξεις κάθε συνόλου 2.500 λέξεων σε 2,5 δευτερόλεπτα, δηλαδή κάθε λέξη χρειάζεται 1ms για να κωδικοποιηθεί, ενώ το *Soundexnaive* σε 0,4 δευτερόλεπτα, δηλαδή χρειάζεται 0,016 ms ανά λέξη. Δεδομένου ότι *το* $\text{Soundex}^{comp}_{GR}$ χρησιμοποιεί και τις δύο υλοποιήσεις για την κωδικοποίηση μιας λέξης, χρειάζεται 1,016 ms ανά λέξη.

Για τον υπολογισμό των κωδικών Soundex_{GR} για κάθε λέξη του λεξικού που περιγράφεται στην §4.7,

Δηλαδή, για περισσότερες από μισό εκατομμύριο λέξεις, η υλοποίησή μας (χρησιμοποιώντας Java 8) χρειάζεται λιγότερο από 2 δευτερόλεπτα (συγκεκριμένα 1,684 msecs) χρησιμοποιώντας ένα μηχάνημα με i7 1,9 GHz, 8MB cache και 16 GB RAM.

Η υλοποίηση όλων των αλγορίθμων, καθώς και τα σύνολα δεδομένων αξιολόγησης, είναι δημόσια διαθέσιμα στη διεύθυνση https://github.com/YannisTzitzikas/SoundexGR. Επιπλέον, παρέχεται επίσης ένα εργαλείο (edi- tor) που βοηθά τον σχεδιαστή να επιλέξει τη μέθοδο που θα εφαρμοστεί: εμφανίζει όλους τους κωδικούς για τις λέξεις του κειμένου εισόδου, ένα στιγμιότυπο οθόνης δίνεται στην Εικόνα 16.



Εικόνα **16**: Ένα εργαλείο για την οπτική επιθεώρηση των παραγόμενων κωδικών, την κατά προσέγγιση αντιστοίχιση και άλλα

4.12 Εφαρμογές

Η απλότητα και η αποτελεσματικότητα του προτεινόμενου αλγορίθμου τον καθιστά εφαρμόσιμο σε ένα ευρύ φάσμα εργασιών. Μπορεί να αξιοποιηθεί όποτε θέλουμε να βρούμε αντιστοιχίες μεταξύ (γραπτών ή προφορικών) περιγραφών στα ελληνικά. Γενικά, αυτοί οι φωνητικοί κώδικες μπορούν να χρησιμοποιηθούν για την αντιμετώπιση *λέξεων*

εκτός λεξιλογίου (Out-Of-Vocabulary - OOV), ένα πρόβλημα που εμφανίζεται συχνά και σε ποικίλα συμφραζόμενα. Πράγματι, οι φωνητικοί κώδικες μπορούν να αξιοποιηθούν για την υποστήριξη διαφόρων ειδών αντιστοίχισης, ανάλογα με το πλαίσιο. Όπως φαίνεται στην ενότητα 4.10, ο τρόπος χειρισμού των OOV

πρόβλημα εξαρτάται από διάφορους παράγοντες (μέγεθος συλλογής, είδος και ποσοστό σφαλμάτων, μήκος λέξεων). Για να το επαληθεύσουμε σε ένα καθαρό πλαίσιο αντιστοίχισης, υλοποιήσαμε μια πρωτότυπη υπηρεσία αντιστοίχισης όπου ο χρήστης εισάγει μια λέξη και το σύστημα εκτελεί αναζήτηση στο λεξικό ελληνικών λέξεων (που αναφέρεται στην ενότητα 4.7 και περιέχει 574.883 διαφορετικές λέξεις), και αν η λέξη δεν βρεθεί, τότε προτείνει στο χρήστη έναν αριθμό προσεγγιστικών αντιστοιχιών. Σημειώστε ότι το πρόβλημα αυτό είναι ευκολότερο σε ένα πλαίσιο όπου είναι διαθέσιμες και οι συχνότητες των λέξεων (π.χ. στην αυτόματη συμπλήρωση ερωτημάτων στην αναζήτηση στο διαδίκτυο), ωστόσο θέλαμε να εξετάσουμε την συμπεριφορά της αντιστοίχισης αν δεν υπάρχουν πληροφορίες χρήσης. Υλοποιήσαμε την κατά προσέγγιση αντιστοίχιση επιστρέφοντας όλες τις λέξεις του λεξικού που έχουν τον ίδιο κωδικό Soundex_{GR} με τη λέξη που εισήγαγε ο χρήστης. Όπως ήταν αναμενόμενο, οι επιστρεφόμενες λέξεις εξαρτώνται από το μήκος των κωδικών που χρησιμοποιούνται. Για παράδειγμα, για την ανορθόγραφη λέξη μοίνειμα το σύστημα, με μήκος κωδικού Soundex_{GR} ίσο με 12, επιστρέφει δύο προτάσεις μήνυμα, μήνυμα. Παρατηρήστε ότι η απόσταση επεξεργασίας αυτών των λέξεων είναι 4 και 5, καθιστώντας σαφή τη διαφοροποίηση (και το όφελος) αυτής της αντιστοίχισης σε σύγκριση με την αντιστοίχιση που βασίζεται στην απόσταση επεξεργασίας. Λαμβάνουμε τις ίδιες δύο προτάσεις για οποιοδήποτε μήκος κώδικα μεταξύ 7 και 12.

Ωστόσο, αν μειώσουμε περαιτέρω το μήκος σε 6, τότε έχουμε τις 23 προτάσεις που φαίνονται στην Εικόνα 17.

