NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCES
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

PROGRAM OF POSTGRADUATE STUDIES

MASTER'S THESIS

Named Entity Recognition and Linking in Greek Legislation

Iosif E. Angelidis

SUPERVISORS: **Manolis Koubarakis**, Professor
Ilias Chalkidis, PhD Candidate

ATHENS

MAY 2018

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αναγνώριση Ονομασμένων Οντοτήτων και Σύνδεση στην Ελληνική Νομοθεσία

Ιωσήφ Ε. Αγγελίδης

ΕΠΙΒΛΕΠΟΝΤΕΣ: **Μανόλης Κουμπαράκης**, Καθηγητής
Ηλίας Χαλκίδης, Υποψήφιος Διδάκτωρ

ΑΘΗΝΑ

ΜΑΙΟΣ 2018

MASTER'S THESIS

Named Entity Recognition and Linking in Greek Legislation

Iosif E. Angelidis
ID: M1477

SUPERVISORS: **Manolis Koubarakis**, Professor
Ilias Chalkidis, PhD Candidate

EXAMINING COMMITTEE: **Manolis Koubarakis**, Professor
Dimitrios Gunopulos, Professor

Examination Date: May 16th, 2018

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αναγνώριση Ονομασμένων Οντοτήτων και Σύνδεση στην Ελληνική Νομοθεσία

Ιωσήφ Ε. Αγγελίδης
Α.Μ.: Μ1477

ΕΠΙΒΛΕΠΟΝΤΕΣ: **Μανόλης Κουμπαράκης**, Καθηγητής
Ηλίας Χαλκίδης, Υποψήφιος Διδάκτωρ

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ: **Μανόλης Κουμπαράκης**, Καθηγητής
Γουνόπουλος Δημήτριος, Καθηγητής

Ημερομηνία Εξέτασης: 16 Μαΐου, 2018

ABSTRACT

We show how entity recognition in Greek legislation texts can be achieved by utilizing a named entity recognizer (NER). Our work is the first of its kind for the Greek language in such an extended form and one of the few that examines legal text. We apply grid search on multiple neural network architectures and combination of hyper-parameters to maximize the efficiency of our approach. We show that, utilizing a big legal corpus we built word/token-shape embeddings using `Word2Vec`, and finally achieve 86% accuracy on average in recognition of organizations, legal references, geographical landmarks, persons, geo-political entities (GPEs) and public documents. The evaluation of our methodology is based on the metrics of precision, recall, f_1-score per entity type for each neural network. Finally, we measure the ratio of correctly guessed links for the interlinking of RDF datasets produced by our approach with well-known public datasets and how new knowledge can be inferred indirectly by our approach from DBpedia, ELI (Europeal Legislation Identifier) and GAG (Greek administrative geography) of Kallikratis.

SUBJECT AREA: Natural Language Processing, Semantic Web, Artificial Intelligence

KEYWORDS: Named Entity Recognition and Linking, Legislative Knowledge Representation, Entity Reference Representation, Linked Open Data, Deep Learning, Entity Generation

ΠΕΡΙΛΗΨΗ

Δείχνουμε πώς η αναγνώριση οντοτήτων σε κείμενα Ελληνικής νομοθεσίας μπορεί να επιτευχθεί με την χρήση ενός αναγνωριστή ονομασμένων οντοτήτων (named entity recognizer, NER). Η δουλειά μας είναι η πρώτη του είδους της που ασχολείται με την ελληνική γλώσσα σε τόσο βάθος και μία από ελάχιστες που μελετούν νομικό κείμενο. Εφαρμόζουμε αναζήτηση δικτύου (grid search) σε πολλαπλές αρχιτεκτονικές νευρωνικών δικτύων και συνδυασμούς υπερ-παραμέτρων (hyper-parameters) για να μεγιστοποιήσουμε την αποτε- λεσματικότητα της προσέγγισής μας. Δείχνουμε ότι, χρησιμοποιώντας ένα μεγάλο νομικό λεξικό χτίσαμε ενσωματωμένες/συμβολικές λέξεις (word/token-shaped embeddings) χρη- σιμοποιώντας το `Word2Vec` και τελικά πετυχαίνουμε κατά μέσο όρο 86% ακρίβεια σε ανα- γνώριση οργανισμών, νομικών αναφορών, γεωγραφικών τοποθεσιών, ανθρώπων, γεω-πολιτικών οντοτήτων (GPEs) και δημοσίων εγγράφων. Η αξιολόγηση της μεθοδολογίας μας βασίζεται στις μετρικές της ακριβείας (precision), της ανάκλησης (recall) και της f_1 μετρικής (f1-score) ανά τύπο οντότητας για κάθε νευρωνικό δίκτυο. Τέλος, μετράμε την αναλογία των σωστά προβλεπόμενων συνδέσμων για την διασύνδεση RDF συνόλων δεδομένων (datasets) που παράγονται από την προσέγγισή μας με άλλα γνωστά σύνολα δεδομένων που έχουν εκδοθεί δημόσια και πώς μπορούμε να εξάγουμε νέα γνώση έμμεσα με την προσέγγισή μας από την DBpedia, το ELI (Europeal Legislation Identifier) και το GAG (Greek administrative geography, Ελληνική διοικητική γεωγραφία) του Καλλικράτη.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Επεξεργασία Φυσικής Γλώσσας, Σημασιολογικός Ιστός, Τεχνητή Νοημοσύνη

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Αναγνώριση Ονομασμένων Οντοτήτων και Σύνδεση, Αναπαράσταση Νομικής Γνώσης, Αναπαράσταση Αναφορών Οντοτήτων, Ανοιχτά Συνδεδεμένα Δεδομένα, Βαθιά Μάθηση, Παραγωγή Οντοτήτων

This thesis is dedicated to my parents. Thank you for your patience and feedback.
Without your continuous support I simply would not have made it this far in life.

ACKNOWLEDGEMENTS

I would like to thank my professor and supervisor Manolis Koubarakis for offering me the chance to tackle the interesting problem presented here and for his invaluable guidance throughout the thesis.

I would also like to thank my supervisor, Ilias Chalkidis, for his patience and support throughout the entire thesis. His contribution was critical for its completion.

Finally, many thanks to my colleagues at Pyravlos team for their positive energy and useful feedback as well as my parents for their unconditional support.

ΣΥΝΟΠΤΙΚΗ ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Πρόσφατα, υπάρχει έντονο ενδιαφέρον στην ενσωμάτωση τεχνολογιών Τεχνητής Νοημοσύνης στην περιοχή της νομοθεσίας όσον αφορά την επεξεργασία κειμένου, την αναπαράσταση γνώσης και τον συμπερασμό. Η επεξεργασία νομικού κειμένου [1] αποτελεί μια αναπτυσσόμενη ερευνητική περιοχή, αποτελούμενη από εργασίες όπως νομικές ερωταπαντήσεις [2], εξαγωγή νομικών οντοτήτων [3, 4] και παραγωγή νομικού κειμένου [5]. Το ίδιο ισχύει και στην περιοχή αναπαράστασης νομικής γνώσης, όπου νέες νόρμες έχουν αναπτυχθεί και αρχίζουν να υιοθετούνται βασισμένες σε τεχνολογίες σημασιολογικού ιστού. Σχετικές συνεισφορές είναι το European Legislation Identifier (ELI) [6, 7, 8] για νομοθεσία, το European case Law Indentifier (ECLI) [9, 10] για υποθέσεις δικαστηρίων, καθώς επίσης και το Legal Knowledge Interchange Format (LKIF) [11, 12] και το LegalRule ML [13, 14] για κωδικοποίηση προχωρημένων νομικών αναφορών, όπως κανόνες και νόρμες. Η ακαδημαϊκή κοινότητα σκοπεύει να αναπτύξει εργαλεία και εφαρμογές για να βοηθήσει επαγγελματίες νομικούς (π.χ., δικαστές, δικηγόρους κτλ.) καθώς επίσης και πολίτες. Με βάση αυτές τις πρακτικές, η ομάδα μας δημιούργησε το Nomothesi@[1] [15], μια πλατφόρμα που προσφέρει στον Ιστό την Ελληνική νομοθεσία ως συνδεδεμένα δεδομένα για να βοηθήσει στην υποβολή σύνθετων ερωτήσεων SPARQL και στην ανάπτυξη σχετικών εφαρμογών.

Πηγαίνοντας ένα βήμα παραπέρα και προκειμένου να δημιουργήσουμε μια σύνδεση, ως σημείο αναφοράς, μεταξύ αυτών των σχετικών ερευνητικών περιοχών της επιστήμης των δεδομένων (της επεξεργασίας φυσικής γλώσσας και του σημασιολογικού ιστού), αναπτύξαμε έναν Named Entity Recognizer (NER) και Linker (NEL) για την Ελληνική νομοθεσία. Για το πρώτο έργο, θα συγκρίνουμε και αποτιμήσουμε state-of-the-art αρχιτεκτονικές νευρωνικών δικτύων (RNNs) για να αναγνωρίσουμε τους παρακάτω τύπους οντοτήτων: άτομα, οργανισμούς, γεωπολιτικές οντότητες, νομικές αναφορές, γεωγραφικά τοπωνύμια και αναφορές εγγράφων του δημοσίου από Ελληνική νομοθεσία. Χρησιμοποιούμε τον καλύτερο entity recognizer στο dataset της Ελληνικής νομοθεσίας [15] και παράγουμε νέα γνώση για οντότητες κωδικοποιημένης σε RDF χρησιμοποιώντας ένα καινούριο λεξικό. Δεδομένων αυτών των τριπλετών, χρησιμοποιούμε hand-crafted κανόνες και το entity linking εργαλείο Silk [16, 17] προκειμένου να κάνουμε normalize και να συνδέσουμε τις αναφορές του κειμένου που εξάγουμε με οντότητες από δημόσια ανοιχτά datasets (Greek administrative units and Greek politicians). Επίσης, δημοσιοποιούμε ένα νέο RDF dataset για Ελληνικά γεωγραφικά τοπωνύμια, τα οποία συνήθως εμφανίζονται σε νομοθεσία σχετική με αστικό, αγροτικό και περιβαλλοντικό σχεδιασμό. Οι βασικές συνεισφορές παραθέτονται παρακάτω:

- Μελετούμε το έργο της εξαγωγής ονομασμένων οντοτήτων στην Ελληνική Νομοθεσία εφαρμόζοντας και αξιολογώντας state-of-the-art αρχιτεκτονικές νευρωνικών δικτύων [4], ενώ μελετάμε και μια κάπως συνθετότερη, που πηγαίνει καλύτερα από τις υπόλοιπες έστω και για λίγο.

[1] http://legislation.di.uoa.gr

- Παρουσιάζουμε ένα καινούριο RDF λεξικό για την αναπαράσταση και σύνδεση αναφορών κειμένου σε οντότητες Ελληνικής νομοθεσίας. Θεωρούμε το RDF ως ένα μονό μοντέλο δεδομένων για την αναπαράσταση μετα-δεδομένων ενός νομικού κειμένου και γνώσης που κωδικοποιείται στο κείμενο.

- Χρησιμοποιούμε το Nomothesi@ NER, βασισμένο στο καλυτερο μοντέλο BILSTM-BILSTM-LR στο dataset της Ελληνικής νομοθεσίας και παράγουμε νέα δεδομένα για αναφορές οντοτήτων, τις οποίες περιγράφουμε χρησιμοποιώντας το νέο RDF λεξικό.

- Συνδέουμε τις αναφορές με τα datasets χρησιμοποιώντας τεχνικές βασισμένες σε κανόνες και το εργαλείο Silk.

- Προσφέρουμε δημόσια ένα νέο benchmark dataset 276 σημειωμένων νομικών κειμένων, τα οποία μπορούν να επαναχρησιμοποιηθούν για να εκπαιδεύσουμε και να δοκιμάσουμε διαφορετικούς αλγορίθμους που σχετίζονται με named entity recognition και linking. Επίσης προσφέρονται προ-εκπαιδευμένα word embeddings ειδικευμένα σε Ελληνικό νομικό κείμενο.

- Παράγουμε ένα νέο dataset Ελληνικών γεωγραφικών τοπωνυμίων βασισμένοι στα αποτελέσματα του Nomothesi@ NER εφαρμόζοντας ευριστικούς κανόνες. Σε ένα ερευνητικό project που ξεκίνησε η ομάδα μας, αυτό το dataset θα επαυξηθεί με επιπλέον γεωγραφική πληροφορία (π.χ., χωρικές σχέσεις και γεωμετρίες) των τοπωνυμίων προκειμένου να υποστηρίξουμε μια υπηρεσία που θα πληροφορεί επαγγελματίες, όπως τοπογράφουμς μηχανικούς, καθώς και απλούς πολίτες, σχετικά με νομοθεσία που αναφέρεται σε ειδικές γεωγραφικές περιοχές της Ελλάδας.

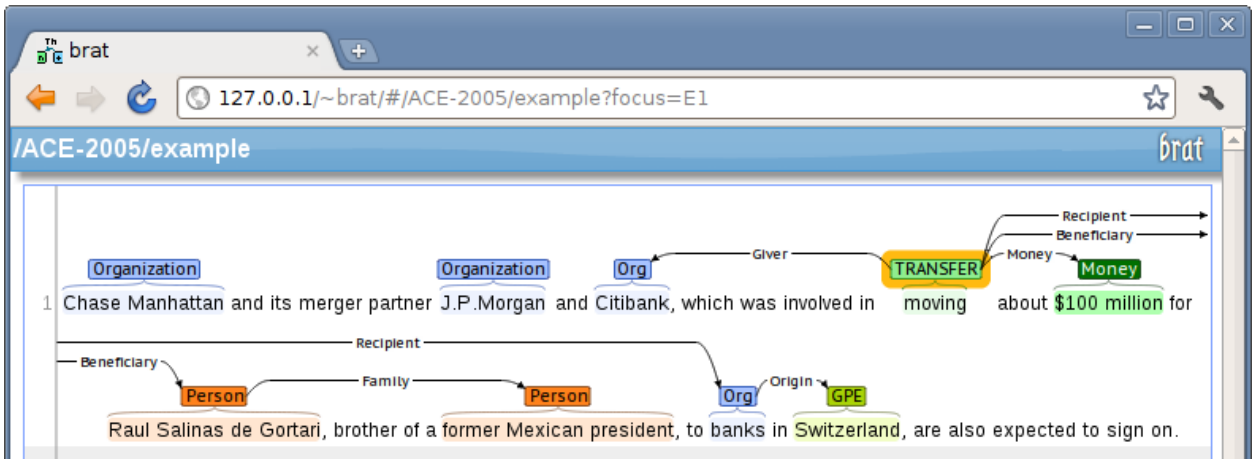
- Βασισμένοι στις παραπάνω διαδικασίες, επαυξάνουμε τη βάση γνώσης και τις δυνατότητες της πλατφόρμας Nomothesi@ με δυο σημαντικούς τρόπους: εντοπισμός νομικών citation δικτύων και αναζήτηση χρησιμοποιώντας κριτήρια βασισμένα σε οντότητες.

Αυτή η δουλειά είναι η πρώτη του είδους της για την Ελληνική γλώσσα σε τόσο εκτεταμένη μορφή και μια από λίγες που αναλύει νομικό κείμενο πλήρως τόσο για την αναγνώρισή όσο και για την σύνδεση οντοτήτων.

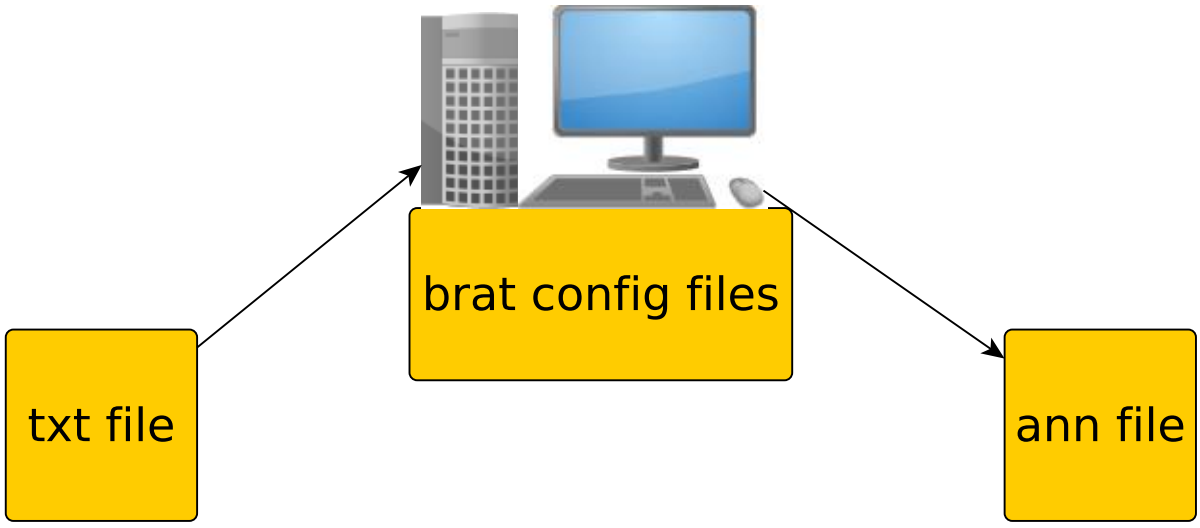
Όταν ετοιμάζουμε datasets για NLP εκπαίδευση, χρειάζεται να παρέχουμε παραδείγματα λέξεων και των τύπων τους, έτσι ώστε να τροφοδοτήσουμε αυτή την πληροφορία στο νευρωνικό δίκτυο. Για να επιτευχθεί αυτό, η κοινότητα έχει αναπτύξει μερικά εργαλεία αφιερωμένα στο έργο του *annotation*. Για τους σκοπούς μας, θα εστιάσουμε στο *brat* (brat rapid annotation tool) [18]. Αρχικά, είχε δημιουργηθεί ως επέκταση του visualizer annotations κειμένου stav[2], ένα εργαλείο visualization annotations που αναπτύχθηκε από τους Pontus Stenetorp, Goran Topić, Sampo Pyysalo και Tomoko Ohta (τότε ήταν μέλη του εργαστηρίου Tsujii του Πανεπιστημίου του Tokyo).

[2] https://github.com/TsujiiLaboratory/stav

Το brat δέχεται ως είσοδο ένα σύνολο txt αρχείων, οπτικοποιώντας σε ένα άρτιο web client. Έπειτα, μπορούμε να ορίσουμε κλάσεις οντοτήτων και πιθανών συσχετίσεων μεταξύ τους ως πιθανές ετικέτες επισημείωσης. Για να προετοιμάσουμε τα datasets, χρειάζεται μόνο να κάνουμε annotate τα tokens που επιθυμούμε να δώσουμε μια συγκεκριμένη ετικέτα. Όλη αυτή η πληροφορία γράφεται σε .ann αρχεία τα οποία περιέχουν γραμμές με πληροφορία όπως ο κωδικός του annotation, το ίδιο το κείμενο, την κλάση και τις θέσεις αρχής-τέλους στο κείμενο.



Σχήμα 1: Οπτικοποίηση στο brat.



Σχήμα 2: Το workflow του brat.

Τα benchmark datasets για τα πειράματά μας περιέχουν 276 ΦΕΚ από τα τεύχη Α και Δ του Εθνικού Τυπογραφείου για τα έτη 2000-2017. Κάθε τεύχος περιέχει πολλαπλά ΦΕΚ. Το τεύχος Α αφορά πρωταρχική νομοθεσία που δημοσιεύεται από την Ελληνική κυβέρνηση (π.χ., νόμοι, προεδρικά διατάγματα, υπουργικές αποφάσεις, εγκύκλιοι, κτλ.). Το τεύ-

T6	PERSON	13807	13832	ΣΤΑΜΑΤΗΣ ΚΑΡΜΑΝΤΖΗΣ
T7	LOCATION-UNK	75	126	θέση «συστάδα 2β» του Συνιδιόκτητου Δάσους Πλατάνης
T11	GPE	667	691	ΑΠΟΚΕΝΤΡΩΜΕΝΗΣ ΔΙΟΙΚΗΣΗΣ
T12	GPE	693	712	ΜΑΚΕΔΟΝΙΑΣ - ΘΡΑΚΗΣ
T14	LEG-REFS	804	815	ν. 998/1979

Σχήμα 3: Παράδειγμα ενός .ann αρχείου που παράγεται από το brat.

χος Δ αφορά αποφάσεις σχετικές με αστικό, αγροτικό και περιβαλλοντικό planning (π.χ. αναδασώσεις,απαλλοτριώσεις, κτλ.).

Μοιράσαμε ομοιόμορφα τα ΦΕΚ σε training (184, 60%), validation (45, 20%) και test (47, 20%) datasets ως προς το τεύχος και έτος χρονιάς. Έτσι, η πιθανότητα overfitting λόγω ειδικών γλωσσολογικών ιδιοσυγκρασιών στην γλώσσα μιας κυβέρνησης ή λόγω ειδικών οντοτήτων και πρακτικών ελαχιστοποιείται. Κάναμε annotate όλα τα παραπάνω έγγραφα για τους 6 τύπους οντοτήτων που εξετάζουμε, χρησιμοποιώντας το *brat*.

Ο βασικός λόγος που τα (BI)LSTM (που είναι μια πιο εξελιγμένη μορφή RNN δικτύων) χρησιμοποιούνται για NLP είναι η ικανότητά τους να χειρίζονται πληροφορία που απαιτεί απομνημόνευση και δομή. Πληθώρα παραδειγμάτων όπως του Andrej Karpathy[3] δείχνουν πολλές τέτοιες εφαρμογές. Μερικά παραδείγματα περιλαμβάνουν εκμάθηση ενός RNN ώστε να μάθει αγγλικές λέξεις και να γράφει από μόνο του τμήματα Shakespeare, συντακτικές δομές από τη Wikipedia, να γράφει LaTeX κώδικα που μεταγλωττίζεται ή ακόμα και να γράφει κώδικα Linux.

Επιπλέον, οι δουλειές [3, 4] έχουν δείξει πως τα BILSTM μοντέλα μπορούν να εφαρμοστούν σε συμβόλαια για την εξαγωγή χρήσιμης πληροφορίας. Προσαρμόζοντας και εξελίσσοντας αυτές τις τεχνικές, αποσκοπούμε να επιτύχουμε εξαγωγή πληροφορίας και οντοτήτων από έγγραφα ελληνικής νομοθεσίας, αναμένοντας αντίστοιχη επιτυχία της διαδικασίας. Εμείς στην παρούσα διπλωματική θα εξετάσουμε τις αρχιτεκτονικές BILSTM-LR, BILSTM-LSTM-LR, BILSTM-BILSTM-LR, BILSTM-CRF.

Ας παραθέσουμε συνοπτικά το workflow της προσέγγισής μας για το Named Entity Recognition τμήμα:

1. Παίρνουμε ένα σύνολο από κείμενα ελληνικής νομοθεσίας σε PDF format, τα μετατρέπουμε σε text και ετοιμάζουμε τα δεδομένα έτσι ώστε κάθε γραμμή να περιέχει μια μόνο πρόταση.

2. Κάνουμε tokenize το κείμενο έτσι ώστε κάθε token να είναι μια λέξη. Σημεία στίξης είναι επίσης tokens (με την εξαίρεση της στίξης που χρησιμοποιείται σε συντμήσεις).

3. Κάνουμε εκπαίδευση `Word2Vec` ή/και `FastText` για να πάρουμε τα word embeddings που είναι απαραίτητα για να διεξάγουμε πειράματα νευρωνικών δικτύων.

4. Κάνουμε χειροκίνητα annotate έγγραφα ελληνικής νομοθεσία του Εθνικού Τυπογραφείου χρησιμοποιώντας το brat[4] [18] προκειμένου να αρχίσουμε supervised εκπαί-

[3] Δείτε http://karpathy.github.io/2015/05/21/rnn-effectiveness/
[4] http://brat.nlplab.org/

δευση.

5. Τροφοδοτούμε τα word embeddings μαζί με τα shapes κάθε token για καθένα από τα 4 προτεινόμενα μοντέλα μας και διεξάγουμε grid search έτσι ώστε να αποφασίσουμε την βέλτιστη τιμή των παραμέτρων.

6. Αξιολογούμε την επίδοση των νευρωνικών δικτύων για όλες τις παραμέτρους που ρυθμίζουμε στο grid search.

Αφού χρειάζεται να δώσουμε τα tokens στο νευρωνικό δίκτυο και για την εκπαίδευση `Word2Vec/FastText`, πρέπει να κάνουμε tokenize το κείμενο που παίρνουμε από το αρχικό PDF format. Είναι πολύ πιθανό να συναντήσουμε στίξη όπως θαυμαστικά, τελείες, κόμματα κτλ., που πρέπει χειροκίνητα να διαχωριστούν από τις λέξεις που βρίσκονται δίπλα τους. Για να το πετύχουμε αυτό, χτίσαμε ένα συστατικό Tokenizer που βασίζεται στο NLTK[5]. Ωστόσο, έπρεπε να χειριστούμε χειροκίνητα ειδικές περιπτώσεις όπως οι παραπάνω καθώς το parsing δέντρο που παρέχεται από τη βιβλιοθήκη κάνει ελαφρώς διαφορετικό χειρισμό της στίξης. Επιπλέον, μετατρέπουμε όλα τα ψηφία που συναντούμε σε “d”, μια μετατροπή που είναι απαραίτητη για εκπαίδευση `Word2Vec`. Τέλος, είναι απαραίτητο να κάνουμε normalize και capitalize όλου του κειμένου, αντιστοιχίζοντας όλες τις Αγγλικές λέξεις στην μοναδική λέξη που κωδικοποιείται ως “ENGLISH_WORD”.

Εφαρμόζουμε το `Word2Vec` (skip-gram model) [19, 20] σε έναν unlabeled σύνολο κειμένων, που περιέχει:

- 150,000 ΦΕΚ του Εθνικού Τυπογραφείου για τα χρόνια 1990-2017.
- όλα τα τμήματα νομοθεσίας από την ίδρυση του Ελληνικού Έθνους από το 1821 μέχρι το 1990, αθροιζόμενα σε περίπου 50,000.
- 1,500 δικαστικές υποθέσεις που έχουν δημοσιευτεί από Ελληνικά Δικαστήρια.
- τις περισσότερες Συνθήκες της Ε.Ε., Εγκυκλίους και Αποφάσεις, που έχουν μεταφραστεί στα Ελληνικά και δημοσιευτεί στο EUR-Lex.
- το Ελληνικό τμήμα των Πρακτικών του Ευρωπαϊκού Κοινοβουλίου.

Παράγουμε word embeddings 100 διαστάσεων για ένα λεξικό 428,963 λέξεων (τύπων), βασισμένο σε 615 εκατομμύρια tokens (λέξεις), που περιλαμβάνονται στο unlabelled σύνολο κειμένων. Χρησιμοποιήσαμε την υλοποίηση `Word2Vec` του Gensim[6], απαιτώντας τουλάχιστον 10 εμφανίσεις ανά λέξη, εκπαίδευση για 20 εποχές και προεπιλεγμένες τις υπόλοιπες παραμέτρους. Λέξεις εκτός λεξικού αντιστοιχήθηκαν σε ένα μοναδικό embedding, το “UNK”.

Η εκπαίδευση του `Word2Vec` μοντέλου έγινε σε υπολογιστή με έναν Intel® Xeon® E5-4603 v2, μια CPU συχνότητας of 2.20GHz, μια 10.24 MB L3 cache, μια μνήμη RAM συνολικής μνήμης 128 GB DDR3 1600 MHz και σε λειτουργικό Linux Debian 8.6 (Jessie) x86 64.

[5] http://www.nltk.org
[6] http://radimrehurek.com/gensim/

Επίσης, πειραματιστήκαμε με δημόσια διαθέσιμα γενικευμένα και προ-εκπαιδευμένα word embeddings 200 διαστάσεων, τα οποία έχουν παραχθεί με το FastText [21][7], έχοντας ως βάση ένα πολύ μεγαλύτερο σύνολο κειμένων με Ελληνικά άρθρα Wikipedia. Όπως θα δείξουμε, τα πειραματικά αποτελέσματα ήταν χειρότερα σε ειδικευμένους τύπους οντοτήτων που εξάγουμε με τα νευρωνικά μας δίκτυα, μάλλον εξαιτίας της έλλειψης εκπροσώπων νομικών εκφράσεων σε γενικό κείμενο (π.χ., άρθρα wikipedia ή νέων).

