National Technical University of Athens
School of Electrical and Computer Engineering
Division of Computer Science

Automatic Summarization of Court Judgements using Machine Learning

with applications to summarizing Greek Court Judgements

Diploma Thesis

of

DIMITRIS P. GALANIS

Supervisor: Panagiotis Tsanakas
Professor

Athens, October 2022

National Technical University of Athens
School of Electrical and Computer Engineering
Division of Computer Science

Automatic Summarization of Court Judgements using Machine Learning

with applications to summarizing Greek Court Judgements

Diploma Thesis

of

DIMITRIS P. GALANIS

Supervisor: Panagiotis Tsanakas
Professor

Approved by the examination committee on 11th November 2022.

(Signature)	*(Signature)*	*(Signature)*
.....	.....	.....
Panagiotis Tsanakas	Vasiliki (Verena) Kadere	Eugenia Tzannini
Professor	Assistant Professor	Assistant Professor

Athens, October 2022

National Technical University of Athens
School of Electrical and Computer Engineering
Division of Computer Science

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism.

(Signature)

.....

Dimitris P. Galanis
Electrical & Computer Engineer, NTUA
31th October 2022

Abstract

The rapid increase of digitized text documents has accentuated the need for reliable automatic methods that discern the important information from the unimportant. In the legal domain of court judgements, this process is done mostly manually by specialized legal editors, which is a time-consuming process. However, court judgement summaries are an essential part of a legal practitioner's workflow, as they are shorter in length, thus enabling faster and more specific search for relevant case-laws. Furthermore, summarized versions of court judgements allow the legal practitioner to intuitively focus on its main points and thus acquire a better understanding of it.

Recent advances in Machine Learning have enabled better performance in Automatic Text Summarization (ATS) systems, in terms of automatic evaluation metrics. Moreover, deep pre-trained Language Models enable the use of ATS without large amounts of training data. However, most methods are trained and evaluated for the news-article domain, which differs from the court-judgements domain as the latter includes longer documents, having significantly different structure and making use of specialized legal terminology.

In our work, we attempt to automatically summarize Greek court judgements using machine learning methods. To that end, we first conduct an extended survey of the automatic text summarization literature; the methods, the datasets and evaluation metrics used and the criticism that has been applied to them. Then we proceed by constructing a dataset of Greek court judgement texts and their summaries. We build an *extractive summarization* system, based on the LexRank algorithm, that extracts the most important sentences from a judgement. We train an Encoder-Decoder Deep Learning model based on the BERT architecture, using open-sourced checkpoints trained on Greek parliamentary corpora and use it to model *abstractive summarization* as a sequence generation task. We evaluate our methods using the ROUGE-family of automatic evaluation metrics and also conduct a human evaluation study.

We show that domain informed preprocessing and including judgement classification information can increase the performance of our *abstractive summarization* methods. We provide a comparison of different variations of our *extractive summarization* methods. Legal experts' evaluation shows our extractive methods perform average, and our abstractive methods, while generating moderately fluent and coherent text, have low scores in the relevance and consistency metrics, indicating the need of methods factually aligned to the judgement text.

Keywords: Automatic Text Summarization, Court Judgements, Machine Learning, Neural Networks, Natural Language Processing, BERT, ROUGE, Legal-AI

Περίληψη

Η ταχεία αύξηση των ψηφιοποιημένων κειμένων έχει εντείνει την ανάγκη για αξιόπιστες αυτόματες μεθόδους που μπορούν να ξεχωρίσουν τις σημαντικές από τις μη-σημαντικές πληροφορίες. Στην νομολογία, η διαδικασία αυτή είναι χρονοβόρα καθώς γίνεται κυρίως μη-αυτοματοποιημένα από εξειδικευμένους νομικούς συντάκτες. Όμως, οι περιλήψεις δικαστικών αποφάσεων είναι απαραίτητο κομμάτι της ροής εργασίας ενός νομικού, αφού λόγω του μικρότερου μήκους δίνουν την δυνατότητα για γρηγορότερη και πιο στοχευμένη αναζήτηση σχετικής νομολογίας και πληρέστερη κατανόηση των κεντρικών σημείων τους. Πρόσφατες εξελίξεις στον χώρο της Μηχανικής Μάθησης, δίνουν την δυνατότητα για καλύτερες επιδόσεις στα συστήματα Αυτόματης Περίληψης Κειμένου (ΑΠΚ), βάσει αυτομάτων μετρικών αξιολόγησης. Επιπλέον, τα βαθιά προ-εκπαιδευμένα Γλωσσικα Μοντέλα επιτρέπουν την χρήση ΑΠΚ συστημάτων χωρίς μεγάλο αριθμό δεδομένων με εφαρμογές κυρίως σε άρθρα ειδήσεων τα οποία, όμως, έχουν μικρότερο μέγεθος, διαφορετική δομή και ελάχιστη νομική ορολογία.

Στην παρούσα εργασία, προσπαθούμε να παραγάγουμε αυτόματες περιλήψεις Ελληνικών δικαστικών αποφάσεων χρησιμοποιώντας μεθόδους Μηχανικής Μάθησης. Για αυτό τον σκοπό, διεξάγουμε εκτεταμένη βιβλιογραφική μελέτη των μεθόδων Αυτόματης Περίληψης Κειμένου, των συνόλων δεδομένων και των μετρικών αξιολόγησης που χρησιμοποιούνται. Στην συνέχεια, συλλέγουμε ένα σύνολο δεδομένων αποτελούμενο από Ελληνικές Δικαστικές Αποφάσεις. Κατασκευάζουμε ένα σύστημα *εξαγωγικής* περίληψης, βασιζόμενο στον αλγόριθμο LexRank, το οποίο εξάγει από τις αποφάσεις τις πιο σημαντικές προτάσεις. Εκπαιδεύουμε ένα μοντέλο Κωδικοποιητή-Αποκωδικοποιητή Βαθιάς Μάθησης που βασίζεται στην αρχιτεκτονική BERT, χρησιμοποιώντας προ-εκπαιδευμένες σε Ελληνικά νομικά κείμενα παραμέτρους, που διατίθενται ελεύθερα, και το χρησιμοποιούμε για την μοντελοποίηση του προβλήματος της *ελεύθερης περίληψης* σαν ένα πρόβλημα παραγωγής ακολουθίας κειμένου. Οι μέθοδοί μας αξιολογούνται κάνοντας χρήση της οικογένειας αυτομάτων μετρικών ROUGE και μέσω μελέτης ανθρώπινης αξιολόγησης από νομικούς.

Δείχνουμε ότι η εξειδικευμένη για δικαστικές αποφάσεις προ-επεξεργασία κειμένου και η συμπερίληψη πληροφορίας κατηγοριοποίησης των δικαστικών αποφάσεων βελτιώνει την επίδοση των μεθόδων μας *ελεύθερης περίληψης*. Παρέχουμε μια μελέτη αξιολόγησης διαφόρων παραλλαγών των μεθόδων *εξαγωγικής περίληψης*. Η αξιολόγηση από νομικούς δείχνει πως οι *εξαγωγικές* μέθοδοι αποδίδουν μέτρια, ενώ οι μέθοδοι *ελεύθερης* περίληψης παράγουν περιλήψεις μέτριας ευφράδειας και συνοχής αλλά χαμηλής σχετικότητας και συνέπειας με το κείμενο της δικαστικής απόφασης, υποδεικνύοντας την ανάγκη για μεθόδους περίληψης που συμφωνούν πραγματολογικά με το προς περίληψη κείμενο.

Λέξεις Κλειδιά: Αυτόματη Περίληψη Κειμένου, Δικαστικές Αποφάσεις, Μηχανική Μάθηση, Νευρωνικά Δίκτυα, Επεξεργασία Φυσικής Γλώσσας, Τεχνητή Νοημοσύνη και Δίκαιο

To my laptop, Dell Inspiron 5737-4417

Ευχαριστίες

Θα ήθελα πρωτίστως να ευχαριστήσω τον Δρ. Μάριο Κόνιαρη για την πολύτιμη βοήθεια του κατά την εκπόνηση της εργασίας αυτής. Η ενθάρρυνση, τα σχόλια και οι παρατηρήσεις του ήταν καθοριστικές στην εκπόνηση της εργασίας. Επίσης, θα ήθελα να ευχαριστήσω τον καθηγητή κ. Παναγιώτη Τσανάκα για το ότι μου έδωσε την δυνατότητα να ασχολήθώ με αυτό το θέμα, αλλά και για την εξαιρετική συνεργασία μας κατά την επίβλεψη της εργασίας αυτής.

Η οικογένειά μου μού έδωσε τα μέσα, την άνεση, τον χώρο και την ψυχική ενθάρρυνση που χρειαζόμουν για να ολοκληρώσω την φοίτησή μου στην σχολή. Τους ευχαριστώ απεριόριστα για αυτό.

Νιώθω υποχρέωση να ευχαριστήσω ταπεινά την παρέα που με ανέχτηκε καθ'όλα τα χρόνια των σπουδών μου. Μαζί προβληματιστήκαμε και κάναμε του εαυτούς μας καλύτερους, ακαδημαϊκά αλλά και ευρύτερα. Τα περισσότερα πράγματα, από τα λίγα που έμαθα τα χρόνια αυτά, τα έμαθα χάρις σε αυτήν.

Τέλος, θα ήθελα να δώσω τις πιο εγκάρδιες ευχαριστίες μου στο προσωπικό μου λάπτοπ Dell Inspiron 5737-4417. Κατά τη διάρκεια των σπουδών μου ενηλικιωθήκαμε και τελικά γεράσαμε μαζί. Υπό τον ήχο του χαλασμένου ανεμιστήρα σου λύθηκαν και γράφτηκαν όλες οι ασκήσεις της σχολής. Όλα αυτά, πάντα δίπλα από μια μπρίζα, διότι η μπαταρία σου στέρεψε.

Αθήνα, Οκτώβριος 2022

Contents

List of Figures

List of Tables

Εκτεταμένη Ελληνική Περίληψη

0.1 Εισαγωγή

Ζούμε στην εποχή της πληροφορίας, όπου η ποσότητα της προσλαμβάνουσας πληροφορίας και η ανάγκη επεξεργασίας της αυξάνονται σταθερά. Όλο και περισσότερα κείμενα ψηφιοποιούνται και ενσωματώνονται σε διάφορες εφαρμογές όπου ο χρήστης μπορεί να κατεβάσει, ψάξει και επεξεργαστεί μεγάλο πλήθος δεδομένων. Πολλές εταιρείες Τεχνητής Νοημοσύνης (ΤΝ) εκμεταλλεύονται τα αυξανόμενα σε πλήθος δεδομένα για να εκπαιδεύσουν αλγορίθμους Τεχνητής Νοημοσύνης οι οποίοι μπορούν να εφαρμοστούν σε διάφορους τομείς. Όμως, τα δεδομένα είναι χρήσιμα - για τις μηχανές και για τους ανθρώπους - μόνο αν μπορούν να χρησιμοποιηθούν είτε απευθείας σε κάποια άλλη εφαρμογή μετά από επεξεργασία. Επομένως, η περίληψη ενός κειμένου - όπως ορίζεται από τον Radev: «ένα κείμενο που παράγεται από ένα ή περισσότερα άλλα κείμενα και το οποίο περιέχει την σημαντικά πληροφορία από αυτό/ά χωρίς να έχει πάνω από το μισό μέγεθός του(ς)» [110] - μπορεί να είναι πολύ χρήσιμη αφού μειώνει σημαντικά το μέγεθος του κειμένου χωρίς να χαθεί το νόημά του, καθιστώντας έτσι εφικτή την χρήση μεγαλύτερων κειμένων.

Η ερευνητική κοινότητα στον χώρο της Τεχνητής Νοημοσύνης έχει διαχρονικά δείξει ενδιαφέρον στο να βρεθούν μέθοδοι Αυτόματης Περίληψης Κειμένου (ΑΠΚ), κυρίως για κείμενα ειδήσεων. Στην προσπάθεια αυτή έχουν χρησιμοποιηθεί πολλές μέθοδοι προερχόμενες από την Υπολογιστική Γλωσσολογία μέχρι την Μηχανική Μάθηση. Οι περισσότερες μέθοδοι προσπαθούσαν να εξάγουν τις πιο σημαντικές προτάσεις του κειμένου (*εξαγωγική περίληψη*), αφού το να παραχθεί νέο κείμενο περίληψης συναρτήσει του αρχικού κειμένου (*ελεύθερη περίληψη*) δεν είναι ήταν δυνατόν με συμβατικές μεθόδους Μηχανικής Μάθησης ή Υπολογιστικής Γλωσσολογίας. Εντούτοις, η πρόσφατη αύξηση του ερευνητικού ενδιαφέροντος για τον χώρο των Δικτύων Βαθιάς Μηχανικής Μάθησης - η οποία ήρθε ως αποτέλεσμα της αυξανόμενες υπολογιστικής ισχύος των σύγχρονων υπολογιστών και των Μεγάλων Δεδομένων (Big Data) - έχει δώσει την δυνατότητα κατασκευής μεθόδων με καλύτερες ικανότητες Παραγωγής Γλώσσας (Language Generation) καθιστώντας την *ελεύθερη περίληψη* εφικτή. Τα Δίκτυα Βαθιάς Μηχανικής Μάθησης εκπαιδεύονται κάνοντας χρήση μεγάλου πλήθους κειμενικών δεδομένων και, συχνά, ο κώδικας αλλά και τα ίδια τα δίκτυα, διατίθενται σαν λογισμικό *ανοικτού κώδικα*, δίνοντας σε περισσότερα άτομα την δυνατότητα να χρησιμοποιήσουν προ-εκπαιδευμένα Γλωσσικά Μοντέλα σε επιμέρους εφαρμογές, όπως η περίληψη κειμένου. Πρόσφατες εξελίξεις στις αρχιτεκτονικές δικτύων μηχανικής μάθησης, όπως η αρχιτεκτονική Transformer [137], είναι πιο αποδοτικές υπολογιστικά και επομένως μπορούν να εκπαιδευτούν με μεγαλύτερο πλήθος δεδομένων. Η αρχιτεκτονική Γλωσσικού Μοντέλου BERT (Bidirectional Encoder Representations from Transformers) [29], η οποία βασίζεται στην

αρχιτεκτονική Transformer, έχει βρει εφαρμογή σε πλήθος ερευνητικών και βιομηχανικών τομέων.

Στην παρούσα εργασία, αξιολογούμε τους αλγορίθμους αυτόματης περίληψης κειμένου σε δικαστικές αποφάσεις. Ο τομέας της νομολογίας, συγκριτικά με άλλους τομείς, έχει υπάρξει αργός στο να εκμεταλλευτεί τις ωφέλειες της ψηφιοποίησης των κειμένων. Όμως, τα τελευταία χρόνια, όλο και περισσότερα δικαστήρια ψηφιοποιούν τις αποφάσεις τους και τις διαθέτουν ελεύθερα μέσω του Διαδικτύου, δίνοντας έτσι την δυνατότητα σε νομικούς, δικαστές και δικηγόρους να έχουν πρόσβαση σε τεράστιο πλήθος δικαστικών αποφάσεων. Εντούτοις, ένας νομικός ο οποίος ψάχνει παλιές δικαστικές αποφάσεις, πρέπει να εξετάσει εξαντλητικά ποιες του είναι χρήσιμες ώστε από αυτές - μέσω των νομικών του γνώσεων αλλά και εμπειρίας - να εξάγει τα κειμενικά αποσπάσματα που του είναι χρήσιμα και να κατανοήσει εις βάθος την εφαρμοστέα νομολογία. Οι περιλήψεις δικαστικών αποφάσεων συγκεντρώνουν τα κυρίως σημεία της αποφάσης σε κείμενο μικρότερου μεγέθους, δίνοντας έτσι την δυνατότητα πιο αποδοτικής (σε θέμα χρόνου) αναζήτησης σχετικής νομολογίας.

Μέχρι την δεκαετία του 90, το μεγαλύτερο μέρος της περιλήψεων νομικών κειμένων αλλά και της διαδικασίας εξαγωγής πληροφορίας από νομικά κείμενα, γίνονταν από εξειδικευμένους νομικούς συντάκτες. Όμως έκτοτε, λογισμικά φιλικά προς τους χρήστες [109, 36] έχουν αναπτυχθεί ώστε αυτές οι διαδικασίες να μπορούν να αυτοματοποιηθούν μερικώς. Στις μέρες μας, πάνω από 2000 νεοφυείς επιχειρήσεις δραστηριοποιούνται στον χώρο της Τεχνητής Νοημοσύνης για Νομικά Κείμενα [130].

0.1.1 Νομικά Κείμενα

Τα νομικά κείμενα είναι κείμενα που έχουν συνταχθεί για διαφόρους σκοπούς, αλλά όλα σχετίζονται σε κάποιο βαθμό με τον νόμο - είτε λόγω της αρμοδιότητας του συντάκτη (π.χ Δικαστής, Νομοθέτης), λόγω των αναφορών που περιέχουν σε άλλα νομικά κείμενα, ή λόγω της θεματολογίας τους που αφορά την ρύθμιση των δικαιωμάτων και των υποχρεώσεων ιδιωτών και θεσμών. Οι πιο συχνοί τύποι νομικών κειμένων [133], ιεραρχημένοι βάσει της ισχύος τους ως πηγών δικαίου [131], είναι οι παρακάτω:

- **Συντάγματα:** τα οποία αποτελούν τις θεμελιώδεις αρχές που αφορούν το πώς διοικείται ένα κράτος.

- **Νομοθεσία:** η οποία περιλαμβάνει τους νόμους τους οποίους θεσπίζει ένα νομοθετικό σώμα και ρυθμίζουν τι είναι επιτρεπτό από τον νόμο. Συνήθως, οι νόμοι οργανώνονται θεματικά σε κώδικες παρόμοιας θεματολογίας, όπως επί παραδείγματι ο Ποινικός Κώδικας.

- **Δικαστικές αποφάσεις:** οι οποίες περιλαμβάνουν τα 1) ενδιάμεσα ή τελικά αποτελέσματα μιας δίκης, 2) την κρίση του δικαστηρίου για τα γεγονότα και τα επιχειρήματα της κάθε πλευράς καθώς και των εφαρμοστέων νόμων, 3) την απόφαση του δικαστηρίου σε γραπτή μορφή.

- **Συμβόλαια:** τα οποία είναι αμοιβαίες συμφωνίες μεταξύ συμβαλλόμενων μερών, μέσω των οποίων παράγονται αμοιβαίες υποχρεώσεις.

Τα νομικά κείμενα έχουν εξελιχθεί σημαντικά μέσα στις χιλιετίες που παρήλθαν από την πρώτη χρήση τους. Τα πρώτα κείμενα ιδιωτικού δικαίου ήταν συμβόλαια, διαθήκες και νομικές πράξεις εγγεγραμμένες σε πινακίδες από πηλό στην Σουμερία περίπου 5000 χρόνια πριν. Παρομοίως, τα πρώτα κείμενα δημοσίου δικαίου, όπως νόμοι, εμφανίστηκαν στην Μεσοποταμία με τους νόμους του βασιλιά Ουρ Ναμμού και αργότερα τον κώδικα του Χαμουραμπί να αποτελούν γνωστά παραδείγματα.