μηνυμάτων, μινιμαλισμού, μινιμαλιστικός, μινιμαλιστικά, μινιμαλιστικό, μινιμαλιστικής, μινιμαλιστικέ, μινιμαλιστικές, μηνύματος, μινιμαλισμό, μινιμαλισμός, μηνύματα, μινιμαλιστή, μηνύματά, μινιμαλιστής, μηνύματός, μινιμαλιστική, μινιμαλιστικού, μινιμαλιστικών, μήνυμα, μινιμαλιστικοί, μινιμαλιστικούς, μινιμαλιστικούς, μήνυμάματα

Σχήμα 17: Προτάσεις για την ανορθόγραφη λέξη μοίνειμα με βάση τον κωδικό μήκους = 6

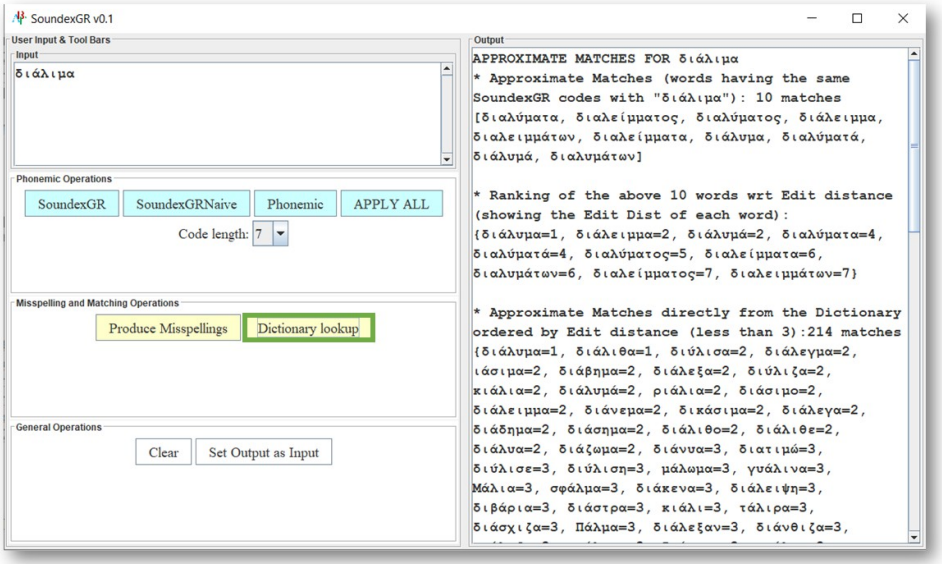
Αυτό υποδηλώνει ότι οι φωνητικοί κώδικες μπορούν να χρησιμοποιηθούν και για πιο εξελιγμένες υπηρεσίες, π.χ. αν ο αριθμός των λέξεων με τον ίδιο κωδικό είναι μεγάλος, τότε μπορούμε να τις κατατάξουμε ανάλογα με την απόσταση επεξεργασίας τους. Ο επιστρεφόμενος ταξινομημένος κατάλογος θα περιλαμβάνει λέξεις που ακούγονται το ίδιο αλλά μπορεί να έχουν πολλά ορθογραφικά λάθη (επομένως δεν θα επιστρεφόταν από την απόσταση επεξεργασίας), οι οποίες στη συνέχεια θα ταξινομηθούν σε σχέση με την απόσταση επεξεργασίας επιτρέποντας με αυτόν τον τρόπο τον έλεγχο του αριθμού των προτάσεων. Ένα παράδειγμα για τη λέξη διάλιμα παρουσιάζεται στην Εικόνα 18, αποδεικνύοντας ότι η κατάταξη με την απόσταση επεξεργασίας επί των κωδικών Soundex_{GR} δίνει καλύτερα αποτελέσματα από την εφαρμογή απευθείας της απόστασης επεξεργασίας, καθώς η τελευταία περιλαμβάνει εντελώς άσχετες λέξεις.

Επιπλέον, δεδομένου ότι οι κωδικοί μπορούν να υπολογιστούν μία φορά (κάτι που δεν είναι δυνατό με την απόσταση επεξεργασίας), αυτό προσφέρει μια πιο αποτελεσματική μέθοδο για τον υπολογισμό προσεγγιστικών αντιστοιχιών.

Για την υποστήριξη της διαδικασίας σχεδιασμού τέτοιων υπηρεσιών, η εφαρμογή επιτρέπει τη δοκιμή των παραπάνω υπηρεσιών με τη χρήση διαφόρων μηκών κώδικα. Προσφέρει επίσης μια μέθοδο που δέχεται ως είσοδο μια λέξη και παράγει διάφορα ορθογραφικά λάθη, επιτρέποντας στον χρήστη να επιλέξει εύκολα ορθογραφικά λάθη για τον έλεγχο της κατά προσέγγιση αντιστοίχισης (όπως φαίνεται στο κάτω μέρος της Εικόνας 16).

4.12.1 Ενδεικτικά πλαίσια εφαρμογής

Παρακάτω περιγράφουμε πώς μπορούν να χρησιμοποιηθούν αυτοί οι κωδικοί για την αντιμετώπιση του προβλήματος των λέξεων *εκτός λεξιλογίου (OOV)* σε διάφορα πλαίσια.