Χρησιμοποιούμε token shape embeddings [4, 22] που παριστάνουν τα ακόλουθα 7 shapes των tokens:

- token που αποτελείται από αλφαβητικούς κεφαλαίους χαρακτήρες, πιθανώς συμπεριλαμβανομένων και τελειών ή αποστρόφων/καθέτων (π.χ., “ΠΡΟΕΔΡΟΣ”, “Π.Δ.”, “ΠΔ/ΤΟΣ”)
- token που αποτελείται από αλφαβητικούς πεζούς χαρακτήρες, πιθανώς συμπεριλαμβανομένων και τελειών ή αποστρόφων/καθέτων (π.χ., “νόμος”, “ν.”, “υπερ-φόρτωση”)
- token με τουλάχιστον 2 χαρακτήρες, που αποτελείται από αλφαβητική λέξη που ξεκινά με κεφαλαίο και ακολουθείται από πεζούς χαρακτήρες πιθανώς συμπεριλαμβανομένων και τελειών ή αποστρόφων/καθέτων (π.χ., “Δήμος”, “Αναπλ.”)
- token που αποτελείται από ψηφία πιθανώς συμπεριλαμβανομένων και τελειών ή αποστρόφων/καθέτων (π.χ., “2009”, “12,000”, “1.1”)
- αλλαγές γραμμής
- οποιοδήποτε άλλο token που περιέχει μόνο μη αλφαριθμητικούς χαρακτήρες (π.χ., “.”, “€”)
- οποιοδήποτε άλλο token (π.χ., “1ο”, “ΟΙΚ/88/4522”, “EU”)

Γενικά, το shape (μορφή) του token εξαρτάται από την ύπαρξη και σχετική θέση αλφαβητικών χαρακτήρων, ψηφίων και στίξης. Διαισθητικά, αυτή η πληροφορία θα βοηθήσει το νευρωνικό να διεξάγει αναγνώριση οντοτήτων πιο αποτελεσματικά αφού δίνουμε word embedding και shape για κάθε token.

Πειραματιζόμενοι, καταλήξαμε στο να μην ενημερώνουμε τα προ-εκπαιδευμένα word embeddings κατά τη φάση της εκπαίδευσης, ενώ τα shape embeddings των tokens δεν είναι προ-εκπαιδευμένα. Τα αντίστοιχα διανύσματα shape μαθαίνονται στη φάση της εκπαίδευσης. Χρησιμοποιήσαμε Glorot αρχικοποίηση [23], binary cross-entropy απώλεια, και τον Adam optimizer [24] για να εκπαιδεύσουμε τους recognizers, χρησιμοποιώντας early stopping εξετάζοντας το validation loss. Οι υπερ-παράμετροι ρυθμίστηκαν κάνοντας grid-search στα ακόλουθα σύνολα παραμέτρων, επιλέγοντας τις τιμές με το καλύτερο validation loss: hidden units {100, 150}, batch size {16, 24, 32}, dropout rate {0.4, 0.5}.

[7] https://fasttext.cc

Ένα νευρωνικό, ειδικά αν έχει πολλαπλά layers, αποτελείται από εκατομμύρια παραμέτρους και η βελτιστοποίηση όλων είναι πρακτικά αδύνατη. Εστιάζουμε στο ποσοστό dropout (dropout είναι η αφαίρεση ενός ποσοστού των μονάδων του νευρωνικού και επανεκπαίδευσή τους έτσι ώστε όλοι οι νευρώνες να είναι ενεργοί και να μην είναι προκατειλημμένοι) και στο batch size (αριθμός δειγμάτων που διαδίδεται στο νευρωνικό, μεγάλες τιμές υποδηλώνουν γρηγορότερη εκπαίδευση αλλά μικρότερη ακρίβεια συνήθως). Τα νευρωνικά εκπαιδεύτηκαν για 30 εποχές. Η εκπαίδευση έγινε σε υπολογιστή με The neural networks are trained for 30 epochs. Intel® Core™ i5-7600, με CPU συχνότητας 3.50GHz, 6.144 MB L3 cache, συνολική μνήμη RAM 32 GB DDR4 2400 MHz, μια AORUS GeForce® GTX 1080 Ti με 11264 MB μνήμη, 3584 CUDA cores και λειτουργικό Linux Ubuntu Gnome 16.04.3 LTS (Xenial Xerus) x86 64. Τα `Word2Vec` embeddings είναι διανύσματα 100 διαστάσεων. Το νευρωνικό μας χρησιμοποιεί την βιβλιοθήκη της Python Keras 2.1.3[8], με το tensorflow-gpu 1.4.1[9] ως backend.

Για καθεμιά από τις 4 μεθόδους μετρήσαμε την επίδοση της ακριβείας, της ανάκλησης και του F_1 score μετρημένα ανά λέξη. Όπως προτείνεται στο [3], μια αξιολόγηση ανά στοιχείο, εννοώντας ανά οντότητα, μπορεί να παρέχει μια περισσότερο ακριβή εκτίμηση της επίδοσης της κάθε μεθόδου. Ωστόσο, η σύνθετη σύνταξη του νομικού κειμένου και ειδικά η ομαδοποίηση πολλαπλών οντοτήτων σε μεγάλες φράσεις (π.χ., “Οι δήμοι Αθηνών, Δάφνης-Υμηττού και Βάρης-Βούλας-Βουλιαγμένης θα οργανώσουν [...]”) δεν παρέχει μια ξεκάθαρη διαχώριση μεταξύ των επιμέρους οντοτήτων [10] (π.χ., Δήμος Αθηνών, Δήμος Δάφνης-Υμηττού, Δήμος Βάρης-Βούλας-Βουλιαγμένης), έτσι ώστε να μπορούμε να βασιστούμε σε μια τόσο υψηλής τάξης αξιολόγηση. Ο παρακάτω πίνακας δείχνει τα αποτελέσματα αυτής της ομάδας πειραμάτων (οι αριθμοί προέρχονται είναι μέσοι όροι από 5 εκτελέσεις των πειραμάτων):

Πίνακας 1: Ακρίβεια (P), Ανάκληση (R), και F_1 score, *μετρημένων ανά λέξη*. Οι καλύτερες τιμές F_1 για κάθε τύπο οντότητας φαίνονται με bold.

Entity Type	BILSTM-LR			BILSTM-LSTM-LR			BILSTM-CRF			BILSTM-BILSTM-LR		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Person	0.89	0.90	0.89	0.89	0.94	**0.91**	0.88	0.92	0.90	0.89	0.93	**0.91**
Organization	0.77	0.73	0.75	0.77	0.78	0.77	0.72	0.74	0.73	0.78	0.77	**0.78**
GPE	0.80	0.87	0.84	0.83	0.89	0.86	0.81	0.86	0.83	0.84	0.90	**0.87**
GeoLandmark	0.67	0.82	0.73	0.72	0.86	**0.78**	0.64	0.83	0.72	0.70	0.86	0.77
Legislation Ref.	0.85	0.81	0.83	0.87	0.85	**0.86**	0.80	0.79	0.80	0.88	0.85	**0.86**
Public Document	0.81	0.75	0.78	0.85	0.81	0.82	0.72	0.75	0.74	0.84	0.81	**0.83**
Macro AVG	0.82	0.84	0.83	0.84	0.87	**0.86**	0.79	0.84	0.81	0.85	0.87	**0.86**

Τα αποτελέσματα είναι πολύ ανταγωνιστικά για όλες τις μεθόδους που συγκρίνουμε. Τα καλύτερα αποτελέσματα βασίζονται στα macro-averaged F_1 που προέρχονται από το BILSTM-LSTM-LR και το BILSTM-BILSTM-LR (0.86), κάτι που δείχνει ότι η προσθήκη επιπλέον LSTM chains που βαθαίνουν το μοντέλο, επαυξάνουν την αποτελεσματικότητα έστω και λίγο, σε σχέση με τα BILSTM-LR (0.83) και BILSTM-CRF (0.81). Η αναποτελεσματικότητα της state-of-the-art NER μεθόδου BILSTM-CRF, η οποία δοκιμάστηκε με

[8] https://keras.io/

[9] https://www.tensorflow.org/

[10] Αυτό το πρόβλημα ισχύει και για τα IO και BIO annotation schemes, τα οποία έχουν εφαρμοστεί ευρέως σε sequence labelling έργα.

όλους τους δυνατούς συνδυασμούς υπερ-παραμέρτων, είναι εντυπωσιακή. Πιστεύουμε σθεναρά ότι αυτό το θέμα είναι άμεσα συσχετισμένο με την πολυπλοκότητα των αναφορών των γεωγραφικών τοπωνυμίων, των νομικών αναφορών και των αναφορών σε έγγραφα του δημοσίου, ειδικά σε περιπτώσεις με ομαδοποιήσεις αναφορών σε οντότητες.

Πίνακας 2: Ακρίβεια, Ανάκληση και F_1 score για το FastText, *μετρημένων ανά λέξη* με το BILSTM-BILSTM-LR.

Entity Type	Precision	Recall	F_1-score
Person	0.89	0.88	0.88
Organization	0.75	0.70	0.72
GPE	0.85	0.78	0.81
GeoLandmark	0.64	0.76	0.70
Legislation Ref.	0.82	0.82	0.82
Public Document	0.77	0.74	0.76
Macro AVG	0.81	0.81	0.81

Από αυτό το σημείο, θα βασιστούμε στον BILSTM-BILSTM-LR recognizer γιατί είναι πιο αποδοτικός από τον BILSTM-LSTM-LR κατά 1% στην F_1 για οργανισμούς (0.78 vs 0.77), Γεωπολιτικές Οντότητες (0.87 vs 0.86) και έγγραφα του δημοσίου (0.83 vs 0.82), ενώ είναι μόνο 1% χειρότερος σε γεωγραφικά Τοπωνύμια (0.77 vs 0.78). Αν λάβουμε υπόψιν μας τα γενικευμένα και προ-εκπαιδευμένα `FastText` embeddings αντί για τα ειδικά στοχευμένα σε νομικό κείμενο που παράγουμε, λαμβάνουμε ένα macro-averaged F_1 score με τιμή 0.81 για την καλύτερη μας μέθοδο BILSTM-BILSTM-LR, ειδικά στις τελευταίες 4 κατηγορίες οντοτήτων, όπου η γνωστική περιοχή έχει τη μεγαλύτερη σημασία (π.χ., γεωγραφικές πληροφορίες και κωδικοποίηση εγγράφων).

Ας παραθέσουμε συνοπτικά το workflow της προσέγγισής μας για το Named Entity Linking τμήμα:

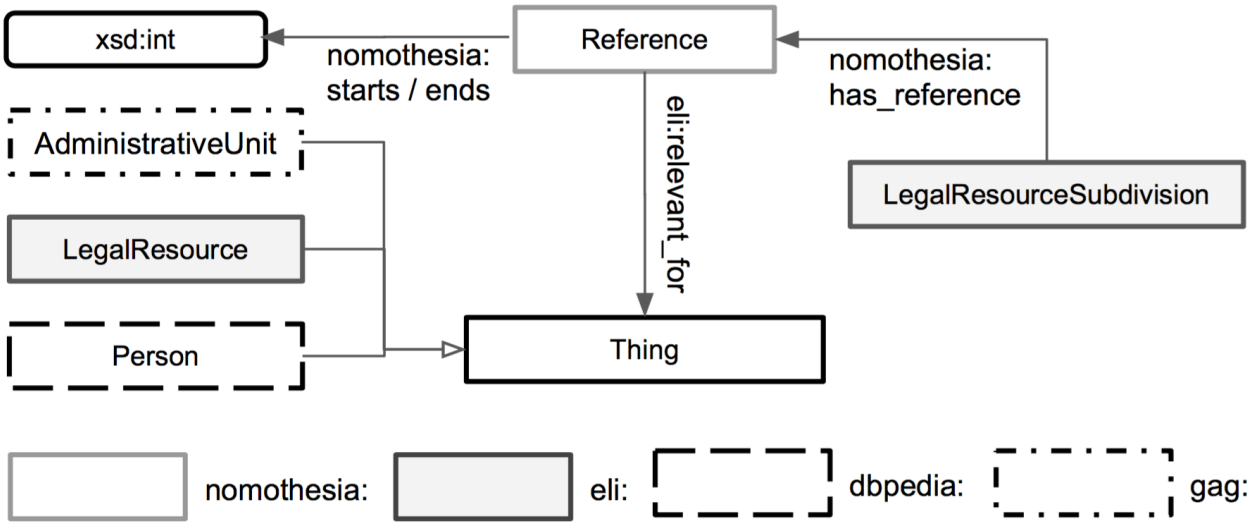
1. Εφαρμόζουμε τεχνικές μετα-επεξεργασίας με hand-written κανόνες και κανονικές εκφράσεις για να κάνουμε normalize και να επεξεργαστούμε τις εξαγόμενες οντότητες με σκοπό την παραγωγή ευπαρουσίαστων labels.

2. Μαζί με τα labels, παράγουμε RDF δεδομένα που αφορούν τις ονομασμένες οντότητες. Χρήσιμες ιδιότητες περιλαμβάνουν το τμήμα του αρχείου που βρέθηκε η αναφορά, την θέση στο κείμενο (για annotation ιστοσελίδας) κτλ.

3. Διασυνδέουμε *Γεωπολιτικές οντότητες*, άτομα και νομικές αναφορές με τα datasets Kallikratis (GAG), Dbpedia persons και ELI, αντίστοιχα, χρησιμοποιώντας το εργαλείο Silk. Επιπλέον, παράγεται ένα ενδιάμεσο dataset αποτελούμενο από `owl:sameAs` τριπλέτες.

4. Παράγουμε χειροκίνητα ένα dataset τοπωνυμίων που συνήθως αναφέρονται σε νομοθεσία σχετική με αστικό, αγροτικό και περιβαλλοντικό planning και, βασιζόμενοι σε ευριστικούς κανόνες και σχετική θέση των οντοτήτων μέσα στο κείμενο, τα διασυνδέουμε με σχέσεις belongs_to στις αντίστοιχες Γεωπολιτικές οντότητες.

Το πρώτο βήμα για την σύνδεση αναφορών (από τον Named Entity Recognizer) σε οντότητες που εξάγουμε με τις οντότητες που περιγράφονται στα δημόσια ανοικτά datasets είναι να αναπαραστήσουμε αυτές τις αναφορές χρησιμοποιώντας το RDF specification. Το *νομικό κείμενο* ενός εγγράφου περιέχει υποδιαιρέσεις (τμήματα επιμέρους νόμων) που ορίζουμε ως *LegalResourceSubdivisions* με βάση την οντολογία της Ελληνικής νομοθεσίας. Αφού κάποιες υποδιαιρέσεις περιέχουν κείμενο, είναι επίσης πιθανό να περιέχουν (*has_reference to*) μια *Αναφορά* σε μια οντότητα (π.χ., ένα τμήμα νόμου που αναφέρεται σε τροποποίηση νόμου). Αυτή η αναφορά υφίσταται σε ένα διάστημα χαρακτήρων. Με άλλα λόγια, *ξεκινά* και *τελειώνει* σε μια συγκεκριμένη ακολουθία χαρακτήρων μέσα στο κείμενο της υποδιαίρεσης. Αυτή η *Αναφορά* πιθανώς αναφέρεται (ή με άλλα λόγια είναι *relevant_for*) σε μια οντότητα, η οποία μάλλον περιγράφεται σε ανοικτά δημόσια datasets. Συνεπώς, ένα *LegalResourceSubdivision* περιέχει αναφορές σε άτομα, διοικητικές μονάδες και νομικούς πόρους (π.χ., νόμους, αποφάσεις κτλ.).



Σχήμα 4: RDF λεξικό νομικών αναφορών σε κείμενο.

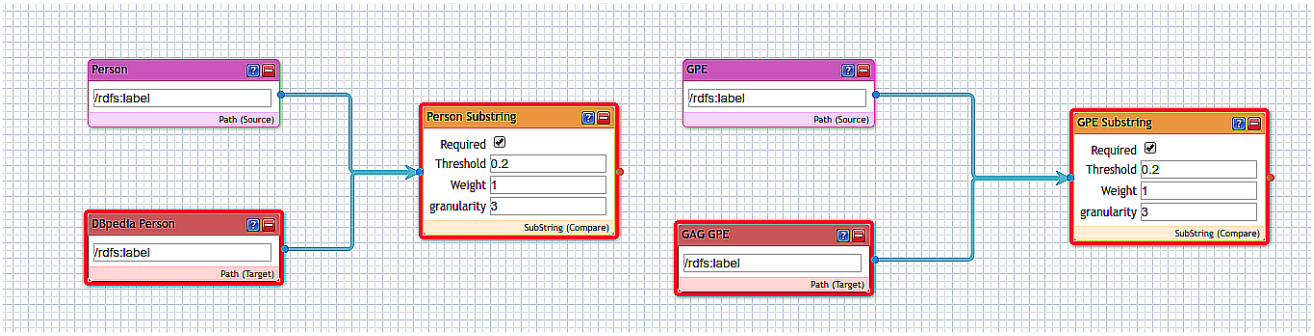
Συνδέσαμε νομικές αναφορές με νομικά έγγραφα που παρέχονται από το dataset Ελληνικής νομοθεσίας[11]. Βασιστήκαμε σε ευριστικούς κανόνες για να ερμηνεύσουμε απευθείας το σχετικό URI εντοπίζοντας τον τύπο, έτος δημοσίευσης και τον σειριακό αριθμό.

Συνδέσαμε αναφορές σε πρόσωπα με Έλληνες πολιτικούς που εξάγουμε από το dataset της Ελληνικής DBpedia[12] και αναφορές σε γεωπολιτικές οντότητες με τις Ελληνικές διοικητικές μονάδες, όπως περιγράφονται στο dataset Greek Administrative Geography (GAG). Διασυνδέουμε και τους δυο τύπους οντοτήτων με τα αντίστοιχα datasets χρησιμοποιώντας το εργαλείο Silk. Πειραματιστήκαμε με δυο διαφορετικές μετρικές σύνδεσης κειμένου: την απόσταση Levenshtein και την απόσταση Substring [25] επί των τιμών `rdfs:label` στο εκάστοτε dataset. Σχετικά με την περίπτωση των Ελληνικών Διοικητικών Μονάδων, δίνουμε επίσης τον τύπο των διοικητικών μονάδων (π.χ., τοπική κοινότητα, δήμος, πε-

[11] Δημοσιευμένο στο http://legislation.di.uoa.gr/legislation.n3.
[12] http://el.dbpedia.org/

ριφέρεια, κτλ.) βασισμένοι στις συμβάσεις ονομάτων που προσδιορίσαμε στο validation τμήμα του labeled dataset.

Για κάθε μέθοδο διασύνδεσης που δοκιμάσαμε, εξετάζουμε την αποδοτικότητα της διασύνδεσης σε όρους *ακριβείας*, *ανάκλησης* και του F_1 score *μετρημένου για κάθε ζευγάρι οντοτήτων* στο test τμήμα του labeled dataset. Εδώ, τα true positives (*TP*) είναι αναφορές που έχουν αντιστοιχηθεί σωστά με μια οντότητα από κάθε σύνολο, τα *false positives* (*FP*) είναι αναφορές που έχουν αντιστοιχηθεί λάθος με οντότητες, και τα *false negatives* (*FN*) είναι αναφορές που λανθασμένα δεν αντιστοιχήθηκαν με κάποια σχετική οντότητα. Ρυθμίσαμε το αποδεκτό φράγμα ανοχής και για τις δύο προσεγγίσεις σύνδεσης στο validation τμήμα των datasets, ενώ τα ζευγάρια οντοτήτων που παρουσιάζονται είναι αυτά που υπάρχουν στο test τμήμα.



Σχήμα 5: Η διαδικασία διασύνδεσης στο Silk για άτομα και GPEs.

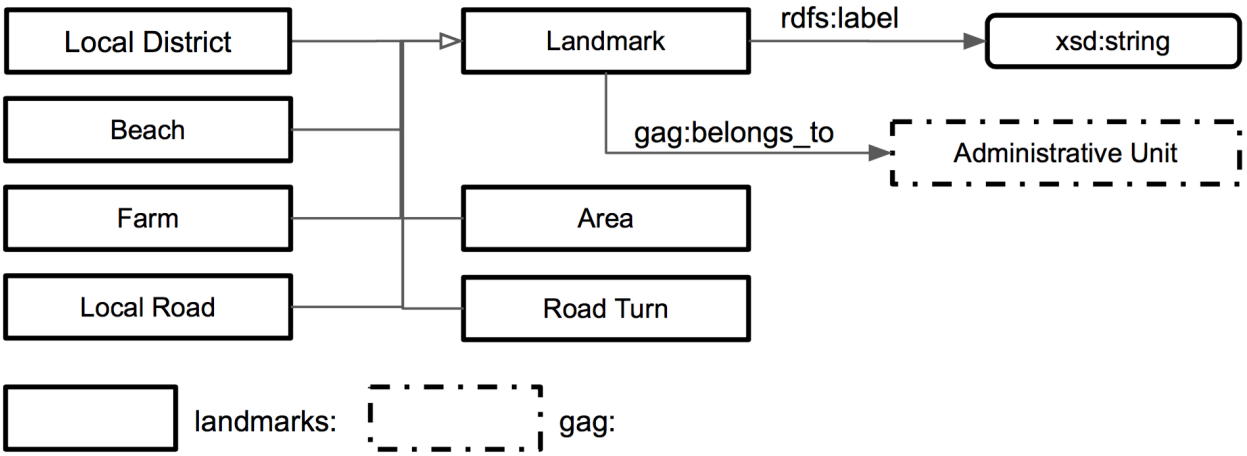
Η σύνδεση ατόμων ήταν μεγάλη πρόκληση για το σύστημά μας, κυρίως γιατί οι νομικοί και το Εθνικό Τυπογραφείο τείνει να αναφέρεται σε πρόσωπα με ακρωνύμιο στο μικρό όνομα (π.χ., Α. Τσίπρας), επομένως ένα μεγάλο σύνολο αναφορών σε άτομα έχει γίνει misclassified (ακρίβεια: 0.71) για άτομα με το ίδιο επώνυμο. Συνδέσαμε επιτυχώς γεωπολιτικές οντότητες με Ελληνικές διοικητικές μονάδες (F_1: 0.92). Συναντήσαμε μικρά θέματα σχετικά με τον διαχωρισμό μαζεμένων αναφορών πολλαπλών διοικητικών μονάδων. Τα αποτελέσματα των νομικών αναφορών είναι άριστα (F_1: 0.98), ενώ ένα πολύ μικρό σύνολο εγγράφων συνδέθηκαν λάθος λόγω του ότι υπουργικές αποφάσεις δεν έχουν μια καθολική κωδικοποίηση (ούτε κάποιο πρότυπο αναφοράς), οπότε διαφέρουν από υπουργείο σε υπουργείο.

Πίνακας 3: Ακρίβεια (P), Ανάκληση (R), και F_1 score, *μετρημένων ανά ζευγάρι οντοτήτων*.

metrics	linking technique								
Entity	rules			levenshtein			Substring		
Type	P	R	F1	P	R	F1	P	R	F1
Person	-	-	-	0.99	0.55	0.71	0.90	0.68	**0.77**
GPE	-	-	-	0.99	0.79	0.88	0.95	0.92	**0.94**
Legislation Ref	0.99	0.97	**0.98**	-	-	-	-	-	-

Τα Ελληνικά γεωγραφικά τοπωνύμια είναι ένα μεγάλο ατού του νομικού recognizer αφού σχετίζονται με planning και αρχιτεκτονικά συμφέροντα. Ωστόσο, δεν υπάρχει τέτοιο dataset

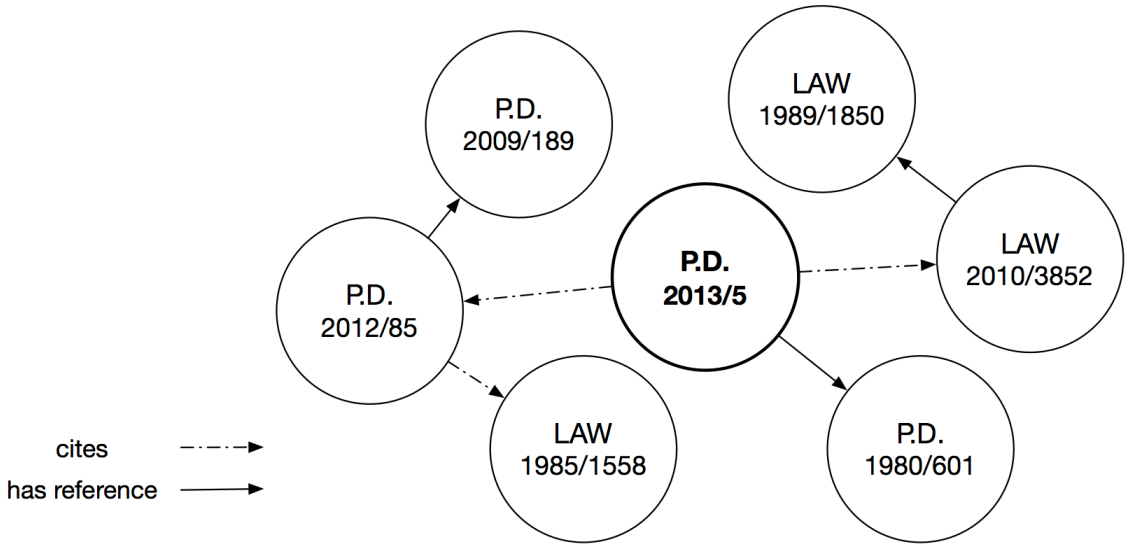
δημόσια για να διασυνδέσουμε τις αναφορές του κειμένου με πραγματικές οντότητες. Επομένως, παράγουμε ένα νέο dataset εφαρμόζοντας γλωσσολογικές ευριστικές για να δημιουργήσουμε ένα σύνολο μοναδικών τοπωνυμίων, παράγοντας την οντολογία που φαίνεται παρακάτω:



Σχήμα 6: RDF λεξικό Γεωγραφικών Τοπωνυμίων.

Έπειτα, διασυνδέουμε το νέο dataset με τις Ελληνικές διοικητικές μονάδες σε περίπτωση που υπάρχει σχέση μεταξύ τους (belongs_to) που προκύπτουν στο κείμενο (π.χ., "Η παραλία Καβούρι στον Δήμο Βάρης-Βούλας-Βουλιαγμένης.").

Επιπλέον δυνατότητα που παρέχει το εργαλείο που αναπτύξαμε είναι η δυνατότητα να μπορεί ένας επαγγελματίας νομικός να εξάγει citation δίκτυα χτισμένα γύρω από ένα νομικό έγγραφο το οποίο πιθανώς περιλαμβάνει αναφορές σε άλλα νομικά έγγραφα:



Σχήμα 7: Δίκτυο citations βασισμένο σε citations και αναφορές γύρω από το Προεδρικό Διάταγμα 2013/5.

Τέλος, παρέχουμε τη δυνατότητα υποβολής ερωτήσεων στον RDF γράφο που προκύπτει. Παρακάτων βλέπουμε δυο ενδεικτικές ερωτήσεις σε φυσική γλώσσα και σε SPARQL:

Πίνακας 4: Ερωτήσεις βασισμένες σε οντότητες.