Στην Αρχαία Ελλάδα τα νομικά κείμενα μεταξύ ιδιωτών αφορούσαν θέματα όπως κληρονομιά, εμπόριο και λοιπά συμβόλαια, ενώ τα κείμενα που ρύθμιζαν πώς οι αρχαίοι Έλληνες *πολίτες* ζούσαν σε κάθε Πόλη-κράτος (*πολιτεία*) διέφεραν από περιοχή σε περιοχή λόγω των διαφορών στην επικράτηση της Ρητορικής τέχνης, το πόσο διαδεδομένη ήταν η νομομάθεια και τα δικαστήρια [141] και άλλων κοινωνικο-πολιτικών λόγων που ρύθμιζαν ποιος θεωρούνταν πολίτης. Γνωστοί νόμοι περιλαμβάνουν του νόμους του Δράκοντα (620 π.Χ), γνωστοί για την αυστηρότητά τους και οι οποίοι μεταρρυθμίστηκαν από τους νόμους του Σόλωνος (593 π.Χ).

Η νομική παράδοση της Ρωμαϊκής Αυτοκρατορίας, αν και αρχικά βασίστηκε στην Αρχαία Ελληνική, στην συνέχεια εξελίχθηκε ανεξάρτητα, με την επιρροή της να αναγράφεται έντονα στο δικαιϊκό σύστημα της Βυζαντινής Αυτοκρατορίας αλλά και πολλών σύγχρονων κρατών. Τα σημερινά νομικά κείμενα διαφέρουν στην δομή τους και στο πως αναφέρουν άλλα κείμενα. Επίσης κάνουν χρήση επίσημης γλώσσας και νομικής ορολογίας.

Σε κάθε περίπτωση, η εύρυθμη λειτουργία των σύγχρονων κοινωνιών προϋποθέτει μια κουλτούρα σεβασμού των κανόνων και των θεσμών που επιβλέπουν την τήρησή τους (*κράτος δικαίου*). Μάλιστα, θα μπορούσε να υποστηριχτεί πως ο σύγχρονος πολιτισμός, όπως τον γνωρίζουμε, βασίζεται στο κράτος δικαίου. Επομένως, οι νόμοι αλλά και η ερμηνεία τους, μέσω των δικαστικών αποφάσεων, έχουν τεράστια βαρύτητα για την ρύθμιση της καθημερινής ζωής του κάθε πολίτη και υπαρξιακή σημασία για τον πολιτισμό κάθε λαού.

0.1.2 Κίνητρο και Συνεισφορά Εργασίας

Τα κίνητρα στο να διερευνήσουμε τρόπους αυτόματης περίληψης κειμένων δικαστικών αποφάσεων στα ελληνικά είναι τα παρακάτω:

- Τα νομικά κείμενα έχουν μεγάλες διαφορές με τα κείμενα που χρησιμοποιούνται συνήθως στην ΑΠΚ: τα άρθρα ειδήσεων. Είναι μεγαλύτερα σε μέγεθος, περιέχουν νομική ορολογία και πολλές αναφορές σε άλλα κείμενα. Μέθοδοι που δεν λαμβάνουν υπόψιν αυτές τις διαφορές μπορούν να χάσουν σημαντικές πληροφορίες ή να μην είναι καν εφαρμόσιμες λόγω του μεγέθους των κειμένων.

- Ο μέσος νομικός ξοδεύει πολλές ώρες κάθε εβδομάδα στο να ψάχνει, να διαβάζει και να αναλύει δικαστικές αποφάσεις που είναι σχετικές με την υπόθεση που δουλεύει, Παράλληλα, στην Ελλάδα - με εξαίρεση πρόσφατες προσπάθειες μοντελοποίησης και εύκολης διάθεσης κειμένων στον χώρο της νομοθεσίας [66, 5] - δεν υπάρχουν συγκρίσιμα εργαλεία για Δικαστικές αποφάσεις. Αναπτύσσοντας και αξιολογώντας μεθόδους αυτόματης περίληψης Ελληνικών Δικαστικών Αποφάσεων, ευελπιστούμε πως θα παρέχουμε χρήσιμα εργαλεία και αποτελέσματα που θα βοηθήσουν την δημιουργία τέτοιων πλατφορμών.

Η συνεισφορά της εργασίας μας έγκειται στα παρακάτω:

- Δημιουργία αυτόματων προγραμμάτων συλλογής δεδομένων δικαστικών αποφάσεων από το Ελληνικό Συμβούλιο της Επικρατείας και τον Άρειο Πάγο. Διαθέτουμε τα δεδομένα ελεύθερα αφού τα οργανώσαμε σε σύνολο δεδομένων[1]. Το σύνολο δεδομένων αναλύεται και συγκρίνεται με άλλα σύνολα δεδομένων για ΑΠΚ, χρησιμοποιώντας μετρικές που έχουν προταθεί στην σχετική βιβλιογραφία.

- Υλοποιούμε μεθόδους *Εξαγωγικής Περίληψης* βασισμένες 1) στον αλγόριθμο LexRank [34] και 2) στην παραλλαγή του Biased LexRank [101], κάνοντας χρήση μεθόδων προ-επεξεργασίας κειμένου εξειδικευμένες στον τομέα των Δικαστικών Αποφάσεων.

- Υλοποιούμε μέθοδο *Ελεύθερης Περίληψης* κάνοντας χρήση της αρχιτεκτονικής Κωδικοποιητή - Αποκωδικοποιητή με BERT. Χρησιμοποιούμε βάρη για το BERT μοντέλο που έχουν προ-εκπαιδευτεί σε Ελληνικά νομικά κείμενα και την Ελληνική wikipedia και τα οποία έχουν διατεθεί ως ανοιχτός κώδικας [69]. Δοκιμάζουμε διάφορες μεθόδους προ-επεξεργασίας κειμένου εξειδικευμένες στις δικαστικές αποφάσεις, με σκοπό την συμπερίληψη όσο το δυνατόν περισσότερης σημαντικής πληροφορίας στο (περιορισμένου μήκους) κείμενο εισόδου του μοντέλου. Επιπλέον, αξιολογούμε την συνεισφορά της εισαγωγής των επισημειώσεων κατηγορίας (class tags) της δικαστικής απόφασης, στην παραγωγή της περίληψής της.

- Αξιολογούμε τους αλγορίθμους μας κάνοντας χρήση αυτόματων μετρικών αξιολόγησης περιλήψεων και διεξαγάγουμε μελέτη ανθρώπινης αξιολόγησης με νομικούς. Επίσης, μετράμε την συσχέτιση μεταξύ των δύο.

- Για τους σκοπούς της μελέτης ανθρώπινης αξιολόγησης και της διάδοσης των συλλεχθέντων δικαστικών αποφάσεων, αναπτύσσουμε μια εφαρμογή διαδικτύου και κάνουμε τα δεδομένα διαθέσιμα μέσω μιας REST διεπαφής λογισμικού (API).

0.2 Μηχανική Μάθηση και Επεξεργασία Φυσικής Γλώσσας

0.2.1 Ορισμοί

Ο όρος Μηχανική Μάθηση αναφέρεται στον χώρο της Τεχνητής Νοημοσύνης, όπου αλγόριθμοι (μοντέλα) μαθαίνουν οι ίδιοι πώς να λύσουν αποδοτικά ένα πρόβλημα χωρίς την ανάγκη περαιτέρω ανθρώπινου προγραμματισμού για να οριστούν κανόνες επίλυσής του. Στις περισσότερες εφαρμογές αυτό γίνεται κάνοντας χρήση συνόλων δεδομένων τα οποία χωρίζονται σε υποσύνολα *εκπαίδευσης*: που χρησιμοποιείται για την εκπαίδευση του μοντέλου, *επικύρωσης*: που χρησιμοποιείται για να επικυρώσουμε αρχιτεκτονικές επιλογές του μοντέλου που δεν μπορούν να εκπαιδευτούν κατά την διαδικασία εκπαίδευσης, και *ελέγχου*: το οποίο χρησιμοποιείται για τον έλεγχο της επίδοσης του μοντέλου και το οποίο δεν είναι γνωστό σε κανένα από τα προαναφερθέντα στάδια. Στην περίπτωση που το σύνολο δεδομένων, πέραν των δεδομένων εισόδου, διαθέτει και επιπλέον δεδομένα επιθυμητής εξόδου για κάθε είσοδο

[1] https://github.com/DominusTea/LegalSum-Dataset/releases/tag/v1.0.0

τότε έχουμε *Επιβλεπόμενη Μηχανική Μάθηση*, ενώ αλλιώς *Μη-επιβλεπόμενη*. Ο τομέας της Μηχανικής Μάθησης βιώνει μια ερευνητική αναζωπύρωση, αφού η αύξηση της υπολογιστικής ισχύος και του μεγέθους των διαθέσιμων ανοιχτών δεδομένων, δίνουν την δυνατότητα εκπαίδευσης μοντέλων με περισσότερες παραμέτρους, τα οποία μπορούν να κατασκευάσουν «βαθύτερες αναπαραστάσεις» των δεδομένων εισόδου και εξόδου (Βαθιά Μηχανική Μάθηση). Ιδιαίτερο ερευνητικό ενδιαφέρον έχει δοθεί, προσφάτως, στην *Αυτο-επιβλεπόμενη Μηχανική Μάθηση* όπου το μοντέλο μαθαίνει από δεδομένα επιθυμητής εξόδου που έχουν παραχθεί αυτομάτως από τα δεδομένα εισόδου.

Ο όρος Επεξεργασία Φυσικής Γλώσσας (ΕΦΓ) αναφέρεται στον επιστημονικό τομέα που διερευνά μεθόδους κατανόησης των φυσικών γλωσσών, δηλαδή γλωσσών που έχουν προκύψει χωρίς σχεδιασμό. Διαχρονικά, κάνει χρήση μεθόδων της Υπολογιστικής Γλωσσολογίας, της Επιστήμης Υπολογιστών και της Γνωσιακής Επιστήμης. Το σύγχρονο παράδειγμα, όμως, εφαρμογών ΕΦΓ κάνει χρήση μεθόδων Βαθιάς Μηχανικής Μάθησης για την αναπαράσταση λέξεων και προτάσεων παράγοντας εντυπωσιακά αποτελέσματα σε τομείς όπως η μηχανική μετάφραση [153], η αυτόματη περίληψη κειμένων [77] αλλά και η κατασκευή γλωσσικών μοντέλων [112] τα οποία μοντελοποιούν την πιθανότητα μιας ακολουθίας λέξεων δεδομένης μιας ακολουθίας που έχει προηγηθεί.

Transformer

Το Transformer [137] αποτελεί μια αρχιτεκτονική νευρωνικού δικτύου που βρίσκει εφαρμογή σε πλήθος διαφόρων τομέων όπως η αυτόματη περίληψη κειμένου [77], η κατανόηση βίντεο [14] και η μηχανική μετάφραση [153]. Η αρχιτεκτονική του δίνεται στο Σχήμα 1. Η πρωτοτυπία της αρχιτεκτονικής αυτής έγκειται στο ότι δεν αποτελείται από ούτε αναδρομικά δίκτυα αλλά ούτε συνελικτικά δίκτυα. Αντιθέτως, για την μοντελοποίηση του context κατά την κατασκευή αναπαραστάσεων για κάθε token εισόδου, χρησιμοποιείται μόνο μια παραλλαγή του μηχανισμού προσοχής: η *αυτο-προσοχή* με πολλαπλές κεφαλές.

Υποθέτοντας εισόδους $\mathbf{K}$, $\mathbf{V}$, $\mathbf{Q}$, ο μηχανισμός αυτο-προσοχής υπολογίζει το διάνυσμα προσοχής:

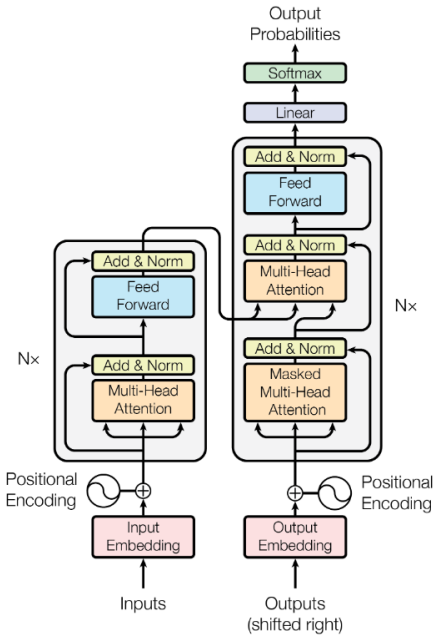
$$Attention(\mathbf{K},\ \mathbf{V},\ \mathbf{Q}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\mathbf{V}) \tag{1}$$

όπου $\mathbf{K} = \mathbf{V} = \mathbf{Q}$ στην περίπτωση της αυτο-προσοχής. Το Transformer υπολογίζει h κεφαλές (δηλαδή διανύσματα προσοχής), για την ίδια είσοδο, προβάλλοντας τα διανύσματα εισόδου χρησιμοποιώντας διαφορετικούς πίνακες για κάθε κεφαλή:

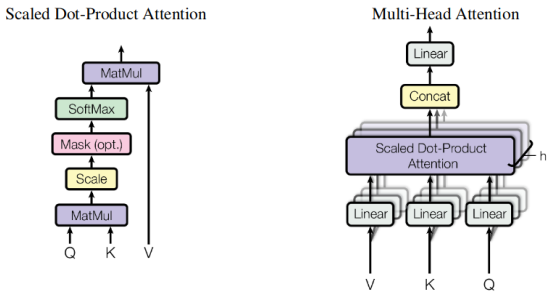
$$\mathbf{head}_i = Attention(\mathbf{K}\mathbf{W}_i^K,\ ,\mathbf{V}\mathbf{W}_i^V,\ ,\mathbf{Q}\mathbf{W}_i^Q,\ ) \tag{2}$$

$$MultiHeadAttention(\mathbf{K},\ \mathbf{V},\ \mathbf{Q}) = [\mathbf{head}_1, \ldots, \mathbf{head}_h]\mathbf{W}^O \tag{3}$$

όπου ο πίνακας $\mathbf{W}^O$ απορροφά την διάσταση των h κεφαλών. Επειδή το δίκτυο Transformer δεν είναι αναδρομικό και επομένως το διάνυσμα εισόδου εισάγεται μία μόνο φορά στο δίκτυο, υιοθετείται ένας πρωτότυπος τρόπος εισαγωγής της πληροφορίας τοποθεσίας σε κάθε token του διανύσματος εισόδου. Συγκεκριμένα, αν το μοντέλο έχει εσωτερική διάσταση d_{model} και μέγιστο μήκος εισόδου N, τότε παράγουμε διανύσματα αναπαράστασης τοποθεσίας $\mathbf{PE} \in$



(α′) *Η πλήρης αρχιτεκτονική ενός μοντέλου κωδικοποιητή-αποκωδικοποιητή βασισμένο στο transformer*



(β′) *Αριστερά: ο μηχανισμός προσοχής scaled dot-product. Δεξιά: ο μηχανισμός προσοχής με πολλαπλές κεφαλές*

Σχήμα 1. *Γραφική απεικόνιση των (α) ενός tranformer μοντέλου κωδικοποιητή-αποκωδικοποιητή, και (β) της μεθόδου προσοχής με πολλαπλές κεφαλές.*

$\mathbb{R}^{N \times d_{\text{model}}}$ ως εξής:

$$\mathbf{PE}_{i,j} = \begin{cases} sin(i/10000^{2j/d_{model}}) & , j \; even \\ cos(i/10000^{(2j-1)/d_{model}}) & , j \; odd \end{cases}$$

τα οποία αθροίζονται με το διάνυσμα εισόδου.

Η αρχιτεκτονική των transformers έχει κυρίως δύο πλεονεκτήματα έναντι προγενέστερων αρχιτεκτονικών: 1) έχει γραμμική πολυπλοκότητα ως προς το μέγεθος εισόδου - έναντι της τετραγωνικής των αναδρομικών δικτύων 2) Λόγω της μη ύπαρξης αναδρομής, οι διεργασίες υπολογισμού του μοντέλου μπορούν να παραλληλοποιηθούν σε μεγάλο βαθμό και επίσης αποφεύγεται προβλήματα μηδενισμού ή απειρισμού του σήματος ανανέωσης των βαρών κατά την εκπαίδευση. Όμως, η έλλειψη αναδρομής επιβάλει ένα άνω όριο μεγέθους στο διάνυσμα εισόδου.

0.2.2 Αναπαραστάσεις Λέξεων και κειμένων

TF-IDF

Η μέθοδος αναπαράστασης λέξεων tf-idf είναι μια απλή αλλά διαδεδομένη μέθοδος που βασίζεται στην συχνότητα κάθε λέξης σε ένα κείμενο αλλά και σε μια συλλογή κειμένων γενικότερα. Με αυτόν τον τρόπο λέξεις που εμφανίζονται σε πολλά κείμενα έχουν μικρότερη τιμή. Έτσι μια πρόταση $S = \{1, \ldots, w_i, \ldots w_T\}$ ενός κειμένου d_i σε ένα σύνολο κειμένων D μπορεί να αναπαρασταθεί ως ένα Bag of Word διάνυσμα μήκους όσο το πλήθος των διαφορετικών λέξεων στο λεξικό και τιμή το tf-idf σκορ της κάθε λέξης στην αντίστοιχη θέση

του διανύσματος η οποία προκύπτει από την θέση της λέξης στο λεξικό DC επί όλων των κειμένων D, εφόσον αυτή η λέξη υπάρχει στο d_i, αλλιώς 0. Μαθηματικά:

$$
\begin{aligned}
tf(w,d) &= log(Count(w,d)+1) \\
df(w,D) &= \sum_{d\in D\,:\,w\in d} 1 \\
idf(w,D) &= \frac{|D|}{df(w,D)} \\
tf-idf(w,d,D) &= tf(w,d)\cdot idf(w,D) \\
tf-idf(S,d,D) &= [(tf-idf(w_i,d,D))^{w_1\in S},\ldots,(tf-idf(w_{|DC|},d,D))^{w_{|DC|}\in S}]
\end{aligned}
$$

Word2Vec

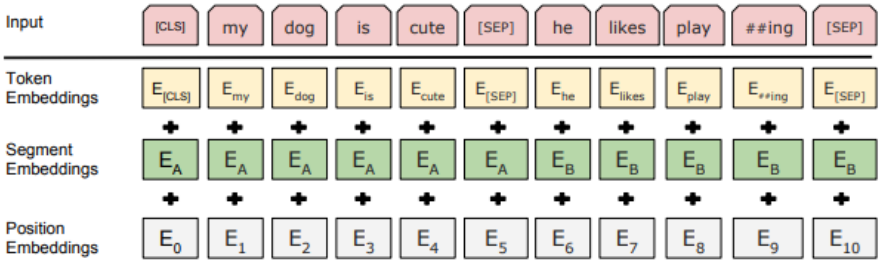
Οι Word2Vec αναπαραστάσεις [91] είναι μια μέθοδος αναπαράστασης που βασίζεται στην υπόθεση ότι η λέξεις αποκτούν το νόημα τους βάσει των συγκείμενων λέξεων τους. Έτσι εκπαιδεύοντας έναν αλγόριθμο αυτο-επιβλεπόμενης μάθησης ο οποίος μαθαίνει παράλληλα να ξεχωρίζει ποιες λέξεις θα μπορούσαν να είναι συγκείμενες και ποιες μη-συγκείμενες μια λέξης. Για να το κάνει αυτό κάθε λέξη προβάλλεται σε ένα ενδιάμεσο διανυσματικό χώρο ο οποίος αποτελεί και το word2vec διάνυσμα της λέξης. Εκπαιδεύοντας τον αλγόριθμο με διάφορες παραλλαγές σε μεγάλο πλήθος κειμένων, καταλήγουμε με αναπαραστάσεις λέξεων οι οποίες έχουν ενσωματώσει της σημασιολογία της κάθε λέξης όπως αυτή εκφράζεται από την πιθανότητας εμφάνισής της λέξης με συγκείμενες άλλες λέξεις.