Σχήμα **18**: Επίδειξη μεθόδων προσεγγιστικής αντιστοίχισης

- **Υπηρεσίες αυτόματης συμπλήρωσης.** Κάθε έργο w στον κατάλογο των πιθανών συμπληρωμάτων ερωτημάτων (που αντιστοιχούν στα συχνά ερωτήματα σύμφωνα με τα αρχεία καταγραφής ερωτημάτων) μπορεί να συνοδεύεται από τον κωδικό Soundex_{GR} . Εάν η είσοδος του χρήστη περιέχει μια λέξη w' που δεν υπάρχει στο C, αντί να αναζητούνται λέξεις με μικρή απόσταση επεξεργασίας, μπορούν να ζητηθούν και οι λέξεις που έχουν τον ίδιο Soundex_{GR} . Για την υποστήριξη προτάσεων με βάση τα γράμματα, μπορεί να χρησιμοποιηθεί και μια δομή δεδομένων trie (όπως αυτή των Fafalios and Tzitzikas (2015)) των κωδικών Soundex_{GR} για παράλληλη διάσχιση, δηλαδή για κάθε γράμμα που πληκτρολογεί ο χρήστης διατρέχουμε τόσο την trie των συχνών ερωτημάτων όσο και την trie των κωδικών Soundex_{GR} των ερωτημάτων αυτών και τελικά προτείνουμε στο χρήστη συμπληρώσεις με βάση το περιεχόμενο και των δύο προσπαθειών.
- **Υπηρεσίες ανάκτησης.** Κάθε έργο w $στο$ λεξιλόγιο V ενός ανεστραμμένου αρχείου, μπορεί να συνοδεύεται από τον κωδικό Soundex_{GR} . Εάν το ερώτημα του χρήστη περιέχει μια λέξη w' που δεν υπάρχει στο V (για παράδειγμα, οι Cucerzan and Brill (2004) αναφέρουν ότι ορθογραφικά λάθη εμφανίζονται σε ποσοστό έως και 15% των ερωτημάτων αναζήτησης στο διαδίκτυο), αντί να αναζητούνται μόνο λέξεις με μικρή απόσταση επεξεργασίας, μπορούν να χρησιμοποιηθούν και οι λέξεις που έχουν τον ίδιο Soundex_{GR} . Στη συνέχεια, οι κωδικοί Soundex_{GR} των λέξεων μπορούν επίσης να αξιοποιηθούν για την παραγωγή των αποσπασμάτων των επιτυχιών που θα εμφανίζονται στα αποτελέσματα της αναζήτησης. Το απόσπασμα μιας επιτυχίας είναι ένα μικρό απόσπασμα του εν λόγω εγγράφου που περιέχει τις περισσότερες από τις λέξεις του ερωτήματος, το οποίο υπολογίζεται κατά τη στιγμή του ερωτήματος με τη χρήση διαδοχικής αναζήτησης κειμένου. Κατά συνέπεια, εάν το τοπικά αποθηκευμένο κειμενικό περιεχόμενο των ευρετηριασμένων εγγράφων κωδικοποιείται με τη χρήση του Soundex_{GR} , τότε αυτό θα επιταχύνει τη διαδοχική αναζήτηση που απαιτείται για την επιλογή του αποσπάσματος προς εμφάνιση.
  Άλλες σύγχρονες εφαρμογές της αναζήτησης σε πραγματικό χρόνο, π.χ. μέθοδοι για τη σύνδεση κειμένου με μια βάση γνώσης με επαληθευμένους ισχυρισμούς (όπως στο Maliaroudakis et al. (2021)), για την υποβοήθηση της ανίχνευσης ψευδών ειδήσεων, μπορούν επίσης να επωφεληθούν από τη φωνητική αντιστοίχιση.

- **Αναγνώριση ονομαστικών οντοτήτων.** Οι σύγχρονες μέθοδοι για την Εξαγωγή Ονοματοποιημένων Οντοτήτων βασίζονται σε καθαρές μεθόδους NLP και σε μεθόδους βασισμένες στη γνώση (Mountantonakis and Tzitzikas (2020)). Η εξαγωγή ονομαστικών οντοτήτων, βασίζεται συνήθως σε λίστες οντοτήτων (π.χ. Χώρες κ.λπ.) οι οποίες περιλαμβάνουν τα ονόματα των οντοτήτων (και εναλλακτικά ονόματα, όπως στα συνδεδεμένα ανοικτά δεδομένα). Οι λίστες αυτές μπορεί επίσης να περιέχουν τους φωνητικούς κώδικες αυτών των

ονόματα για την επιτάχυνση της αντιστοίχισης και την αντιμετώπιση των μορφολογικών παραλλαγών. Πράγματι, η έρευνα re- cent των Singh κ.ά. (2020) δείχνει ότι τα στοιχεία των σύγχρονων συστημάτων απάντησης ερωτήσεων (που βασίζονται σε μεγάλο βαθμό στην αναγνώριση οντοτήτων) είναι πολύ ευάλωτα στις μορφολογικές παραλλαγές των λέξεων στις ερωτήσεις που αναφέρονται σε οντότητες.