Q1: *Επέστρεψε πράξεις νομικών εγγράφων που αναφέρονται σε τοπικές κοινότητες, που ανήκουν στην περιφερειακή ενότητα Λάρισσας.*

```
SELECT DISTINCT
?act_id ?local_district_name
WHERE {
 ?act eli:local_id ?act_id.
 ?act leg:has_reference ?reference.
 ?reference eli:relevant_for ?local_district.
 ?local_district rdfs:label ?local_district_name.
 ?local_district a landmark:LocalDistrict.
 ?local_district leg:belongs_to ?regional_unit.
 ?regional_unit rdfs:label
"REGIONAL UNIT OF LARISSA"@en.
} LIMIT 5
```

Act ID	Local District
Dec. 2015/1882	"RIGEOU"
Dec. 2015/1827/109821	"ROUMANI"
Dec. 2015/1629/99573	"LIKOVOUNI ST. CHARALAMPOU"
Dec. 2013/1002/65288	"KLARAKI"
Dec. 2013/1154/74937	"PALIOMANDRIA ST. CHARALAMPOU"

Q2: *Επέστρεψε πράξεις νομικών εγγράφων που περιέχουν αναφορές σε άτομα που γεννήθηκαν στην Αθήνα.*

```
SELECT DISTINCT
?act_id ?person_name
WHERE {
 ?act eli:local_id ?act_id.
 ?act leg:has_reference ?reference.
 ?reference eli:relevant_for ?person.
 ?person rdfs:label ?person_name.
 ?person dbpedia:birthplace ?birthplace.
 ?birthplace rdfs:label "Athens"@en.
} LIMIT 5
```

Act ID	Local District
Dec. 2014/16591/943	"KIRIAKOS K. MITSOTAKIS"
P.D. 2002/73	"KONSTANTINOS STEFANOPOULOS"
Dec. 2011/23564	"LOUKAS PAPADIMOS"
Dec. 2015/Y58	"ALEXIS TSIPRAS"
Dec. 2009/1059423	"GIANNIS PAPATHANASIOU"

Συνολικά, αναπτύξαμε, δοκιμάσαμε και αξιολογίσαμε ένα συστατικό για Named Entity Recognition και ένα για Named Entity Linking, εφαρμοσμένα στην ελληνική νομοθεσία. Η Ελληνική γλώσσα αποτελεί πρόκληση για NLP εργασίες, ενώ ο επιπρόσθετος θόρυβος από εξωτερικές πηγές (αφού το αρχικό σύνολο κειμένων των εγγράφων είναι διαθέσιμο μόνο σε PDF format) μας έδωσε μια ενδιαφέρουσα πρόκληση για να αντιμετωπίσουμε.

Όσον αφορά το NER συστατικό, αξιολογήσαμε όλες τις παραπάνω LSTM μεθόδους στο έργο του Named Entity Recognition σε ένα dataset Ελληνικής νομοθεσίας, το οποίο δημοσιεύσαμε για περαιτέρω ακαδημαϊκή έρευνα. Η διαδικασία είχε μεγάλη διάρκεια και ήταν δύσκολη, καθώς έπρεπε να μετατρέψουμε τα PDF αρχεία σε TXT format, να τα επεξεργαστούμε ώστε να είναι σε μορφή κατάλληλη για εκπαίδευση, να κάνουμε χειροκίνητα annotate ένα υποσύνολο των εγγράφων για να παράγουμε τα test, train και validation τμήματα του dataset μας, προτού μπορέσουμε να διεξάγουμε τα πειράματά μας. Όπως αναφέραμε και παραπάνω, τα πειράματά μας έδειξαν μερικά ενδιαφέροντα και ακόμα και απρόσμενα ευρήματα.

Όσον αφορά το NEL συστατικό, αξιολογήσαμε την σύνδεση οντοτήτων μεταξύ αναφορών κειμένου και οντοτήτων από ανοιχτά datasets από τρίτους. Η απόκτηση συνδέσμων είναι σημαντική καθώς μπορούμε έτσι να συμπληρώσουμε πληροφορίες οντοτήτων που εξάγουμε από το κείμενο με τις αντίστοιχες οντότητες που ταιριάζουν στα γνωστά datasets.

Τέλος, παρουσιάσαμε και εφαρμόσαμε ένα καινούριο λεξικό για την αναπαράσταση αναφορών κειμένου και παράγουμε ένα νέο dataset για Ελληνικά γεωγραφικά τοπωνύμια. Όπως εξηγήσαμε και παραπάνω, αγροτική/αρχιτεκτονική πληροφορία αυτού του είδους δεν έχει εξαχθεί ποτέ σε κάποιο dataset, επομένως είναι μια σημαντική συνεισφορά αφού μας προσφέρει άφθονες δυνατότητες.

Τα μελλοντικά μας σχέδια περιλαμβάνουν περαιτέρω πειραματισμό επί των LSTM μεθόδων χρησιμοποιώντας word embeddings που έχουν εκπαιδευθεί με τον αλγόριθμο FastText, το οποίο λαμβάνει υπόψιν του πληροφορία sub-words. Πιστεύουμε ότι θα ήταν ωφέλιμο με βάση το γεγονός ότι η Ελληνική γλώσσα περιέχει πολλές κλίσεις σε ενικό και πληθυντικό, πτώσεις (ονομαστική, υποτακτική, γενική, προστακτική), και γένη. Για τους ίδιους λόγους, σκοπεύουμε επίσης να αντικαταστήσουμε τα shapes των embeddings με ένα δυναμικό RNN ή CNN επιπέδου χαρακτήρων, για να ενσωματώσουμε πληροφορία σχετική με τα shapes των tokens, προθέματα, καταλήξεις, όπως περιγράφεται από τους Ma και Hovy [26]. Επίσης, ένα RNN ή CNN μοντέλο επιπέδου χαρακτήρα θα εξεταστεί ως εναλακτική μέθοδος (διαδικασία) για τη σύνδεση οντοτήτων.

Μια άλλη ενδιαφέρουσα πιθανή κατεύθυνση μελλοντικής έρευνας είναι η εφαρμογή ενός περισσότερο περίπλοκου annotation format με πλουσιότερα σύνολα από labels, βασισμένου στις αρχές των BIO tags, προκειμένου να λύσουμε το πρόβλημα της συνθετότητας του νομικού κειμένου. Επίσης, επιθυμούμε να εξάγουμε (αναγνωρίσουμε) περισσότερη γεωχωρική πληροφορία όπως συντεταγμένες, που παρουσιάζονται σε πίνακες, ή εξαγωγή συσχετίσεων μεταξύ τοπωνυμίων προκειμένου να επαυξήσουμε την πληροφορία που έχουμε στο νέο dataset που παράγουμε.

LIST OF PUBLICATIONS

[1] P. Liakos, I. Angelidis, and A. Delis, “Cooperative routing and scheduling of an electric vehicle fleet managing dynamic customer requests,” in *On the Move to Meaningful Internet Systems: OTM 2016 Conferences - Confederated International Conferences: CoopIS, C&TC, and ODBASE 2016, Rhodes, Greece, October 24-28, 2016, Proceedings* (C. Debruyne, H. Panetto, R. Meersman, T. S. Dillon, eva Kühn, D. O’Sullivan, and C. A. Ardagna, eds.), vol. 10033 of *Lecture Notes in Computer Science*, pp. 118–135, 2016.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

The present thesis is part of the requirements for the acquisition of a Master's degree in the Department of Informatics and Telecommunications of the National and Kapodistrian University of Athens. The main goal is to provide a way to extract (semi-)automatically entities from Greek legislation, encode using the RDF specification and then linking them to well-known datasets which contain additional information about these entities. For instance, we can link the extracted entity "Municipality of Athens" found in a passage of a legal document containing a presidential decree with the "Municipality of Athens" found in the GAG (Greek administrative geography) dataset[13], gaining additional information such as the coordinates of its geometry, population status etc.

Extracting entities from legal documents and linking them is particularly useful, since this augments the potential of information extraction in Greek legal documents. Since our entire approach is based on neural networks which are oblivious to their input specifics, it can be generalized to any language and type of document with the expectancy of obtaining similar results.

Working on this subject was a most interesting experience as I managed to gain a wealth of knowledge about things I had never heard about, while also expanding my knowledge on various scientific topics. Neural networks are often misunderstood in the sense that it is hard (if not outright impossible) to properly pinpoint the success and failures of different models deployed to tackle specific problems, but on the same token this is what makes this field so interesting. That being said, I will do my best to interpret the behavior and results of the experiments conducted, aiming to be as thorough as possible.

Tackling a problem this challenging motivated me to try and design a component that can be incorporated seamlessly into Nomothesi@[14]. Nomothesi@ is a broader initiative of the University of Athens and platform which makes Greek legislation available on the Web as linked data to aid its sophisticated querying using SPARQL and the development of relevant applications.

[13] http://linkedopendata.gr/dataset
[14] http://legislation.di.uoa.gr

1. INTRODUCTION

Recently, there has been an increased interest in the adaptation of Artificial Intelligence technologies to the legal domain including text processing, knowledge representation and reasoning. Legal text processing [1] is a growing research area, comprising of tasks such as legal question answering [2], legal entity extraction [3, 4] and legal text generation [5]. The same applies to the area of legal knowledge representation, where new standards have been developed and started to be adopted based on semantic web technologies. Relevant contributions here are the European Legislation Identifier (ELI) [6, 7, 8] for legislation, the European Case Law Identifier (ECLI) [9, 10] for case law, as well as Legal Knowledge Interchange Format (LKIF) [11, 12] and LegalRule ML [13, 14] for the codification of advanced legal concepts, such as rules and norms. The research community aims to develop tools and applications to help legal professionals (e.g., judges, lawyers, etc.) as well as ordinary citizens. Based on these principles our group created Nomothesi@[1] [15], a platform which makes Greek legislation available on the Web as linked data to aid its sophisticated querying using SPARQL and the development of relevant applications.

Deepening this effort in order to build a bridge, as a point of reference, between those relative research fields of data science (natural language processing and semantic web), we developed a Named Entity Recognizer (NER) and Linker (NEL) for Greek legislation. For the first task, we will compare and evaluate state-of-the-art neural architectures (RNNs) to recognise the following types of entities: persons, organizations, geopolitical entities, legal references, geographical landmarks and public document references from Greek legislation. We deploy our best entity recognizer on the Greek legislation dataset [15] and produce new entity knowledge encoded in RDF using a novel vocabulary. Given those triples, we use hand-crafted rules and the entity linking framework Silk [16, 17] in order to normalize and link the extracted textual references with entities in public open datasets (Greek administrative units and Greek politicians). We also publish a new RDF dataset for Greek geographical landmarks, that can usually be noted in legislation related to urban, rural and environmental planning. The main contributions are listed below:

- We study the task of named entity extraction in Greek Legislation by applying and evaluating state-of-the-art neural architectures [4], while we also examine a somewhat more complicated one, which outperforms the rest of them even by a short margin.
- We introduce a novel RDF vocabulary for the representation and linking of textual references to entities in Greek legislation. We consider RDF as a single data model for representing both metadata of a legislative document and knowledge that is encoded in the text.
- We deploy Nomothesi@ NER, based on the best model BILSTM-BILSTM-LR in the

[1] http://legislation.di.uoa.gr

Greek legislation dataset and produce new data for entity references, that we describe using the new RDF vocabulary.

- We link the references with datasets using rule-based techniques and the Silk framework.
- We make publicly available a new benchmark dataset of 276 annotated legal documents, which can be reused to train and test different algorithms related to named entity recognition and linking. Pre-trained word embeddings specialized in Greek legal text are also provided.
- We generate a new dataset of Greek geographical landmarks based on the results of Nomothesi@ NER by applying heuristic rules. In a research project that our group has started this dataset will be enhanced further with additional geographical information (e.g., spatial relations and geometries) of the landmarks in order to support a service informing professionals, such as landscape engineers, as well as ordinary citizens about legislation that refers to specific geographical areas of Greece.
- Based on the above procedures, we augment the knowledge base and the querying capabilities of the Nomothesi@ platform in two significant ways: tracing legislation citation networks and searching using entity-based criteria.

This work is the first of its kind for the Greek language in such an extended form and one of the few that examines legal text in a full spectrum for both recognizing and linking entities.

In chapter 2, we provide background information about the problem at hand, related work and the main building blocks that compose our approach.

In chapter 3, we describe the challenge this thesis tackles in more detail.

In chapter 4, we train neural networks and conduct training, testing and interlinking experiments.

In chapter 5, we conduct interlinking experiments and show the potential of indirect inferencing of entity relations by using rule-based techniques.

In chapter 6, we showcase real-life use cases as examples, indicating the value of the component developed in this thesis.

In chapter 7, we summarize what was contributed and the potential future work may yield.

2. BACKGROUND AND RELATED WORK

In this chapter we provide a historical background on neural networks, their origins and their evolution. Further on, we provide the building blocks that we require for our experiments and explain the usage of each component. Additionally, we provide in-depth theory about their functionality.

2.1 Machine Learning and Neural Networks

2.1.1 History of neural networks

2.1.1.1 Local Search and Perceptron

In order to tackle difficult machine learning/classification tasks, numerous algorithms (both approximate and exact ones) have been developed. One of the first and most frequently used for NP-hard problems such as Mobile Facility Location, Bin Packing and Prize-Collecting Steiner Tree is Local Search. Local Search is being used so much as it provides a good measurement to compare with other approaches and it is based on human intuition so it is natural and relatively easy to implement. Also, since it is not fine-tuned to a specific problem (with the exception of mixing it with heuristics and meta-heuristics), it remains generic enough to be useful for virtually any algorithmic problem.

Around 1957, Frank Rosenblatt invented the *Perceptron* algorithm at the Cornell Aeronautical Laboratory. The project was funded by the United States Office of Naval Research and the goal was to develop a generic algorithm for supervised learning[1] of binary classifiers [2]. This is considered to be one of the first neural networks.

2.1.1.2 The Boltzmann machine and AI Winter

While the idea was promising and much of the academia's focus was on neural networks, ameliorating Local Search more and more in the process, the first limitations of neural networks began to become public. In 1969, Marvin Minsky, founder of the MIT AI Lab, and Seymour Papert, director of the lab at the time, published a book [27] where they expressed their skepticism about the potential limitations of Perceptron. Their focus was on the XOR classification problem. Around the same time, a few papers that speculated about the theoretical construct of the Boltzmann machine [3] were published. Further on, due

[1] In plain terms, supervised learning is the process of mapping an input to an output based on example input-output pairs. In human and animal psychology, it is often referred to as concept learning.

[2] A binary classifier is a system that maps inputs to their correct class among two. The concept can be obviously generalized for classifiers of many classes.

[3] Boltzmann machines are a type of stochastic recurrent neural network. They can be seen as the stochastic, generative counterpart of Hopfield nets. They were one of the first neural networks capable of learning

to the neural networks' dependency on the dataset they are applied on and the difficulty of classification, they can be seen as a different way of doing Local Search and therefore they are also subject to the limitations of Local Search, as a result.

When the theoretic construct of the Boltzmann machine was proven to have limitations, it had a profound effect in the direction of research, since a major part of academia began to doubt the neural networks' potential. This is known the be the first of the "AI Winter" periods of history, where all initial enthusiasm about the capabilities of neural networks diminishes and academia focuses on other topics.

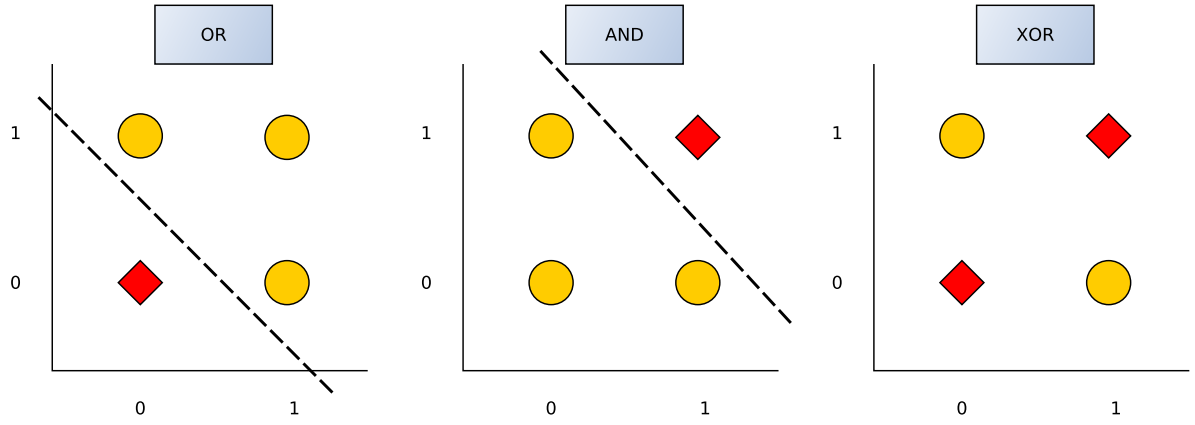


Figure 8: The limitations of Perceptrons. Finding a linear function on the inputs X, Y (resp., a *hyper-plane*) to correctly classify is equivalent to drawing a line (resp., a *hyper-surface*) on the graph separating the classes. It is impossible to separate the classes in XOR's case with a *single* linear function.

2.1.1.3 The XOR problem and multi-layer neural networks

Despite that first "AI Winter" period, the problem of XOR classification also provided invaluable clues that would shape how we view neural networks today. Minsky and Papert's analysis also indicated that it was simply the way *Perceptron* was learning that prevented it from conducting XOR classification. Put simply, *Perceptron* learns to compute some function with the following steps:

1. A set of *Perceptrons*, equal to the number of the function's outputs, start off with small initial weights.
2. For the inputs of a sample in the training set, compute the *Perceptrons*' output.
3. For each *Perceptron* unit, if the output does not match the sample's output, adjust the weights accordingly.

internal representations, and are able to represent and (given sufficient time) solve difficult combinatoric problems.

4. Go to the next sample in the training set and repeat steps 2-4 until the *Perceptrons* no longer make mistakes.

Since the great interest in neural networks lies in combining components in multiple layers to compute something complex, we need to understand why “Perceptron”’s learning process will not work in a multi-layer setting. The samples only specify the correct output for the final output layer, so how can we know how to adjust the weights of Perceptrons in layers before that? This very question troubled academics for a long time, until math provided the solution, the *chain rule*.

2.1.1.4 Stochastic Gradient Descent

Stochastic gradient descent (SGD), also known as incremental gradient descent, is a stochastic approximation of the gradient descent optimization and iterative method for minimizing an objective function that is written as a sum of differentiable functions. Simply put, it tries to find minima or maxima iteratively. When combined with the backpropagation algorithm, it is the de facto standard algorithm for training artificial neural networks. Stochastic gradient descent has been used since at least 1960 for training linear regression models, originally under the name Adaline [28].

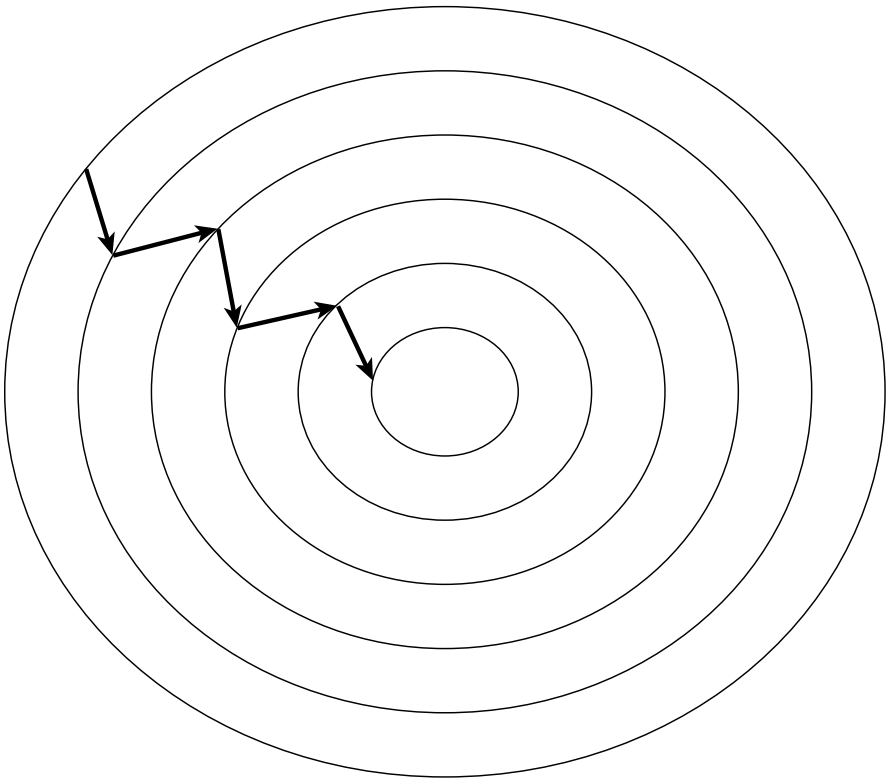


Figure 9: Stochastic Gradient Descent.

2.1.1.5 The Backpropagation algorithm

If we have neurons that do not behave as *Perceptrons*, but are made to compute the output with an activation function that is still non-linear but also differentiable (as with Adaline [28]), not only can the derivative be used to adjust the weight to minimize error, but the chain rule can also be used to compute the derivative for all the neurons in a prior layer and thus the way to adjust the weights is also known. In summary: we can use calculus to assign some of the blame for any training set mistakes in the output layer to each neuron in the previous hidden layer, and then we can further split and cascade blame if there is another hidden layer, and so on; we backpropagate the error. As a result, we can find how much the error changes if we change any weight in the neural net, including those in the hidden layers, and use an optimization technique (for a long time, that would be *stochastic gradient descent*) to find the optimal weights to minimize the error.

The above methodology is the *Backpropagation* algorithm [29, 30]. The interesting fact, however, is that while it was derived by multiple researchers in the early 60's and implemented to run on computers (pretty much as it is today) as early as 1970 by Seppo Linnainmaa [31], Paul Werbos proposed its application on neural networks after a thorough analysis in his PhD thesis [32] in 1982. The reason it took so many years to do so was, predictably, the effects of the "AI Winter". Further publications from David Rumelhart et al. [29] state the ideas of backpropagation so clearly that even modern textbooks on machine learning are virtually identical to their description and further on, they address *Perceptron*'s problems by explaining how multilayer neural nets could be trained to tackle complex learning problems.

2.1.1.6 Convolutional Neural Networks

Academia found value in multilayer neural nets, especially after having mathematically proved [33] that multiple layers allow them to theoretically implement both linear and non-linear functions (XOR as well). This gave rise to numerous applications for *Backpropagation*, such as handwritten zip code recognition by Yann LeCun et al. [34] in 1989.

Such applications began to showcase the need of something more than backpropagation in neural nets. With the input being images, the first hidden layer of the neural net was *convolutional*. This means that, instead of having a different weight for each pixel of the image, only a small set of weights (and therefore a significantly less number of neurons are needed) are being applied to small subsets of the image. The reason this is more promising than a plain neural network is that "local features" found in previous layers rather than pixels are being forwarded and as a result the network sees progressively larger and more complete parts of the image.

In addition, since neurons focus on learning specific local features instead of learning everything over and over (e.g., degree lines in images, small shapes etc) for each pixel, they gain a considerable speedup. Also, since the focus now is entire subsets of the image and not pixels, keeping all values can be redundant and therefore it is possible to gain

even more in speedup if subsampling takes place. Layers that do that are called *pooling* layers and together with the *convolutional* layers they distinguish plain neural nets from *Convolutional Neural Networks (CNNs)*.

2.1.1.7 Unsupervised Learning

When more and more applications that showcased the value of neural networks appeared, it only made sense to find ways to design networks that can train *without supervision*. So far we have discussed how supervised learning works: we provide a network with the input-output pairs and expect the network to approximate the function that maps the inputs to the outputs. Unsupervised learning can be achieved by providing a small hidden layer that outputs the input, in essense forcing the neural network to learn on its own.

Unsupervised learning is particularly important for certain tasks. For example, learned compression can outperform stock compression techniques by finding specific features in the data. One method for learning compression is the utilization of a network called *autoencoder*, which encodes input to a compressed format and then back to itself. Ideally, the final output should match the initial input as much as possible but in this case we achieve that only by using the encoded input of the intermediate layer.

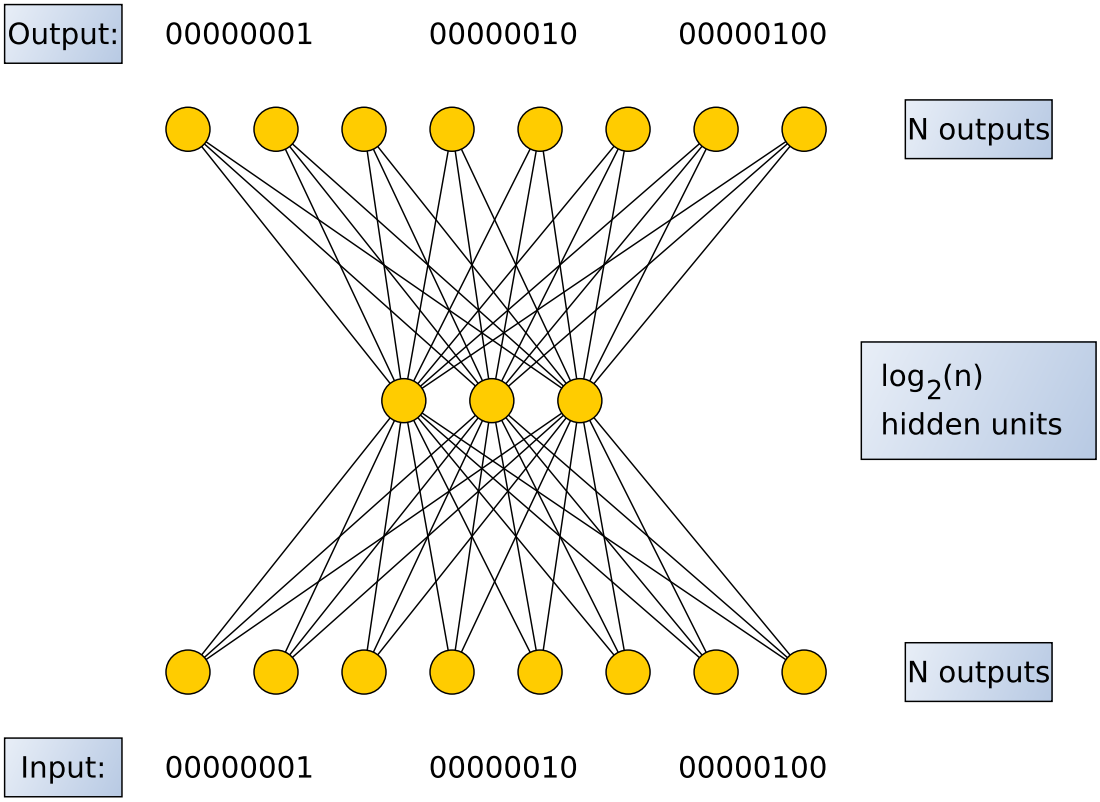


Figure 10: An autoencoder neural net.

Since we are trying to encode the information of the input data into a compressed form, we can exploit the fact in order to conduct clustering tasks (as we merely keep the “essence” of the data instead of that and noise). Apart from that, unsupervised learning has the advantage of requiring just the input samples, not the outputs. One should always be careful, however, as encoding input in a compressed form can cause a network’s efficiency to fall dramatically if the data is complex/random enough.

2.1.1.8 Time-Delay Neural Networks

Despite the great interest of research in proposing applications of neural networks utilizing backpropagation, another challenge arose for neural nets: speech recognition. Due to the nature of speech and possible combinations one can derive a given context or how a word is spoken, the task is quite challenging indeed. Additionally, having to deal with long input sequences does not make things any easier. To make matters worse, since the input is voice samples, lots of noise is added on top of the already long input sequences. Even if we try to separate characters to reduce the input size in a similar fashion to what is done when separating specific characters from text for OCR (Optical Character Recognition), it will be intuitivelly harder to understand context when we split the input.

Another major challenge in speech recognition is being able to recognize when a certain input can effect another following after it. So far, no neurons understand the concept of memory. This means that, in order to adequately tackle the problem of speech recognition, it is imperative to somehow adapt the design of neurons so that they can process input in stream form instead of batch.

To that end, Alexander Waibel et. al [35] introduced a new type of neural network, *Time-Delay Neural Networks (TDNNs)*. While they share many similarities with conventional neutal networks, they differ in the fact that each neuron only processes a subset of the input while also containing a set of weights for different delays of the input. Since the input is a *sequence of audio* and the network accepts only part of it at any given time, we can imagine a “rolling window” moving towards the future as the actual input. Due to the window rolling forward,the same bits of the audio are being processed by each neuron but with different sets of weights depending on their relevant position within the window.