BERT

Οι αναπαραστάσεις BERT (Bidirectional Encoder Representations from Transformers) αποτελούν μια αλλαγή παραδείγματος στην βιβλιογραφία των αναπαραστάσεων κειμένου για μηχανική μάθηση, αφού κάνουν χρήση ενός κωδικοποιητή Transformer μέσω του οποίου μπορούν να παραχθούν αμφίδρομες αναπαραστάσεις για κάθε λέξη του κειμένου εισόδου, οι οποίες μάλιστα εξαρτώνται και από όλες τις άλλες συγκείμενες λέξεις - δηλαδή τα συμφραζόμενα.

Το μοντέλο εκπαιδεύεται 1) στο πρόβλημα της εύρεσης κρυμμένων λέξεων του κειμένου, δηλαδή λέξεων που έχουν αντικατασταθεί από ένα [MASK] token και 2) στο πρόβλημα πρόβλεψης αν μια πρόταση είναι η επόμενη μιας άλλης. Και για τα δύο προβλήματα μπορούν να κατασκευαστούν αυτομάτως σύνολα δεδομένων καθιστώντας έτσι την εκπαίδευση του μοντέλου ένα πρόβλημα αυτο-επιβλεπόμενης μάθησης.

Για να μπορούν να μοντελοποιηθούν και τα δύο προβλήματα από την αρχιτεκτονική του δικτύου, το διάνυσμα εισόδου μετασχηματίζεται προσθέτοντας ένα token κατηγοριοποίησης που μπορεί να χρησιμοποιηθεί για να εξαχθεί πληροφορία κατηγορίας στο πρόβλημα πρόβλεψης επόμενης πρότασης, και τα tokens διαχωρισμού εισόδου, με τα οποία μπορούν να διαφοροποιηθούν τα (ενδεχομένως) ξεχωριστά κομμάτια του διανυσμάτων εισόδου (Σχήμα 2). Επίσης ανάλογα με το τμήμα (που ορίζεται μεταξύ των διανυσμάτων διαχωρισμού εισόδου) προστίθεται σε κάθε token ένα διάνυσμα τμήματος, το οποίο είναι εκπαιδεύσιμη παράμετρος του δικτύου.



Σχήμα 2. *Το σύστημα μετασχηματισμού εισόδου της αρχιτεκτονικής BERT. Σε κάθε token εισόδου προστίθεται τα διανύσματα τοποθεσίας και τα διανύσματα τμήματος. Πηγή: [29]*

Η αρχιτεκτονική BERT λόγω της δυνατότητας εκπαίδευσης σε πολύ μεγάλα σύνολα δεδομένων τα οποία κατασκευάζονται επίσης εύκολα, έδωσε την δυνατότητα εκπαίδευσης πολύ μεγάλων μοντέλων τα οποία χρησιμοποιούνται για να εξαχθούν βαθιές αναπαραστάσεις κειμενικών δεδομένων.

0.3 Αυτόματη Περίληψη Νομικού Κειμένου

0.3.1 Εισαγωγή

Η αυτόματη περίληψη κειμένων, από την δημιουργία της σαν ερευνητικός τομέα την δεκαετία του '50 μέχρι τις μέρες μας, έχει βρει εφαρμογή σε πολλά είδη κειμένων - κυρίως άρθρα ειδήσεων και επιστημονικά άρθρα. Η χρησιμότητα της είναι προφανής από το γεγονός ότι τα ολοένα και αυξανόμενου πλήθους ψηφιοποιημένα δεδομένα είναι χρήσιμα, αν και μόνο αν, μπορούν να χρησιμοποιηθούν από τους χρήστες των αντιστοίχων εφαρμογών ή από άλλες διαδικασίες επεξεργασίας τους. Στον τομέα της νομολογίας, η ψηφιοποίηση των πληροφοριών ήταν συνήθως αργή, κάτι που μπορεί να αποδοθεί στην εκ᾽φύσεως εμπιστευτικότητα των εγγράφων που χρησιμοποιούνται, στην ανάγκη σεβασμού των νόμων περί προσωπικών δεδομένων, αλλά και στην αδράνεια που συχνά χαρακτηρίζει παραδοσιακούς τομείς όπως η νομική.

Η χρησιμότητα εφαρμογών ΑΠΚ για δικαστικές αποφάσεις μπορεί να γίνει κατανοητή μέσω των παρακάτω παραδειγμάτων: 1) Δικαστήρια και οργανισμοί έχουν εξειδικευμένες ομάδες νομικών συντακτών για να συντάσσουν περιλήψεις των δικαστικών αποφάσεων. Αυτοματοποιώντας μέρος της σχετικής διαδικασίας θα δίνονταν η δυνατότητα στους νομικούς συντάκτες να αφιερώσουν περισσότερο χρόνο σε άλλες αρμοδιότητές τους. 2) Οι δικηγόροι και νομικοί χρειάζεται να διαβάσουν ολόκληρες τις δικαστικές αποφάσεις ώστε να μπορέσουν να εξάγουν τα υπερασπιστικά επιχειρήματα της κάθε πλευράς. Οι περιλήψεις αυτών των κειμένων μπορούν να τους εξοικονομήσουν χρόνο αλλά και να επιτρέψουν ποσοτικές μελέτες στον τομέα των νομικών επιχειρημάτων οι οποίες δεν θα ήταν εφικτές μόνο με μη-αυτοματοποιημένες περιλήψεις. 3) Οι δικηγόροι που προετοιμάζουν τα επιχειρήματά τους για μια υπόθεση χρειάζεται να ψάξουν για παρόμοιες δικαστικές υποθέσεις, επιλέγοντας τα αντίστοιχα αποσπάσματα κάνοντας χρήση της εμπειρίας και των γνώσεών τους, έτσι ώστε να μπορέσουν αν αποκτήσουν μια εις βάθος κατανόηση της εφαρμοστέας νομολογίας. Η ανάγνωση των περιλήψεων είναι πιο εύκολος και λιγότερο χρονοβόρος τρόπος για να πραγ-

ματωθεί αυτή η ανάγκη, συγκριτικά με την ανάγνωση όλης της δικαστικής απόφασης.

0.3.2 Σύνολα δεδομένων

Τα σύνολα δεδομένων για ΑΠΚ προέρχονται κυρίως από τον τομέα των άρθρων ενημέρωσης. Χαρακτηριστικά αναφέρουμε τα σύνολα δεδομένων NYT [120]: με άρθρα ειδήσεων από το αντίστοιχο περιοδικό, CNN/Daily Mail [94]: με άρθρα ειδήσεων από τους αντίστοιχους ενημερωτικούς ιστοτόπους, και το Newsroom σύνολο δεδομένων [46] το οποίο εμπεριέχει άρθρα από 38 διαφορετικά δημοσιογραφικά έντυπα. Τα σύνολα δεδομένων αυτά, λόγω της φύσης των δημοσιογραφικών άρθρων, εμπεριέχουν «κλισέ» (*biases*): 1) *εξαγωγικό bias* [70]: το οποίο σημαίνει ότι οι περιλήψεις συχνά μπορούν να συνταχθούν ικανοποιητικά εξάγοντας μόνο λέξεις (ή και ολόκληρες) προτάσεις του αρχικού κειμένου, 2) *bias στην τοποθεσία των σημαντικών πληροφοριών:* [61, 125] καθώς οι σημαντικές πληροφορίες ενός άρθρου συνήθως βρίσκονται στην αρχή του. Επίσης, τα σύνολα δεδομένων ΑΠΚ που εμπεριέχουν ανθρώπινη πολλαπλές ανθρώπινες περιλήψεις για το ίδιο άρθρο συχνά αναδεικνύουν την διαφωνία των ανθρώπινων συντακτών στο τι αποτελεί μια καλή περίληψη [70].

Τα σύνολα δεδομένων για αυτόματη περίληψη νομικών κειμένων είναι ομολογουμένως λιγότερα. Ενδεικτικά, αναφέρουμε τα Billsum [67]: με σύνολο δεδομένων από νόμους του Αμερικανικού Κογκρέσου, δικαστικές αποφάσεις από Καναδικά δικαστήρια [145] και δικαστικές αποφάσεις από το Ανώτατο Δικαστήριο της Ινδίας [16]. Στο χώρο των περιλήψεων νομικών κειμένων, δεν γνωρίζουμε αντίστοιχη συστηματική μελέτη που να μετράει ποσοτικά την ύπαρξη των προαναφερθέντων biases.

0.3.3 Αλγόριθμος LexRank

Ο αλγόριθμος LexRank [34] ανήκει στους αλγορίθμους που βασίζονται σε γράφους και οι οποίοι χρησιμοποιούνται για την εξαγωγή σημαντικών προτάσεων/αποσπασμάτων από ένα κείμενο. Ο αλγόριθμος LexRank προτάθηκε ταυτόχρονα και ανεξάρτητα από άλλη ερευνητική ομάδα με την ονομασία TextRank [89]. Ο αλγόριθμος βασίζεται στον PageRank [103] αλγόριθμο, για εύρεση σημαντικών κόμβων σε ένα γράφο βάσει της ανα-ζεύγη ομοιότητάς τους.

Έστω μια συνάρτηση ομοιότητας μεταξύ δύο προτάσεων sim(u,v). Το σκορ σημαντικότητα κάθε πρότασης u δίνεται από την παρακάτω σχέση η οποία εφαρμόζεται επαναληπτικά για κάθε πρόταση μέχρι να υπάρξει σύγκλιση:

$$p(u) = \frac{d}{N} + (1-d) \sum_{v \in \text{adj}[u]} \frac{\text{sim}(u,v)}{\sum_{z \in \text{adj}[v]} sim(z,v)} p(v) \tag{4}$$

όπου ο πρώτος αθροιστικός όρος $\frac{d}{N}$ χρειάζεται για την σύγκλιση του αλγορίθμου και προτάσεις με μικρή ομοιότητα θεωρούνται μη-γειτονικοί. Αυτή η μέθοδος επιτρέπει την ύπαρξη πολλαπλών συστάδων με σημαντικές προτάσεις, σε αντίθεση με προγενέστερους αλγορίθμους που βασίζονταν μόνο σε μια κεντρική πρόταση.

Μια παραλλαγή του αλγορίθμου η οποία είναι χρήσιμη για περίληψη κειμένων η οποία εξειδικεύεται σε ένα θέμα (query) είναι ο Biased LexRank αλγόριθμος [101], ο οποίος τροποποιεί την εξίσωση 4 αυξάνοντας τον όρο $\frac{d}{N}$ και επομένως την σημαντικότητα των προτάσεων

που είναι σχετικές με το θέμα q βάσει μιας συνάρτησης σχετικότητας rel:

$$p(u) = d\frac{rel(u,q)}{\sum_{z\in \text{Corpus}} rel(z,q)} + (1-d)\sum_{v\in \text{adj}[u]} \frac{\text{sim}(u,v)}{\sum_{z\in \text{adj[v]}} sim(z,v)} p(v) \tag{5}$$

Οι συναρτήσεις σχετικότητας που μπορούν να χρησιμοποιηθούν είναι είτε συναρτήσεις συχνότητας λέξεων:

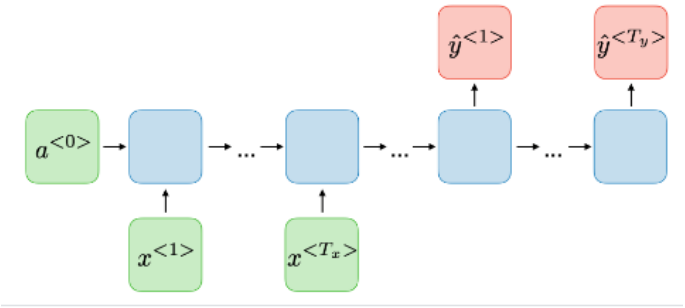
$$rel(u,q) = \sum_{w\in q} log(tf(w,u)+1)\cdot log(tf(w,q)+1)\cdot idf(w)$$

είτε συναρτήσεις ομοιότητας (όπως πχ απόσταση συνημιτόνου) σε κάποιο χώρο διανυσματικής αναπαράστασης των προτάσεων

$$rel(\mathbf{u},\mathbf{q}) = \frac{\mathbf{u}\cdot\mathbf{q}}{\|\mathbf{u}\|\cdot\|\mathbf{q}\|}$$

0.3.4 Αρχιτεκτονική Κωδικοποιητή-Αποκωδικοποιητή

Η αρχιτεκτονική κωδικοποιητή-αποκωδικοποιητή, αποτελεί μια ευρέως διαδεδομένη μέθοδο για ανάλυση σειριακών δεδομένων και παραγωγή σειριακών εξόδων (Σχήμα 3). Συνήθεις εφαρμογές είναι η μηχανική μετάφραση, η αυτόματη περίληψη και η πρόβλεψη χρονοσειρών.

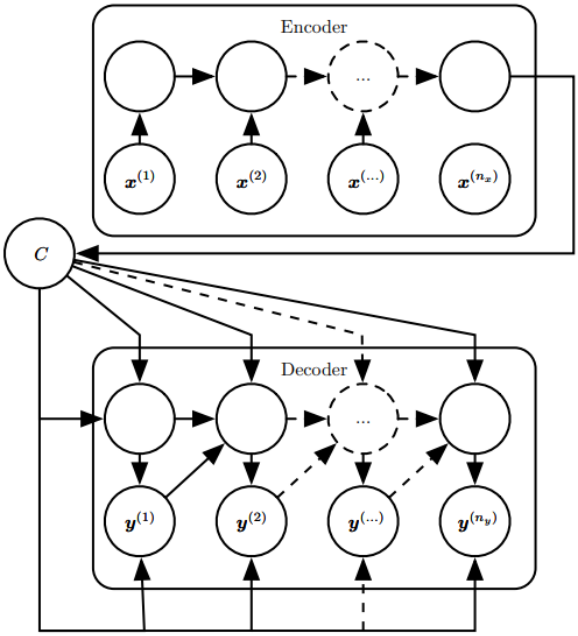


Σχήμα 3. *Γραφική απεικόνιση της λειτουργίας ενός μοντέλου αρχιτεκτονικής Κωδικοποιητή-Αποκωδικοποιητή. Πηγή: [3]*

Η αρχιτεκτονική αποτελείται από δύο αναδρομικά μοντέλα: τον κωδικοποιητή και τον αποκωδικοποιητή. Ο κωδικοποιητής κατασκευάζει μια αναπαράσταση (διάνυσμα context) της ακολουθίας εισόδου, αφού την επεξεργαστεί σειριακά, παίρνοντας διαφορετικές εσωτερικές καταστάσεις (hidden states) - με την τελευταία εσωτερική κατάσταση να αντιστοιχεί στο διάνυσμα context. Ο αποκωδικοποιητής λαμβάνει και επεξεργάζεται σειριακά αυτή την αναπαράσταση την οποία και χρησιμοποιεί για να παραγάγει επίσης σειριακά την ακολουθία εξόδου.

Πολλές φορές, η παραπάνω αρχιτεκτονική εμπλουτίζεται με έναν μηχανισμό προσοχής [9], έτσι ώστε το διάνυσμα context να βασίζεται σε όλες τις ενδιάμεσες hidden states του κωδικοποιητή (και όχι μόνο στην τελευταία) και μάλιστα με τρόπο διαφορετικό για κάθε στάδιο αποκωδικοποίησης (Σχήμα 4).

Χρησιμοποιώντας αυτές τις αρχιτεκτονικές με νευρωνικά δίκτυα, εφόσον υπάρχουν σύνο-



Σχήμα 4. *Γραφική απεικόνιση της λειτουργίας ενός μοντέλου αρχιτεκτονικής Κωδικοποιητή-Αποκωδικοποιητή με προσοχή. Πηγή: [45]*

λα δεδομένων με ελεύθερες περιλήψεις αναφοράς των δικαστικών αποφάσεων, μπορούμε να εκπαιδεύσουμε το δίκτυο στο να παράγει ελεύθερες περιλήψεις, κάτι που έχει δοκιμαστεί με επιτυχία για άρθρα ειδήσεων [77].

0.3.5 Μετρικές Αξιολόγησης

Αυτόματες Μετρικές

Οι αυτόματες μετρικές αξιολόγησης που χρησιμοποιούνται για την αξιολόγηση μεθόδων αυτόματης περίληψης κειμένων είναι πολλές. Οι περισσότερες βασίζονται στην λεξικογραφική ομοιότητα των περιλήψεων που έχουν παραχθεί αυτόματα και των περιλήψεων που έχουν συντάξει άνθρωποι. Οι μετρικές ROUGE (Recall Oriented Understudy for Gisting Evaluation) [73] βασίζονται ποικιλοτρόπως σε αυτήν την λεξικογραφική ομοιότητα και αποτελούν τις μετρικές αξιολόγησης που συναντώνται συχνότερα στην σχετική βιβλιογραφία. Οι μετρικές ROUGE ορίζονται ως εξής:

- **ROUGE-N:** η οποία μετράει την N-gram επικάλυψη μεταξύ αυτόματα παραγόμενης περίληψης S και μιας περίληψης αναφοράς R:

$$ROUGE - N(R, S) = \frac{\sum_{g \in GRAM(N,R)} count_{matching}(g)}{\sum_{g \in GRAM(N,R)} count(g)}$$

  Η μετρική είναι βασισμένη στην ανάκληση αφού το ποσοστό των επικαλυπτόμενων N-gram υπολογίζεται επί των N-gram της περίληψης αναφοράς.