- **Ενσωματώσεις λέξεων και ML.** Όπως αναφέρεται στο Piktus et al. (2019), οι υπάρχουσες προσεγγίσεις για την παραγωγή ενσωματώσεων λέξεων δεν μπορούν να παρέχουν ενσωματώσεις για λέξεις που δεν έχουν παρατηρηθεί κατά τη στιγμή της εκπαίδευσης. Για παράδειγμα, για την αγγλική γλώσσα, οι Satapathy κ.ά. (2017) χρησιμοποίησαν τον αλγόριθμο Soundex για να μετατρέψουν τις εκτός λεξιλογίου έννοιες σε εντός λεξιλογίου και ανέλυσαν τον αντίκτυπό του στην εργασία ανάλυσης συναισθήματος, ενώ οι Satapathy κ.ά. (2019) πρότειναν ένα λεξικό βασισμένο σε έννοιες που εκμεταλλεύεται τα φωνητικά χαρακτηριστικά για να κανονικοποιήσει τις εκτός λεξιλογίου έννοιες σε εντός λεξιλογίου έννοιες (Huang κ.ά. (2020)). Μια ανάλογη κατεύθυνση θα μπορούσε να διερευνηθεί για την ελληνική γλώσσα, καθώς υπάρχουν ήδη προτάσεις για τη δημιουργία ενσωματώσεων για την ελληνική γλώσσα,
π.χ. η μέθοδος ensemble που περιγράφεται στους Lioudakis et al. (2019), η μέθοδος αναγνώρισης ονομαστικών οντοτήτων από την ελληνική νομοθεσία που περιγράφεται στους Angelidis et al. (2018), ενώ μια αξιολόγηση των ελληνικών Word Embeddings περιγράφεται στους Outsios et al. (2019), που δεν περιλαμβάνει το πιο πρόσφατο ελληνικό BERT Koutsikakis et al. (2020). Οι λέξεις εκτός λεξιλογίου (Out-Of- Vocabulary - OOV) πρέπει να αντιμετωπιστούν σε όλες τις περιπτώσεις, για παράδειγμα το λεξικό που χρησιμοποιήσαμε περιέχει περίπου 500K ελληνικές λέξεις, ενώ το ελληνικό BERT Koutsikakis et al. (2020) περιέχει ενσωματώσεις μόνο για 35K λέξεις.

Γενικά, οι εφαρμογές των αλγορίθμων φωνητικής κωδικοποίησης χρησιμοποιούνται ευρέως στη σύγχρονη τεχνολογία διαμόρφωσης, τόσο στην αρχική όσο και στην τροποποιημένη μορφή τους, αναλυτικός κατάλογος παρατίθεται στο Vykhovanets et al. (2020).

5. Συμπέρασμα

Παρουσιάσαμε μια οικογένεια φωνητικών αλγορίθμων για την Ελληνική Γλώσσα προσαρμόζοντας τον αρχικό Soundex στα χαρακτηριστικά της Ελληνικής Γλώσσας και διευρύνοντας τους κανόνες, όπως έκαναν οι περισσότεροι σύγχρονοι φωνητικοί αλγόριθμοι. Συγκεκριμένα, παρουσιάσαμε το Soundex_{GR} και μια απλούστερη παραλλαγή που ονομάζεται $Soundex^{naive}_{GR}$, και οι δύο παράγουν κώδικες 4 χαρακτήρων. Εν συντομία, πριν κωδικοποιηθεί μια λέξη, γίνεται προεπεξεργασία και η προεπεξεργασία αυτή περιλαμβάνει: εντοπισμό των περιπτώσεων που ένα φωνήεν ακούγεται ως σύμφωνο στα ελληνικά, ομαδοποίηση φωνηέντων που κάνουν διαφορετικό ήχο όταν ζευγαρώνουν μεταξύ τους, αφαίρεση του τονισμού και αποσυναρμολόγηση των διγραμμάτων σε μεμονωμένα γράμματα. Επιπλέον, ορίσαμε το $\text{Soundex}^{comp}_{GR}$ που συνδυάζει τα δύο προηγούμενα στο διαδικασία αντιστοίχισης.

Για να προσδιορίσουμε ποιοι κανόνες έχουν θετικό αντίκτυπο στον αλγόριθμο, σε διαφορετικά σενάρια σφαλμάτων, αξιολογήσαμε συγκριτικά αυτούς τους αλγορίθμους. Για το σκοπό αυτό, κατασκευάσαμε τέσσερα σύνολα δεδομένων αξιολόγησης: ένα με παρόμοιες ηχητικά ελληνικές λέξεις και άλλα τρία ανάλογα με το είδος του λάθους που μπορεί να συμβεί σε μια λέξη (προσθήκη, διαγραφή ή αντικατάσταση γραμμάτων), που

περιέχουν συνολικά 7.650 λέξεις. Οι αλγόριθμοι επιτυγχάνουν μετρικές (ακρίβεια, ανάκληση) που κυμαίνονται μεταξύ (0,90-0,96, 0,40-0,98) για το Soundex_{GR}, (0,69-0,88, 0,34-0,92) για το $Soundex^{naive}$ και (0,66-0,86, 0,50-0,98) για το Soundex^{comp}. Συνοπτικά, το Soundex^{comp} επιτυγχάνει F-Score ίσο σε 0,91 στο σύνολο δεδομένων με τις similar λέξεις με παρόμοιο ήχο. Έχουμε verified επίσης δει ότι αυτές οι αλ-

συμπεριφέρονται καλύτερα (στη συλλογή αξιολόγησης) από έναν ελληνικό stemmer και δοκιμάσαμε την αποτελεσματικότητά τους σε ένα ελληνικό λεξικό που περιλαμβάνει περισσότερες από μισό εκατομμύριο λέξεις. Επιπλέον, είδαμε ότι το Soundex_{GR} έχει πολύ καλύτερες επιδόσεις σε σύγκριση με μια πλήρη φωνητική μεταγραφή. Σε ένα εκτεταμένο σύνολο δεδομένων που περιέχει κοινά λάθη, είδαμε ότι το Soundex_{GR} επιτυγχάνει το υψηλότερο F-Score (0,97), ξεπερνώντας επίσης την αντιστοίχιση με βάση την Edit Distance. Σε μεγαλύτερα σύνολα δεδομένων (που περιλαμβάνουν μεγάλες λέξεις) το Soundex_{GR}