So, since TDNNs seem to be handling parts of the input at any given time, a careful reader might wonder, *how are they different from CNNs then*? The main difference lies in the fact that CNNs do not have the concept of time at all and that, while the rolling window of input is always moved across the entire input image to compute a result in CNNs, in TDNNs the input and output of data is sequential.

2.1.1.9 Recurrent Neural Networks

As academia experimented more with TDNNs, a new model managed to surpass it, the *Recurrent Neural Networks (RNNs)*. So far we have seen *feedforward* networks, which

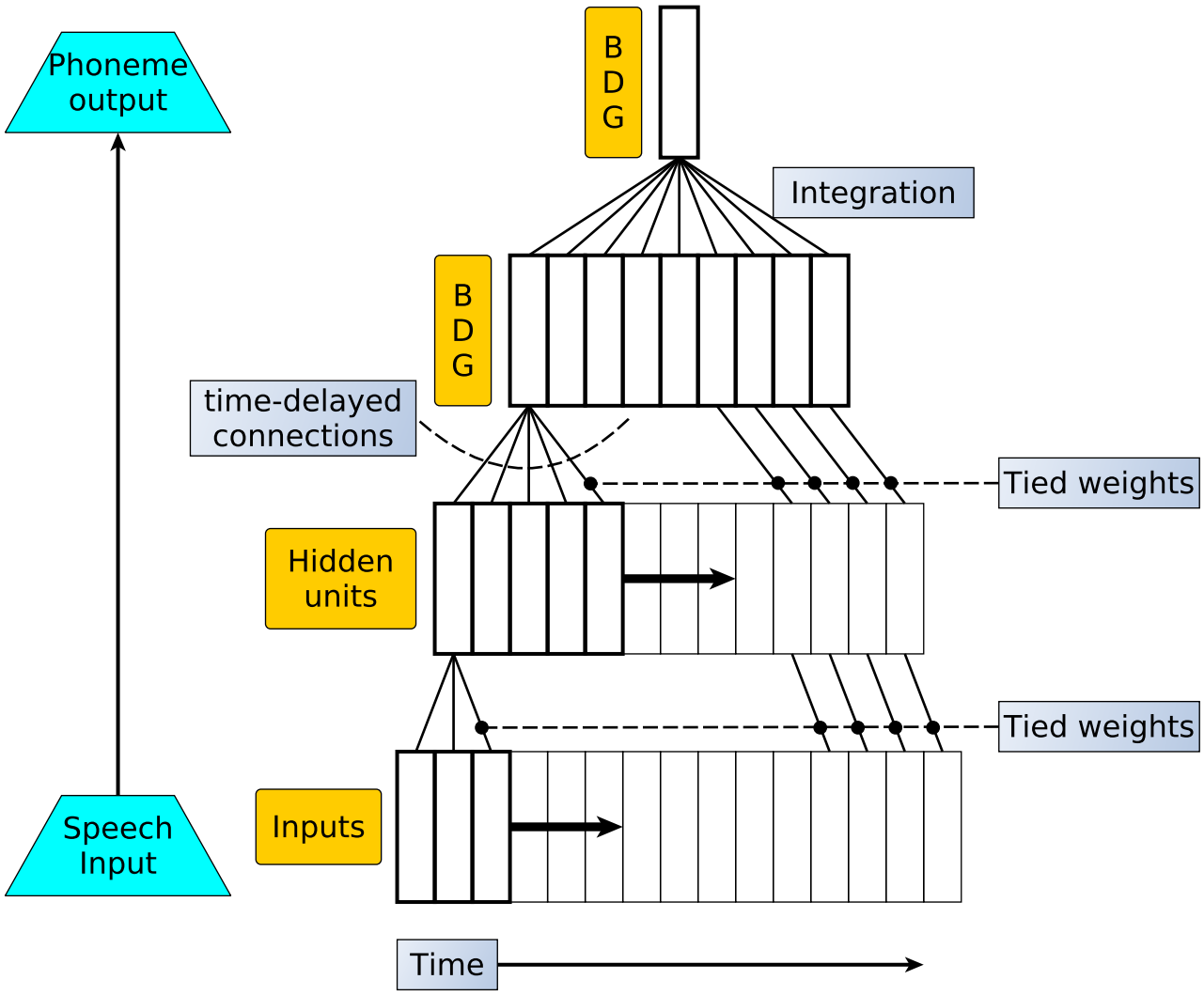


Figure 11: A time delay neural network (TDNN).

means that outputs act as inputs only in forward layers. The innovation of RNNs lies in the fact that we remove that restriction; it is acceptable to have outputs used as inputs to both previous layers as well as to the neurons they came from (looped inputs). Interestingly enough, this also solves the problem of maintaining memory in neural networks based on past inputs.

Despite solving two problems with one major innovation, there is now a problem in our math. Since we now allow loops and outputs of later layers being inputs of previous ones, how can backpropagation and the chain rule function when it's possible to have infinite loops? The error would be cascaded infinitely throughout the network via loops like that. That issue is also resolved by *backpropagating through time* [36]. This means that we unravel the loops and treat the result as a normal network, but we do the unravelling a limited number of times.

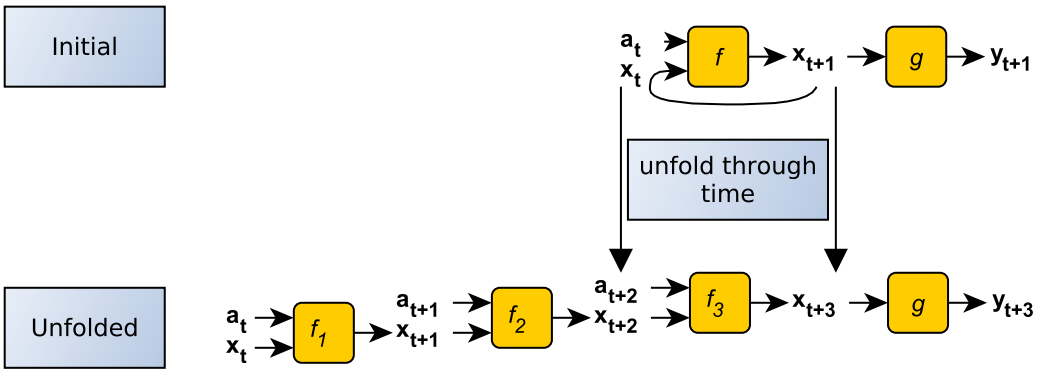


Figure 12: Backpropagation through time.

2.1.1.10 Long Short-Term Memory Neural Networks

Having solved all issues of RNNs, all that is left is to see them in action. Unfortunately, papers from major figures such as Yoshua Bengio et al. [37, 38] in 1993 revealed the new challenge for neural networks. Training RNNs seems to be particularly tough because their parameters settle in a suboptimal solution which takes into account *short term dependencies* but not *long term ones*.

The true problem lies in backpropagation which initially solved so many issues in neural nets and the reason is backpropagation splits the blame of the error for previous layers. Taking into account the fact that we now have loops and outputs used as inputs in previous layers, we can observe that backpropagation will yield either tiny or huge numbers. This is called the *vanishing or exploding gradient problem* as explained by Jurgen Schmidhuber.

Predictably, the solution came once more in the form of a new kind of neural network; the *Long Short-Term Memory (LSTM)* network [39]. The main idea is to have some units called *Constant Error Carousels (CECs)* which have the identity function as an ativator with a weight of 1.0. This trick ensures that errors backpropagated through CECs will not vanish or explode (with the exception of flowing out of a CEC component into other, adaptive parts of the network). CECs are then connected to non-linear adaptive units (multiplicative activation functions), a necessary inclusion for learning non-linear behavior. Error signals backpropagated in time through CECs enhance the weight changes of these units. As a result, LSTM networks are now capable of memorizing and discovering the importance events that happened a long time in the past.

Each LSTM unit consists of a self-connected memory cell and three multiplication units. These are the input, output and forget gates that represent writing, reading and resetting operations taking place in each cell for each timestep (word/token).

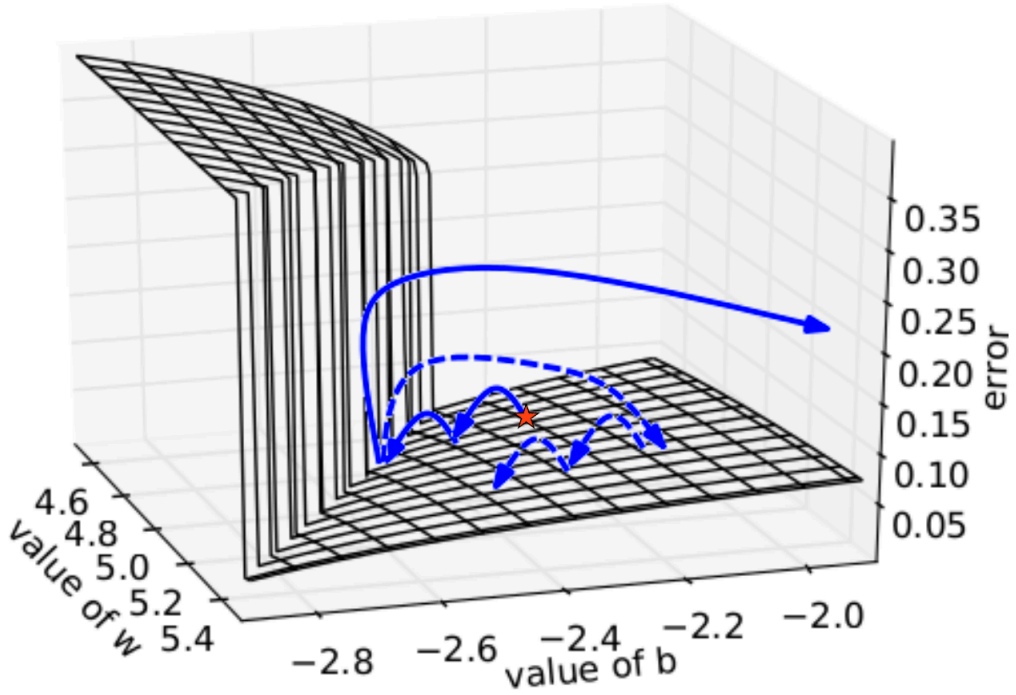


Figure 13: Pascanu, Razvan, Tomas Mikolov, and Yoshua Bengio. "On the difficulty of training recurrent neural networks."

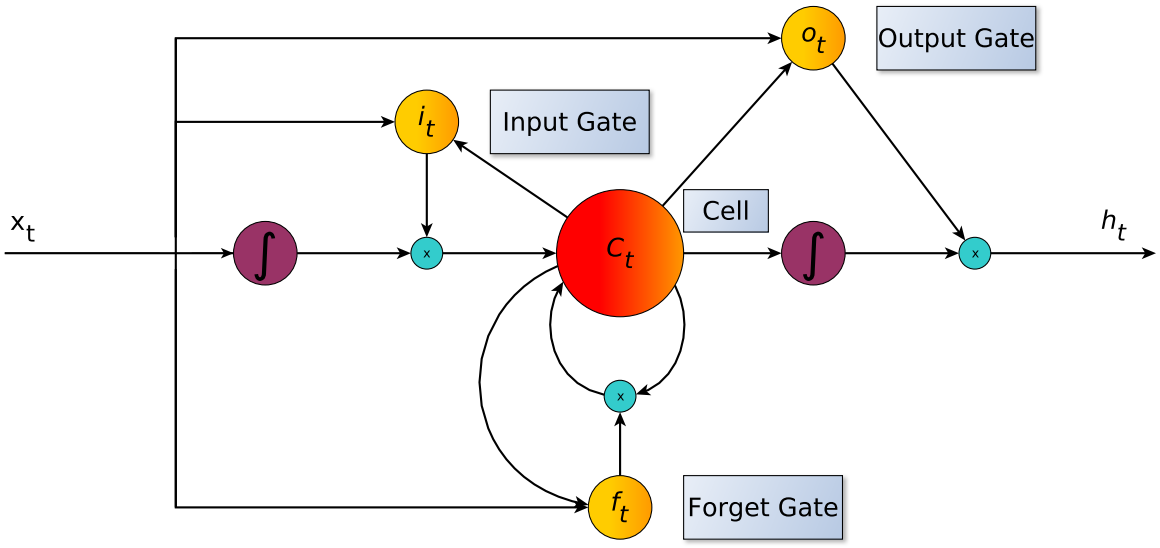


Figure 14: A Long Short-Term Memory cell.

2.1.1.11 BILSTM

When it is necessary to consider both the previous and next timesteps (words/tokens) in each single step (word/token), bidirectional LSTM chains can be used instead of unidirectional ones. Although more computationally expensive, taking into consideration both the past and the future can help capture more complex meanings in sequences.

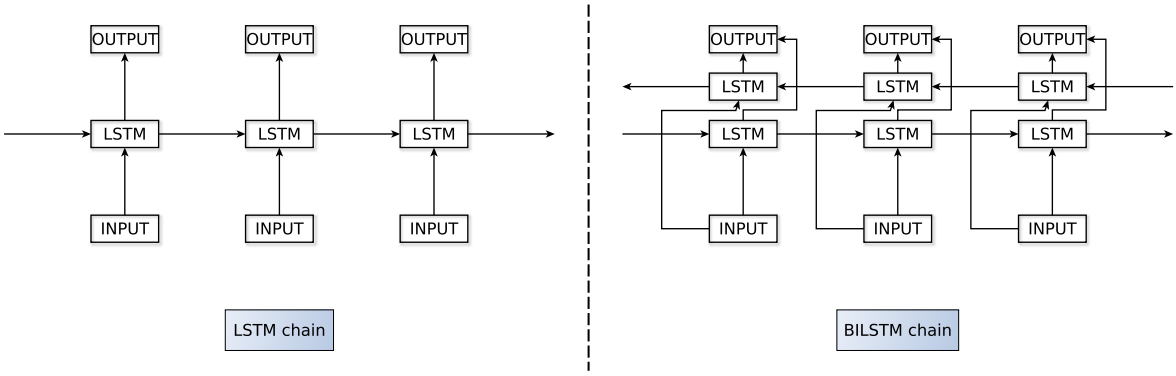


Figure 15: LSTM and BILSTM chains comparison.

2.1.1.12 Second AI Winter and Support Vector Machines

Despite managing to overcome all these difficulties, the trend around the mid 90's dictated that neural networks required too much computational power to be useful, or they took too long to produce results. As a result, another AI Winter era began.

Furthermore, a major hit came in the form of a method called *Support Vector Machines (SVMs)*. Simply put, it is a mathematically optimal way to train a two-layer neural network. This simplicity and the frustration of the inflexibility of sophisticated neural network models encouraged the use of SVMs. LeCun et al. [40] explain how SVMs are highly competitive against neural nets. In addition, even more competitive methods with strong mathematical ideas as their background began to appear such as *Random Forests*.

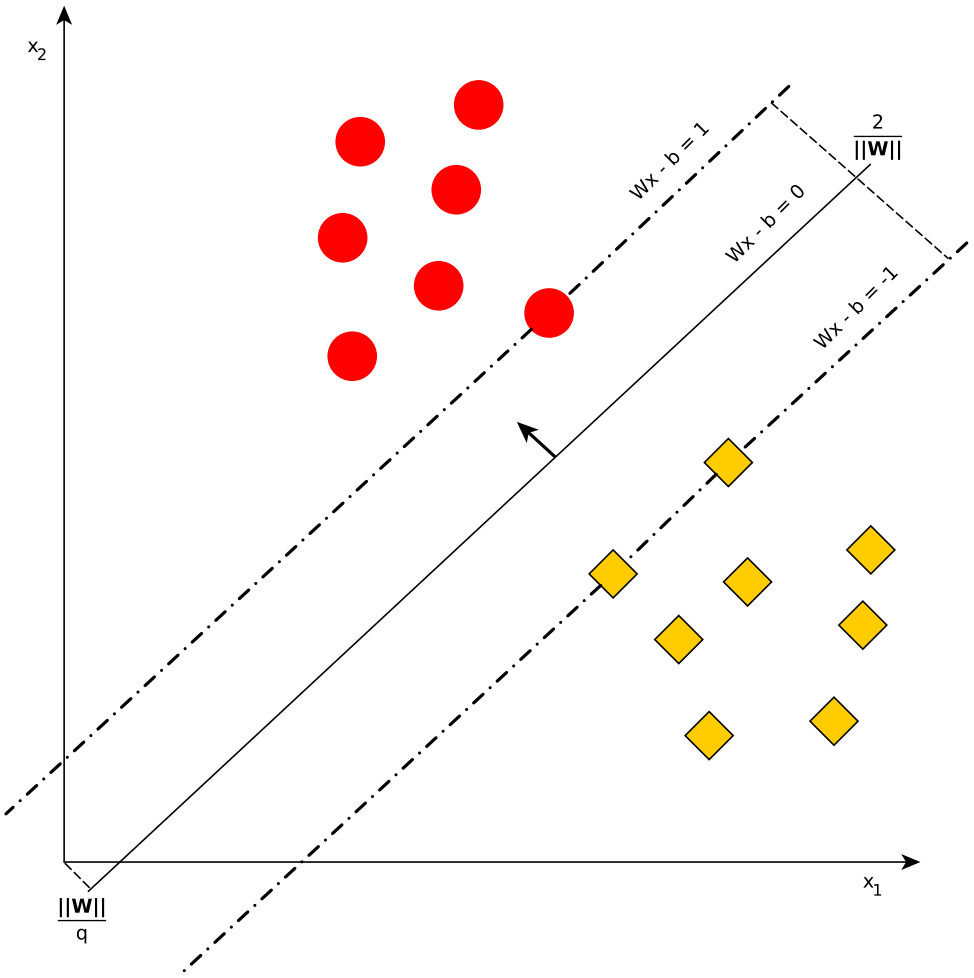


Figure 16: Maximum-margin hyper-plane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors.

2.1.1.13 The comeback of Deep Learning

For a few years, research on neural networks took a major hit, until Hinton et al. published a paper [41] that proposes yet another novel idea; if weights of the neural networks are cleverly initialized instead of randomly, they can be trained well. Clever initialization is possible if we separate the layers, train them independently using unsupervised learning and then conducting one round of supervised learning. These separate layers (when they do not have connections between hidden and visible units) are called *Random Boltzmann Machines (RBMs)* and methods like this that combine supervised and unsupervised learning are classified as semi-supervised learning methods.

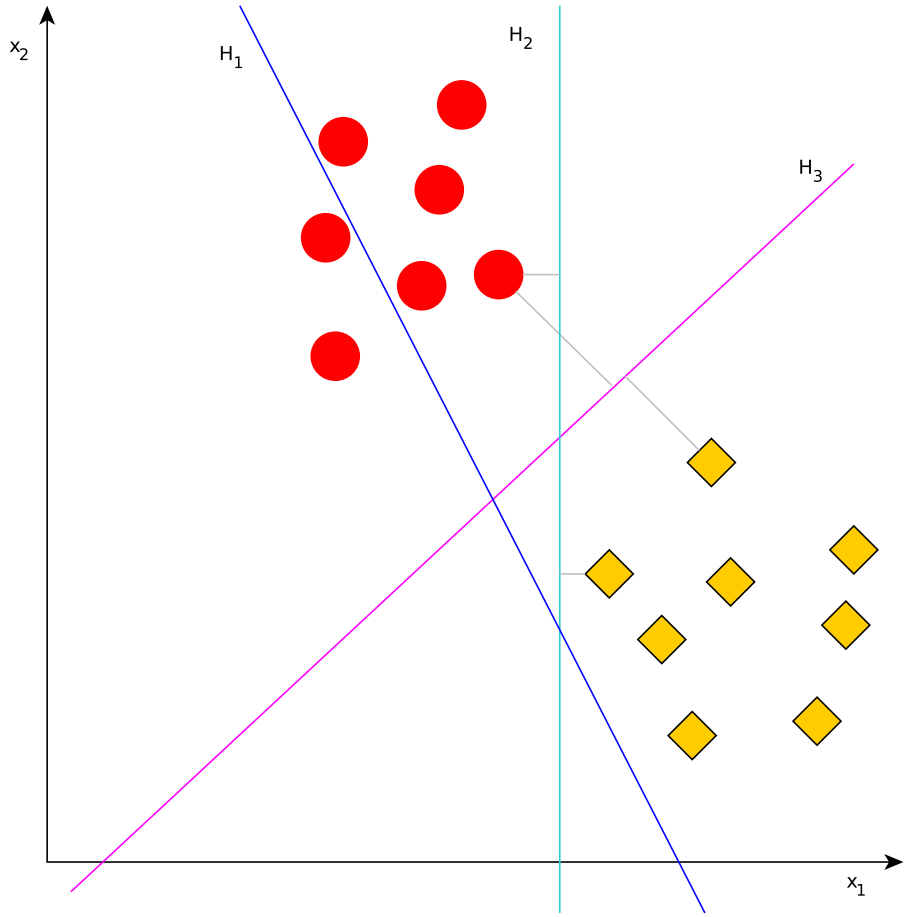


Figure 17: SVM optimization. H1 does not separate the classes. H2 does, but only with a small margin. H3 separates them with the maximum margin.

It was shown by Hinton [42] that this form of Boltzmann Machine can be efficiently trained. The reason for this is that maximization focuses on something other than the probability of the units generating the training data and therefore we get an approximation that works rather well in practice.

Further research attempts which focused on improving neural networks were rather impressive but the computational power required to achieve better results still made people skeptical. This trend began to shift when works like the one by Hinton et al. [43] indicated that neural networks are now ready to tackle very challenging AI tasks such as speech recognition and even doing that while breaking performance records that took decades to surpass. Further on, Raina et al. [44] and others show that with the introduction of massive parallelization via the utilization of GPU power can achieve a speedup of at least 70 times for neural networks training.

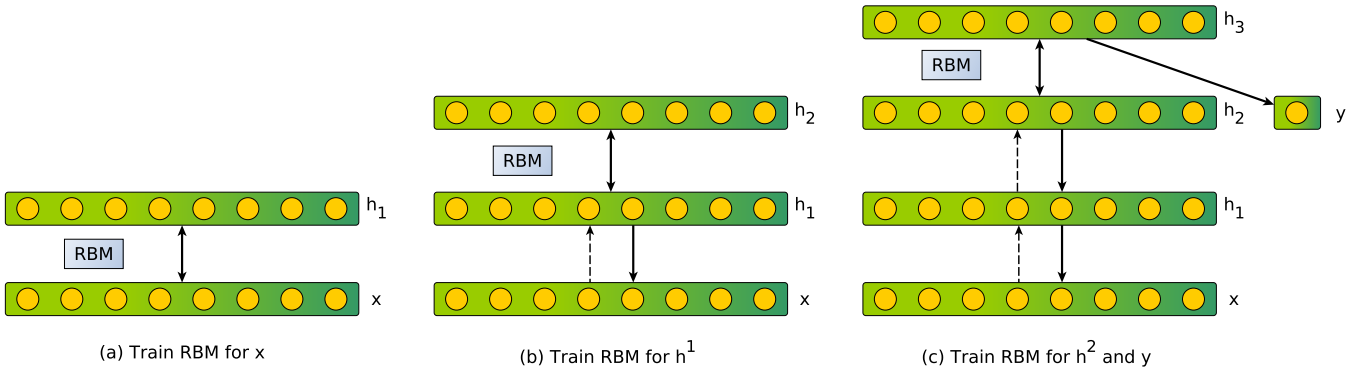


Figure 18: Layerwise pre-training for RBM.

2.1.1.14 Activation functions and Dropout

At this point, researchers began to wonder why the old approaches did not work while the new ones achieved significant results. Xavier Glorot and Yoshua Bengio [23] discuss two major findings regarding that:

- The non-linear activation function chosen for the neurons of a neural net is very important for performance.
- The vanishing gradient problem occurs because backpropagation involves a series of multiplications which result in smaller derivatives for earlier layers. So, while choosing random weights in general might not be very problematic, choosing random weights without taking into consideration the layer the weights are for, is.

LeCun et al. [45], Hinton et al. [46] and Bengio et al. [47] separately tried to compare different activation functions to find which one is the best one and all three groups came to the same, surprising, conclusion. The non-differentiable and very simple function $f(x) = max(0, x)$, which is also called *Rectified Linear Unit (ReLU)*, tends to be the best. While a non-differentiable function being the best in a task that requires differentiazation is surprising in itself, the greatest question is why such a simple function can work so well. The mathematical probability of having to deal with values at zero is negligible so in practice the former is not that problematic. However, the latter is not precisely answered, but academia has a few well-established ideas:

- The simplicity of the function and its derivatives make it computationally cheap and therefore essential in order to scale neural networks for Big Data.
- Andrew Ng et al. [48] have provided an analysis where it is explained that ReLU's form can actually help to tackle the issue of gradient vanishing, while also providing distributed representations, avoiding localization.
- ReLU produces sparse representations, meaning that few neurons will output something other than zero. This means that information representation is robust, but we

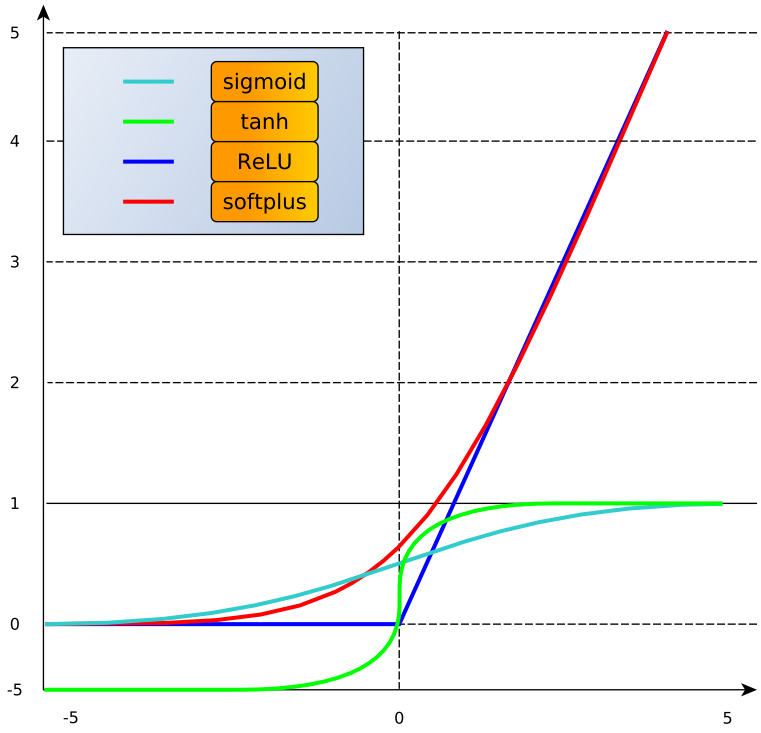


Figure 19: Activation functions.

gain the additional advantage of being able to gain significant computational speed (since most outputs are zero they can just be ignored and other computations can take their place).

With Deep Learning being popular again, more research focused on improving neural nets and tackling potential problems that take place during training. A very significant one is *overfitting*. Overfitting occurs when a neural network learns a bit “too well” the training dataset so it is fine-tuned to that. This is bad as it means that when we try to use the trained network on anything else, the results will leave much to be desired. Kingma et al. [49] proposed an idea to tackle this problem, *Dropout*. The idea is to simply pretend at random that some neurons are not present during training. This means that we utilize a more powerful form of learning, since we learn things about the training data in different ways each time, without focusing too much on specific features.

2.1.2 Linear Classifiers

2.1.2.1 Logistic Regression

Logistic regression (LR) is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured

with a binary variable (meaning there can only exist two possible outcomes).

The goal of logistic regression is to find the best fitting but biologically reasonable model to describe the relationship between the binary characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables. Logistic regression generates the coefficients, its standard errors as well as the significance levels of a formula to predict a *logit transformation*[4] of the probability of presence of the characteristic of interest:

$$logit(p) = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + \ldots + b_kX_k$$

2.1.2.2 Conditional Random Fields

Conditional Random Fields (CRFs) are a type of discriminative undirected probabilistic graphical model. They are used to encode known relationships between observations and construct consistent interpretations. CRFs [50] have been widely used in traditional NLP sequence labeling tasks (e.g., pos tagging, named entity recognition). They have also shown promising results if applied on top of LSTM, BILSTM in sequence labeling [51, 52, 53, 54, 26] and parsing [55]. In our case, the CRF layer jointly selects the assignment of positive or negative labels to the entire token sequence, which allows taking into account the predicttions of neighboring tokens. For example, if both the previous and the next token of the current token are classified as parts of a legislation reference, this may be an indication that the current token is also part of the same legislation reference.