- **ROUGE-L:** η οποία μετράει το μήκος της μεγαλύτερης κοινής υπακολουθίας λέξεων που υπάρχουν μεταξύ προτάσεων της αυτόματης περίληψης και της περίληψης αναφο-

ράς. Για να ορίσουμε την ROUGE-L για μια περίληψη αναφοράς R και μια αυτόματη περίληψη C, οι περιλήψεις χωρίζονται στις επιμέρους προτάσεις τους και στην συνέχεια υπολογίζεται το μέσο μήκος των Μέγιστων Κοινών Υπακολουθιών (ΜΚΥ) τους ως εξής:

$$ROUGE - L(R, C) = \frac{\sum_{r \in R} | \cup_{c \in C} \{MKY(r, c)\}|}{|R|}$$

Η μετρική αυτή δίνει την δυνατότητα ποσοτικοποίησης της ευφράδειας της αυτόματης περίληψης και μάλιστα σε επίπεδο προτάσεων και όχι απλά διαδοχικών λέξεων. Όμως, δεν λαμβάνει υπόψιν 1) την ύπαρξη κοινών υπακολουθιών μικρότερου μήκους από την ΜΚΥ και 2) το πλήθος των κενών διαστημάτων μεταξύ των λέξεων-μελών των ΜΚΥ.

- **ROUGE-W:** γενικεύει την ROUGE-L μετρική δίνοντας διαφορετικό βάρος σε κάθε κοινή υπακολουθία ανάλογα με το πόσο συνεχόμενες είναι οι λέξεις της, αποδίδοντας έτσι λιγότερο βάρος σε κοινές υπακολουθίες με πολλές μη-συνεχόμενες λέξεις.

Οι παραπάνω μετρικές υπολογίζουν την ανάκληση, αλλά συχνότερα στην βιβλιογραφία χρησιμοποιείται το F1-score τους το οποίο υπολογίζεται ως ο αρμονικός μέσος των ROUGE μετρικών για ακρίβεια και ανάκληση.

Μετρικές ανθρώπινης αξιολόγησης

Οι αυτόματες μετρικές μας δίνουν την δυνατότητα γρήγορης αξιολόγησης των μεθόδων αυτόματης περίληψης κειμένου σε μεγάλα σύνολα δεδομένων. Όμως, πρόσφατες μελέτες [35, 70] δείχνουν πως η συσχέτιση των ανθρώπινων μετρικών με τις αυτόματες μετρικές είναι μέτρια ή και ασθενής. Για αυτό θεωρείται απαραίτητη η επιπλέον επικύρωση των χρησιμοποιούμενων μεθόδων με χρήση ανθρώπινης αξιολόγησης. Τα κριτήρια που συνήθως χρησιμοποιούνται [70] είναι:

- **Σχετικότητα**: το κατά πόσο η περίληψη έχει συλλάβει τις σημαντικές πληροφορίες του κειμένου.
- **Συνέπεια:** το κατά πόσο το κείμενο και η περίληψη συμφωνούν πραγματολογικά.
- **Ευφράδεια:** το κατά πόσο το κείμενο περιλαμβάνει προτάσεις που είναι υψηλής ποιότητας, η κάθε μία ξεχωριστά.
- **Συνοχή:** το κατά πόσο βάσιμη είναι η δομή με την οποία οι ιδέες του κειμένου οργανώνονται σε προτάσεις στο κείμενο της περίληψης.

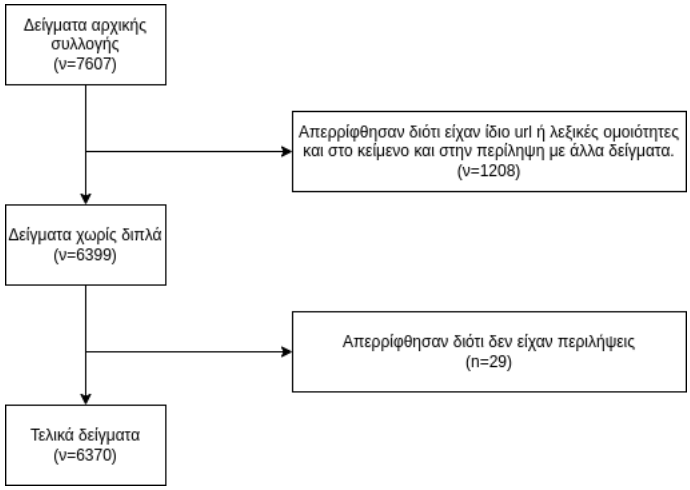
Όμως, η κατασκευή συνόλων δεδομένων ανθρώπινης αξιολόγησης περιλήψεων είναι δύσκολο πρόβλημα, καθώς απαιτείται μεγάλο πλήθος αξιολογητών οι οποίοι καλούνται να αφιερώσουν μεγάλο χρονικό διάστημα για την ανάγνωση των κειμένων και των περιλήψεών τους. Επίσης, στη περίπτωση νομικών κειμένων όπως οι δικαστικές αποφάσεις, κρίνεται απαραίτητο οι αξιολογητές να έχουν επαρκή νομική κατάρτιση για να μπορούν να κατανοήσουν τα κείμενα και την ποιότητα των περιλήψεών τους.

0.4 Προτεινόμενες Μέθοδοι και Πειραματικά Αποτελέσματα

0.4.1 Κατασκευή Συνόλου δεδομένων

Με έναυσμα 1) το αυξημένο πλήθος εργατοωρών που δαπανούν οι Έλληνες νομικοί στην ανάγνωση και ανάλυση δικαστικών αποφάσεων αλλά και 2) την έλλειψη αντίστοιχου συνόλου δεδομένων για Ελληνικά δικαστήρια, κατασκευάζουμε προγράμματα συλλογής δεδομένων από τους ιστότοπους των δικαστηρίων του Αρείου Πάγου και του Συμβουλίου της Επικρατείας. Στην περαιτέρω ανάλυση, χρησιμοποιούμε μόνο τα δεδομένα από το πρώτο, καθώς τα δεδομένα από το Συμβούλιο της Επικρατείας δεν περιείχαν περιλήψεις και επίσης κρίθηκαν χαμηλής ποιότητας λόγω προβλημάτων στην ψηφιοποίησή τους από το δικαστήριο. Αφήνουμε για μελλοντική δουλειά την αποθορύβωση αυτών των δεδομένων και την αξιολόγηση χρησιμότητας των δεδομένων που προκύπτουν από αυτή την διαδικασία.

Κάνοντας χρήση ενός πρωτοκόλλου διαγραφής μη-χρήσιμων αποφάσεων (Σχήμα 5) κατασκευάζουμε ένα σύνολο δεδομένων 6,370 δειγμάτων το οποίο, πέραν του κειμένου και της περίληψης της δικαστικής απόφασης, εμπεριέχει μεταδεδομένα σχετικά με την απόφαση και τα οποία δίνονται στον Πίνακα 1. Κάνουμε διαθέσιμο το σύνολο δεδομένων μας με την ελπίδα να αποτελέσει κίνητρο για περισσότερη έρευνα στην ερευνητική περιοχή της επεξεργασίας δικαστικών αποφάσεων[2].



Σχήμα 5. *Γραφική απεικόνιση του πρωτοκόλλου δημιουργίας συνόλου δεδομένων που ακολουθήθηκε. Το v αναφέρεται στο πλήθος κειμένων που απομένουν/απορρίπτονται σε κάθε στάδιο.*

0.4.2 Προτεινόμενες μέθοδοι αυτόματης περίληψης

Εξαγωγική περίληψη

Χρησιμοποιούμε τον αλγόριθμο LexRank και αξιολογούμε τους παρακάτω συνδυασμούς συναρτήσεων ομοιότητας προτάσεων και διανυσματικών αναπαραστάσεων των προτάσεων. Έστω s_1, s_2 δύο προτάσεις:

[2] https://github.com/DominusTea/LegalSum-Dataset/releases/tag/v1.0.0

Μεταδεδομένα Δικαστικών Αποφάσεων Αρείου Πάγου

Μεταδεδομένα	Τύπος δεδομένου	Συναγόμενο	Περιγραφή
Κατηγορία υπόθεσης	String		Η γενική κατηγορία στην οποία ταξινομήθηκε η απόφαση από τους νομικούς συντάκτες του Αρείου Πάγου. Κάθε υπόθεση ανήκει σε μία ακριβώς κατηγορία.
Ετικέτες υπόθεσης	String		Οι ετικέτες που αντιστοιχούν στην κάθε υπόθεση, όπως επισημειώθηκαν από τους νομικούς συντάκτες του Αρείου Πάγου. Κάθε υπόθεση μπορεί να έχει πολλαπλές ετικέτες.
Τμήμα δικαστηρίου	String	✓	Το συγκεκριμένο τμήμα και το είδος του (πχ. Ποινικό, Πολιτικό, κτλπ.) που δίκασε την συγκεκριμένη υπόθεση.
Έτος έκδοσης	Integer	✓	Το έτος που εξεδόθη η απόφαση του δικαστηρίου.
Αναγνωριστικό απόφασης	String	✓	Το αναγνωριστικό που απεδόθη στην απόφαση από το δικαστήριο. Είναι μοναδικό ανά απόφαση που εξήγχθει από το ίδιο τμήμα του δικαστηρίου.
Πηγαίο URL	String	✓	Ο σύνδεσμος στην HTML ιστοσελίδα του Αρείου Πάγου από την οποία αντλήθηκαν τα δεδομένα.

Πίνακας 1. *Τα μεταδεδομένα που συλλέχθηκαν για το σύνολο δεδομένων δικαστικών αποφάσεων του Αρείου Πάγου. Τα μεταδεδομένα που συνάγονται αυτομάτως από εμάς, βάσει του κειμένου εισόδου, επισημειώνονται με ✓στην αντίστοιχη στήλη.*

- Ομοιότητα κοινών λέξεων: η συνάρτηση ομοιότητας που χρησιμοποιείται στον TextRank και υπολογίζει το πλήθος των κοινών λέξεων:

$$sim_{cw}(s_1, s_2) = \frac{|s_1 \cap s_2|}{log(|s_1|) + log(|s_2|)}$$

- Απόσταση συνημιτόνου σε Tf-Idf BoW αναπαραστάσεις: η συνάρτηση ομοιότητας και η μέθοδος αναπαράστασης των προτάσεων που υιοθετήθηκε στον LexRank, όπου χρησιμοποιείται η απόσταση συνημιτόνου πάνω σε Tf-Idf BoW αναπαραστάσεις

$$sim_{cos\_tfidf}(s_1, s_2) = 1 - \frac{\mathbf{Tf-Idf}(s_1) \cdot \mathbf{Tf-Idf}(s_2)}{\|\mathbf{Tf-Idf}(s_1)\| \times \|\mathbf{Tf-Idf}(s_2)\|}$$

- Απόσταση συνημιτόνου σε Word2Vec+idf BoW αναπαραστάσεις: όπου χρησιμοποιούμε προ-εκπαιδευμένα unigram word2vec διανύσματα προτάσεων τα οποία προκύπτουν ως μέσα διανύσματα των λέξεων που τις αποτελούν σταθμισμένες επί το idf σκορ της κάθε μίας.

$$sim_{w2v}(s_1, s_2) = 1 - \frac{\mathbf{W2V}(s_1) \cdot \mathbf{W2V}(s_2)}{\|\mathbf{W2V}(s_1)\| \times \|\mathbf{W2V}(s_2)\|}$$

Επίσης, αξιολογούμε και τον αλγόριθμο Biased LexRank όπου για λέξεις θέματα κάνουμε χρήση των tags της υπόθεσης και για συνάρτηση σχετικότητας την συνάρτηση κοινών λέξεων.

Ελεύθερη Περίληψη

Υλοποιούμε το μοντέλο Κωδικοποιητή-Αποκωδικοποιητή με χρήση BERT που προτάθηκε για παραγωγή ελεύθερων περιλήψεων στο [77]. Τα βάρη του κωδικοποιητή αρχικοποιούνται με τα προ-εκπαιδευμένα βάρη του Ελληνικού BERT που διατίθενται ελεύθερα. Μέρος των δεδομένων προ-εκπαίδευσης υπήρξε η ελληνική μετάφραση των κοινοβουλευτικών πρακτικών του Ευρωκοινοβουλίου, τομέας σχετικός με τον τομέα των δικαστικών αποφάσεων.

0.4.3 Πειραματικά Αποτελέσματα

Αυτόματες Μετρικές

Αξιολογούμε τις μεθόδους αυτόματης περίληψης δικαστικών αποφάσεων που αναπτύξαμε μέσω των μετρικών ROUGE, χρησιμοποιώντας ως περιλήψεις αναφοράς τις περιλήψεις του συλλεχθέντος συνόλου δεδομένων. Τα αποτελέσματα δίνονται στον πίνακα 2[3]:

[3]Χρησιμοποιούμε την τελεία «.» ως υποδιαστολή, αν και δεν συνηθίζεται στα Ελληνικά, για την ομοιόμορφη παρουσίαση των αποτελεσμάτων καθ'όλη την εργασία μας.

Μετρικές	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-L	ROUGE-W
$\text{LexRank}_{\text{tf-idf}}$	71.46	**42.90**	**23.78**	**17.29**	**8.35**
$\text{LexRank}_{\text{com}}$	**71.51**	42.10	22.02	15.09	7.09
$\text{LexRank}_{\text{w2v}}$	69.63	39.63	19.69	12.93	6.05
Biased Lexrank	67.73	41.05	22.06	15.50	7.33
Random Sentence	70.64	40.26	19.77	13.41	6.28
BERT	62.90	38.52	20.64	14.37	5.28
BERT(RE)	62.80	38.51	20.39	14.19	5.21
BERT(RE+RM)	62.08	38.83	21.26	14.42	5.28
BERT(RE+RM+C)	**64.24**	**40.40**	**22.27**	**15.34**	**5.64**
BERT(RE+RM+LR)	62.01	37.89	20.32	13.71	4.99
BERT(RE+RM+LR+C)	63.98	39.85	21.90	15.33	5.64

Πίνακας 2. *Αποτελέσματα αυτόματης αξιολόγησης σε δύο τμήματα α) εξαγωγικές μέθοδοι, β) μέθοδοι ελεύθερης περίληψης. Οι εξαγωγικές μέθοδοι εξάγουν προτάσεις μέχρι να φτάσουν το τριπλάσιο μήκος της περίληψης αναφοράς. Στις μεθόδους ελεύθερης περίληψης, μετασχηματίζουμε το κείμενο εισόδου και ονοματίζουμε αντιστοίχως τα μοντέλα: RM: περιττά αποσπάσματα του κειμένου αφαιρούνται, RE: αναδιάταξη του κειμένου ώστε το αποτέλεσμα της δίκης να περιλαμβάνεται πάντα στην αρχή του, LR: μείωση του κειμένου στο μισό μήκος μέσω του αλγορίθμου $\text{LexRank}_{\text{tf-idf}}$, C: συμπερίληψη των ετικετών κατηγορίας στην αρχή του κειμένου εισόδου. Στον πίνακα αναφέρονται οι ποσοστιαίες F-1 τιμές των ROUGE μετρικών. Η καλύτερη μέθοδος για κάθε μετρική ανά κατηγορία αλγορίθμου περίληψης, επισημειώνεται με* ***έντονους χαρακτήρες****.*

Παρατηρούμε πως ο εξαγωγικός αλγόριθμος που αποδίδει καλύτερα είναι ο LexRank με τις tf-idf αναπαραστάσεις προτάσεων. Η κακή επίδοση των word2vec αναπαραστάσεων οφείλεται στην κακή ποιότητα των δεδομένων που χρησιμοποιήθηκαν για την εκπαίδευσή τους. Η μέθοδος Biased LexRank είναι καλύτερη από την τυχαία επιλογή προτάσεων, όμως χειρότερη από την απλή LexRank κάτι που υποδεικνύει ότι οι προτάσεις με υψηλή λεξικογραφική ομοιότητα με τις ετικέτες κατηγορίας δεν είναι κατανάγκην οι χρησιμότερες στην εξαγωγή περίληψης.

Στις μεθόδους ελεύθερης περίληψης, δοκιμάζουμε διάφορες μεθόδους που έχουν σκοπό την συμπερίληψη στο διάνυσμα εισόδου, όσο το δυνατόν μεγαλύτερου μέρους από τις χρήσιμες πληροφορίες του κειμένου[4]: 1) Αναδιάταξη του κειμένου εισόδου ώστε το αποτέλεσμα της απόφασης να είναι πάντα στην αρχή του κειμένου και επομένως να εισάγεται πάντα στο μοντέλο, 2) Αφαίρεση γενικού/περιττού κειμένου της απόφασης που είναι άσχετο με την ουσία της, όπως ημερομηνίες, ονόματα δικαστών, κλπ. 3) Συμπερίληψη των ετικετών κατηγορίας στο κείμενο εισόδου, 4) Μείωση του μεγέθους του κειμένου εισόδου στο μισό με χρήση του αλγορίθμου $\text{LexRank}_{\text{tf-idf}}$.

Παρατηρούμε πως η αναδιάταξη κειμένου και η αφαίρεση περιττού κειμένου βελτιώνουν ελαφρώς την επίδοση του μοντέλου. Μεγάλη βελτίωση της επίδοσης σε όλες της μετρικές υπάρχει όταν συμπεριλαμβάνουμε στο κείμενο εισόδου τις ετικέτες κατηγορίας, κάτι που σε συνάρτηση και των αποτελεσμάτων των εξαγωγικών αλγόριθμων, μας οδηγεί στο συμπέρασμα ότι: αν και η λεξικογραφική ομοιότητα προτάσεων με τις ετικέτες κατηγορίας δεν αρκεί για

[4]Η προ-εκπαιδευμένη αρχιτεκτονική BERT που χρησιμοποιούμε, επιβάλλει το διάνυσμα εισόδου να έχει μέγιστο μήκος 512 tokens. Σε περίπτωση μεγαλύτερης εισόδου, κρατούνται μόνο τα 512 πρώτα tokens

τις εξαγωγικές περιλήψεις, είναι όμως χρήσιμη για την παραγωγή συγκειμενικών αναπαραστάσεων για το κείμενο και βοηθάει έτσι στην παραγωγή ελεύθερων περιλήψεων. Τέλος, η μείωση του μεγέθους του κειμένου εισόδου μέσω του αλγορίθμου $\text{LexRank}_{\text{tf-idf}}$, φαίνεται να μειώνει την απόδοση κάτι που αποδίδουμε στο ότι το παραχθέν κείμενο εισόδου έχει χάσει μεγάλο μέρος της εσωτερικής συνοχής του, και επομένως διαφέρει αισθητά από τα κείμενα τα οποία χρησιμοποιήθηκαν για την προ-εκπαίδευση του BERT μοντέλου.