διατηρεί την υπεροχή του, αλλά με μήκος κώδικα ίσο ή μεγαλύτερο από 6, ενώ το μήκος που δίνει το βέλτιστο F-Score είναι 12. Η αποτελεσματικότητα, η απλότητα και η αποδοτικότητα του προτεινόμενου αλγορίθμου τον καθιστά εφαρμόσιμο σε ένα ευρύ φάσμα εργασιών. Το μήκος των κωδίκων μπορεί να διαμορφωθεί ανάλογα με την επιθυμητή απόδοση ακρίβειας-ανάκλησης και πιστεύουμε ότι τα πειραματικά αποτελέσματα που αναφέρονται σε αυτή την εργασία παρέχουν βοήθεια για μια τέτοια διαμόρφωση- είδαμε ότι το μέγεθος του λεξιλογίου, η κατανομή των μεγεθών των λέξεων, καθώς και το είδος και το ποσοστό των λαθών καθορίζουν το μήκος του κώδικα που δίνει τη βέλτιστη απόδοση. Επιπλέον, είδαμε ότι αυτοί οι κώδικες μπορούν να χρησιμοποιηθούν σε συνδυασμό με άλλες μεθόδους προσεγγιστικής αντιστοίχισης για την επίτευξη πιο εξελιγμένων μεθόδων αντιστοίχισης που μπορούν να είναι πιο αποτελεσματικές και ακόμη πιο αποδοτικές. Η υλοποίηση του αλγορίθμου, μια αυτόνομη εφαρμογή για προσεγγιστικό ταίριασμα που μπορεί να υποστηρίξει τον υπογράφοντα στην επιλογή του μήκους κώδικα που θα χρησιμοποιήσει, καθώς και τα σύνολα δεδομένων αξιολόγησης, είναι διαθέσιμα στη διεύθυνση https://github.com/YannisTzitzikas/SoundexGR. Επιπλέον, και για τη διευκόλυνση των συγκριτικών αποτελεσμάτων, δημιουργήσαμε και δημοσιοποιήσαμε το σύνολο δεδομένων GMW (Greek Misspelled Words) Tzitzikas (2021), ένα σύνολο δεδομένων που περιέχει 574.883 διακριτές ελληνικές λέξεις και 4,32 ορθογραφικά λάθη ανά λέξη κατά μέσο όρο, συνολικά περισσότερες από 3 εκατομμύρια μορφές ελληνικών λέξεων.

Μια κατεύθυνση που αξίζει να ερευνηθεί είναι η διερεύνηση του κατά πόσον αυτοί οι φωνητικοί κώδικες θα μπορούσαν να αξιοποιηθούν σε διάφορα μοντέλα βαθιάς μάθησης για NLP για την ελληνική γλώσσα (π.χ. Lioudakis et al. (2019) για word embeddings, Angelidis et al. (2018) για named entity recognition από την ελληνική νομοθεσία), για να γίνουν αυτά τα μοντέλα πιο ανεκτικά σε ανορθόγραφες ή λανθασμένα προφερόμενες λέξεις. Ένα άλλο θέμα που αξίζει να ερευνηθεί είναι ο υπολογισμός n-γραμμάτων τέτοιων φωνητικών κωδίκων σε διάφορα σώματα κειμένων και στη συνέχεια η αξιολόγηση του κατά πόσον μπορούν να βελτιώσουν περαιτέρω τον χειρισμό των εκτός λεξιλογίου λέξεων. Στο ίδιο μήκος κύματος, δεδομένου ότι η εργασία μας δεν αφορά την αποσαφήνιση του νοήματος των λέξεων, π.χ. στη λέξη λόγια στις δύο φράσεις "λόγια τωz azθρώπωz" και "p λόγια παράδοσp" θα αποδοθεί ο ίδιος φωνητικός κώδικας ακόμη και αν η σημασία είναι διαφορετική, Τα N-grams και άλλες πιο πρόσφατες μέθοδοι, είτε πάνω στις αρχικές λέξεις είτε πάνω στη φωνημική μεταγραφή τους, θα μπορούσαν να διερευνηθούν στο μέλλον για τον εντοπισμό του σωστού νοήματος μιας εμφάνισης λέξης.

Αναγνώριση

Πολλές ευχαριστίες στην Κατερίνα Παπαντωνίου για τα σχόλιά της και για τη διόρθωση της εργασίας, καθώς και στους ανώνυμους κριτές για τα γόνιμα σχόλια και τις προτάσεις τους.

Αναφορές

Ahmed, A. F., **Sherif, M. A.**, **and Ngonga Ngomo, A.-C.** 2019**.** Do your resources sound similar? on the impact of using phonetic similarity in link discovery. In *Proceedings of the 10th International Conference on Knowledge Capture*, pp. 53-60.

Angelidis, I., **Chalkidis, I.**, **and Koubarakis, M.** 2018**.** Αναγνώριση, σύνδεση και παραγωγή ονομαστικών οντοτήτων για την ελληνική νομοθεσία. In *Proceedings of the 31st International Conference on Legal Knowledge and Information Systems (JURIX)*, pp. 1-10.

Arvaniti, A. 2007. Ελληνική φωνητική: Φωνητική: Η κατάσταση της τέχνης. *Εφημερίδα της ελληνικής γλωσσολογίας*, 8(1):97-208.

Baruah, D. και Mahanta, A. K. 2015**.** Σχεδιασμός και ανάπτυξη soundex για τη γλώσσα Assamese. *International Journal of Computer Applications*, 117(9).

Beider, A. 2008. Φωνητικό ταίριασμα Beider-morse: Μια εναλλακτική λύση στο soundex με λιγότερα ψευδή

αποτελέσματα.