2.1.3 Feature representation for NLP tasks

A major task to perform in Information Representation (IR) and NLP is to find a way to properly represent words/tokens in such a way that we capture as much information as possible. To this end, shallow neural networks which are pre-trained using unsupervized algorithms [19, 20, 21] on large corpora are usually employed. In the past, sparse feature representations were employed but they proved ineffective. Here, will explain the fundamentals of how they work and the intuition behind their success.

2.1.3.1 One-hot vectors

One of the first representations for tokens is a *one-hot vector*. A one-hot vector is a $1 \times N$ matrix (vector) used to distinguish each word in a vocabulary from every other word in the vocabulary. The vector consists of zeros in all cells with the exception of a single 1 in a cell

[4] The *logit function* is the inverse of the sigmoidal "logistic" function or logistic transform used in mathematics, especially in statistics. When the function's variable represents a probability p, the logit function gives the logarithm of the odds as $ln\left(\frac{p}{1-p}\right)$.

used uniquely to identify the word. It is intuitively easy to see why this is very problematic; having a vocabulary of a million words would mean we would need a million dimensions to map each word. Even worse, not only are we using so much memory to utilize this kind of representation, but we also fail to get any useful context information about the actual words' relations to each other, potential semantic similarities etc. Finally, exactly because each word vector has a 1 on the dimension it differs from all other words in the vocabulary, the cosine similarity[5] would always be zero.

2.1.3.2 TFxIDF

Term Frequency times Inverse Document Frequency (*TFxIDF*), is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The TFxIDF value increases proportionally to the number of times a word appears in the document and is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general.

2.1.3.3 Pointwise mutual information

Pointwise mutual information (*PMI*) [56], or point mutual information, is a measure of association used in information theory and statistics. Given two discrete random variables X and Y and assuming independence of the variables, PMI quantifies the discrepancy between the probability of their coincidence given their joint distribution and their individual distributions for a pair of outcomes x and y belonging to X and Y, respectively. Mathematically, it can be seen as:

$$pmi(x;y) \equiv log\frac{p(x,y)}{p(x)p(y)} = log\frac{p(x|y)}{p(x)} = log\frac{p(y|x)}{p(y)}$$

2.1.3.4 Word Embeddings

The more modern approaches try to address the issues of all these cases by training shallow neural networks carefully in order to properly map words into meaningful vectors that will be fast, efficient, requiring little memory and capturing essential context between words. Here, we will discuss `Word2Vec`, `FastText` and `GloVe`.

2.1.3.4.1 Word2Vec A popular algorithm by Google [19], named `Word2Vec`, was invented. Is utilizes a shallow two-layer architecture (an input layer and a hidden layer)

[5] $similarity = cos(\theta) = \frac{A \cdot B}{||A||||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$

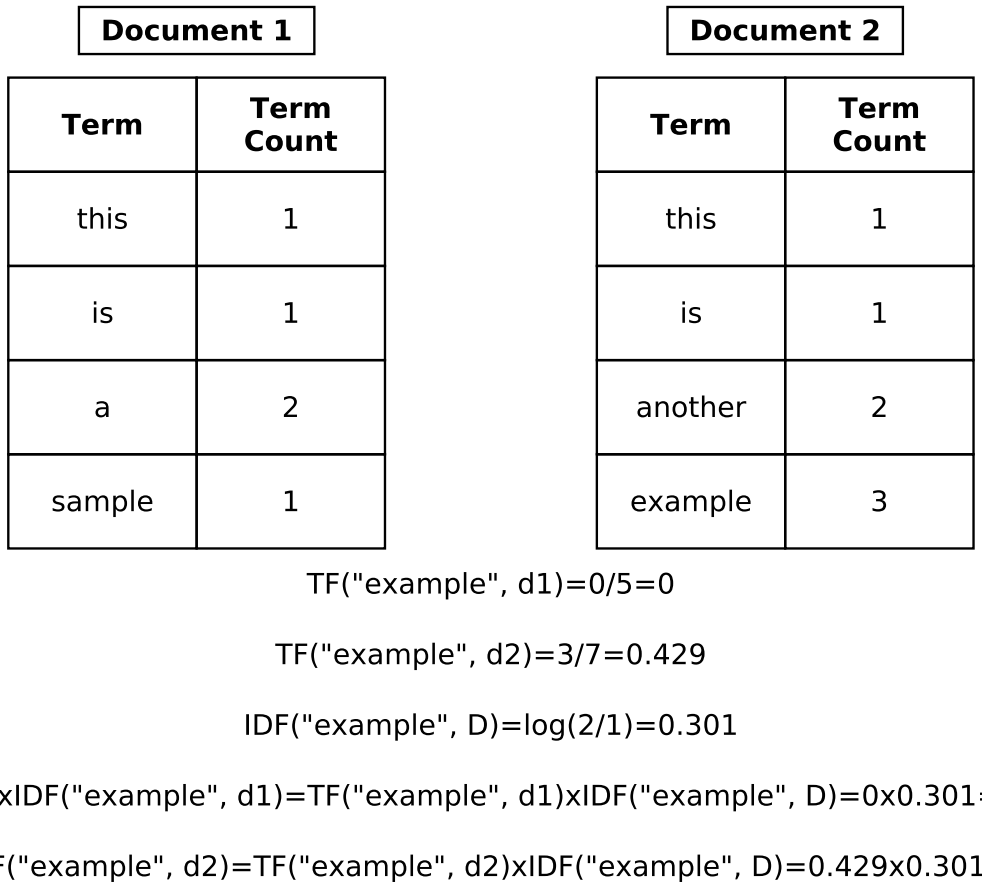


Figure 20: TFxIDF example.

to produce word embeddings. It is an efficient way to produce a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Numerous implementations of the algorithm exist currently, most notably gensim's implementation in Python [6].

When representing tokens, it is important to map them into vectors of many dimensions. Ideally, we would want as many dimensions as all possible features that distinguish them, but that is obviously impossible due to hardware constraints. The best we can do is randomly select a representative number of dimensions and hope for the best. So, in order to minimize the chance of choosing all the less useful dimensions, we need a way to properly and generally capture the context and "distance" of words, especially since we can observe that, frequently, similar words tend to co-occur in similar contexts (phrases). Properly mapped vectors, for instance, would encode $||\text{“}man\text{”} - \text{“}woman\text{”}|| = ||\text{“}king\text{”} - \text{“}queen\text{”}||$.

Since `Word2Vec` is based on a shallow two-layer model, there are a few options available, the most well-known of which being CBOW and Skip-gram.

[6] See https://radimrehurek.com/gensim/models/word2vec.html.

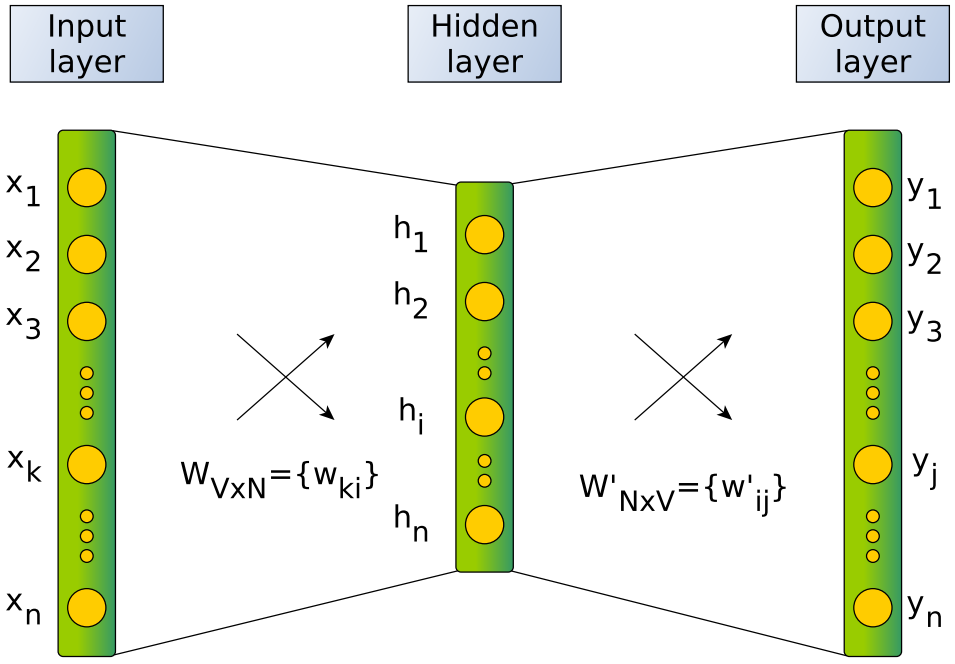


Figure 21: Generic Word2Vec architecture.

2.1.3.4.1.1 Continuous bag of words and Skip-gram In order to achieve this, we can use two shallow two-layer neural network models: the Continuous Bag of Words (CBOW) and the Skip-gram:

- **CBOW**. The input to the model could be $w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}$, the preceding and following words of the current word we are at. The output of the neural network will be w_i. Hence you can think of the task as "predicting the word given its context".

  Example: "The cat ate ____." (food).

- **Skip-gram**. The input to the model is w_i , and the output could be $w_{i-1}, w_{i-2}, w_{i+1}, w_{i+2}$. So the task here is "predicting the context given a word". In addition, more distant words are given less weight by randomly sampling them.

  Example: "___ ___ ___ food." (the cat ate).

Therefore, word embeddings capture both semantic and syntactic information as well as correlations between words in both models.

At this point, a careful reader might wonder, *which of the two models could yield better results or run faster*? In general, it is not easy to answer that, however it is believed that CBOW is faster while skip-gram is slower but does a better job for infrequent words. Knowing we are dealing with the greek language and an incosistent input, in this thesis we focus on **skip-gram**.

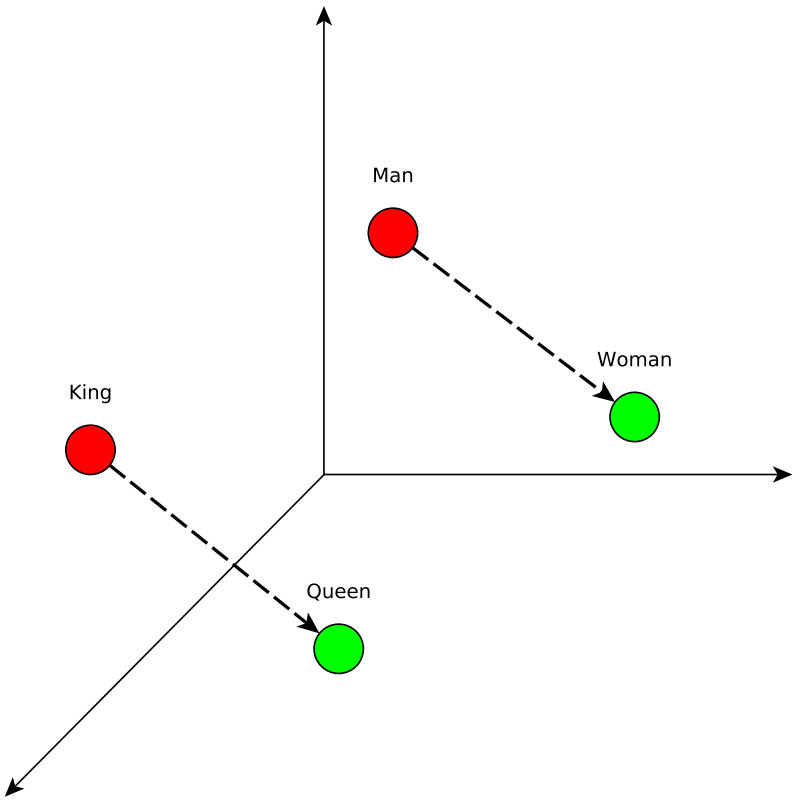


Figure 22: Vector representations of words.

2.1.3.4.2 FastText Another attempt to achieve the same result came from `FastText`, a library for learning of word embeddings and sentence classification created by Facebook's AI Research (FAIR) lab [21, 57]. The model is an unsupervised learning algorithm for obtaining vector representations for words. Facebook makes available pretrained models for 294 languages, including greek. The intuition behind the success of `FastText` is that it is based on a huge corpora of training data and also from multiple languages. As we have already established, the more training data a neural network is given, the more chances it has to be highly accurate.

`FastText` is considered superior to `Word2Vec`, because it solves a few major issues `Word2Vec` has. For instance, `Word2Vec` can only map words into vectors for words that are known to it. New/unknown words cannot be represented so that we can extract useful information (such as how close an unknown word is to the existing ones vector-wise). `FastText` is structured in such a way that it captures sub-word information by analyzing n-grams relations. As a result, it can even produce a sensible word embedding even for unknown words based on proximity to the existing vocabulary.

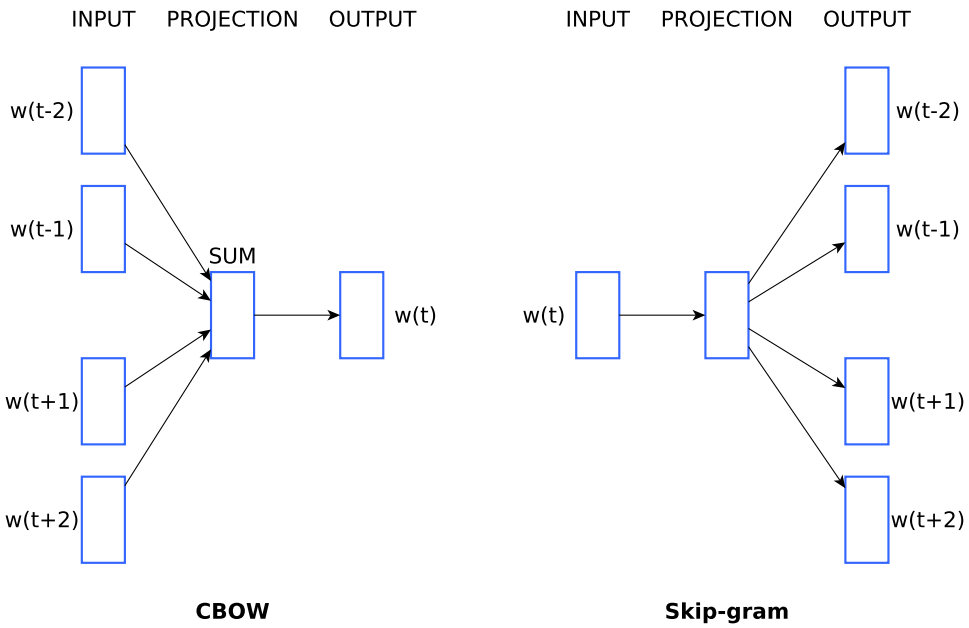


Figure 23: CBOW vs Skip-gram.

2.1.3.4.3 GloVe Yet another competitive attempt, originating from Stanford University, is the *Global Vectors (GloVe)* [58] algorithm. Again, we obtain vector representations for words, but this time training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear sub-structures of the word vector space, aiming to capture deeper meanings and contexts, if possible.

2.2 Semantic Web and Linked Data

The Semantic Web is an extension of the World Wide Web through standards by the World Wide Web Consortium (W3C). The term was coined by Tim Berners-Lee for a web of data that can be processed in a machine-readable format. It has the main goal of providing a common ground in production of data so that they can be shared and reused across contributors with minimal changes, therefore providing the opportunity to easily enhance and enrich large knowledge bases with even more data.

2.2.1 The RDF model and OWL

The Resource Description Framework (RDF) is a family of W3C specifications originally designed as a metadata data model. It has come to be used as a general method for conceptual description or modeling of information that is implemented in web resources, using a variety of syntax notations and data serialization formats. The main idea is that all

Semantic Web data are being described in $< s, p, o >$ triples[7], where s stands for *subject*, p stands for *predicate* and o stands for *object*. Having a set of triples we can describe an entire dataset of entities in a way that s and o are all nodes in a graph and they are being linked by edges with p properties, assembling a big Semantic Web graph.

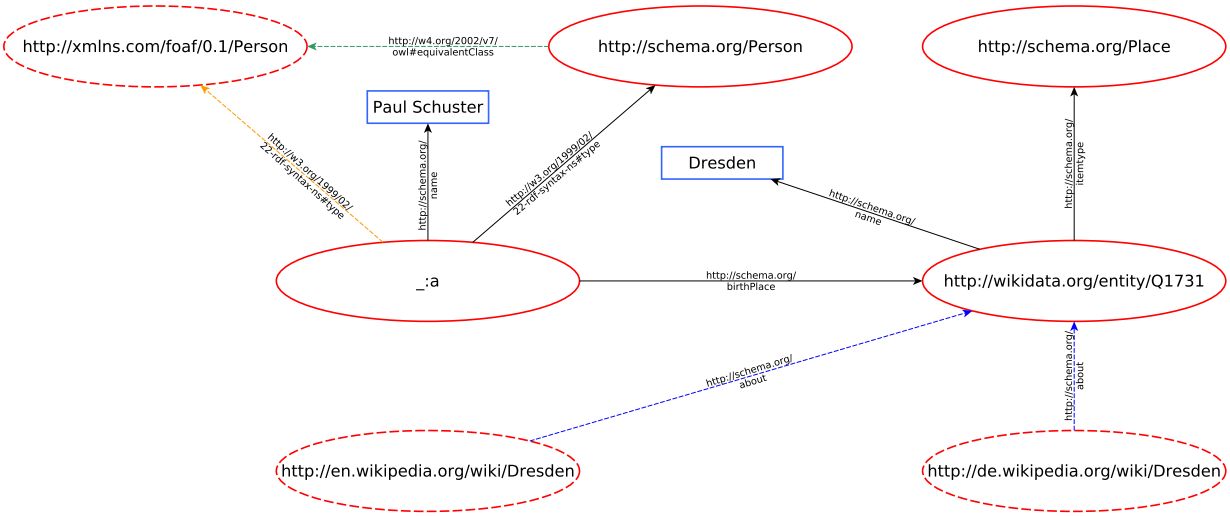


Figure 24: A graph of RDF data from a single source. Dashed edges and nodes show external resources linked to the original.

The Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontologies. Ontologies are a formal way to describe taxonomies and classification networks, essentially defining the structure of knowledge for various domains. It is necessary to accompany a dataset with its corresponding ontology because this enables us to derive relations and axioms among properties and even enhance its querying potential by adding a semantic reasoner[8]. For example, assuming we have a dataset representing the bloodline of the royal family of England and having `siblingOf` relations that are defined as transitive in the ontology, we can infer that if person A has a sibling B and sibling B has a sibling C, then person A also has a sibling C although it was not stated in the set of triples of the dataset. Respectively, if `siblingOf` was also symmetric, knowing that person A has a sibling B would also mean that person B has a sibling A.

2.2.2 Linked Data

The Semantic Web is closely tied to Internet of Things (IoT). IoT is the network of physical devices, vehicles, home appliances and other items embedded with electronics, software, sensors, actuators, and connectivity which enables these objects to connect and exchange data. Each thing is uniquely identifiable through its embedded computing system but is

[7] In recent suggested extensions of the specification quadruples have also been introduced, the fourth part representing temporal information.

[8] A semantic reasoner is a piece of software able to infer logical consequences from a set of asserted facts or axioms.

able to inter-operate within the existing Internet infrastructure. The Semantic Web defines each entity with an Internationalized Resource Identifier (IRI). The IRI was defined by the Internet Engineering Task Force (IETF) in 2005 as a new internet standard to extend upon the existing URI[9] scheme.

2.2.3 Linking related work

Ensuring that each entity in each dataset has a unique identifier, it means we can easily distinguish them from one another. However, this also means that when we wish to enrich knowledge bases, we need to make sure that entities of dataset A match with the corresponding ones of dataset B so that each “real entity” can become enriched with all its relevant information derived from both datasets and nothing more.

When producing RDF data, it is always useful to have a way to link it with corresponding entities of other existing datasets. The reason is simple. Properties of one dataset can complement the properties of another’s for the same entity, therefore enriching the knowledge we can infer about an entity.

In order to interlink datasets, we need to:

- define the relevant entity types from both datasets. For example dataset A might contain singers and actors while dataset B might contain actors and authors, so linking the authors means we need to restrict the entity type to author to reduce noise. Naturally, if both sides got exactly the entity types needed, this is not necessary.
- utilize a property of both datasets that distinguishes entitites from each other and also is likely to match across datasets. It is important to note that it is *not* necessary to use the same property on both sides, though we usually do. Further on, the most frequent property used for this is the label of the entity. Of crucial importance is the similarity metric and threshold used as it can greatly effect the quality of the discovered links.
- produce links that relate an entity from one side with one of the other. Most frequently, it will be `owl:sameAs` links.

However, this is no easy task, in no small part due to the freedom and flexibility of the RDF specifications. Throughout the years, numerous tools developed by academics have been introduced to tackle this problem. `Silk` [10] is a well-known workbench with numerous capabilities, all geared towards handling rdf datasets and making interlinking easier. It provides the *Link Specification Language (LSL)* so that a user can specify the exact properties, types and relation types they wish to include in a linking task. Additionally, it provides many functions that can manipulate input (tokenizers, regexes, capitalizers, etc.), encodings (utf-8, ISO, etc.) or measure distances (Levenshtein, Jaro-Wrinkler, Substring,

[9] In information technology, a Uniform Resource Identifier (URI) is a string of characters used to identify a resource.
[10] http://silkframework.org

etc.). Since it is open source, the user can even extend Silk's functionality by adding custom plugins.

2.3 Evaluation metrics

Apart from the methodology followed and the implementation of the code, it is particularly important to properly and fairly evaluate our networks' performance. So, before we delve into the proposed models, we need to showcase the necessary metrics which compose a proper, unbiased, evaluation.

The first metric we need to utilize is *Precision*, also known as *PPV (Positive Predicted Value)*. It is the fraction of relevant instances among the retrieved instances:

$$PPV = \frac{TP}{P} = \frac{TP}{TP + FN}$$

Intuitively, precision shows the ratio between correctly identified predictions and wrong identified predictions.

Secondly, we need to complement precision with *Recall*, also known as *sensitivity* and *TPR (True Positive Rate)*. It is the fraction of relevant instances retrieved over the total amount of relevant instances:

$$TPR = \frac{TP}{TP + FP}$$

Recall shows the ratio between correct identified predictions and correct unidentified predictions.

A careful reader might wonder, *why do we need both of these metrics and not just one*? The answer is that they complement each other and better showcase the true capabilities of a neural network. Regarding precision, finding many true positives and very few false positives is useless if the neural network does not find many true positives compared to all positives (the complement of true positives for the positives set are the false negatives). Respectively, regarding recall, finding many true positives and leaving out very few false negatives is useless if the neural network also finds too many false positives. Therefore, both metrics are required to avoid biased results and conclusions.

However, even the two metrics together might yield biased results. The reason is that there might be biased within the datasets themselves and therefore, they are also subject to a different metric that evaluates the accuracy of the test itself: $F_1 - score$ *(also known as F-measure)*. It takes into consideration both precision and recall, while this score's value is in [0, 1]. In most cases, we use $F_1 - score$, but in general we can use $F_\beta - score$, where β is a parameter that indicates a different balancing of importance between Precision and Recall among the F-family. $F_1 - score$ sets $= 1$:

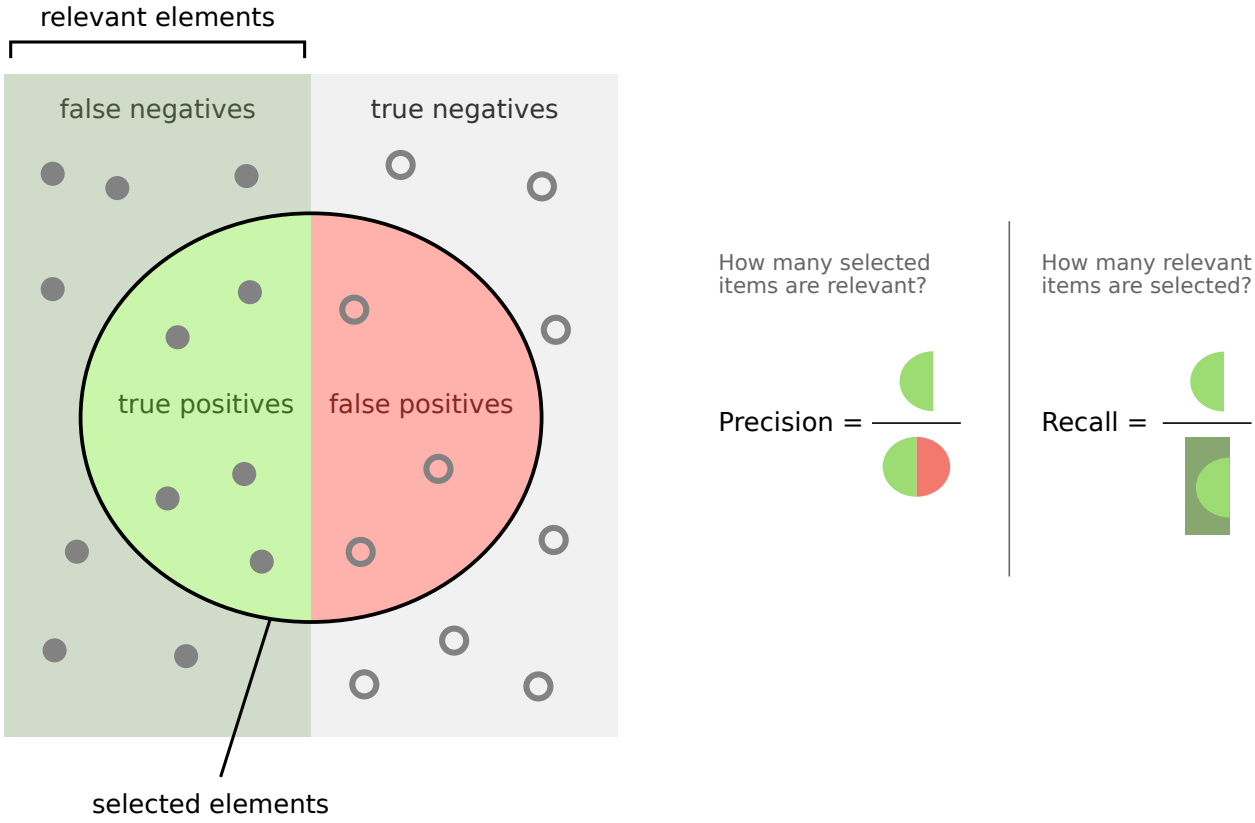


Figure 25: Precision and Recall visually. The circle square region shows the entire dataset, the circle is the neural network's classified data. Precision, Recall and True/False Positives/Negatives are shown with intuitive colors.

$$F_\beta = (1+\beta^2) \cdot \frac{precision \cdot recall}{precision + recall} \; (in \; precision - recall \; terms)$$

$$F_\beta = \frac{(1+\beta^2) \cdot TP}{(1+\beta^2) \cdot TP + \beta^2 \cdot FN + FP} \; (in \; TP, \; FP, \; FN \; terms)$$

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall} = \frac{2 \cdot TP}{2 \cdot TP + FN + FP}$$

Finally, in order to provide a general score for all entity types recognized, we average the precision, recall and $F_1 - score$ individually, obtaining 3 values. These averages can be produced either by micro-averaging or macro-averaging. Both of these are necessary for a proper evaluation because while the former aggregates the contributions of all classes to compute the average metric, the latter computes the metric independently for each class and then takes the average with the intention of treating all classes equally.

3. TASK DEFINITION

In this chapter we will describe the challenge this thesis tackles in more detail.