Μετρικές Ανθρώπινης Αξιολόγησης

Μελετούμε επιπλέον την επίδοση των αλγορίθμων μας σε πέντε (5) τυχαία επιλεγμένες από το υποσύνολο δεδομένων ελέγχου δικαστικές αποφάσεις, μέσω ανθρώπινης αξιολόγησης από έξι (6) νομικούς. Οι αξιολογητές, αξιολογούν 1) τις ελεύθερες περιλήψεις βάσει των μετρικών σχετικότητας, ευφράδειας, συνοχής και συνέπειας και 2) τις εξαγωγικές περιλήψεις βάσει μόνο της σχετικότητάς τους. Τα αποτελέσματα δίνονται στον Πίνακα 3

Περίληψη	Σχετικότητα	Ευφράδεια	Συνοχή	Συνέπεια
Αναφοράς	3.9	3.7	3.7	3.7
BERT(παραγχθείσα)	1.9	3.1	3.3	1.8
$\text{Lexrank}_{\text{tf-idf}}$	2.9	-	-	-
Biased LexRank	3.0	-	-	-

Πίνακας 3. *Αποτελέσματα ανθρώπινης αξιολόγησης σε κλίμακα Λίκερτ [72] 1-5. Στο πρώτο μέρος του πίνακα, συγκρίνονται οι ελεύθερες περιλήψεις αναφοράς με τις ελεύθερες περιλήψεις που παράγονται από το BERT μουτέλο. Στο δεύτερο μέρος του πίνακα, συγκρίνονται οι διάφορες εξαγωγικές μέθοδοι αυτόματης περίληψης μεταξύ τους.*

Παρατηρούμε πως, όσον αφορά τις μετρικές ευφράδειας και συνοχής, οι περιλήψεις του BERT μοντέλου αξιολογούνται ως παρόμοιες αλλά λίγο χειρότερες από τις περιλήψεις αναφοράς. Αυτό υποδεικνύει πως ο αλγόριθμός μας δίνει την δυνατότητα παραγωγής κειμένου που διαβάζεται εύκολα και έχει εσωτερική συνοχή. Όμως, οι περιλήψεις του BERT μοντέλου είναι αρκετά χειρότερες όσον αφορά τις μετρικές σχετικότητας και συνέπειας. Αυτό σημαίνει ότι, συγκριτικά με τις περιλήψεις αναφοράς, οι αυτόματα παραχθείσες ελεύθερες περιλήψεις δεν συμπεριλαμβάνουν όλα τα σημαντικά δεδομένα της δικαστικής απόφασης και επίσης είναι πιο πιθανόν να είναι πραγματολογικά ασυνεπείς με αυτές - κάνοντας αναφορά σε ανύπαρχτα γεγονότα ή σε λάθος νόμους.

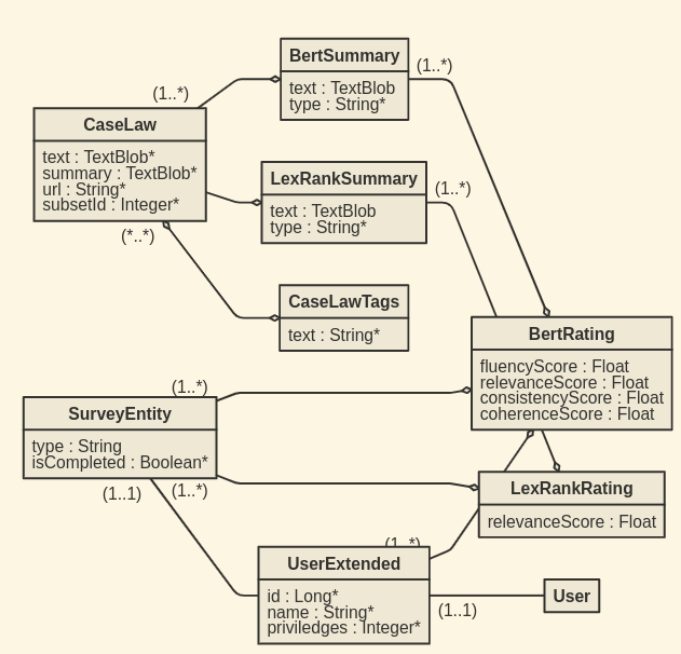
Οι εξαγωγικές μέθοδοι έχουν μέτρια αποτελέσματα, παράγοντας σκορ σχετικότητας συγκρίσιμα με αυτά των περιλήψεων αναφοράς. Ο απλός LexRank αλγόριθμος και η biased παραλλαγή του δεν φαίνεται να έχουν ουσιαστικές διαφορές.

0.5 Εφαρμογή Διαδικτύου

Για τους σκοπούς της εύκολης προγραμματιστικής διάθεσης του συλλεχθέντος συνόλου δεδομένων καθώς και της διεξαγωγής μελέτης ανθρώπινης αξιολόγησης, αναπτύξαμε μια εφαρμογή διαδικτύου χρησιμοποιώντας το λογισμικό **JHipster**. Η αναπτυχθείσα εφαρμογή έχει ενσωματωμένες δυνατότητες αυτόματου building και ελέγχου.

0.5.1 Μοντέλο Δεδομένων Εφαρμογής

Δομούμε τα δεδομένα κάνοντας χρήση της γλώσσας JDL (JHipster Domain Language ώστε να μπορούμε να αξιοποιήσουμε τις δυνατότητες αυτόματης παραγωγής κώδικα του JHipster. Η δομή των δεδομένων δίνεται στο Σχήμα 6



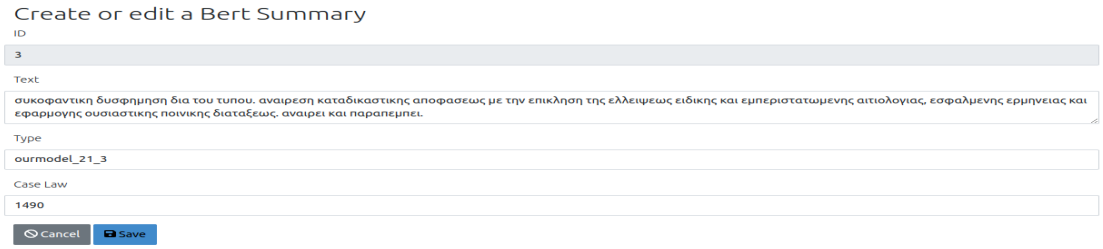
Σχήμα 6. *Αναπαράσταση του μοντέλου των δεδομένων μας, όπου αναγράφονται τα επιμέρους πεδία του κάθε αντικειμένου καθώς και οι σχέσεις μεταξύ των αντικειμένων.*

Κάθε μελέτη αξιολόγησης αντιστοιχίζεται μοναδικά με έναν χρήστη και πολλαπλές BERT ή/και LexRank αξιολογήσεις από τις οποίες η καθε μία αντιστοιχίζεται (με μια σχέση Πολλά-προς-Ένα) με μια περίληψη. Κάθε περίληψη αντιστοιχίζεται με μια δικαστική απόφαση, μέσω μιας σχέσης Πολλά-προς-Ένα. Τέλος κάθε δικαστική απόφαση, σχετίζεται με μια σχέση Πολλά-προς-Πολλά με τις ετικέτες κατηγορίας.

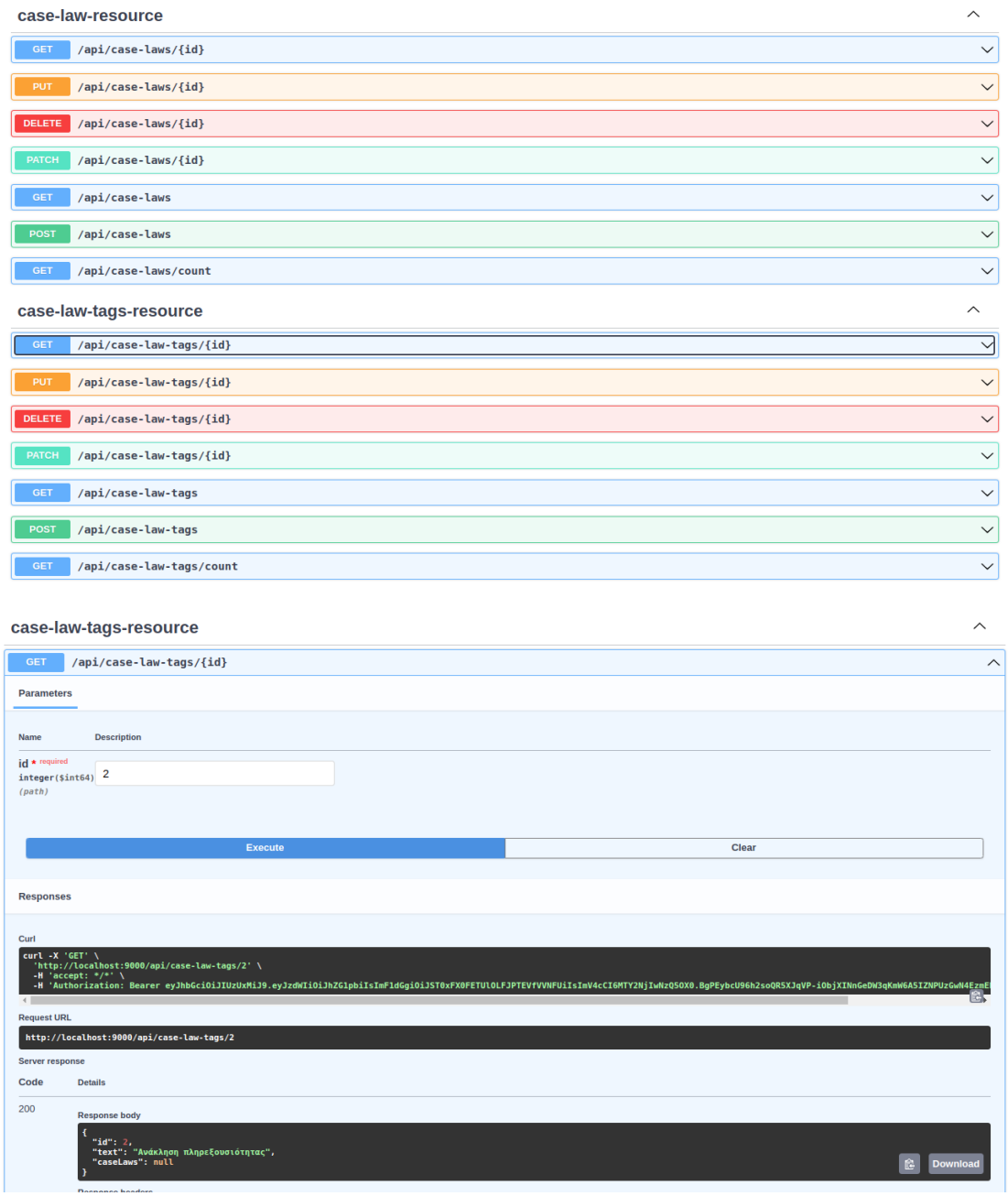
Το παραπάνω σχήμα, δίνει την δυνατότητα διεξαγωγής μελετών όπου στον κάθε χρήστη αντιστοιχεί μια προσωποποιημένη (και ενδεχομένως διαφορετική από τους άλλους χρήστες) μελέτη αξιολόγησης.

0.5.2 Προγραμματιστική Διεπαφή Εφαρμογής

Έχοντας ορίσει το μοντέλο δεδομένων, σχεδιάσαμε και υλοποιήσαμε ένα RESTful API για εισαγωγή, ανάγνωση, ενημέρωση και διαγραφή δεδομένων από την βάση (CRUD operations). Οι κατάλληλα διαπιστευμένοι χρήστες μπορούν να έχουν πρόσβαση σε μια φιλική για τον χρήστη διεπαφή διαδικτύου, μέσω της οποίας μπορούν να κάνουν χρήση των προαναφερθέντων λειτουργιών (Σχήμα 7)



Σχήμα 7. *Δημιουργία/ενημέρωση μιας περίληψης BERT από την διεπαφή της διαδικτυακής εφαρμογής μας.*

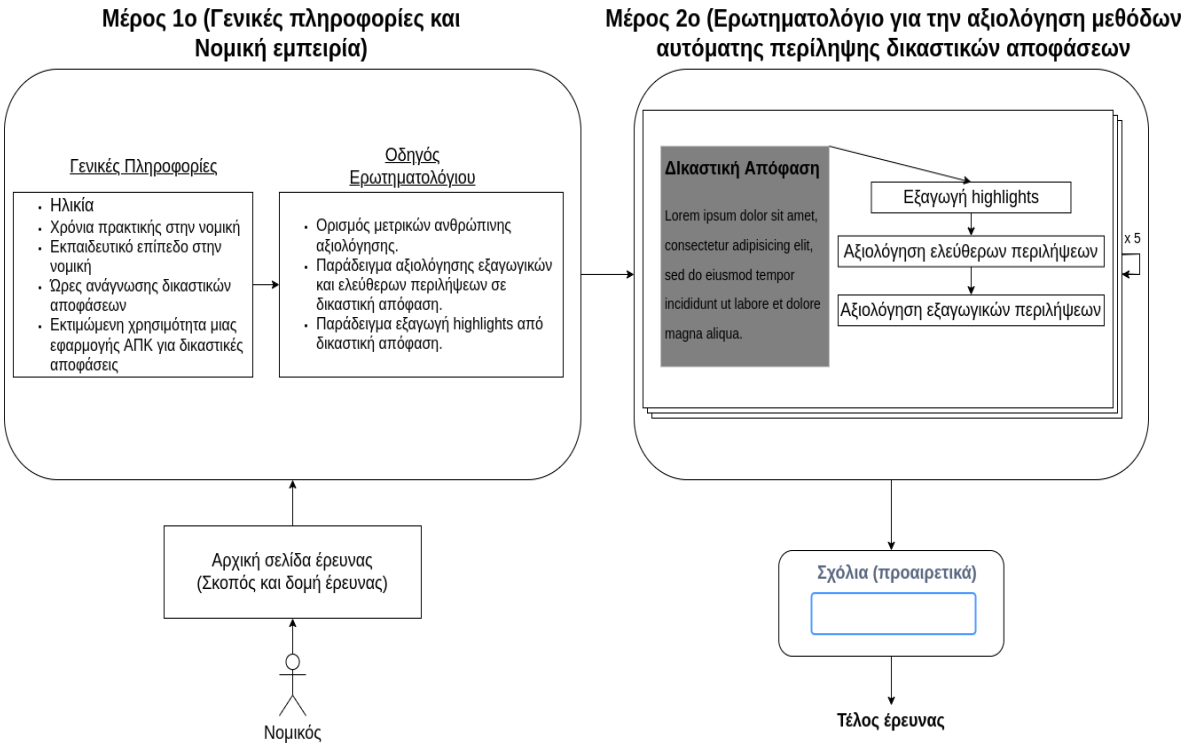


Σχήμα 8. *Αποτύπωση οθόνης των εγγράφων τεκμηρίωσης του API (πάνω) και της εκτέλεσης ενός API request μέσω της διεπαφής του SwaggerUI (κάτω).*

Οι χρήστες μπορούν να έρθουν σε απευθείας επαφή με το αναπτυχθέν REST API και τα πλήρη έγγραφα τεκμηρίωσής του, όπως φαίνεται στο Σχήμα 8, μέσω της διεπαφής που αναπτύξαμε κάνοντας χρήση του λογισμικού **SwaggerUI**.

0.5.3 Σελίδα διεξαγωγής μελέτης ανθρώπινης αξιολόγησης

Για την παραγωγή της ιστοσελίδας διεξαγωγής της μελέτης ανθρώπινης αξιολόγησης, κάνουμε χρήση της βιβλιοθήκης δημιουργίας διαδικτυακής διεπαφής για διεξαγωγή μελετών SurveyJS. Το διάγραμμα ροής της διαδικτυακής εφαρμογής για την διεξαγωγή της μελέτης που αναπτύξαμε, δίνεται στο Σχήμα 9. Στο πρώτο μέρος της μελέτης αξιολόγησης που αναπτύξαμε εξηγείται η δομή της μελέτης, δίνεται ένας οδηγός απάντησης των ερωτήσεων και τα συμμετέχοντα άτομα απαντούν ερωτήσεις σχετικές με το γνωστικό τους επίπεδο στον χώρο της νομικής. Στο δεύτερο μέρος, οι συμμετέχουσες 1) αρχικά εξάγουν επισημειώσεις για το ποια είναι τα σημαντικά αποσπάσματα του κειμένου 2) αξιολογούν ελεύθερες περιλήψεις και 3) αξιολογούν εξαγωγικές περιλήψεις, με την διαδικασία να επαναλαμβάνεται για όλες τις δικαστικές αποφάσεις που έχουν ανατεθεί στο κάθε άτομο.



Σχήμα 9. *Διάγραμμα ροής του διαδικτυακού περιβάλλοντος διεπαφής στην σελίδα διεξαγωγής της μελέτης ανθρώπινης αξιολόγησης.*

0.6 Επίλογος

0.6.1 Συζήτηση

Η Αυτόματη Περίληψη Κειμένου (ΑΠΚ) είναι μια ενεργή ερευνητική περιοχή, όπου διάφορες μέθοδοι αξιοποιούνται για να συνοψιστεί ένα κείμενο, χωρίς να χαθούν οι πιο σημα-

ντικές πληροφορίες εντός του. Οι μέθοδοι ΑΠΚ μπορούν να είναι χρήσιμες σε περιπτώσεις όπου η ανάγνωση ολόκληρου του κειμένου είναι εξαιρετικά χρονοβόρα, αλλά και σε διαδεδομένες εφαρμογές όπως η παραγωγή περιλήψεων διαδικτυακών ιστοσελίδων για μηχανές αναζήτησης, περίληψη βιβλίων/σεναρίων για την ευκολότερη παραγωγή μεταδεδομένων και σημασιολογική σύνδεση/αναζήτηση κειμένων, αυτόματη παραγωγή περιλήψεων για επιχειρήσεις βασισμένη σε πολλές, ξεχωριστές κριτικές, κλπ.

Στην νομολογία, η ανάγκη για αξιόπιστα συστήματα ΑΠΚ είναι μεγάλη. Νομικοί, δικαστές και ερευνητές χρειάζεται να ψάχνουν μη-αυτοματοποιημένα για νόμους και νομολογίες δικαστηρίων σχετικούς με την υπόθεση στην οποία δουλεύουν. Η περίληψη νομικών κειμένων είναι δύσκολη υπόθεση, αφού τα κείμενα αυτά έχουν μεγάλο μέγεθος, περιέχουν νομική ορολογία και προϋποθέτουν νομική γνώση για την κατανόησή τους. Στην περίπτωση της περίληψης δικαστικών αποφάσεων, η δουλειά αυτή συνήθως εναποτίθεται σε εξειδικευμένους νομικούς συντάκτες ή δικαστές που εργάζονται στο εκάστοτε δικαστήριο.

Στην παρούσα εργασία, υλοποιούμε διάφορες μεθόδους αυτόματης περίληψης δικαστικών αποφάσεων από Ελληνικά δικαστήρια. Επειδή δεν υπήρχε αντίστοιχο σύνολο δεδομένων, αναπτύξαμε λογισμικό αυτόματης συλλογής δεδομένων δικαστικών αποφάσεων από τις ιστοσελίδες των δικαστηρίων: 1) του Αρείου Πάγου και 2) του Συμβουλίου της Επικρατείας (ΣΤΕ). Το σύνολο δεδομένων από το Συμβούλιο της Επικρατείας δεν χρησιμοποιείται για την εκπαίδευση ή αξιολόγηση των μεθόδων μας καθώς 1) δεν περιλαμβάνει περιλήψεις των δικαστικών αποφάσεων και 2) περιέχει λάθη κατά την διαδικασία της ψηφιοποίησης των κειμένων. Συγκρίνουμε το σύνολο δεδομένων του Αρείου Πάγου με αντίστοιχα σύνολα δεδομένων για ΑΠΚ χρησιμοποιώντας σχετικές μετρικές της βιβλιογραφίας.