Avotaynu: Διεθνής Επιθεώρηση Εβραϊκής Γενεαλογίας, 24(2):12.

Christian, P. 1998. Soundex - μπορεί να βελτιωθεί; *Computers in Genealogy*, 6:215-221.

Cucerzan, S. και Brill, E. 2004. Διόρθωση ορθογραφίας ως επαναληπτική διαδικασία που εκμεταλλεύεται τη συλλογική γνώση των χρηστών του διαδικτύου. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 293-300.

da Silva, M. H. L. F., **da Silva Leite, M. T.**, **Sampaio, V.**, **Lynn, T.**, **Endo, P. T. και άλλοι** 2020. Εφαρμογή και ανάλυση τεχνικών σύνδεσης αρχείων για την ολοκλήρωση των βραζιλιάνικων βάσεων δεδομένων υγείας. Το *2020*

Διεθνές συνέδριο για την επίγνωση της κατάστασης στον κυβερνοχώρο, την ανάλυση δεδομένων και την αξιολόγηση (CyberSA),
σελ. 1-2. IEEE.

del Pilar Angeles, M., **Espino-Gamez, A.**, **and Gil-Moncada, J.** 2015. Σύγκριση των λειτουργιών κωδικοποίησης Modified Spanish Phonetic, Soundex και Phonex κατά τη διαδικασία αντιστοίχισης δεδομένων. In *2015 International Conference on Informatics, Electronics & Vision (ICIEV)*, pp. 1-5. IEEE.

Devlin, J., **Chang, M.-W.**, **Lee, K.**, **and Toutanova, K.** 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dimitrakis, E., **Sgontzos, K.**, **Papadakos, P.**, **Marketakis, Y.**, **Papangelis, A.**, **Stylianou, Y.**, **and Tzitzikas, Y.** 2018. On finding the relevant user reviews for advancing conversational faceted search. In **Recupero, D. R.**, **Dragoni, M.**, **Buscaldi, D.**, **Alam, M.**, **and Cambria, E.**, editors, *Proceedings of 4th Workshop on Sentic Computing, Sentiment Analysis, Opinion Mining, and Emotion Detection (EMSASW 2018) Co-located with the 15th Extended Semantic Web Conference 2018 (ESWC 2018), Heraklion, Greece, June 4, 2018*, volume 2111 of *CEUR Workshop Proceedings*, pp. 22-31. CEUR-WS.org.

Dimitrakis, E., **Sgontzos, K.**, **and Tzitzikas, Y.** 2019**.** A survey on question answering systems over linked data and documents. *Journal of Intelligent Information Systems*.

Elmagarmid, A. K., **Ipeirotis, P. G.**, **and Verykios, V. S.** 2006. Ανίχνευση διπλών εγγραφών: Μια έρευνα.
IEEE Transactions on knowledge and data engineering, 19(1):1-16.

Επιτροπάκης, Γ., **Γιούργαλης, Ν.**, **και Κοκκινάκης, Γ.** 1993. Αλγόριθμος υψηλής ποιότητας τονισμού για το σύστημα Greek TTS. In *ESCA Workshop on Prosody*.

Fafalios, P., **Kitsos, I.**, **and Tzitzikas, Y.** 2012**.** Κλιμακούμενη, ευέλικτη και γενική άμεση αναζήτηση επισκόπησης. In *Proceedings of the 21st International Conference on World Wide Web*, pp. 333-336. ACM.

Fafalios, P. and Tzitzikas, Y. 2015**.** Διερευνητική αναζήτηση με προγενέστερη πληκτρολόγηση μέσω αυτόματης συμπλήρωσης με ανοχή σε τυπογραφικά λάθη και σειρά λέξεων . *J. Web Engineering*, 14:80-116.

Φουράκης, Μ., **Μποτίνης, Α.**, **και Κατσαΐτη, Μ.** 1999. Ακουστικά χαρακτηριστικά των ελληνικών φωνηέντων. *Phonetica*, 56(1-2):28-43.

Gautam, V., **Pipal, A.**, **and Arora, M.** 2019. Επανεξέταση του αλγορίθμου Soundex για την ινδική γλώσσα. In
Διεθνές συνέδριο για την καινοτόμο πληροφορική και τις επικοινωνίες, σσ. 47-55. Springer.

Hood, D. 2002. Caverphone: Αλγόριθμος φωνητικής αντιστοίχισης. *Τεχνικό έγγραφο CTP060902, Πανεπιστήμιο Otago, Νέα Ζηλανδία*.

Huang, L., **Zhuang, S.**, **και Wang, K.** 2020. Μια μέθοδος κανονικοποίησης κειμένου για σύνθεση ομιλίας που βασίζεται σε μηχανισμό τοπικής προσοχής. *IEEE Access*, 8:36202-36209.

Karakasidis, A. και Verykios, V. S. 2009**.** Σύνδεση αρχείων με τη χρήση φωνητικών κωδίκων. Στο *2009 Fourth Balkan Conference in Informatics*, σ. 101-106. IEEE.

Karamaroudis, C. και Markidakis, Y. 2006**.** Mitos Greek Stemmer. `https://github.com/YannisTzitzikas/GreekMitosStemmer`. Φοιτητές του CSD-UOC στο πλαίσιο του μαθήματος CS463 Συστήματα Ανάκτησης Πληροφοριών.

Καρανικόλας, Ν. Ν. 2019. Μηχανική εκμάθηση κανόνων φωνητικής μεταγραφής για την ελληνική γλώσσα. In *AIP Conference Proceedings*, τόμος 2116. AIP Publishing LLC.