3.1 Entity extraction in legal text

As explained in the introduction, we want to find a way to extract named entitites from greek legal text. Traditionally, there are two main approaches one can take to tackle the problem:

- use gazetteers and heuristics, possibly hand-written rules and regular expressions (regexes) in order to extract the desired information. When applicable, it is often the desired approach due to the relative simplicity and time required to implement.
- use machine learning and some sort of neural network specialized in natural language processing (NLP) so that we can train an architecture well-enough that it can learn to distinguish and successfully extract named entities. Despite being a more generalized effort, it will only be considered as a “last resort”, mostly due to the requirement of providing large training datasets; sometimes no such datasets exist and need to be manually generated which in turn could mean that, due to human errors, the training datasets themselves contain mistakes that the network will itself learn, as a result.

Based on all that, it seems that gazetteers and heuristics would be much simpler to use. So, *why did we choose to tackle the problem with neural networks*? In essense, there are multiple reasons behind this decision:

- greek legal documents rarely follow a strict template which means that even if we decided to utilize heuristics, we would just need to employ too many and also making sure they would not interfere with each other.
- since, among other types of entities, we also want to extract organizations and legal references, we do *not* have a finite set of vocabulary so that a gazetteer could be utilized. Taking into consideration the multiple ways the same phrase can be expressed in the greek language as well as possible abbreviations, it becomes obvious that a more general method is required.
- greek legal documents, especially older ones (before 1990) are available in PDF format only which means that in order to get the actual text, OCR techniques need to be employed, adding even more noise to the input data. In this setting, gazetteers will be much less accurate to find matches. Furthermore, due to the text existing in a double-column format, lines are being continued after line breaks and it is also challenging to be able to distinguish the text between two columns.

- The English language has a very wide range of resources (neural networks included) and tools that can efficiently process most types of Natural Language input. From a linguistic point of view, english is very structured and has good properties (essentially english can be structured as a tree that expands on the bottom right continuously, most of the time), which means that we are already expecting good results. Greek is considered one of the most difficult languages worldwide and therefore achieving a good accuracy here is one of the ultimate tests a neural network can tackle.

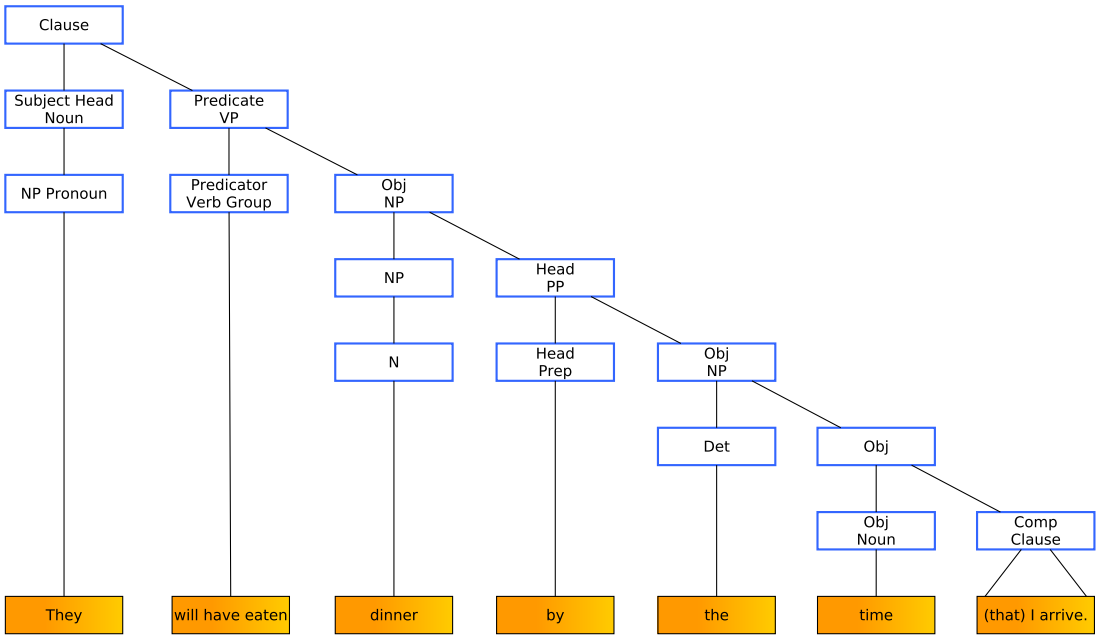


Figure 26: Parsing tree of an english sentence.

3.1.1 Classes

For our experiments, we focus on extracting 6 entity types, when present:

- **Person.** Any formal name of a person mentioned in the text. There are most probably Greek government members.
- **Geopolitical Entity.** Any reference to a geopolitical entity (e.g., country, city, Greek administrative unit, etc.)
- **Organization.** Any reference to a public or private organization, such as: international organizations (e.g., European Union, United Nations, etc.), Greek public organizations (e.g., Social Insurance Institution) or private organiza- tions (e.g., companies, NGOs, etc.).

- **Geographical Landmark.** References to geographical entities such as local districts, roads, farms, beaches, which are mainly included in legislation related to topographical procedures and urban planning.
- **Legislation Reference.** Any reference to Greek or European legislation (e.g., Presidential Decrees, Laws, Decisions, EU Regulations and Directives etc.)
- **Public Document Reference.** Any reference to documents or decisions that have been published by a public institution (organization) that are not considered primary source of legislation (e.g., local decisions, announcements, memorandums, directives).

3.1.2 Annotation and datasets

When preparing datasets for NLP training, we need to provide examples of tokens and their labels so that we can feed this information into a neural network to properly train. To this end, the community has developed some tools dedicated to the task of *annotation*. For our purposes, we focus on *brat* (brat rapid annotation tool) [18]. Initially, it was created as an extension of the stav text annotation visualizer[1], an annotation visualization tool created by Pontus Stenetorp, Goran Topić, Sampo Pyysalo and Tomoko Ohta (then members of the Tsujii laboratory of the University of Tokyo).

Simply put, brat accepts as input a set of txt files, visualizing in a robust web client. Then, we can define classes of entities and any relations linking them as possible annotation labels. To prepare the datasets, all we need to do is annotate the tokens that we wish to give a label to; all this information is being written in .ann files which contain lines with information such as the annotation id, the actual text , the class and the offsets (start and end).

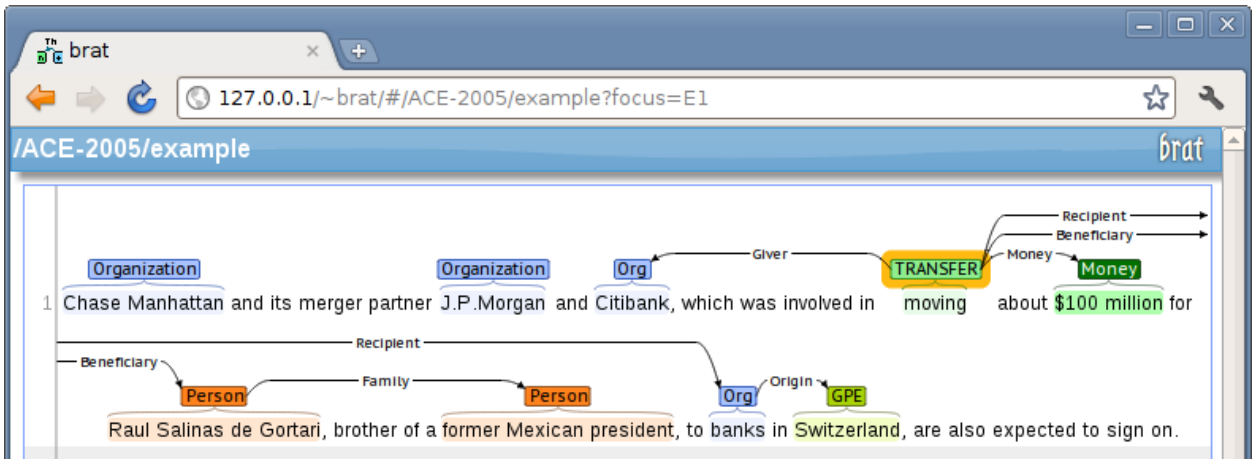


Figure 27: Visualization in brat.

[1] https://github.com/TsujiiLaboratory/stav

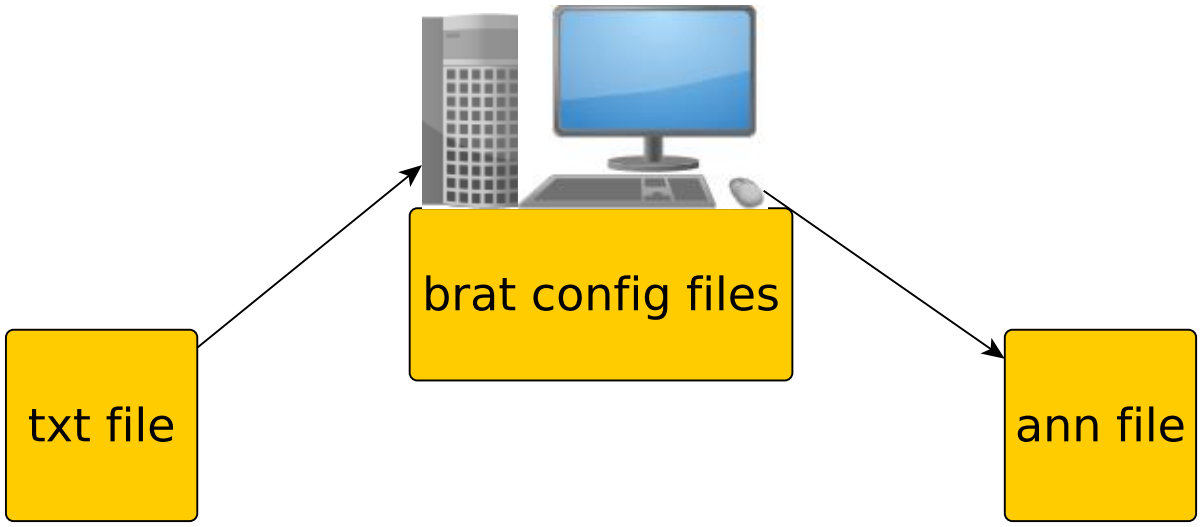


Figure 28: brat's workflow.

T6	PERSON	13807	13832	ΣΤΑΜΑΤΗΣ ΚΑΡΜΑΝΤΖΗΣ
T7	LOCATION-UNK	75	126	θέση «συστάδα 2β» του Συνιδιόκτητου Δάσους Πλατάνης
T11	GPE	667	691	ΑΠΟΚΕΝΤΡΩΜΕΝΗΣ ΔΙΟΙΚΗΣΗΣ
T12	GPE	693	712	ΜΑΚΕΔΟΝΙΑΣ - ΘΡΑΚΗΣ
T14	LEG-REFS	804	815	v. 998/1979

Figure 29: A .ann file example produced by brat.

Generally speaking, when annotating entities in text for training, we use BIO tags. BIO stands for beginning, inside and outside and it is necessary when an entity consists of multiple tokens. If, for instance, we had the text "Municipality of Athens belongs to", "Municipality" would be B, "of" and "Athens" would be I and "belongs" and "to" would be O.

The benchmark datasets for our experiments contain 276 daily issues for class A and D of the Greek Government Gazette over the period 2000-2017. Every issue contains multiple legal texts. Class A issues concern primary legislation published by the Greek government (e.g., laws, presidential decrees, ministerial decisions, regulations, etc.). Class D issues concern decisions related to urban, rural and environmental planning (e.g., reforestations, declassifications, expropriations, etc.).

We uniformly splitted the issues across training (184, 60%), validation (45, 20%), and test (47, 20%) in terms of publication year and class. Thus the possibility of overfitting due to specific linguistic idiosyncrasies in the language of a government or due to specific entities and policies is minimal. We annotated all of the above documents for the 6 entity types that we examine, using *brat*.

3.2 Public open datasets to link

In this section we briefly discuss third-party datasets that we will try to interlinking our extracted entities with.

3.2.1 GAG - Kallikratis

Our group has made publicly available the dataset of Kallikratis[2], which contains RDF data about the geographical boundaries of all administrative divisions of Greece, as well as other useful information such as estimations of population number for each division. Since it is a complete dataset with relevant information to our work, we can utilize this dataset to interlink extracted GPEs from greek legislation and, as a result, obtain all the information Kallikratis contains about them.

3.2.2 DBpedia Persons

DBpedia is a project aiming to extract structured content from the information created in the Wikipedia project. This structured information is made available on the World Wide Web. DBpedia allows users to semantically query relationships and properties of Wikipedia resources, including links to other related datasets. It is considered by Tim Berners-Lee and others as one of the most famous parts of the decentralized Linked Data effort. DBpedia integrates many large datasets such as Yago[3], Wikidata[4], Wordnet[5], schema.org[6] etc. As a result, it contains a wealth of information about entities. Of interest to us are the entities of greek politicians and some of their properties (such as their birthplace, the political party they belong to etc.). Therefore, we extract this small subset of DBpedia for our purposes.

3.2.3 ELI - Nomothesi@

As explained in the introduction, our group created Nomothesia following initiatives such as Holland's Metalex[7] and UK's legislation [8] in order to provide an enriched ontology capable of properly capturing the semantics of greek legislation documents. With the addition of our NER/NEL component, we will have recognized legislation references inside passages which will be then encoded into RDF.

[2] http://linkedopendata.gr/dataset/greek-administrative-geography
[3] https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/
[4] https://www.wikidata.org/wiki/Wikidata:Main_Page
[5] https://wordnet.princeton.edu/
[6] http://schema.org/
[7] http://doc.metalex.eu/
[8] http://www.legislation.gov.uk/

4. NER EXPERIMENTS

In this chapter we focus on describing the workflow of the named entity recognition. We provide the experimental evaluation specifications, as well as outline the neural networks we conducted experiments on.

4.1 NER state-of-the-art

Having established the reasons to employ a neural network for the task of named entity recognition, we are ready to consider our options among the available architectures. For starters, since the task at hand belongs to the field of natural language processing, it makes sense to consider good NLP networks. Recurrent models have been shown to produce state-of-the-art results for language modeling [59], as well as for sequence tagging [54, 26], machine translation [60, 61], dependency parsing [55] and sentiment analysis [62].

The main reason (BI)LSTMs (which are a more advanced form of RNN networks) are used for NLP is their ability to deal with information memorization and structure. Numerous examples such as Andrej Karpathy's[1] show many such applications. Examples involve teaching an RNN to learn english words and write Shakespeare parts on its own, syntactic structures from Wikipedia, writing LaTeXcode that compiles or even writing Linux code.

Furthermore, the work of Chalkidis et al. [3, 4] has shown how BILSTM models can be applied on contracts to extract useful information. Adapting and evolving these techniques, we endeavor to achieve information and entity extraction from greek legislation documents, expecting similar success in the process.

4.2 Workflow

Let's begin by showcasing the summarized workflow of our approach, since that will make the following sections easier to comprehend (each step will be analyzed further):

1. We begin by taking a set of greek legislation documents in PDF format, convert them into text and prepare the data so that each line contains a single sentence.

2. We tokenize the text so that each token is a single word. Punctuation are also tokens (with the exception of punctuation used in abbreviations).

3. We conduct `Word2Vec` and/or `FastText` training to obtain the word embeddings necessary to run neural network experiments.

[1] See http://karpathy.github.io/2015/05/21/rnn-effectiveness/

4. We manually annotate greek legislation documents of the National Printing House using brat[2] [18] so that we can begin supervised training.

5. We feed the word embeddings in addition to embeddings shapes for each token into each of the four proposed models and do grid search so that we can determine the optimal set of parameters.

6. We evaluate the performance of the neural networks for all parameters calibrated during grid search.

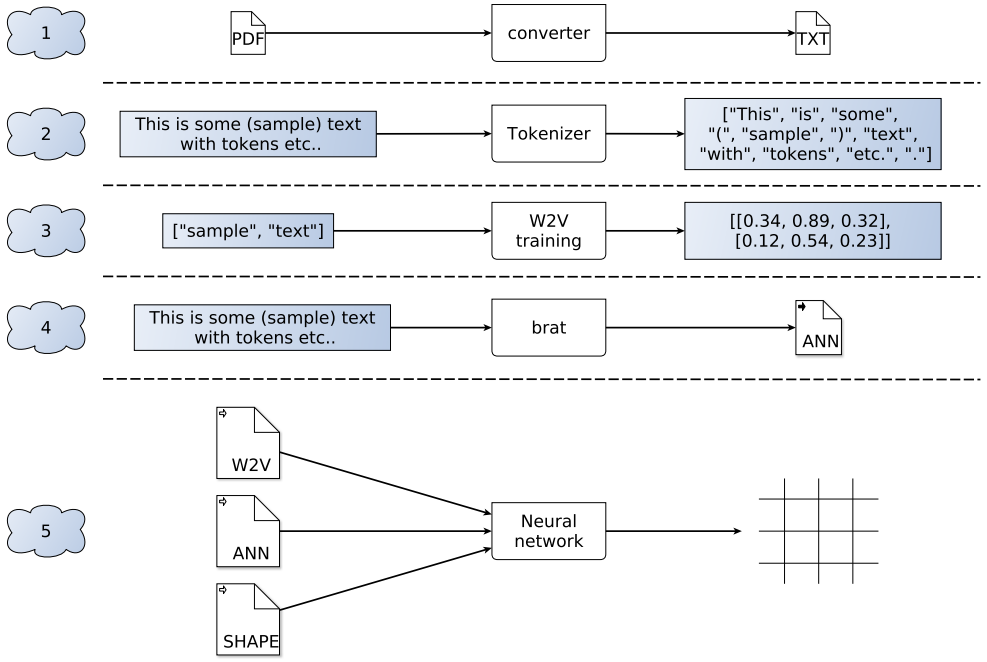


Figure 30: NER workflow.

4.2.1 Extracting entities from a document (workflow example)

Here, we will describe what happens in each step of the workflow in a small example (the numbers correspond to the processing steps of Figure 30):

1. We download 10 documents from the National Printing House of Greece, which are in PDF format and we convert them into TXT. The text obtained originates from a two-column document, therefore some words might break into new lines. The resulting text files are being processed so that the columns are positioned one under the other (so that the text has the intended flow) and word breaks are being corrected. Further on, the files are structured in such a way that each line of the final document will contain a single sentence. This is needed to enhance the performance of the

[2] http://brat.nlplab.org/

neural networks during training, as we wish to provide an entire sentence as input per sample.

Κήρυξη ως αναδασωτέας εκτάσεως 3.557,00 τ.μ. ⏎
στη θέση «συστάδα 2β» του Συνιδιόκτητου Δά- ⏎
σους Πλατάνης,του Δήμου Έδεσσας, της ΠΕ Πέλ- ⏎
λας.

Κήρυξη ως αναδασωτέας εκτάσεως 3.557,00 τ.μ. στη θέση «συστάδα 2β» του Συνιδιόκτητου Δάσους Πλατάνης, του Δήμου Έδεσσας, της ΠΕ Πέλλας.

Figure 31: PDF to TXT conversion. On the left we can see how the original text is found in a PDF file and on the right we can see the processed text in the TXT file. Notice the newline characters present on the left side and the word breaks (the right side text is in a single line).

2. The text obtained will now be tokenized, meaning we are going to generate lists of tokens for each token found in the text, alongside their offsets in the text. Punctuation is also considered as separate tokens with a few exceptions such as it being present in abbreviations (“π.δ.”).

3. The entire tokenized corpus from all relevant documents will be fed into Word2Vec and Fasttext for training.

4. The corpus will be annotated using brat to obtain labels for supervised training. As explained before, when annotating, we select the tokens that represent a specific entity and then we pick its type from a list of relevant types (Person, Organization, GPE, Geographical Landmark, Legislation Reference, Public Document). That set of tokens is then provided with a label (class type), the offsets found and the actual string as shown in the text (all this information is written in the corresponding .ann file by brat).

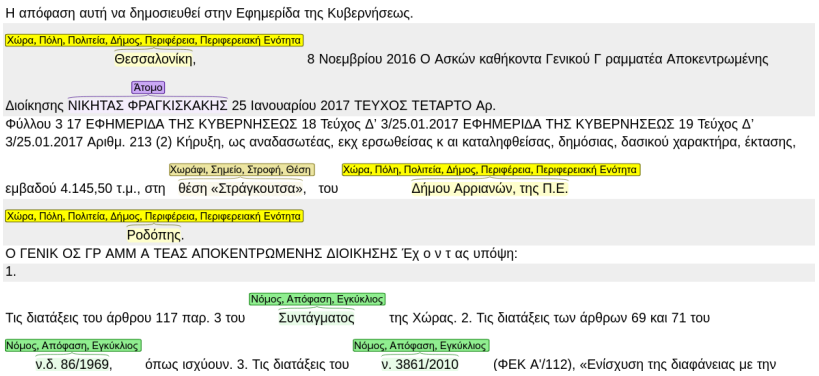


Figure 32: Annotating a document with brat.

5. We split the documents into one of three categories: test, train, validation. We pick 60% for training (6 documents), 20% for validation (2 documents) and 20% for testing (2 documents). We utilize Word2Vec to convert all tokens of the original text (of the training dataset) into vectors, providing their labels obtained from the previous step. Further on, we provide shape embeddings depending on the form of the tokens

themselves, feeding all this information into our neural networks. After training is completed, we use the trained models to predict the labels of the validation dataset (this is possible since we also got the correct labels from the annotation phase) for evaluation of the training process. Finally, we predict the labels of the test dataset as well to measure the performance of the models.

4.3 Word Embeddings

The first step towards our goal is text tokenization. Since we need to feed tokens into the neural network as well as for `Word2Vec/FastText` training, we need to tokenize the text obtained from the original PDF format. It is highly likely that we encounter certain punctuation like quotation marks, full stops, commas etc, all of which need to be artificially separated from any token/word they are next to. To achieve this, we have built a Tokenizer module that is based on NLTK[3]. However, we had to manually handle special cases like the above since the parsing tree provided by the library handles punctuation slightly differently. Furthermore, we convert all digits encountered in the text into “d”, a necessary mapping for `Word2Vec` training. Lastly, it is necessary to normalize and capitalize the entire text (see subsection 4.3.1) and map all English words to a single word named “ENGLISH_WORD”.

4.3.1 Word2Vec Training

Having obtained the tokenized text, we can begin `Word2Vec` training. As explained before, a `Word2Vec` model will map each word/token into a proper word embedding/vector that captures its meaning and places it in a hyper-space in such a position that it can be related to words with similar meanings. However, during tokenization, it is important to map all digits into “d” in order to better train our model.

The reason for this mapping is simple: throughout the text we are guaranteed to encounter multiple law ids, dates and numerals in general. Since each different token is mapped into a different vector, strings such as “13/12/2005”, “26/10/2014” would be mapped into different vectors. This would mean that `Word2Vec` training would be biased in its effort to accomodate these mappings, sacrificing more efficient mappings for all other kinds of tokens. By converting the digits into “d”, the above would be “dd/dd/dddd”, meaning they would have just one vector as a representation.

Another necessary transformation is the normalization and capitalization of all tokens. Since many words can be encountered in all-capitals, first capital and all-lowercased in some parts of the text, we again encounter a similar problem as with the digits cases; multiple vectors are generated for the same token. This becomes even more problematic if we consider the fact that the greek language also includes accents. So, we can address these issues by capitalizing and normalizing all tokens. The reason for capitalizing against

[3] http://www.nltk.org

lowercasing is so that we eliminate all accents and also because while the greek language has 25 lower-case letters (because of σ/ς), it has 24 upper-case ones.

Lastly, since the text might have some english words/references, we also need to map all these into a single word named “ENGLISH_WORD”. This is necessary because we are not interested in english text in our experiments. Therefore we bias the `Word2Vec` training to map all english words into a single vector representation so that it focuses on greek.

In our work, we applied `Word2Vec` (skip-gram model) [19, 20] to an unlabelled corpus, which contains:

- 150,000 issues of Greek Government Gazette in the period of 1990-2017.
- all pieces of legislation from the foundation of the Greek Nation in 1821 until 1990, which sum up to 50,000.
- 1,500 case laws published online by Greek Courts.
- most EU Treaties, Regulations and Decisions, that have been translated in Greek and published in EUR-Lex.
- the Greek part of the European Parliament Proceedings Parallel Corpus.

We produced 100-dimensional word embeddings for a vocabulary of 428,963 words (types), based on 615 millions of tokens (words), included in the unlabelled corpus. We used Gensim’s implementation of `Word2Vec` (http://radimrehurek.com/gensim/), with 10 minimum occurrences per word, 20 epochs and default values for other parameters. Out of vocabulary words were mapped to a single “UNK” embedding.

The `Word2Vec` model training was carried out on a computer with an Intel® Xeon® E5-4603 v2, with a CPU frequency of 2.20GHz, a 10.24 MB L3 cache, a total of 128 GB DDR3 1600 MHz RAM and the Linux Debian 8.6 (Jessie) x86 64 OS.

4.3.2 FastText experimentation

We also experimented with publicly available generic pre-trained 200-dimensional word embeddings, which have been built with FastText [21] (https://fasttext.cc), based on a much larger corpus with Greek Wikipedia articles. As we will show, the experimental results were worse in specific entity types extracted by our neural networks, possibly because legal expressions are under-represented (or do not exist) in generic corpora (e.g., wikipedia or news articles).

One of the key features of `FastText` word representation is its ability to produce vectors for any words, even made-up ones which differentiates it from `Word2Vec` (since it only maps existing words into vectors). Indeed, `FastText` word vectors are built from vectors of substrings of characters contained in it. This allows to build vectors even for misspelled words or concatenation of words.

4.4 Token Shape Embeddings

We use token shape embeddings [4, 22] that represent the following seven possible shapes of tokens:

- token consisting of alphabetic upper-case characters, possibly including periods and hyphens (e.g., “ΠΡΟΕΔΡΟΣ”, “Π.Δ.”, “ΠΔ/ΤΟΣ”)
- token consisting of alphabetic lower-case characters, possibly including periods and hyphens (e.g., “νόμος”, “ν.”, “υπερ-φόρτωση”)
- token with at least two characters, consisting of an alphabetic upper-case first character, followed by alphabetic lower-case characters, possibly including periods and hyphens (e.g., “Δήμος”, “Αναπλ.”)
- token consisting of digits, possibly including periods and commas (e.g., “2009”, “12,000”, “1.1”)
- line break
- any token containing only non-alphanumeric characters (e.g., “.”, “€”)
- any other token (e.g., “1ο”, “ΟΙΚ/88/4522”, “EU”)

In general, the shape (form) of its token relies on the existence and relative position of alphabetic characters, digits and punctuation. Intuitively, this information is going to help the neural network conduct entity recognition more efficiently since we provide word embeddings and also shapes for each token.

4.5 POS tag embeddings

A promising component that could perhaps improve the quality of our results is a *Part Of Speech (POS)* tagger. Despite our wishes to try incorporating such a component due to potential performance increase, we were unable to embed the part-of-speech tag of each token due to the fact that so far there is no currently available POS tagger for the Greek language that can cover all aspects of the present research. We verified this by experimenting with the NLTK (http://www.nltk.org) POS tagger and, as also the one provided by CLTK (http://cltk.org), but both of them had a vast amount of wrong predictions, a fact that is even more profound in legal text.

4.6 BILSTM-based architectures

Here, we will showcase the models we experimented on. The architectures are considered state-of-the-art and necessary for the task of Named Entity Recognition so that we can

extract the desired entities. We will also explain how we conducted grid search to fine-tune their hyper-parameters. Finally, we provide a thorough evaluation report on our results and speculate about possibly unexpected findings.

4.6.1 BILSTM-LR

The first LSTM-based method that we have used, called BILSTM-LR (Figure 33) uses a bidirectional LSTM (BILSTM) chain, to convert the concatenated word and token shape embeddings of each token in each sentence to context-aware token embeddings, which better describe the semantics of each token given the specific task. Each context-aware token embedding is then passed on to the logistic regression layer (including the softmax activation) to estimate the probability that the corresponding token belongs to each of the examined categories (e.g., person, organization, etc.).