Αναπτύξαμε ένα σύστημα *εξαγωγικής περίληψης* βασισμένο στον αλγόριθμο LexRank, το οποίο εξάγει τις σημαντικές προτάσεις του κειμένου. Ακόμη, υλοποιούμε και συγκρίνουμε διάφορες παραλλαγές της συνάρτησης ομοιότητας προτάσεων που χρησιμοποιεί ο αλγόριθμος.

Ακόμη, αναπτύξαμε ένα σύστημα *ελεύθερης περίληψης* βασισμένο στο σχήμα Κωδικοποιητή - Αποκωδικοποιητή με αρχιτεκτονική BERT. Το μοντέλο κάνει χρήση βαρών προεκπαιδευμένων σε νομικά κείμενα στην Ελληνική γλώσσα, τα οποία διατίθενται ελεύθερα σαν λογισμικό ανοιχτού κώδικα [69]. Το μοντέλο επανεκπαιδεύεται στο πρόβλημα της *ελεύθερης περίληψης* κάνοντας χρήση του συνόλου δεδομένων από τον Άρειο Πάγο.

Οι μέθοδοί μας αξιολογούνται αυτόματα μέσω των ROUGE μετρικών. Βρίσκουμε ότι το εξαγωγικό σύστημα αυτόματης περίληψης που υλοποιήσαμε, αποδίδει καλύτερα σε σχέση με ένα σύστημα τυχαίας εξαγωγής προτάσεων, αλλά έχει περιθώρια βελτίωσης. Ακόμη, Βρίσκουμε πως η εξειδικευμένη σε δικαστικές αποφάσεις προ-επεξεργασία κειμένου καθώς και η συμπερίληψη των μεταδεδομένων κατηγορίας της απόφασης στην δημιουργία ελεύθερης περίληψης, βελτιώνει την επίδοση του μοντέλου.

Διεξάγουμε μελέτη ανθρώπινης αξιολόγησης με νομικούς οι οποίοι αξιολογούν τις περιλήψεις (αυτόματες & αναφοράς) βάσει της σχετικότητας, συνέπειας, συνοχής και ευφράδειάς τους. Τα αποτελέσματα για τις μεθόδους *εξαγωγικής περίληψης* δείχνουν υποσχόμενα, αφού φαίνεται να καταφέρνουν να εξάγουν κάποια από τα σχετικά αποσπάσματα των κειμένων. Οι μέθοδοι *ελεύθερης περίληψης* παράγουν εύγλωττες, έχουσες συνοχή περιλήψεις οι όποιες όμως, συχνά, δεν είναι συνεπείς πραγματολογικά με το κείμενο της δικαστικής απόφασης

και δεν εμπεριέχουν μεγάλο μέρος σημαντικής πληροφορίας που βρίσκεται σε αυτήν. Δεδομένου του πόσο χρονοβόρα είναι για τους νομικούς που συμμετείχαν στην έρευνα μας η εργασιακή τους υποχρέωση να διαβάζουν δικαστικές αποφάσεις, πιστεύουμε πως η περαιτέρω έρευνα για μεθόδους ΑΠΚ νομικών κειμένων που είναι καλύτερα στο να συλλαμβάνουν τις σημαντικές πληροφορίες και πιο συνεπή με το δοθέν κείμενο, θα ήταν ιδιαιτέρως χρήσιμη για τους νομικούς στην Ελλάδα.

0.6.2 Μελλοντική Δουλειά

Σαν μελλοντική δουλειά, η παρούσα εργασία μπορεί να επεκταθεί στους παρακάτω τομείς:

- Δοκιμές με διαφορετικές αρχιτεκτονικές νευρωνικών δικτύων: όπως ιεραρχικά μοντέλα Transformer που επεξεργάζονται τμηματικά και ιεραρχικά κάθε μέρος του κειμένου [152], ή διαφορετικοί μηχανισμοί προσοχής οι οποίοι μειώνουν την τετραγωνική υπολογιστική πολυπλοκότητα του μηχανισμού self-attention [12, 65].

- Εισάγωντας περιορισμούς κατά την εκπαίδευση του μοντέλου που ευνοούν την παραγωγή περιλήψεων που είναι πραγματολογικά συνεπείς με το προς σύνοψη κείμενο. Αυτό μπορεί να επιτευχθεί μέσω εισαγωγικής «αρνητικών παραδειγμάτων» (negative samples) κατά την διαδικασία εκπαίδευσης [76], ή μέσω της από κοινού εκπαίδευσης στην περίληψη αλλά και την παραγωγή ερωτήσεων-απαντήσεων σχετικών με το κείμενο εισόδου [139, 95].

- Διεξαγωγή μελέτης μεγαλύτερου δείγματος η οποία μπορεί να δώσει χρήσιμες πληροφορίες για την διάρθρωση των ελληνικών δικαστικών κειμένων, μέσω διαδικασιών εξαγωγής νομικών επιχειρημάτων [145, 144] ή της αιτιολογίας της δικαστικής απόφασης [22]. Το σύνολο δεδομένων αυτό μπορεί να χρησιμοποιηθεί για περαιτέρω πειραματισμούς *εξαγωγικής* αλλά και *ελεύθερης* περίληψης.

Chapter 1

Introduction

We live in the era of information, where both the quantity of information and the need to process it are steadily increasing. More documents are getting digitized and integrated into various applications, where the user is able to download, search and process large amounts of data. Many AI companies utilize the growing size of open data to train Artificial Intelligence algorithms that can be applied in a variety of domains. However, the data are valuable - for both machines and humans - only if they are usable either directly or in a downstream task. Therefore, a summary of an input text - as defined by D.Radev [110] "(...) a text that is produced from one or more texts, that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually significantly less than that." can be very valuable as a summary can reduce significantly the size of a text without losing its meaning.

The Artificial Intelligence (AI) research community has been interested, for many decades, in finding ways of Automatically Summarizing Textual (ATS) data, particularly news articles and scientific papers. In this pursuit, many approaches ranging from computational linguistics to machine learning have been tried. Typical ATS approaches attempted to extract the most important sentences from each text (*extractive summarization*), as generating new text based on an input context (*abstractive summarization*) was not achievable with classical machine learning or computational linguistics methods. However, the recent rise in popularity of Deep Machine Learning (ML) Networks - which was the result of increased computational power in modern day computers and Big Data - has enabled better Language Generation through which *abstractive summarization* is feasible. Deep ML networks are trained on vast amounts of textual data and are, afterwards, open-sourced thereby giving everyone the ability to use pre-trained Language Models for their specific downstream task, such as text summarization. Recent advances in ML Network architectures, such as the Transformer architecture [137] can be more computationally efficient and thus enable efficient training on larger amounts of data. BERT (*Bidirectional Encoder Representations from Transformers*) [29] is a Transformer-based Language Model architecture that has been widely used in research and business applications.

In our work, we evaluate automatic summarization algorithms on Legal Texts; and specifically court decisions. The law domain has been, comparatively to other sectors, slow to embrace the benefits of document digitization. However, in the recent years an

increasing number of courts digitize part of their decisions and make them available through the Internet, providing law practitioners, judges and scholars with quick access to an enormous number of past court decisions. However, a person searching through lists of past-court judgements still has to browse a very large number of them in order to extract, using their law knowledge and experience, relevant passages and gain in-depth understanding of the applicable case-laws. Summaries of court judgements are focused on the main points of the court judgements and considerably shorter in length than the main court judgement text, enabling less time-consuming search for relevant court judgements.

Up until the 1990s, most of the summarization and information retrieval was done manually by specialized legal editors. However, soon after many user-friendly software [109, 36] were developed in order to partially automate those processes. Nowadays, more than 2000 Legal-AI startups are in the market [130].

1.1 Legal Text

Legal text refers to texts, of varying purposes, that are related to law - either by the authority of the author, their topic being connected with the rights and obligations of institutions and individuals, their cross-reference of other legal texts, etc. The most common types of legal texts [133], sorted by their importance as sources of law [131], include:

- **Constitutions:** which consist of fundamental principles that concern the way a state is governed.
- **Statutes:** which are laws which are enacted by a legislative body and regulates what can be done and what cannot be done according to law. Typically, they are organized in **codes** which cover specific subjects such as the Penal Code.
- **Court decisions:** which include 1) *court rulings*: the interlocutory or final outcome of a trial, 2) *orders*: the court ruling in a written form, 3) *opinions/judgements*: which is present in written format the court's opinion about the facts relevant to the trial, applicable laws etc. The aforementioned terms are often used interchangeably[1].
- **Contracts:** which are mutual agreements between parties and by which mutual obligations are generated.

Private legal texts has evolved over thousands of years. [134] trace the first contacts at *Sumer* around 5000 years ago; where contracts, wills, deeds were inscribed in clay tables. Similarly, public legal texts such as statutes first appeared in Mesopotamia with King Ur-Nammu's laws and later the Code of Hammurabi being well known examples.

In Ancient Greece, private law was concerned with inheritance, commerce and contracts, while public law regulated how an Ancient Greek citizen (*politis*) lived his life in

[1] In the rest of our work, we will also use the terms interchangeably as the dataset we have constructed contains documents which include both the court judgement and the court order.

an Ancient Greek city-state (*politeia*). Each city-state had major differences in their laws, based on the way they were organized, the nature of public rhetoric, law literacy and courts[141] and socio-political factors that regulated who was considered a citizen. One of the earliest mentions of Greek law can be found in Homer's *Iliad* and *Odyssey* in which: the transition from informal law to a *politeia*, where an institutionalized legal system is created, is highlighted [2]. The first actual inscriptions of Ancient Greek law can be found in the laws of *Gortyn* and *Dreros*, both ancient cities in Crete. The public law code of *Draco* (620 BC) regulated life in Ancient Athens known for its harshness, until they were reformed by the law code of *Solon* (593 BC). Many philosophers were interested in law; Plato's *Laws* analyzes practical and philosophical problems in Ancient Greek law, Plato's *Apology of Socrates* is a fictional retelling of Socrate's legal defense during his trial, Aristotle's philosophy of law can be found in his books: *Politics*, *Nicomachean Ethics* and *Rhetoric*.

The first known written legal text in Ancient Rome is the *Law of the Twelve Tables*, which dates from the mid-fifth century BC and arose from the political struggles between the class of *plebeians* and the *patricians*. It was based on Ancient Greek law and governed all areas of law: civil procedure, public law and private law. Other significant laws in the Roman legal system include: *Lex Canuleia*, *Leges Liciinae Sextiae*, *Lex Ogulnia*, *Lex Hortensia* which regulated the rights of the *Plebeians*. The legacy of Rome's legal system cannot be overstated. The Byzantine Empire largely followed Roman Law but adjusted it to the Orthodox and Hellenistic traditions. Rome's public law influenced greatly the public law of many Nations in the Medieval and Modern eras, with the exception of England which developed its own version of *Common Law*.

Modern legal texts can have varying amount of internal structure, and a fair amount of cross-document references. Moreover, the language used is formal and makes use of law specific terminology. Different nations' legal systems and texts although sharing similarities, have major differences.

In any case, a well-functioning society requires a culture of respect for the law and institutions that oversee the observance of those laws (*rule of law*). Indeed, it can be claimed that modern civilization, as we know it, depends on the existence of the rule of law. Therefore, the laws and their interpretations, as they are expressed through court judgements, are very important both as a regulating force of its citizen's everyday life, and as pillar upon which the existence of modern civilization is based.

However, a view of the *rule of law* that is only procedural and thus irrespective of the laws' actual content cannot lead to a liberal society. Economist and legal theorist F.A.Hayek, points [132, p. 230-235] that the transition from customs to modern *rule of law* was a result of laws being general and abstract while also specifying the framework within individuals can make their decisions. Thus, supposing the laws are universally applied, then each individual can determine the legal outcomes of their actions[2].

[2]For a view that accepts the value of both laws-based, formal definitions of the *rule of law*, while also making the case for the importance of legal procedures and institutions (such as courts), the reader is referred to [138].

1.2 Thesis Motivation

In our work, we attempt to build methods for summarizing greek court decisions and evaluate those methods using both automatic metrics and human evaluation. Our work is motivated by the following remarks:

- Legal Texts have significant differences from texts that are typically used in ATS research, news articles. Legal texts are comparatively longer, make use of specialized legal terminology and often cite other legal texts.

- The average law practitioner has to spend many hours per week to search for, browse, read and analyze judicial rulings which are relevant to a court case he works on.

- Applying domain-agnostic ATS methods to legal text, can miss valuable domain-specific information or be downright unfeasible due to significant differences in legal text which require preprocessing and adapting most methods.

- In Greece, the digitization of Greek legal documents is underdeveloped. However, recent attempts to model and mine legal documents [66, 5] make accessing and searching Greek legislature easier, although to this date we do not know of any comparable tools for Greek judicial decisions. By developing methods for automatic summarization of Greek court judgements we aim to evaluate whether those methods can provide valuable information which can then be used for downstream tasks in the aforementioned platforms, such as interlinking and semantic search of court judgements.

1.3 Thesis Contribution

In our work, we evaluate algorithms for both *extractive* and *abstractive* text summarization of legal documents. Our thesis contributions are the following:

- Development of automated scraping scripts for downloading court decisions from the Greek Court of Cassation and the Greek Council of State. After following a data deduplication protocol, the scrapped data are organized into a dataset that we make openly available[3].

- Analysis of the documents and comparison with other text summarization datasets, using metrics proposed in the ATS literature.

- Implementation of an *Extractive Summarizer* using the LexRank [34] algorithm and its variant; the Biased LexRank [101], while using a domain-specific preprocessing pipeline.

[3] https://github.com/DominusTea/LegalSum-Dataset/releases/tag/v1.0.0

- Implementation of an *Abstractive Summarizer* using an Encoder-Decoder BERT architecture. We make use of the BERT model pretrained on legal-texts and wikipedia open-sourced by [69]. We test several methods of domain-informed preprocessing which aim to fit as much relevant information as possible into the model's input which is limited in size. Furthermore, we asses the value of informing our model's summary generation with the inclusion of the court judgement's tags in the input.

- We evaluate all algorithms using automated metrics, whose preprocessing pipeline is also adapted to our Greek legal court decisions domain. We also conduct a human evaluation study of our methods and measure the correlation between human evaluators scores and the scores provided by automated evaluation metrics.

- For the purposes of human evaluation and the dissemination of the collected data, we develop a web application. We also make the data available through a REST API.

1.4 Thesis Outline

The thesis is outlined as following:

- Chapter 2 provides a brief overview of Machine Learning. A short history of the path from *Artificial Intelligence* to *Machine Learning* outlined. General definitions and typology for most current machine learning applications is provided. Subsequently, machine learning methods are defined; starting from simple networks, and moving on to *deep learning*, and their comparative advantages and disadvantages are analyzed. Furthermore, algorithms for training neural networks and regularization methods are explained.

- Chapter 3 provides background knowledge on *Natural Language Processing* (NLP) methods. Common NLP applications for general and legal text are listed and analyzed. The preprocessing interface between a text and a computer is expanded on in detail. Subsequently, the construction of rich word/document representations is examined, through methods common in the literature. Finally, the problem of modeling language for text generation is described.

- Chapter 4 is concerned with *Automatic Text Summarization* (ATS) methods for general-purpose and/or legal text. An extended list of the datasets used for automatic text summarization is provided, while their common biases are discussed. Extractive and abstractive automatic methods of summarizing text are defined. A detailed analysis of automated evaluation metrics for summarization and their correlation with human judgement is included. We make separate mention of how general-purpose text summarization work fits into summarizing legal text.

- Chapter 5 contains our proposed methods and contribution to the court judgements automatic summarization problem. The data collection process is defined and the dataset collected is analyzed. We introduce the methods used for summarizing

legal text in our dataset and explain the pre/post processing pipeline. Experimental results for both automated and human evaluation are provided and discussed.

- Chapter 7 explains the implementation of our Web application. The technology stack used to develop and deploy our application is analyzed. We provide examples of our database schema and the API we developed. Finally, we expand, in detail, on the web interface for our human evaluation study.
- Chapter 8 provides our conclusions on our findings, the limitations of our work, and discusses possible future work.

Chapter 2

A Brief Overview of Modern Machine Learning Methods

2.1 Introduction

From research to commercial applications Artificial Intelligence (AI), and more specifically its subfield; Machine Learning (ML), seems to be everywhere nowadays - capturing the public's interest with its seemingly wondrous achievements, while also receiving criticism when it falls short of expectations. Common modern applications of Machine Learning include among others: text applications such as automatic text translation [9] & summarization [77], image/video automatic generation applications: [113, 51], music generation and accompaniment [30], autonomous driving [148], applications in simulating protein folding [56], material science applications [124].

Artificial Intelligence (AI) can be defined as the scientific discipline, in the field of computer science, that tries to develop and evaluate methods of enabling computers to exhibit behavior corresponding to - but sometimes even surpassing - the human capabilities of learning, reasoning and acting rationally. However, a discipline which deals with such abstract notions cannot be expected to have an all-encompassing definition, as is evident with the various definitions collected in the Artificial Intelligence textbook by Russel & Norvig [119]. Subsequently, many approaches have been applied in Artificial Intelligence research such as: Biological equivalency/Cognitive Science methods, Knowledge-based methods, Symbolic methods, statistical methods, et alia.

Machine Learning (ML) is a subfield of Artificial Intelligence which studies how to best make AI-algorithms adaptive to a set of data, commonly referred to as *training-data*. Either by purely statistical methods or by a combination of statistical and symbolic methods, a machine learning program "learns by experience" by reasoning on patterns found in the *training data* and making inferences from them. A special subcategory of Machine Learning is *Deep Machine Learning*, which leverages very large amounts of data in order to train computer algorithms to generate very resource-heavy representations and reason using them.

In this chapter, we will: 1) give a brief overview of the history of Artificial Intelligence and Machine Learning, 2) provide definitions for commonly used typologies of Machine Learning, 3) describe basic and deep Machine Learning methods and 4) explain how a

neural network is trained.

2.2 A Brief history of Machine Learning

The term *Artificial Intelligence*, at least in its modern sense of the word, was first coined by John McCarthy in the 1956 "*Dartmouth Summer Research Project on Artificial Intelligence*" workshop. The Dartmouth workshop is credited as being the founding event of Artificial Intelligence as a independent research discipline, uniting research previously done under the field terms of: control theory, cybernetics, automata theory, et alia.

However, authors in [119] consider the 1943 paper "A Logical Calculus Of The Ideas Immanent In Nervous Activity" by Warren McCulloch and Walter Pitts [88] as the first research paper in what is today considered Artificial Intelligence. There, the authors inspired by a theoretic neurophysiological model of the brain, describe a model of neurons (later defined as perceptrons in AI literature) that get excited by neighbouring synapses rendering the neuron on or off. Furthermore, they show that their model is computationally equivalent with a Turing Machine with each neuron activation corresponding to the truth of a propositional logic statement, thereby bridging their work with that of Alan Turing and the Propositional Logics developed by Bertrand Russel and Alfred Whitehead.