Karoonboonyanan, T., **Sornlertlamvanich, V.**, **και Meknavin, S.** 1997**.** Ένα ταϊλανδικό σύστημα Soundex για τη διόρθωση της ορθογραφίας. In *Proceeding of the National Language Processing Pacific Rim Symposium*, pp. 633-636.

Kaur, J., **Singh, A.**, **και Kadyan, V.** 2020. Σύστημα αυτόματης αναγνώρισης ομιλίας για τονικές γλώσσες: Κατάσταση προόδου. *Archives of Computational Methods in Engineering*, σ. 1-30.

Koneru, K., **Pulla, V. S. V.**, **and Varol, C.** 2016**.** Performance evaluation of phonetic matching algorithms on English words and street names. In *Proceedings of the 5th International Conference on Data Management Technologies and Applications*, pp. 57-64. SCITEPRESS-Science and Technology Publications, Lda.

Koutsikakis, J., **Chalkidis, I.**, **Malakasiotis, P.**, **and Androutsopoulos, I.** 2020. GREEK-BERT: Οι Έλληνες που επισκέπτονται την οδό Σουσάμι. *11ο Ελληνικό Συνέδριο για την Τεχνητή Νοημοσύνη*.

Kukich, K. 1992. Τεχνικές για την αυτόματη διόρθωση λέξεων σε κείμενο. *Acm Computing Surveys (CSUR)*, 24(4):377-439.

Levenshtein, V. I. 1966. Δυαδικοί κώδικες ικανοί να διορθώνουν διαγραφές, εισαγωγές και αντιστροφές. In *Soviet physics doklady*, volume 10, pp. 707-710.

Li, D. και Peng, D. 2011. Διόρθωση ορθογραφίας για την κινεζική γλώσσα με βάση τον αλγόριθμο pinyin-soundex. In *2011 International Conference on Internet Technology and Applications*, σ. 1-3. IEEE.

Λιουδάκης, Μ., **Όττσιος, Σ.**, **και Βαζιργιάννης, Μ.** 2019. An ensemble method for producing word representations for the Greek language. *arXiv preprint arXiv:1912.04965*.

Maliaroudakis, E., **Boland, K.**, **Dietze, S.**, **Todorov, K.**, **Tzitzikas, Y.**, **and Fafalios, P.** 2021. ClaimLinker: Linking Text to a Knowledge Graph of Fact-checked Claims. Στα *συνοδευτικά πρακτικά του*

το συνέδριο για τον Παγκόσμιο Ιστό 2021 (WWW'2021. ACM.

Medhat, D., **Hassan, A.**, **and Salama, C.** 2015**.** Μια υβριδική διαγλωσσική τεχνική αντιστοίχισης ονομάτων με χρήση νέας τροποποιημένης απόστασης Levenshtein. In *2015 Tenth International Conference on Computer Engineering & Systems (ICCES)*, pp. 204-209. IEEE.

Mikolov, T., **Sutskever, I.**, **Chen, K.**, **Corrado, G. S.**, **and Dean, J.** 2013. Κατανεμημένες αναπαραστάσεις λέξεων και φράσεων και η συνθετικότητά τους. In *Advances in neural information processing systems*,
σελ. 3111-3119.

Mountantonakis, M. και Tzitzikas, Y. 2019. Σημασιολογική ολοκλήρωση συνδεδεμένων δεδομένων μεγάλης κλίμακας: A survey.
ACM Computing Surveys (CSUR), 52(5).

Mountantonakis, M. και Tzitzikas, Y. 2020**.** LODsyndesisIE: εξαγωγή οντοτήτων από κείμενο και εμπλουτισμός με τη χρήση εκατοντάδων συνδεδεμένων συνόλων δεδομένων. In *European Semantic Web Conference*, pp. 168-174. Springer.

Newton, B. 1972. *Η παραγωγική ερμηνεία της διαλέκτου: Μια μελέτη της νεοελληνικής φωνολογίας*, τόμος 8. Αρχείο CUP.

Nguyen, P. H., **Ngo, T. D.**, **Phan, D. A.**, **Dinh, T. P.**, **και Huynh, T. Q.** 2008**.** Ανίχνευση και διόρθωση της βιετναμέζικης ορθογραφίας με τη χρήση των αλγορίθμων Bi-gram, Minimum Edit Distance, SoundEx και ορισμένων πρόσθετων ευρετικών μεθόδων. In *2008 IEEE International Conference on Research, Innovation and Vision for the Future in Computing and Communication Technologies*, pp. 96-102. IEEE.

Ousidhoum, N. D. και Bensaou, N. 2013. Προς την τελειοποίηση του αραβικού soundex. In *International Conference on Application of Natural Language to Information Systems*, σ. 309-314. Springer.

Outsios, S., **Karatsalos, C.**, **Skianis, K.**, **and Vazirgiannis, M.** 2019. Evaluation of Greek Word Embeddings. *arXiv preprint arXiv:1904.04032*.

Papadakos, P., **Vasiliadis, G.**, **Theoharis, Y.**, **Armenatzoglou, N.**, **Kopidaki, S.**, **Marketakis, Y.**, **Daskalakis, M.**, **Karamaroudis, K.**, **Linardakis, G.**, **Makrydakis, G.**, **and others** 2008. The anatomy of mitos web search engine. *arXiv preprint arXiv:0803.2220*.

Παπαντωνίου, Κ. και Τζίτζικας, Γ. 2020. NLP για την ελληνική γλώσσα: Μια σύντομη επισκόπηση. Στο *11ο Ελληνικό Συνέδριο Τεχνητής Νοημοσύνης (SETN 2020)*.