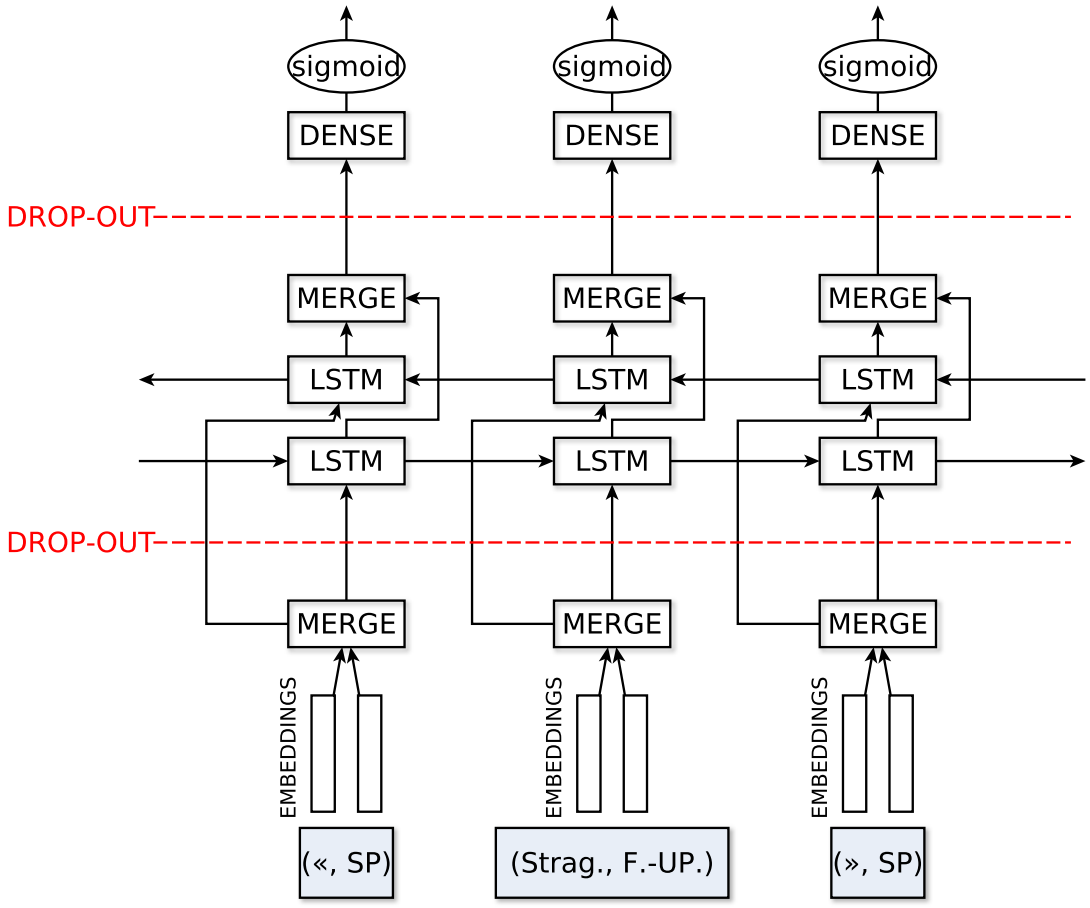


Figure 33: A BILSTM-LR model.

4.6.2 BILSTM-LSTM-LR

The second LSTM-based method, called BILSTM-LSTM-LR, is the same as the previous one, except that it has an additional LSTM chain between the context-aware token embeddings of the lower BILSTM chain and the final logistic regression layer. Stacking LSTM (or BILSTM) chains has been reported to improve efficiency in several natural language processing tasks [63, 60] at the expense of a shortly increased computational cost.

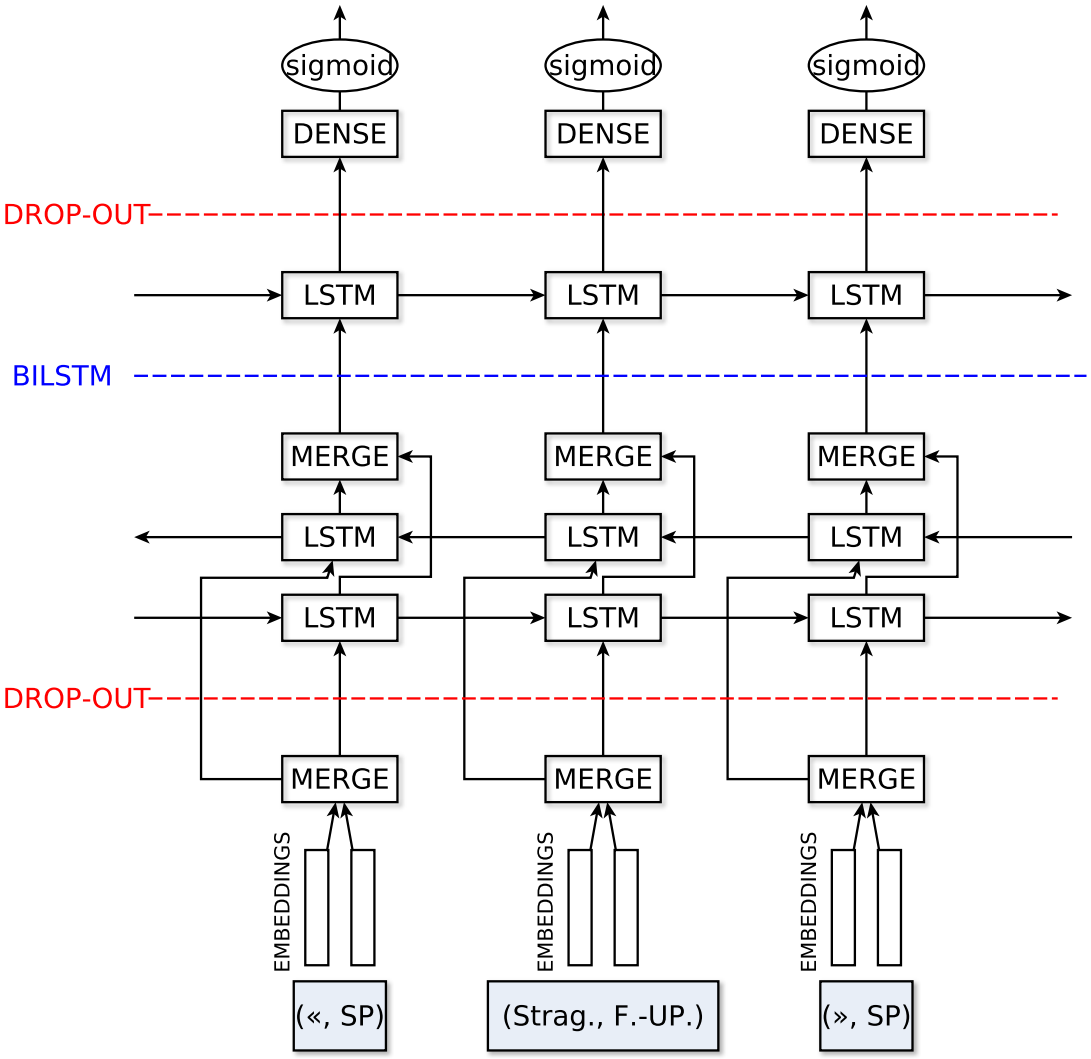


Figure 34: A BILSTM-LSTM-LR model.

4.6.3 BILSTM-BILSTM-LR

The third LSTM-based method, called BILSTM-BILSTM-LR has a BILSTM chain, instead of the single direction LSTM chain of the previous method, between the context-aware

token embeddings of the lower BILSTM chain, and the logistic regression layer.

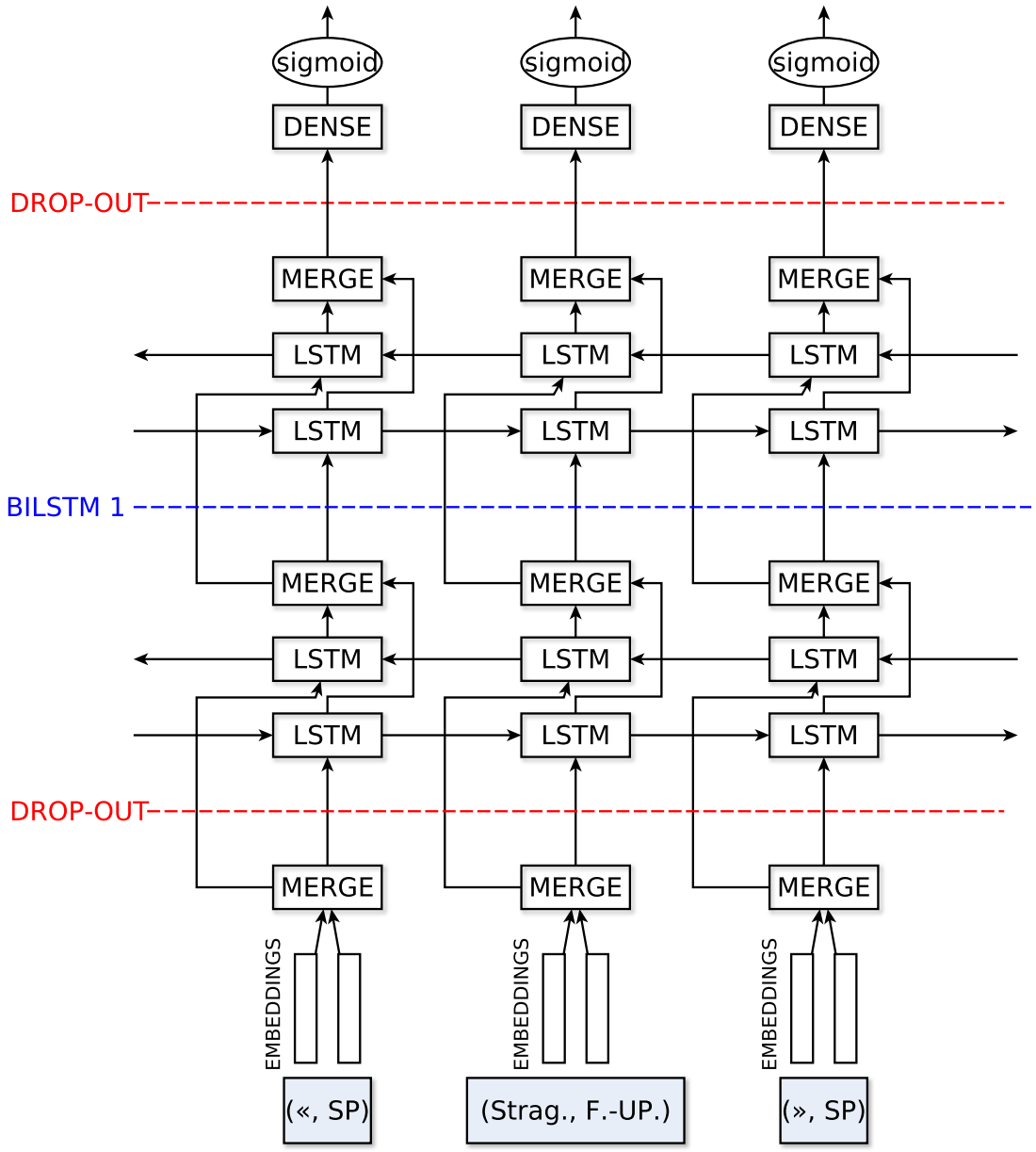


Figure 35: A BILSTM-BILSTM-LR model.

4.6.4 BILSTM-CRF

In the fourth LSTM-based method, called BILSTM-CRF, we replace the upper LSTM chains and the logistic regression layer of the stacked-LSTM method with a *linear-chain Conditional Random Field (CRF)*. In our case, the CRF layer jointly selects the assignment of positive or negative labels to the entire token sequence of each sentence, which

allows taking into account the predicted labels of neighboring tokens [4]. The previous three LSTM-based recognizers still take into account the surrounding tokens (by considering their features from the pre-trained and context-aware embeddings), but they do that in a greedy fashion per token. Given the token sequence of a sentence, the BILSTM-CRF recognizer computes the joint conditional probability for each possible label assignment of the token sequence. As in conventional CRFs, decoding (searching for the optimum label assignment) can be performed via dynamic programming or beam search [52, 55]. Training combines dynamic programming or beam search decoding with backpropagation to maximize the joint conditional log-likelihood [5].

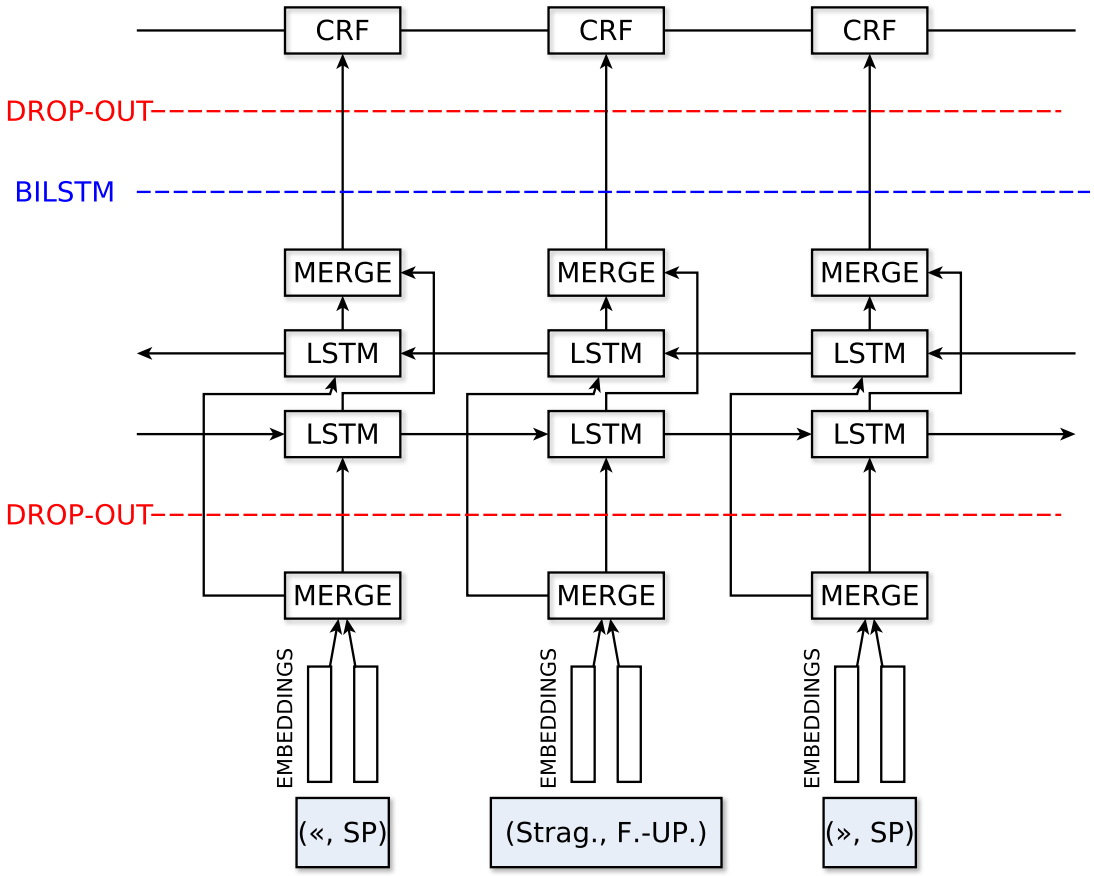


Figure 36: A BILSTM-CRF model.

While all models share some similarities, they also have a few differences. All of them are classified as deep learning architectures and share some of their components, but each one is based on a different intuition to provide good results. BILSTM-CRF and

[4] This is particularly useful for capturing tokens that are apart but part of the same label, e.g., "το υπ' αριθμόν 15 (ΦΕΚ Α' 123) π.δ. του 2013". This is why we explore this model's potential for the task at hand.

[5] We use the CRF layer implementation of keras-contrib (https://github.com/farizrahman4u/keras-contrib/blob/master/keras_contrib/layers/crf.py), with joint conditional log-likelihood optimization and Viterbi best path prediction (decoding).

BILSTM-BILSTM are the most computationally expensive architectures, while BILSTM and BILSTM-LSTM are the cheapest. We need to test less complex architectures such as BILSTM and BILSTM-LSTM so that we can verify how much we lose in performance compared to reducing the computational power and time needed to get the results. Another interesting perspective is the possibility of getting better results with less complex architectures. Such a phenomenon would be very rare and surprising, therefore worth mentioning.

4.7 Hyper-parameter tuning

Based on experimentation, the pre-trained word embeddings are not updated during training on the labeled dataset, while in contrast token shape embeddings are not pre-trained. The corresponding shape vectors are being learned during the actual training. We used Glorot initialization [23], binary cross-entropy loss, and the Adam optimizer [24] to train the recognizers with early stopping by examining the validation loss. Hyper-parameters were tuned by grid-searching the following sets, and selecting the values with the best validation loss: hidden units {100, 150}, batch size {16, 24, 32}, dropout rate {0.4, 0.5}.

A neural network, especially one with multiple layers, consists of millions of parameters and optimizing all of them is nearly impossible. We focus on the dropout percentage (dropout is the act of dropping a percentage of the network's units and retrain them so that all neurons remain active and not biased) and batch size (number of samples propagated through the network, large value indicates faster training but less accuracy usually). The neural networks are trained for 30 epochs. The training was carried out on a computer with an Intel® Core™ i5-7600, with a CPU frequency of 3.50GHz, 6.144 MB L3 cache, a total of 32 GB DDR4 2400 MHz RAM, an AORUS GeForce® GTX 1080 Ti with 11264 MB of memory, 3584 CUDA cores and the Linux Ubuntu Gnome 16.04.3 LTS (Xenial Xerus) x86 64 OS. The `Word2Vec` embeddings are vectors of 100 dimensions. Our neural network utilizes Python's library of Keras 2.1.3[6], with tensorflow-gpu 1.4.1[7] as its backend.

4.8 Evaluation

For each of the four methods we measured the performance on precision, recall, and F_1 scores measured per token. As suggested in [3], an evaluation per element, meaning per entity, can provide a more delicate estimation of each method's performance. Regardless, the complex syntax of the legislation text and more specifically groupings of multiple entities in long phrases (e.g., "The municipalities of Athens, Dafnis-Imittou and Varis-Voulas-Vouliagmenis will organize [...]") does not provide a clear segmentation be-

[6] https://keras.io/
[7] https://www.tensorflow.org/

tween the individual entities[8] (e.g., Municipality of Athens, Municipality of Dafni-Imittos, Municipality of Varis-Voulas-Vouliagmenis), so that we may rely on for such a high-order evaluation. Table 5 lists the results of this group of experiments (the numbers are averaged over 5 runs of experiments).

Table 5: Precision (P), Recall (R), and F_1 score, *measured per token*. Best F_1 per entity type shown in bold font.

Entity Type	BILSTM-LR			BILSTM-LSTM-LR			BILSTM-CRF			BILSTM-BILSTM-LR		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Person	0.89	0.90	0.89	0.89	0.94	**0.91**	0.88	0.92	0.90	0.89	0.93	**0.91**
Organization	0.77	0.73	0.75	0.77	0.78	0.77	0.72	0.74	0.73	0.78	0.77	**0.78**
GPE	0.80	0.87	0.84	0.83	0.89	0.86	0.81	0.86	0.83	0.84	0.90	**0.87**
GeoLandmark	0.67	0.82	0.73	0.72	0.86	**0.78**	0.64	0.83	0.72	0.70	0.86	0.77
Legislation Ref.	0.85	0.81	0.83	0.87	0.85	**0.86**	0.80	0.79	0.80	0.88	0.85	**0.86**
Public Document	0.81	0.75	0.78	0.85	0.81	0.82	0.72	0.75	0.74	0.84	0.81	**0.83**
Macro AVG	0.82	0.84	0.83	0.84	0.87	**0.86**	0.79	0.84	0.81	0.85	0.87	**0.86**

The results are highly competitive for all the examined methods. The best results based on the macro-averaged F_1 are coming from both BILSTM-LSTM-LR and BILSTM-BILSTM-LR (0.86), which indicates that the extra LSTM chains, which deepen the model, expand its capacity even by a short margin, compared to BILSTM-LR (0.83) and BILSTM-CRF (0.81). The deficiency of the current NER state-of-the-art method BILSTM-CRF, which has been validated across all possible hyper-parameter sets, is quite impressive. We strongly believe that this issue is strongly correlated with the complicated references of geographical landmarks, legislation references and public documents references, especially in cases with entity reference groupings under a single keyword as demonstrated above.

Table 6: Precision, Recall, and F_1 score for FastText, *measured per token* with BILSTM-BILSTM-LR.

Entity Type	Precision	Recall	F_1-score
Person	0.89	0.88	0.88
Organization	0.75	0.70	0.72
GPE	0.85	0.78	0.81
GeoLandmark	0.64	0.76	0.70
Legislation Ref.	0.82	0.82	0.82
Public Document	0.77	0.74	0.76
Macro AVG	0.81	0.81	0.81

Further on, we are going to rely on the BILSTM-BILSTM-LR recognizer based on the fact that it outperforms the BILSTM-LSTM-LR by 1% in F_1 in Organizations (0.78 vs 0.77), Geopolitical Entities (0.87 vs 0.86) and Public Documents (0.83 vs 0.82), while it is only 1% worse in Geographical Landmarks (0.77 vs 0.78). Considering the generic `FastText` pre-trained embeddings instead of our domain-specific ones, leads to a macro-averaged F_1 of 0.81 for the best reported method BILSTM-BILSTM-LR (Table 6), especially in the latter four categories, in which domain knowledge matters the most (e.g., geographical aspects and codification of documents).

[8]This shortcoming is true for both IO and BIO annotation schemes, which have been widely applied in sequence labelling tasks.

5. LINKING EXPERIMENTS

In this chapter we describe the workflow necessary to conduct interlinking with other datasets after manipulating the extracted entities from the NER component. In addition, the experimental evaluation specifications and a discussion on the results obtained are provided.

5.1 Workflow

Let's begin by showcasing the summarized workflow of our approach, since that will make the following sections easier to comprehend (each step will be analyzed further):

1. We apply post-processing techniques with hand-written rules and regexes to normalize and process the extracted entities into presentable labels.
2. Alongside the labels, we generate RDF data regarding the named entities. Useful properties include the passage in which they were found, their position in the text (for a web-page annotation).
3. We interlink *geo-political entities (GPEs)*, persons and legislation references with Kallikratis (GAG), Dbpedia persons and ELI, respectively with the Silk framework. An intermediate dataset consisting of `owl:sameAs` is generated, as a result.
4. We manually generate a dataset of landmarks which are usually noted in legislation related to urban, rural and environmental planning and, based on heuristic rules and relative position within passages, interlink them with belongs_to relations to corresponding GPEs.

5.1.1 Extracting entities from a document (workflow example)

Here, we will describe what happens in each step of the workflow in a small example (the numbers correspond to the processing steps of Figure 37):

1. Let's assume that a document/law contains the GPE entity "Αποκεντρωμένη Διοίκηση Μ-Θ" (Decentralized Administration of Macedonia-Thrace). The text needs to be post-processed with hand-written rules so that it is normalized. The previous string will become "ΑΠΟΚΕΝΤΡΩΜΕΝΗ ΔΙΟΙΚΗΣΗ ΜΑΚΕΔΟΝΙΑΣ-ΘΡΑΚΗΣ".
2. Having obtained the processed labels originating from the text of the documents, we proceed to use information such as the offsets of the actual entity within the text, the law/document in which it was found etc., in order to produce RDF data.

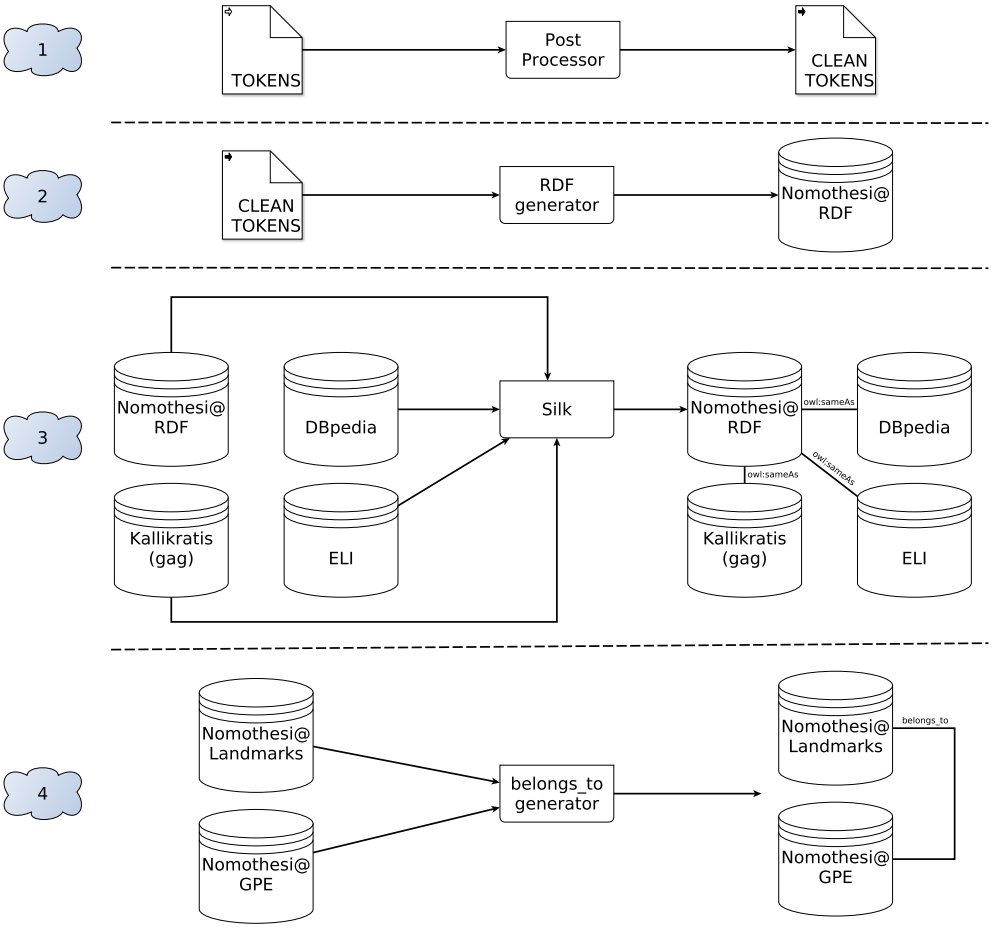


Figure 37: Linking workflow.

3. As it is a GPE entity, it is a candidate for interlinking with Silk. The kallikratis dataset contains an entity with the label “ΑΠΟΚΕΝΤΡΩΜΕΝΗ ΔΙΟΙΚΗΣΗ ΜΑΚΕΔΟΝΙΑΣ ΘΡΑΚΗΣ” (notice the difference), therefore, the substring similarity score between those two entities will be high. An `owl:sameAs` link/triplet will be produced.

4. To complete our example, let’s assume that within the same passage the geographical landmark “Αγρόκτημα Μαυρομιχάλη” was also present (next to the GPE entity). The landmark would also undergo processing like the GPE of the example. Furthermore, since the offsets of the labels found in the original text are “close” (up to 5 characters apart), we can deduce that this landmark belongs to the corresponding GPE entity. Other RDF data will be produced like with all other categories of entities.

5.2 Textual entity references vocabulary

The first step towards linking entity references extracted (by the Named Entity Recognizer) with the entities described in public open datasets is to represent those references using

the RDF specification. The *legal text* of a document contains subdivisions (passages of individual laws) that we define as *LegalResourceSubdivisions* based on the Greek legislation ontology. Since some of those contain text, it is also possible to contain (*has_reference to*) a *Reference* to an entity (e.g., a law passage referring to a specific law that it modifies). This reference is realized in an interval of characters. In other words, it *begins* and *ends* on specific sequential characters inside the text of the subdivision. This *Reference* most likely refers to (or in another sense is *relevant_for*) an Entity, which is probably described in open public datasets. Therefore, a *LegalResourceSubdivision* contains references to persons, administrative units and legal resources (e.g., laws, decisions etc.). The former description is depicted in Figure 38.