A theoretical model of intelligent machines has been proposed and soon the question whether thinking machines can exist becomes a serious scientific and philosophical question. Alan Turing's 1950 paper on "*Computing Machinery and Intelligence*" [135] asks the same question, but also the question of identifying intelligence in a machine (the famous Turing Test), while also introducing early versions of what will later be called *Reinforcement Learning* and *Genetic Algorithms*.

Several distinct research programs began, as optimism for the future of AI was high. In 1958 the first perceptron network was implemented by Frank Rosenblatt in the Cornell Aeronautical Laboratory [116], while in 1962 he proved [117] that some learning rules for the single layer perceptron networks can always converge to a solution, provided that solution existed for the given network. However, as it was shown in the 1969 textbook on Perceptrons [92] by Marvin Minksy and Seymour Papert, in most cases a solution for any single layer perceptron network did not exist. Single-layer networks could not mimic a XOR gate.

The research on Neural Networks slowed significantly, while other research directions were explored. *Expert Systems* were systems where human prior knowledge was encoded in a knowledge database that the system used to produce inferences. These systems used some type of symbolic logic, and often incorporated uncertainty in the form of fuzzy logic. Expert Systems were introduced by the 1965 Stanford *Heuristic Programming Project*, leading to applications in medical consulting [6] or even chatbots [140] With the creation of the LISP and PROLOG Logical Programming languages, the development of expert AI systems became easier and extremely popular, both in research and commercially. However, it soon became evident that knowledge acquisition and programming was not an easy task. Even in the case of purely logical programming AI systems with little prior knowledge required, the state space of solutions that a symbolic logic solver had to search

greatly exceeded the computational resources that existed.

What followed is commonly referred to as "*AI winter*" [93], when public and private funding were largely cut after what were seen as failures of successfully implementing AI systems in real-life. The "*AI winter*" was followed by an "*AI spring*" (1990s-present), as (once again) more researchers became interested in connectionist Machine Learning models, that leveraged statistics to make predictions, over the paradigms of expert systems and symbolic models. Multiple layers of perceptrons (MLP networks) were shown to be universal approximators of every function [54], while the Backpropagation algorithm ; an algorithm to *efficiently* train feedforward MLP networks [1] , became more popular.

Nowadays, Machine Learning is an established and well funded scientific field. Large technology corporations have invested for machine learning solutions in various fields; finance, health, transportation, translation, among others.

2.3 A Taxonomy for Machine Learning

As we have previously discussed, there have been many approaches in Machine Learning. Subsequently, the various conceptualizations of Machine Learning learning lead to different taxonomies. A widely accepted one, is based on the type of feedback the Machine Learning model uses to learn.

2.3.1 Supervised Learning

Supervised Learning is one of the most common type of Machine Learning feedback. The Machine Learning model is presented with input-desired outputs pairs of data (*labeled data)* and is asked to learn the mapping between input and output (reference) data. Supervised Learning can be a useful approach in problems where the output of various types of inputs can be easily observable and thus collectable into a *training data* set. However, this is not always possible.

Supervised Learning can be divided into two categories depending on the type of output data given. In some cases, a model learns better if, in conjuction to learning what to predict, it learns what not to predict (*Constrastive Learning)*.

2.3.2 Unsupervised Learning

Unsupervised Learning involves only a set of input data, without the corresponding desired output data (*unlabeled data*). The Machine Leaning model is asked to learn patterns in the input data, that can be used in order to *cluster* the input data into different classes, or generate new input data by computing a probability density function over the inputs (*generative models*). Unsupervised Learning can be quite useful in problems where human/automatic labeling of the data cannot be done due to the size of data or the computational resources required.

[1] The backpropagation algorithm was discovered independently multiple times, although it's generally accepted that it was popularized by Rumelhart's paper [118]. For the curious reader, we refer to Schmidhuber's post on the history of the Backpropagation algorithm [59]

2.3.3 Semi-supervised Learning

Semi-supervised Learning can be seen as a combination of *Supervised* and *Unsupervised* Learning. It leverages a large amount of *unlabeled data* and a (usually smaller) amount of *labeled data*. The model is trained on the *labeled data* and uses them to label the *unlabeled data*, thereby utilizing the whole dataset.

2.3.4 Self-supervised Learning

Self-supervised Learning is another way of combining an *unlabeled* set of data, with *supervised learning* techniques. (Pseudo-)Labels are automatically generated for the *unlabeled* data by the characteristics of the data themselves and the model is trained to predict those labels. In cases where those pseudo-labels are easy to generate correctly, this approach can be used to train models on very large sets of *unlabeled* data, that couldn't possibly be labeled manually.

In the case of Natural Language Processing, common ways of generating the *pseudo-labels* are: masking certain words found in the text and train the model to predict, giving a part of the text to the model and asking it to predict its next sentence. Those techniques can also work *contrastively* by asking the model not to predict some other random word or sentence that can be found in the text.

2.3.5 Reinforcement Learning

Reinforcement Learning is different from the approaches we have previously discussed, as the model acts as an agent that learns to model the problem's inner states in a sequence of inputs as well as an optimal policy for every step of the sequence. The agent doesn't use labeled data, but tries to maximize a cumulative reward function, which *reinforces* the desired behaviour of our agent (in reaching the end goal using a small number of steps, minimizing other penalty functions, etc.)

2.3.6 Transfer Learning

Transfer learning is a general way of training machine learning models, which enables the utilization of other previously trained models (*pre-trained models*). In essence, a model is trained using as input data the inner representations of the same data produced by a pre-trained model, often trained to solve a different problem. It is based on the assumption that the inner representations learned by the *pre-trained* model, can be generalized into different domains and tasks (*Representation Learning*).

This technique is very useful when the data or the computational resources are limited. It has become an active area of research, as it can be viewed as a way of generalizing & transferring knowledge between different domains.

2.4 Basic Concepts and Methods in Machine Learning

We will now introduce and define several concepts and methods that are essential for an understanding of how most machine learning models are trained. Henceforth, we will denote with $\mathbf{x} \in \mathbb{R}^n$ the input vector of n-dimensions, with $\hat{\mathbf{y}} = f(\mathbf{x}; \boldsymbol{\theta}) \in \mathbb{R}^m$ the output of a machine learning model with parameters $\boldsymbol{\theta}$, and with $\mathbf{y} \in \mathbb{R}^m$ the desired output of the model. Furthermore, let $\mathbf{X}$ denote the set of all input vectors $\mathbf{x}$, and $\mathbf{Y}$, $\hat{\mathbf{Y}}$ the set of all corresponding output reference and predicted vectors, respectively. We further denote with N the cardinality of $\mathbf{X}$.

2.4.1 Loss Functions

A loss function measures how well a machine learning model maps the input data to the desired output data. Higher loss function values correspond to model instances that don't fit the data as well as model instances with lower loss function values.

A per-instance loss function $L(f(\mathbf{x}_i; \boldsymbol{\theta}), \mathbf{y}_i)$ measures the loss in a single instance of input data $\mathbf{x}_i$, model output $\hat{\mathbf{y}}_i$ and reference output $\mathbf{y}_i$. We can also define the loss function over the whole dataset by averaging the single-instance loss values:

$$\mathcal{L}(\hat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{N} \sum_{i=1}^{N} L(\mathbf{y}_i, \hat{\mathbf{y}}_i)$$

The goal of training a machine learning model, is to adjust its parameters $\boldsymbol{\theta}$ so that the loss function is minimized:

$$\boldsymbol{\theta}^* = argmin_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{Y}, \hat{\mathbf{Y}})$$

Different loss functions have been proposed that may fit different types of data better:

- **Mean Square Error:** In regression problems, where the model is asked to predict vector values as output, we can define the means square error (MSE) as follows:

  $$\mathcal{L}_{\text{MSE}}(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|^2$$

  The square on distance metrics assigns proportionally bigger loss values for distances that are bigger, but is susceptible to outlier data.

- **Mean Absolute Error:** Similarly to the MSE error, if we omit the square on the distance metric we can define the mean absolute error (MAE) as follows:

  $$\mathcal{L}_{\text{MAE}}(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|$$

  The MAE loss function has the disadvantage of not being differentiable at $\mathbf{y} = \hat{\mathbf{y}}$, leading to uncomputability of the gradient at this point unless the loss function is altered locally.

- **(Categorical) Cross-Entropy Loss:** In cases where a machine learning model is asked to classify input vectors $\mathbf{x}_i$ into a category c_i from a set of M-categories C, a cross-entropy loss is used. This loss leverages the definition of the cross entropy between two probability density functions:

$$H(f, g) = -\int_x f(x) log(g(x)) dx$$

or in the discrete case:

$$H(p, q) = -\sum_x p(x) log(q(x))$$

which measures the number of bits required to distinguish the true distribution p from the estimated distribution q.

Assuming, the model outputs are the estimated membership probabilities of $\mathbf{x}_i$ to each of the m-classes[2], that is:

$$\hat{\mathbf{y}}_i = [\hat{a_1}, \ldots, \hat{a_M}]$$
$$\text{where: } \hat{a_i} \in [0, 1] \; \forall i \text{ and } \sum_{i=1}^{M} \hat{a_i} = 1$$

and the reference membership distributions is $\mathbf{Y}_i = [a_1, \ldots a_M]$, [3], we can similarly define a cross entropy loss for the model's output:

$$H(\mathbf{y}_i, \hat{\mathbf{y}}_i) = -\sum_{j=1}^{M} a_{i,j} log(\hat{a}_{i,j})$$

2.4.2 Activation Functions

Activation functions are non-linear functions that are used in machine learning models, in order to enable learning classifiers for non-linearly separable classes of data or generally non-linear functions. These functions were conceived as a way of mimicking a neuron's binary state of ON or OFF. In essence, they map the input to a fixed interval [a,b] [4] using the mapping f: $\mathbb{R} \rightarrow [a, b]$. Below, we will list only the most commonly used ones:

- **Signum/Step:** The signum function maps the input to either 0 or 1.

$$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$$

This may be closer to the original conceptualization of a perceptron [88], but it is not commonly used in modern neural networks since the derivative is zero everywhere

[2]This is achieved by applying a softmax function over the outputs of the network. If it has not already been applied to the model's output, then it has to be applied by the loss function.

[3]which is usually to a one-hot vector where the only non-negative element is at the index corresponding to the class that $\mathbf{x}_i$ belongs to

[4]Most commonly a=0,b=1 or a=-1, b=-1. The interval can be also open instead of closed.

except at x=0 where the function is not even differentiable. This prohibits the use of differential algorithms for the learning.

- **Sign:** The sign function maps the input to either -1 or 1.

$$f(x) = \begin{cases} -1, & x < 0 \\ 1, & x \geq 0 \end{cases}$$

The sign function is quite similar to the signum function, and the same reasons for inability to be used in neural networks apply.

- **Sigmoid:** The sigmoid function has the characteristic shape of the greek letter sigma: "ς":

$$f(x) = \frac{1}{1 + e^{-x}}$$

Its main problem is that its derivative has small values when f(x) is close to 0 or 1, which leads to limited learning in those neurons.

- **Hyperbolic Tangent:** The hyperbolic tangent function (tanh) essentially rescales and shifts the sigmoid function:

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

The input is mapped to [-1,1], and the gradient is steeper than the sigmoid's. However the problem of the derivative approaching zero when f(x) is close to -1 or 1, still applies.

- **Rectified Linear Unit:** The Rectified Linear Unit (ReLU) function is one of the most widely used activation functions. It is defined as:

$$f(x) = max(0, x)$$

In contrast with activation functions, ReLU has a scale-invariant derivative and is less computationally expensive. However, its sparse activations come with the cost of neurons with zero valued ReLU activation, getting stuck to a zero-valued gradient.

- **Leaky Rectified Linear Unit:** Leaky ReLU is a version of ReLU which addressing the zero-valued problem of vanilla ReLU when x < 0.

$$f(x) = \begin{cases} ax, & x < 0 \\ x, & x \geq 0 \end{cases}$$

where a is a pre-defined parameter (*leakage factor*).

- **Parametric Rectified Linear Unit:** The Parametric Rectified Linear Unit (PReLU)

function makes PReLU's leakage factor learnable while keeping the same formula:

$$f(x, a) = \begin{cases} ax, & x < 0 \\ x, & x \geq 0 \end{cases}$$

2.4.3 Linear Regression

Linear Regression is a method of modeling linearly the relationship between a number of input variables, $\mathbf{x}$, and a scalar output y. Formally:

$$\hat{y} = w_0 + \sum_{i=1}^{n} w_i x_i = \mathbf{w}^T [1; \mathbf{x}] = \mathbf{w}^T \mathbf{x}_{\text{aug}}$$

where the "aug" subscript denotes a vector extended with the value 1 at index 0. Most commonly, an MSE loss is used to fit the Linear Regression model to the data. Minimizing the loss gives as the optimal linear regression model parameters:

$$\nabla L_{\text{MSE}}(\mathbf{Y}, \hat{\mathbf{Y}}) = 0 \implies$$
$$\mathbf{w}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{Y} = \mathbf{X}^{\dagger} \mathbf{Y}$$

where the † superscript denotes the Moore–Penrose of a matrix.

The optimal parameters can be calculated directly via matrix multiplications and inversions, when the Mooore-Penrose inverse exists. However as more data are inserted to X, it is possible $\mathbf{X}$ loses its full-rank property and, thus $\mathbf{X}^T\mathbf{X}$, becomes non-invertible. Therefore, other optimization approaches can be required such as: SVD factorization, Gradient descent - which can also be computationally more efficient.

Generalized Linear Regression

The Linear Regression model as described above can only act as linear combination of its inputs $\mathbf{X}$. Thus, in order to fit a non-linear function a non-linear transformation of the inputs is needed.

Let $\Phi : \mathbb{R}^n \to \mathbb{R}^M$ be that non-linear transformation:

$$\mathbf{\Phi}(\mathbf{x}) = [\Phi_1(\mathbf{x}), \ldots \Phi_M(\mathbf{x})]^T$$

Similarly to before, the optimal parameters are give by:

$$\mathbf{w}^* = (\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi} \boldsymbol{Y}$$
$$\text{where } \mathbf{\Phi} = [\mathbf{\Phi}(\mathbf{x}_1)^T; \ldots; \mathbf{\Phi}(\mathbf{x}_N)^T]$$

The model can now fit non-linear data if the basis-functions transformation is selected appropriately. However, the model remains linear in its nature.

2.4.4 Logistic Regression

Logistic regression is a method of modeling the relationship between input variables **x** and a categorical output y. In essence, the logistic regression model classifies the input into one class *j*, by modeling the probability of the input belonging to class j:

$$\hat{y}_j = P(C = j \mid \mathbf{x}\,;\, \mathbf{w})$$

Binary Classification

When there are two classes, y can be either 0 or 1 and thus a sigmoid activation function (Section (2.4.2)) suffices the map the model's output to (0,1):

$$\hat{y} = \frac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}}$$

Multinomial Classification

When the model is asked to classify the input to M>2 classes the sigmoid function would not suffice, since the $\hat{y}_j$ probabilities would need to be normalized to have a sum equal to 1. To this end, the softmax function is used as it both maps each $\hat{y}_j$ into the (0,1) interval as well as normalizes all $\hat{y}_j$ to make up a probability function:

$$\hat{y}_j = \frac{e^{\mathbf{w}_j^T\mathbf{x}}}{\sum_j e^{\mathbf{w}_j^T\mathbf{x}}}$$

To measure the discrepancy between our model's predicted class membership distribution and the reference distribution, the cross-entropy loss function (Section: 2.4.1) is used:

$$\mathcal{L}(\mathbf{X}, \mathbf{w}) = -\sum_{i=1}^{N}\sum_{c=1}^{M} y_c^i log(\hat{y}_j(\mathbf{x_i}, \mathbf{w}))$$

where y_c^i is an indicator function of the membership of $\mathbf{x}_i$ to the c-th class. The loss function cannot be minimized analytically and, therefore, a gradient descent-type of method needs to used.

Non-linearly separable classes

The model described so far can only be trained to separate classes that are linearly separable, that is there exists a line, in the case of 2-D data, or a hyperplane, in the case of more dimensions, which can separate the data into two classes. The decision surfaces must be convex, which doesn't always fit the data.

In order to solve this problem, the data can be projected using a non-linear mapping $\Phi(\mathbf{x}) : \mathbb{R}^{\mathbf{n}} \rightarrow \mathbb{R}^{\mathbf{k}}$. The logistic regression model would still be linear in terms of the Φ feature space, but non-linear in the space of the inputs.

2.4.5 Support Vector Machines

Consider the two-classes classification problem. The classification methods we have described so far have attributed equal importance to every point's distance from them decision hyperplane. The SVM algorithm assigns importance only to the points that are "difficult" to classify, i.e the support vectors which are the points closest to the decision hyperplane (see figure 2.1). The goal is to find the maximum-margin hyperplane between

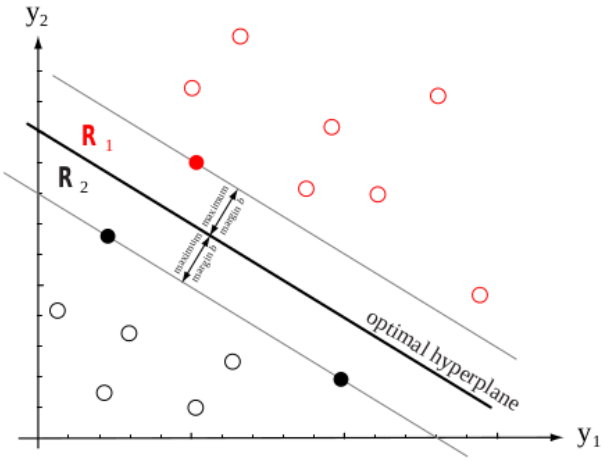


Figure 2.1. *A 2-D classification problem example, and its solution using SVM. Source: [32]*

General Mathematical formulation

Assume a linear perceptron-like model:

$$\hat{y} = \mathbf{w}^T\mathbf{x} + b$$

Also, assume class 1 corresponds to y=1 and class 2 corresponds to y=-1 [5]. Our goal is to find the optimal hyperplane that maximizes the margin between the support vectors of the two classes. If $\mathbf{x}_+$, $\mathbf{x}_-$ are support vectors of class 1 and class 2 respectively then the margin m can by calculated by the formula:

$$m = \frac{2}{||\mathbf{w}||} \text{ since } \mathbf{w}^T\mathbf{x}_+ + b = 1 \text{ and } \mathbf{w}^T\mathbf{x}_- + b = -1$$

Therefore, the optimization goal can be reformulated to

$$min_{\mathbf{w}}||\mathbf{w}||^2$$
$$\text{such that } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \ \forall\, i$$

with the equality holding true for the support vectors.

[5] This is arbitrarily chosen since the equation $c(\mathbf{w}^T\mathbf{x} + b)$ defines the same hyperplane, regardless of c

How to find the support vectors

It can be easily seen, that the support vectors of each class must be in its convex hull[6]. This can't happen due to the convex hull's convexity.

The support vectors of each class can be found by iterating over every possible pairing between a point in the convex hull of class 1 and a point in the convex hull of class 2. The pairing that is selected as the support vectors of each class is the one that corresponds to the largest margin.