Pennington, J., **Socher, R.**, **and Manning, C. D.** 2014. Glove: Global vectors for word representation. Στα *πρακτικά του συνεδρίου 2014 για τις εμπειρικές μεθόδους στην επεξεργασία φυσικής γλώσσας (EMNLP)*,
σελ. 1532-1543.

Philips, L. 1990. Hanging on the metaphone. *Computer Language*, 7(12 (Δεκέμβριος)). **Philips, L.** 2000. Ο αλγόριθμος αναζήτησης διπλού μεταφώνου. *C/C++ users journal*, 18(6):38-43.

Philips, L. 2013. Metaphone 3. http://aspell.net/metaphone/.

Piktus, A., **Edizel, N. B.**, **Bojanowski, P.**, **Grave, É.**, **Ferreira, R.**, **and Silvestri, F.** 2019**.** Ορθογραφικά λάθη σε ενσωματωμένες λέξεις. In *Procs of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pp. 3226-3234.

Pinto, D., **Vilarino, D.**, **Alem´an, Y.**, **G´omez, H.**, **and Loya, N.** 2012. Ο φωνητικός αλγόριθμος soundex revisited for sms-based information retrieval. In *II Spanish Conference on Information Retrieval CERI*.

Raghavan, H. και Allan, J. 2004**.** Χρήση κωδικών soundex για την ευρετηρίαση ονομάτων σε έγγραφα ASR. In *Proceedings of the Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval at HLT- NAACL 2004*, pp. 22-27. Association for Computational Linguistics.

Russell, R. 1918. Δίπλωμα ευρεσιτεχνίας των Ηνωμένων Πολιτειών 1,261,167. *Ουάσιγκτον, Γραφείο Διπλωμάτων Ευρεσιτεχνίας των Ηνωμένων Πολιτειών*.

Russell, R. 1922. Δίπλωμα ευρεσιτεχνίας των Ηνωμένων Πολιτειών 1,435,663. *Ουάσιγκτον, Γραφείο Διπλωμάτων Ευρεσιτεχνίας των Ηνωμένων Πολιτειών*.

Satapathy, R., **Guerreiro, C.**, **Chaturvedi, I.**, **and Cambria, E.** 2017. Κανονικοποίηση μικροκειμένου με βάση τη φωνητική για ανάλυση συναισθήματος στο twitter. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 407-413. IEEE.

Satapathy, R., **Singh, A.**, **and Cambria, E.** 2019. Phonsenticnet: A cognitive approach to microtext normalization for concept-level sentiment analysis. In *International Conference on Computational Data and Social Networks*, pp. 177-188. Springer.

Σφακιανάκη, A. 2002. Ακουστικά χαρακτηριστικά ελληνικών φωνηέντων που παράγονται από ενήλικες και παιδιά. *Selected papers on theoretical and applied linguistics*, 14:383-394.

Shah, R. 2014. Βελτίωση του αλγορίθμου Soundex για την ινδική γλώσσα με βάση τη φωνητική αντιστοίχιση. *International Journal of Computer Science, Engineering and Applications*, 4(3):31.

Shedeed, H. A. και Abdel, H. 2011. Μια νέα ευφυής μεθοδολογία για την αξιολόγηση ερωτήσεων

σύντομης απάντησης μέσω υπολογιστή με βάση έναν νέο βελτιωμένο φωνητικό αλγόριθμο Soundex για την αραβική γλώσσα. *International Journal of Computer Applications*, 34(10):40-47.

Singh, K., **Lytra, I.**, **Radhakrishna, A. S.**, **Shekarpour, S.**, **Vidal, M.-E.**, **και Lehmann, J.** 2020. Κανείς δεν είναι τέλειος: dbpedia

γράφημα γνώσης. *Journal of Web Semantics*, 65.

Θεμιστοκλέους, Γ. 2011. Υπολογιστική Ελληνική Φωνολογία: IPAGreek. Στα *Πρακτικά του 10ου Διεθνούς Συνεδρίου Ελληνικής Γλωσσολογίας* .

Θεμιστοκλέους, Γ . 2017. IPAGreek: Computational Greek Phonology. `https://github.com/themistocleous/IPA_Greek`.

Θεμιστοκλέους, Γ. 2019. Ταξινόμηση διαλέκτων από έναν μόνο ηχητικό ήχο με τη χρήση βαθιών νευρωνικών δικτύων.

Frontiers in Communication, 4:64.

Trudgill, P. 2009. Συστήματα φωνηέντων της ελληνικής διαλέκτου, θεωρία διασποράς φωνηέντων και κοινωνιογλωσσολογική τυπολογία.

Journal of Greek Linguistics, 9(1):165-182.

Tzitzikas, Y. 2021. GMW - Greek Misspelled Words. http://islcatalog.ics.forth.gr/dataset/gmw.

Vykhovanets, V., **Du, J.**, **και Sakulin, S.** 2020. Επισκόπηση των αλγορίθμων φωνητικής κωδικοποίησης.

Automation and Remote Control, 81(10):1896-1910.

Yadav, V. και Bethard, S. 2018**.** Μια επισκόπηση των πρόσφατων εξελίξεων στην αναγνώριση ονομαστικών οντοτήτων από μοντέλα βαθιάς μάθησης. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2145-2158.

Yahia, M. E., **Saeed, M. E.**, **και Salih, A. M.** 2006. Ένας ευφυής αλγόριθμος για την αραβική συνάρτηση soundex με χρήση διαισθητικής ασαφούς λογικής. Στο *2006 3rd International IEEE Conference Intelligent Systems*,
σελ. 711-715. IEEE.