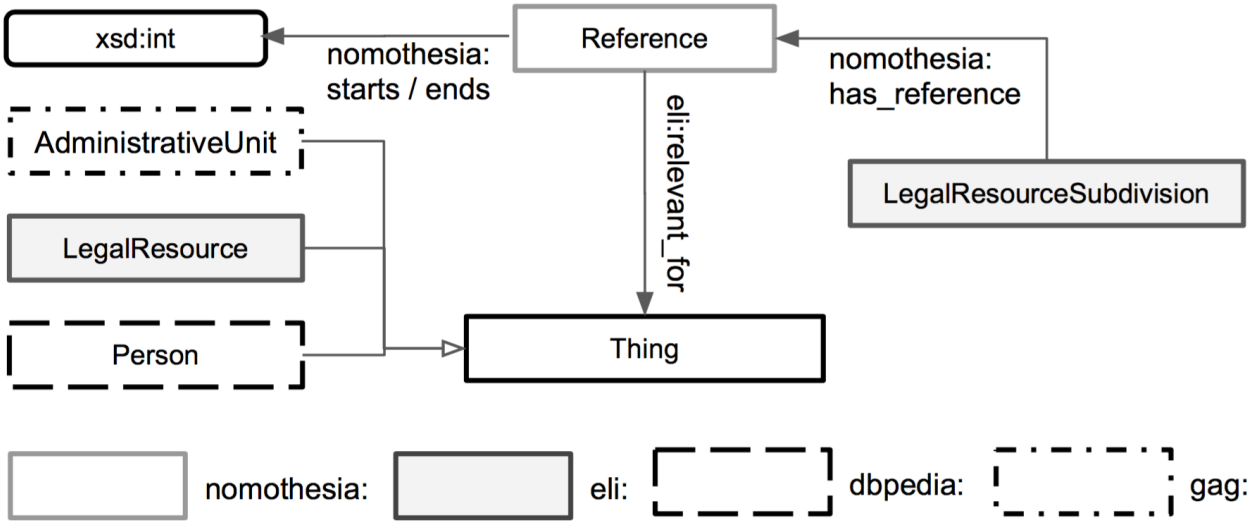


Figure 38: Textual Reference RDF Vocabulary.

5.3 Using Silk and heuristics to generate owl:sameAs links

We linked legal references with legal documents provided by the Greek legislation dataset[1]. We based on heuristic rules to directly interpret the relevant URI by capturing the type, year of publication and the serial number.

We linked person references with Greek politicians retrieved from the Greek DBpedia[2] dataset and geopolitical entity references with the Greek administrative units as they are described in the Greek Administrative Geography (GAG) dataset. For both entity types, we proceed in interlinking the corresponding datasets using the Silk framework. We experimented with two different textual linking operators: Levenshtein and Substring distance [25] over the `rdfs:label` values provided by each dataset. For the case of the Greek Administrative Units, we also provided the type of the administrative units (e.g., local community, municipality, region, etc.) based on the naming conventions that we identified in

[1]Published in http://legislation.di.uoa.gr/legislation.n3.
[2]http://el.dbpedia.org/

the validation part of the labeled dataset.

For each interlinking method that we tried, we examine the performance of the interlinking in terms of *precision*, *recall*, and F_1 score *measured per entity pair* on the test part of our labeled dataset. Here, true positives (*TP*) are references correctly paired with an entity of each set, *false positives* (*FP*) are references incorrectly paired with entities, and *false negatives* (*FN*) are references incorrectly not paired with the relative entities of the examined sets. The acceptance threshold for both linking operators was tuned on the validation part of our datasets, while the entity pairs provided are those presented in the test part. Table 7 lists the results for this group of experiments.

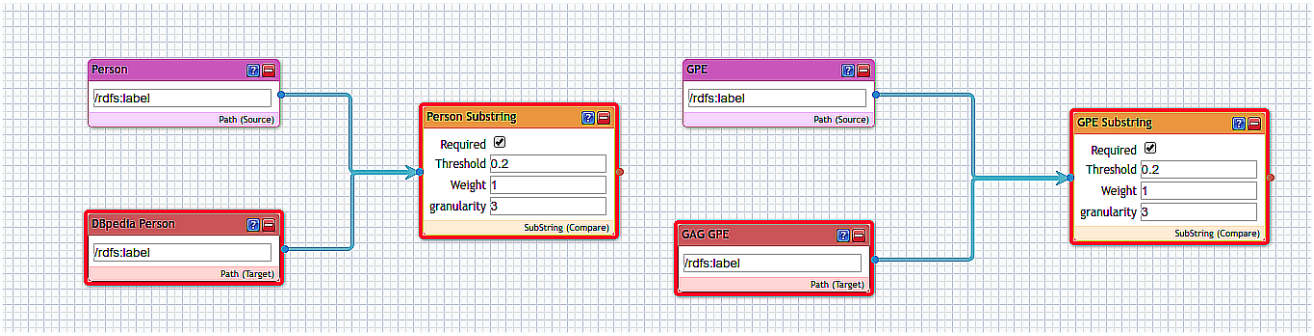


Figure 39: The interlinking process in Silk for persons and GPEs.

5.4 Evaluation

Linking persons was a great challenge for our system, mainly because legislators and the publication office tend to refer to a person's first name by its initials (e.g., A. Tsipras), thus a fair amount of person references have been misclassified (precision: 0.71) for persons with the same surname. We successfully linked the geopolitical entities with the Greek administrative units (F_1: 0.92). Minor issues are related to the segmentation of compound references of multiple administrative units. The results for legislation references are excellent (F_1: 0.98), while a short margin of documents are mis-linked due to the fact that ministerial decisions do not have a standard codification (neither standard reference pattern), which vary from one ministry to another.

Table 7: Precision (P), Recall (R), and F_1 score, *measured per entity pair*.

metrics	linking technique								
Entity Type	rules			levenshtein			Substring		
	P	R	F1	P	R	F1	P	R	F1
Person	-	-	-	0.99	0.55	0.71	0.90	0.68	**0.77**
GPE	-	-	-	0.99	0.79	0.88	0.95	0.92	**0.94**
Legislation Ref	0.99	0.97	**0.98**	-	-	-	-	-	-

5.5 Greek geographical landmarks dataset generation

Greek geographical landmarks are a major asset for our legal recognizer since they are related to planning and architectural interests. However, there is no such public dataset to interlink between the references and the actual entities. We proceed in generating a new dataset by applying linguistic heuristics to create a set of unique landmarks, classified in 5 different main categories (classes):

- **Local District.** Rural districts such as villages and small local communities (e.g., Koukkari Settlement).
- **Area.** Geographical areas, mainly sub-classified in agricultural, forest, coastal and marine areas (e.g., Area Peristeria).
- **Road.** Roads sub-classified in highway, local, bypass roads or hairpin turns (e.g., Kastelia Road).
- **Beach.** Areas that are designated for swimming (e.g., Kavouri Beach).
- **Islet.** Small islands that most possibly are not inhabited (e.g., Poliaegos Islet).

Further on, we interlink the new dataset with the Greek administrative units in case there is a connection between them (belongs_to) indicated in terms of text (e.g., “Beach Kavouri at Municipality of Varis-Voulas-Vouliagmenis”). In Figure 40, we depict the mini-ontology of Greek Landmarks dataset.

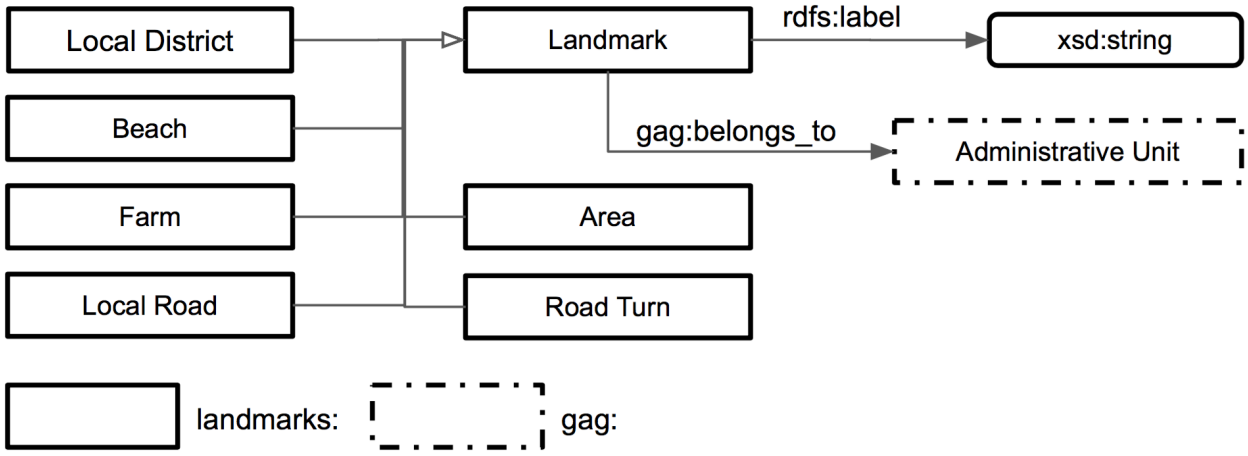


Figure 40: Geographical Landmark RDF vocabulary.

6. DEMONSTRATING THE NER/NEL'S FUNCTIONALITY

In this chapter, we demonstrate new forms of querying the augmented Greek legislation and Greek geographical landmarks datasets.

6.1 Querying the augmented Greek legislation and Greek geographical landmarks datasets

6.1.1 Legislation citation networks

A legal professional may retrieve citation networks built around a legal document, which most likely include legal documents in the same context (see Figure 41).

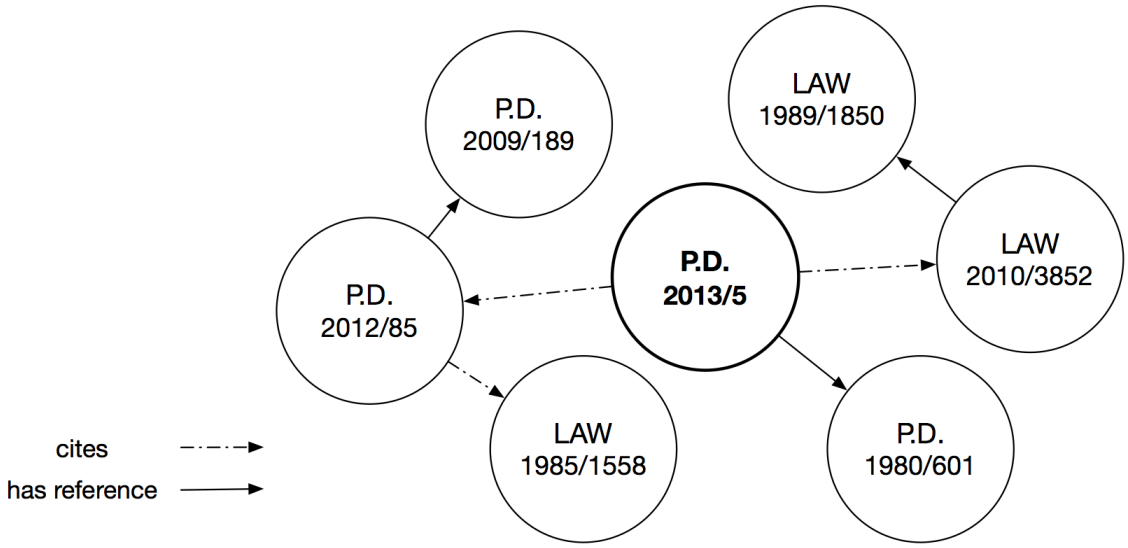


Figure 41: Citation network based on citations and references around Presidential Decree 2013/5.

6.1.2 Entity-based search

Based on the above, we have the ability to pose queries against the resulting RDF graph. Two sample queries are given in natural language together with their expression in SPARQL in Table 8.

Table 8: Entity-based queries.

Q1: *Retrieve any acts of legal documents that refer to local districts, which belong to the regional unit of Larissa.*

```
SELECT DISTINCT
?act_id ?local_district_name
WHERE {
 ?act eli:local_id ?act_id.
 ?act leg:has_reference ?reference.
 ?reference eli:relevant_for ?local_district.
 ?local_district rdfs:label ?local_district_name.
 ?local_district a landmark:LocalDistrict.
 ?local_district leg:belongs_to ?regional_unit.
 ?regional_unit rdfs:label
"REGIONAL UNIT OF LARISSA"@en.
} LIMIT 5
```

Act ID	Local District
Dec. 2015/1882	"RIGEOU"
Dec. 2015/1827/109821	"ROUMANI"
Dec. 2015/1629/99573	"LIKOVOUNI ST. CHARALAMPOU"
Dec. 2013/1002/65288	"KLARAKI"
Dec. 2013/1154/74937	"PALIOMANDRIA ST. CHARALAMPOU"

Q2: *Retrieve any acts of legal documents that contain references to persons that have been born in Athens.*

```
SELECT DISTINCT
?act_id ?person_name
WHERE {
 ?act eli:local_id ?act_id.
 ?act leg:has_reference ?reference.
 ?reference eli:relevant_for ?person.
 ?person rdfs:label ?person_name.
 ?person dbpedia:birthplace ?birthplace.
 ?birthplace rdfs:label "Athens"@en.
} LIMIT 5
```

Act ID	Local District
Dec. 2014/16591/943	"KIRIAKOS K. MITSOTAKIS"
P.D. 2002/73	"KONSTANTINOS STEFANOPOULOS"
Dec. 2011/23564	"LOUKAS PAPADIMOS"
Dec. 2015/Y58	"ALEXIS TSIPRAS"
Dec. 2009/1059423	"GIANNIS PAPATHANASIOU"

7. CONCLUSION AND FUTURE WORK

All in all, we developed, tested and evaluated a Named Entity Recognition and a Named Entity Linking component, applied on greek legislation. Greek is a challenging language for NLP tasks, while the additional noise from external sources (since the original corpus of documents is only available in PDF format) provided an interesting challenge to tackle.

Regarding the NER component, we evaluated all of the above LSTM-based methods in the task of Named Entity Recognition in a Greek legislation dataset, which we made publicly available for further academic research. The process was challenging and lengthy, as we had to convert PDF files into TXT format, process them so that they are suitable for training, manually annotate a subset of the documents to generate the test, train and validation parts of the datasets, before being able to conduct our experiments. As reported above, our experiments yielded some interesting and even unexpected findings.

Regarding the NEL component, we evaluated entity-linking between textual references and entities from open third-party datasets. Obtaining links is important as we can complement the information of the entities extracted from the text with their corresponding/dedicated matches on popular datasets.

Finally, we introduced and applied a novel vocabulary for the representation of textual references and we generated a new dataset for Greek geographical landmarks. As explained before, rural/architectural information of this kind has never been extracted into a dataset of any kind, therefore it is a significant contribution as it provides us with numerous capabilities.

Our future plans include further experimentation on the LSTM-based methods using word embeddings trained with the `FastText` algorithm, which considers sub-words information. We consider that it would be beneficial based on the fact that the Greek language includes multiple declensions in the indication of numbers, cases (nominative, subjective, genitive, possessive), and genders. For the same reasons, we are also planning to replace the shape embeddings with a dynamic character-level RNN or CNN model, to embed information relevant to token shapes, prefixes, suffixes, as described by Ma and Hovy [26].

A character-level RNN or CNN model will also be examined as an alternative method (operation) for entity linking.

Another interesting potential direction is the introduction of a more complicated annotation format with richer sets of labels, based on the principles of BIO tags, in order to address the complexity of the legal text. We also endeavour to extract (recognize) more geospatial information such as coordinates, presented in tables, or extracting relations between landmarks to augment the information in the newly generated dataset.

ABBREVIATIONS - ACRONYMS

AI	Artificial Intelligence
BILSTM	Bidirectional Long Short-Term Memory
BIO	Beginning-Inside-Outside
BPTT	Back Propagation Through Time
CBOW	Continuous Bag of Words
CEC	Constant Error Carousel
CLTK	Classical Language Toolkit
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CRF	Conditional Random Fields
ECLI	European Case Law Identifier
ELI	European Legislation Identifier
EU	European Union
FAIR	Facebook AI (Artificial Intelligence) Research
FN	False Negatives
FP	False Positives
GAG	Greek Administrative Geography
GPE	Geo-Political Entity
GPU	Graphics Processing Unit
GloVe	Global Vectors
IETF	Internet Engineering Task Force
IO	Inside-Outside
IR	Information Representation
IoT	Internet of Things
IRI	Internationalized Resource Identifier
ISO	International Organization for Standardization

LKIF	Legal Knowledge Interchange Format
LR	Linear Regression
LSL	Link Specification Language
LSTM	Long Short-Term memory
NEL	Named Entity Linker
NER	Named Entity Recognition
NGO	Non-governmental Organization
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
OCR	Optical Character Recognition
OWL	Web Ontology Language
PDF	Portable Document Format
POS	Part Of Speech
PPV	Positive Predicted Value
PhD	Philosopher's Doctorate
RBM	Random Boltzmann Machine
RDF	Resource Description Framework
RNN	Recurrent Neural Network
ReLU	Rectified Linear Unit
SGD	Stochastic Gradient Descent
SPARQL	SPARQL Protocol and RDF Query Language
SVM	Support Vector Machine
TDNN	Time-Delay Neural Network
TFxIDF	Term Frequency times Inverse Document Frequency
TN	True Negatives
TP	True Positives
TPR	True Positive Rate
TXT	text

UK	United Kingdom
URI	Uniform Resource Identifier
UTF	Unicode Transformation Format
W2V	Word2Vec
W3C	World Wide Web Consortium
XOR	Exclusive OR
brat	brat rapid annotation tool

BIBLIOGRAPHY

[1] K. Ashley, *Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age*. Cambridge University Press, 2017. 17, 43

[2] M. Kim and R. Goebel, "Two-step cascaded textual entailment for legal bar exam question answering," in *Proceedings of the 4th Competition on Legal Information Extraction/Entailment*, (London, UK), 2017. 17, 43

[3] I. Chalkidis, I. Androutsopoulos, and A. Michos, "Extracting contract elements," in *Proceedings of the 16th International Conference on Artificial Intelligence and Law*, (London, UK), pp. 19–28, 2017. 17, 20, 23, 43, 75, 85

[4] I. Chalkidis and I. Androutsopoulos, "A deep learning approach to contract element extraction," in *Proceedings of the 30th International Conference on Legal Knowledge and Information Systems*, (Luxembourg), pp. 155–164, 2017. 17, 20, 22, 43, 75, 80

[5] W. Alschnerd and D. Skougarevskiy, "Towards an automated production of legal texts using recurrent neural networks," in *Proceeding of the 16th International Conference on Artificial Intelligence and Law*, (London, UK), pp. 159–168, 2017. 17, 43

[6] J. Dann, "European Legislation Identifier "ELI"," tech. rep., European Commission, 2014. 17, 43

[7] ELI Task Force, *ELI - A technical implementation guide*, 2015. 17, 43

[8] ELI Task Force, *ELI implementation methodology: Good practices and guidelines*, 2015. 17, 43

[9] M. V. Opijnen, "European Case Law Identifier: Indispensable Asset for Legal Information Retrieval," in *From Information to Knowledge* (M. A. Biasiotti and S. Faro, eds.), vol. 236 of *Frontiers in Artificial Intelligence and Applications*, pp. 91–103, IOS Press, 2011. 17, 43

[10] T. Agnoloni, L. Bacci, G. Peruginelli, M. van Opijnen, J. van den Oever, M. Palmirani, L. Cervone, O. Bujor, A. A. Lecuona, A. B. García, L. D. Caro, and G. Siragusa, "Linking european case law: Bo-ecli parser, an open framework for the automatic extraction of legal links," in *JURIX*, 2017. 17, 43

[11] J. Breuker, R. Hoekstra, A. Boer, K. van den Berg, R. Rubino, G. Sartor, M. Palmirani, A. Wyner, and T. Bench-Capon, "OWL ontology of basic legal concepts (LKIF-core), deliverable 1.4," tech. rep., ESTRELLA, 2007. 17, 43

[12] R. Hoekstra, J. Breuker, M. Di Bello, and A. Boer, "Lkif core: Principled ontology development for the legal domain," in *Proceedings of the 2009 Conference on Law, Ontologies and the Semantic Web: Channelling the Legal Information Flood*, (Amsterdam, The Netherlands, The Netherlands), pp. 21–52, IOS Press, 2009. 17, 43

[13] T. Athan, H. Boley, G. Governatori, M. Palmirani, A. Paschke, and A. Wyner, "OASIS LegalRuleML," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law*, (Rome), pp. 3–12, 2013. 17, 43

[14] T. Athan, G. Governatori, M. Palmirani, A. Paschke, and A. Z. Wyner, "Legalruleml: Design principles and foundations," in *Reasoning Web. Web Logic Rules*, 2015. 17, 43

[15] I. Chalkidis, C. Nikolaou, P. Soursos, and M. Koubarakis, "Modeling and querying greek legislation using semantic web technologies," in *The Semantic Web - 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28 - June 1, 2017, Proceedings, Part I*, pp. 591–606, 2017. 17, 43

[16] C. B. Robert Isele, Anja Jentzsch, "Silk server - adding missing links while consuming linked data," in *1st International Workshop on Consuming Linked Data*, (Shanghai, China), 2010. 17, 43

[17] C. B. Anja Jentzsch, Robert Isele, "Silk - generating rdf links while publishing or consuming linked data," in *International Semantic Web Conference*, (Shanghai, China), 2010. 17, 43

[18] P. Stenetorp, S. Pyysalo, G. Topic, T. Ohta, S. Ananiadou, and J. Tsujii, "brat: a web-based tool for nlp-assisted text annotation," in *EACL 2012, 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France, April 23-27, 2012* (W. Daelemans, M. Lapata, and L. Màrquez, eds.), pp. 102–107, The Association for Computer Linguistics, 2012. 18, 20, 71, 76

[19] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *CoRR*, vol. abs/1310.4546, 2013. 21, 59, 60, 79

[20] T. Mikolov, W. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA* (L. Vanderwende, H. D. III, and K. Kirchhoff, eds.), pp. 746–751, The Association for Computational Linguistics, 2013. 21, 59, 79

[21] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *CoRR*, vol. abs/1607.04606, 2016. 22, 59, 63, 79

[22] D. Jurafsky, *Speech and language processing: An introduction to natural language processing*. Prentice Hall, 2018. 22, 80

[23] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010* (Y. W. Teh and D. M. Titterington, eds.), vol. 9 of *JMLR Proceedings*, pp. 249–256, JMLR.org, 2010. 22, 57, 85

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations*, (San Diego, CA), 2015. 22, 85

[25] G. Stoilos, G. Stamou, and S. Kollias, "A string metric for ontology alignment," in *4th International Semantic Web Conference*, (Galway, Ireland), pp. 624–637, 2005. 25, 89

[26] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional LSTM-cNNs-CRF," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, (Berlin, Germany), pp. 1064—1074, 2016. 29, 59, 75, 95

[27] M. Minsky and S. Papert, *Perceptrons - an introduction to computational geometry*. MIT Press, 1987. 45

[28] B. Widrow *et al.*, "Adaptive" adaline" neuron using chemical" memistors."," *Technical Report No. 1553-2*, 1960. 47, 48

[29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, p. 533, 1986. 48

[30] Y. LeCun, L. Bottou, G. B. Orr, and K. Müller, "Efficient backprop," in *Neural Networks: Tricks of the Trade - Second Edition* (G. Montavon, G. B. Orr, and K. Müller, eds.), vol. 7700 of *Lecture Notes in Computer Science*, pp. 9–48, Springer, 2012. 48

[31] S. Linnainmaa, "The representation of the cumulative rounding error of an algorithm as a taylor expansion of the local rounding errors," *Master's Thesis (in Finnish), Univ. Helsinki*, pp. 6–7, 1970. 48

[32] P. Werbos, "Beyond regression: new fools for prediction and analysis in the behavioral sciences," *PhD thesis, Harvard University*, 1974. 48

[33] K. Hornik, M. B. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989. 48

[34] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989. 48

[35] A. H. Waibel, T. Hanazawa, G. E. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 37, no. 3, pp. 328–339, 1989. 50

[36] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990. 51

[37] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen netzen," *Diploma, Technische Universität München*, vol. 91, p. 1, 1991. 52

[38] Y. Bengio, P. Y. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994. 52

[39] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. 52

[40] Y. Lecun, L. Jackel, L. Bottou, A. Brunot, C. Cortes, J. Denker, H. Drucker, I. Guyon, U. Muller, E. Sackinger, P. Simard, and V. Vapnik, "Comparison of learning algorithms for handwritten digit recognition," in *International Conference on Artificial Neural Networks, Paris* (F. Fogelman and P. Gallinari, eds.), pp. 53–60, EC2 & Cie, 1995. 54

[41] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, 2012. 55

[42] G. E. Hinton, “Training products of experts by minimizing contrastive divergence,” *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002. 56

[43] A. Mohamed, T. N. Sainath, G. E. Dahl, B. Ramabhadran, G. E. Hinton, and M. A. Picheny, “Deep belief networks using discriminative features for phone recognition,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011, May 22-27, 2011, Prague Congress Center, Prague, Czech Republic*, pp. 5060–5063, IEEE, 2011. 56

[44] R. Raina, A. Madhavan, and A. Y. Ng, “Large-scale deep unsupervised learning using graphics processors,” in *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009* (A. P. Danyluk, L. Bottou, and M. L. Littman, eds.), vol. 382 of *ACM International Conference Proceeding Series*, pp. 873–880, ACM, 2009. 56

[45] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, “What is the best multi-stage architecture for object recognition?,” in *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*, pp. 2146–2153, IEEE Computer Society, 2009. 57

[46] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel* (J. Fürnkranz and T. Joachims, eds.), pp. 807–814, Omnipress, 2010. 57

[47] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011* (G. J. Gordon, D. B. Dunson, and M. Dudík, eds.), vol. 15 of *JMLR Proceedings*, pp. 315–323, JMLR.org, 2011. 57

[48] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *Proc. icml*, vol. 30, p. 3, 2013. 57

[49] D. P. Kingma, T. Salimans, and M. Welling, “Variational dropout and the local reparameterization trick,” *CoRR*, vol. abs/1506.02557, 2015. 58

[50] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in *Proceedings of the 18th International Conference on Machine Learning*, (Williamstown, MA), pp. 282–289, 2001. 59

[51] J. Peng, L. Bo, and J. Xu, “Conditional neural fields,” in *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada.* (Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, eds.), pp. 1419–1427, Curran Associates, Inc., 2009. 59

[52] K. Yao, B. Peng, G. Zweig, D. Yu, X. Li, and F. Gao, “Recurrent conditional random field for language understanding,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, (Florence, Italy), pp. 4077–4081, 2014. 59, 84

[53] Z. Huang, W. Xu, and K. Yu, “Bidirectional LSTM-CRF models for sequence tagging,” *CoRR*, vol. abs/1508.01991, 2015. 59

[54] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (San Diego, California), pp. 260–270, 2016. 59, 75

[55] D. Andor, C. Alberti, D. Weiss, A. Severyn, A. Presta, K. Ganchev, S. Petrov, and M. Collins, "Globally normalized transition-based neural networks," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (long papers)*, (Berlin, Germany), pp. 2442–2452, 2016. 59, 75, 84

[56] K. W. Church and P. Hanks, "Word association norms, mutual information and lexicography," in *27th Annual Meeting of the Association for Computational Linguistics, 26-29 June 1989, University of British Columbia, Vancouver, BC, Canada, Proceedings.* (J. Hirschberg, ed.), pp. 76–83, ACL, 1989. 60

[57] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *arXiv preprint arXiv:1607.01759*, 2016. 63

[58] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL* (A. Moschitti, B. Pang, and W. Daelemans, eds.), pp. 1532–1543, ACL, 2014. 64

[59] T. Wang and K. Cho, "Larger-context language modelling with recurrent neural network," in *54th Annual Meeting of the Association for Computational Linguistics*, (Berlin, Germany), pp. 1319–1329, 2016. 75

[60] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, and G. Kurian, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016. 75, 82

[61] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. B. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *CoRR*, vol. abs/1611.04558, 2016. 75

[62] D. Tang, B. Qin, X. Feng, and T. Liu, "Effective lstms for target-dependent sentiment classification," in *26th International Conference on Compu- tational Linguistics: Technical Papers*, 2016. 75

[63] O. Irsoy and C. Cardie, "Deep recursive neural networks for compositionality in language," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, (Montreal, Canada), pp. 2096–2104, 2014. 82