Solving the optimization problem

In solving the optimization problem, we can make use of a very useful theorem:

Theorem 2.1 (Representer theorem)**.** *The optimal* $\mathbf{w}$ *that maximizes the hyperplane margin subject to the margin classification constraints, is a linear combination of the training points*

$$\mathbf{w}^* = \sum_{i=1}^{N} a_i y_i \mathbf{x}_i$$

The $a_i = 0$ for all non support-vector points, since the insertion of a non support-vector point into the SVM training problem should not alter the optimal hyperplane [7]. Thus the $\mathbf{w}^*$ square of the norm is:

$$\|\mathbf{w}^*\| = \langle \mathbf{w}^*, \mathbf{w}^* \rangle = \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

An equivalent form of the problem using the previous result is the following:

$$\min_{\mathbf{a}} \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

$$\text{such that } y_i \Big(\sum_{j=1}^{N} a_j y_j \langle \mathbf{x}_j, \mathbf{x}_i \rangle + b \Big) \geq 1, : \forall i$$

The solution of the second formulation has $O(N^3)$ complexity (analogous to the cube of the number of samples), while the first formulation has $O(n^3)$ complexity (analogous to the cube of the dimensions of the input samples). In both cases, the optimization problem can be solved using quadratic programming.

The "Kernel Trick"

The kernel trick refers to a trick that reduces the computational complexity of the second formulation of the quadratic optimization that was presented above. Assume, instead of applying SVM to the original input space of the vectors $\mathbf{X}$, we apply SVM to the

[6] The convex hull of a set X is unique and can be defined as the smallest convex set of its points that contains it, i.e that each other point lies inside the convex area defined by the convex hull

[7] The a_i coefficients are the Lagrange coefficients introduced by the Lagrangian function: $L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} a_i [y_i(\hat{y}_i - 1)]$

space of the input vectors transformed by a non-linear transformation $\Phi(\mathbf{x}) : \mathbb{R}^{\mathbf{n}} \rightarrow \mathbb{R}^{\mathbf{M}}$. This, similarly to previous methods, would enable classification of non-linearly seperable classes. The second optimization problem takes the form:

$$\min_{\mathbf{a}} \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{such that } y_i\Big(\sum_{j=1}^{N} a_j y_j k(\mathbf{x}_j, \mathbf{x}_i) + b\Big) \geq 1, : \forall i$$

where $k(\mathbf{x_i}, \mathbf{x_j} = \langle \mathbf{\Phi}(\mathbf{x_i}), \mathbf{\Phi}(\mathbf{x_j}) \rangle$ is a kernel function which measures the distance between the samples $\mathbf{x}_i$, $\mathbf{x}_j$ after they have been transformed by $\mathbf{\Phi}$. By selecting directly a kernel function, we can bypass the computationally expensive calculation of $\mathbf{\Phi}(\mathbf{x}_i)$, $\mathbf{\Phi}(\mathbf{x}_j)$, and their inner product.

2.5 Deep Learning Methods

Having described many common methods for machine learning , we now turn our attention to *deep* methods for machine learning, and particularly neural-network methods for supervised learning problems [8]. The term "*deep*" refers [45] to the multiple layers of (non-linear) modules, that the input data must pass before reaching the output layer of the model. This enables learning of different hierarchies of representation for the same data - that are hopefully richer than the input's original features - while also offering the model complexity needed to fit demanding tasks.

As mentioned briefly in Section 2.2, the rise in deep neural networks came after the re-discovery of the backpropagation algorithm and the trend of collecting bigger datasets. Both are respectively necessary for the efficient and sufficient training of deep neural networks.

We begin our discussion with the *Perceptron* algorithm, explaining its conception and its limitations. Then we will explain how the *Perceptron* algorithm can be used to in a straight-forward deep learning model: the *Multiple Layers Perceptron* (MLP) model.

2.5.1 Perceptron

Perceptrons are the backbones of most modern deep learning networks. They model a conceptualization of how neurons work by linearly combining their inputs, that can be seen as excitation from neighbouring neurons transmitted via the synapses (Figure 2.2), and output an ON/OFF value by applying an activation function over the aforementioned linear combination.

Formally:

$$y = f(\mathbf{W}^T \mathbf{x} + b) \tag{2.1}$$

[8] This is not to say that there cannot exist any deep learning architecture that doesn't use neural networks. On the contrary, methods like deep forests [151] can, similarly to deep neural networks, employ multiple hierarchies of data representations. Furthermore, there are deep unsupervised learning methods, such as Variational AutoEncoders (VAE) [64] and deep Self Organized Maps (SOM) [74]

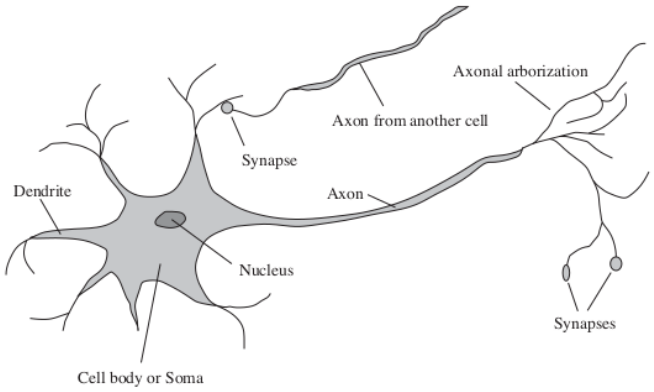


Figure 2.2. *An illustration of the biology of a neuron. Source: [119]*

However, a simple one-layer perceptron network cannot solve a number of non-linear regression or classification tasks. As an example, the XOR logic gate problem is often mentioned. This requires more perceptron layers to be solved, as we will see in 2.5.2

2.5.2 MultiLayer Perceptron Networks (MLP)

The MultiLayer Perceptron (MLP) Networks are composed of a number of individual perceptrons (neurons), that can be connected with other perceptrons directly. If those connections are acyclic, then the network is *feedforward*, as the information goes from the start of the network to its end, without any type of feedback or recurrency. In that case, the neurons can be divided into layers; the input layer which consists of the neurons are presented with the input directly, the output layer which consists of the neurons that generate the network's output, and hidden layer which consists of the intermediate neurons that generate inner representations of the data (see Figure 2.3). This leads us to the following formulation for the activation value of the j-th neuron in the i-th layer:

$$h_j^{(i)}(\mathbf{x}) = f_j^{(i)}\left(\mathbf{w}_j^{(i)T}\mathbf{h}^{(i-1)}(\mathbf{x}) + b_j^i\right) \tag{2.2}$$

Notice once again, the use of an activation function f which is almost always non-linear. In fact, if all the activation functions in the MLP network are linear, then the network has the same estimating power as a 1-layered perceptron networks with sufficient number of neurons[9]. However, a 2-layered MLP network, with sufficient number of neurons, can be much stronger than a single layer perceptron network (see Figure 2.4). In fact for many non-linear activation functions, an MLP with sufficient number of hidden layers can act as a universal approximator of every function [54].

2.6 Training a Neural Network

Having described the architecture of most common Deep Learning models, we will now discuss how these models are *trained*. Deep learning models, regardless of the complex-

[9]This is true, because a composition of linear operations, which is what happens in an MLP network with only linear activation functions, can be decomposed into multiple perceptron linear outputs.

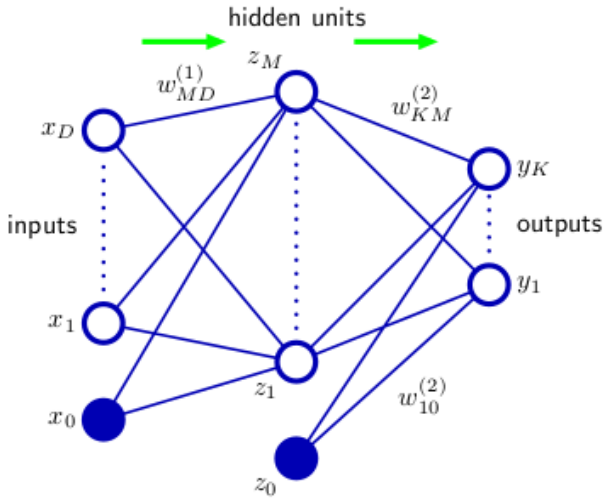


Figure 2.3. *A schema of a 2-layered feedforward MLP network. The information traverses the network from left to right. Source: [18]*

ities of each individual architecture, contain a great amount of learnable parameters[10]. By the term *training* we refer to the algorithm of learning values for those parameters, through data.

In this section, we will mention only learning the parameters of fully differentiable neural networks, since the opposite case is outside of our work's scope[11].

2.6.1 Gradient Descent & Backpropagation Algorithm

The mathematical formulation for training a network f with learnable parameters $\boldsymbol{\theta}$ using a set of inputs $\mathbf{X}$ and corresponding outputs $\mathbf{Y}$[12], optimizing a loss function $\mathcal{L}$, is the following:

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{Y}, f(\mathbf{X}, \boldsymbol{\theta})) = \min_{\boldsymbol{\theta}} \frac{1}{|\mathbf{X}|} \sum_{(\mathbf{x},\mathbf{y})} L(\mathbf{y}, f(\mathbf{x}, \boldsymbol{\theta})) \tag{2.3}$$

where L is the per sample loss function.

The *Gradient Descent* algorithm updates the network's parameters $\boldsymbol{\theta}$ iteratively by moving them towards the direction of $-\nabla_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{Y}, f(\mathbf{X}, \boldsymbol{\theta}))$ which is the direction in which the $\mathcal{L}$ loss function diminishes more quickly:

$$\boldsymbol{\theta}_{\text{new}} = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{Y}, f(\mathbf{X}, \boldsymbol{\theta})) \tag{2.4}$$

where η is a *hyperparameter* called learning rate and regulates how quickly the parameters move towards the direction which minimizes the loss more quickly. As the loss

[10] Of course not all of the model's parameters are learnable. Those which are not, are called *hyperparameters* and they are usually tuned by picking the values that perform best in a subset of the dataset called *validation set*. Common hyper-parameters can include the network's depth, the data's representation length, the optimization algorithm used, etc.

[11] A non-differentiable process that could be employed in a neural network is randomly sampling from a learnt distribution. However this can be made differentiable by the*reparameterization trick* (see [64])

[12] Those outputs can be either predefined, as is the case in *supervised learning*, or generated automatically by using the inputs set $\mathbf{X}$ as is the case in *self-supervised learning* (see Section 2.3)

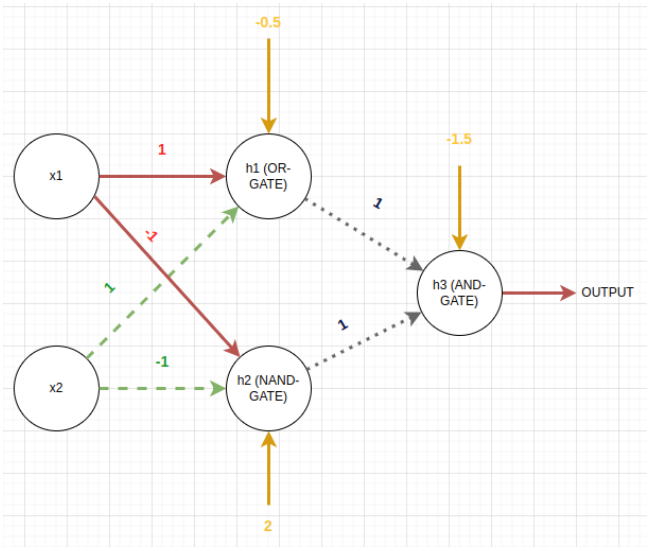


Figure 2.4. *A schema of a 2-layered feedforward MLP network that implements the XOR logical function, using the sign function as activation. Since XOR requires the intermediate calculation of the NAND and OR gate result, it would not be possible to not use an intermediate layer.*

function is minimized, the learning rate η can be decreased to allow for more fine-grained movements in the parameter space. Some important notes regarding the Gradient Descend algorithm are the following:

- If the Loss function over the parameter space θ is not convex: its possible that the algorithm converges to a local minimum point which corresponds to non-optimal parameters θ. Furthermore, if the algorithm reaches a *saddle point* [13], it will finish without reaching even a local minimum.
- The algorithm is not efficient: in order to update the parameters just once, the gradient has to be computed with respect to the whole dataset $\mathbf{X}$. This is not optimal since the dataset may contain duplicates which do not contribute anything towards the direction of the gradient.

Stochastic Gradient Descent

In practice, Gradient Descent cannot be used to train big neural networks with large datasets. A stochastic variation of gradient descent (*Stochastic Gradient Descent*-SGD) aims to solve the problems listed above by iteratively minimizing the loss as a function of a single sample from the dataset.

$$\theta_{\text{new}} = \theta - \eta \nabla_{\theta} L(\mathbf{y}, f(\mathbf{y}, \theta)) \tag{2.5}$$

This enables faster training, and adds variance to the θ movement in the parameter space. However, it also makes the learning curve more noisy.

[13] A saddle point of a smooth function f can be defined as a point which is neither a local minimum nor maximum point but has an indefinite Hessian matrix, i.e the derivatives over each parameter axis is zero.

In practice, in order to take advantage of the parallelization property of our network's forward pass, the gradient is calculated, in parallel, for multiple samples of the dataset referred to as a *batch* and afterwards averaged. In this way, assuming the computer has enough memory to parallelize the operations, the training process can be sped up considerably.

Smarter Descent

Many approaches have been proposed in order to make optimizing using SGD both faster and smoother. The momentum-SGD leverages the values of previous movements of the parameter to inform the current movement: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \cdot \mathbf{m}_t$ where $\mathbf{m}_t = \gamma \mathbf{m}_{t-1} + (1 - \gamma)\nabla_{\boldsymbol{\theta}} L(\mathbf{y}, f(\mathbf{y}, \boldsymbol{\theta}))$. Taking into account the previous movement can speed up convergence. Other, more complex, approaches include RMSprop [50], Adagrad [31] and Adam [63]

Backpropagation Algorithm

So far we have described the way the parameters are updated after the gradient has been calculated. But how can the gradient itself be calculated efficiently in order to update all the parameters in the network? The answer to this question is the *backpropagation* (BP) algorithm re-popularized for modern Neural Networks in [118].

Assume a L-layered neural network f with parameters θ, input $\mathbf{x}$ and reference/generated output $\mathbf{y}$, $\hat{\mathbf{y}}$ respectively. During the forward pass, the input must pass through all of the L-layers of the network:

$$\hat{\mathbf{y}} = f(\mathbf{x}, \boldsymbol{\theta}) = f^{(L)}_{\boldsymbol{\theta}^{(L)}}\left(f^{(L-1)}_{\boldsymbol{\theta}^{(L-1)}}(\ldots f^{(1)}_{\boldsymbol{\theta}^{(1)}}(x))\right)$$

Thus, minimizing the per-sample loss function L over each j-th layer parameter $\theta_i^{(j)}$ using the chain-rule:

$$\frac{\partial L(\mathbf{y}, \hat{\mathbf{y}}(\mathbf{x}))}{\partial \theta_i^{(j)}} = \frac{\partial L(\mathbf{y}, \hat{\mathbf{y}}(\mathbf{x}))}{\partial f^{(L)}_{\boldsymbol{\theta}^{(L)}}} \cdot \cdots \cdot \frac{\partial f^{(j)}}{\partial \theta_i^{(j)}}(\mathbf{x})$$

In terms of network topology, starting from output to input, the intermediate partial derivatives are calculated layer-by-layer and stored for the calculation of the previous layer's partial derivatives in a *dynamic programming* fashion. This update of the network's parameter from end to start is called a *backward-pass*.

Of course, implementing this into a computer is more difficult. The network topology corresponds to a computational graph that must be pre-calculated before the forward and backward passes commence. Furthermore, must common software that implement the BP algorithm do it in a vectorized way, calculating gradients instead of partial derivatives, for for better performance. For a more extensive analysis of the algorithm, we refer the reader to [45].

2.6.2 Regularization

Very important to the training of deep machine learning models is the use of *regularization* techniques. In order to better understand the way regularization affects the

performance of a network we will first explain the concepts of model *overfitting* and *underfitting*.

Model Overfitting/Underfitting

Underfitting refers to the performance of a model which doesn't fit the data well enough. Consider the case of the logical XOR gate, which we described in Section 2.5.1. The perceptron clearly underfits the dataset, despite being trained on every sample from it. It is not computationally complex enough to fully learn the input-output mapping. In general, a model *underfitting* the training data may be because of 1) Not enough training steps over the data, 2) The model having inadequate computational capacity[14] or 3) something is wrong (e.g a bug) with the optimization algorithm.

Overfitting is the opposite of *underfitting* the training data. In fact, the model has more than enough computational capacity to learn the input-output mapping, that learns to mimic it perfectly. This, however may lead to worse results for samples from a *test set* which was not used for training the model (Figure 2.5).

We can express *overfitting* and *underfitting* in terms of the model performance in a training set and a test set. *Bias* refers to bad fit of the training set's data, while *variance* refers to high discrepancy between the model's fit to the training and the test data. *Underfitting* happens when the bias is high and the variance is low, while *overfitting* happens when the bias is low and the variance is high. However, a model with high computational capacity can still achieve a good bias-variance balance using regularization techniques.

Early Stopping

Early stopping is one of the most widely known and practised way of avoiding overfitting the model. The model's loss is evaluated separately in a subset of the whole dataset; the *validation* set. When the model starts overfitting the training data, the loss on the *training* set keeps decreasing but the loss on the *validation* set remains relatively unchanged or even increases (see Figure 2.5). Therefore, it is clearly beneficial to stop the training at this point, in order to ensure that the model will generalize well enough to unseed data.

Weight Regularization

Weight regularization refers to a process where the model's loss function is augmented with a weight penalty term:

$$\mathcal{L}^{'}(\mathbf{Y}, \hat{\mathbf{Y}}, \boldsymbol{\theta}) = \mathcal{L}(\mathbf{Y}, \hat{\mathbf{Y}}) + \lambda \cdot \Omega(\boldsymbol{\theta}) \tag{2.6}$$

[14] By computational capacity we refer to the cardinality of the set of functions that our model can learn. It depends both on the number of its parameters and its architecture as well. For example, an intermediate layer is always needed for constructing a XOR gate using Neural Networks. Furthermore, in regression models a linear model cannot learn to model a - say - 4rth degree polynomial function.

Typical examples of weight penalties include the L_p norms:

$$\Omega(\boldsymbol{\theta}) = ||\boldsymbol{\theta}||^p = \sqrt[p]{\sum_{\partial_i \in \boldsymbol{\theta}} |\partial_i|^p}, \quad p \geq 1 \tag{2.7}$$

A weight penalty forces the model to learn smaller values for its parameters. Most notable cases are p=1 where the type of penalty is called *Lasso*, and p=2 where the type of penalty is called *Ridge* or *weight decay*. A Lasso penalty is not differentiable at $\partial_i = 0$ which can be problematic for gradient descent algorithm. However, due to to having a derivative equal to $sign(\partial_i)$ everywhere else, its time ∂_i is updated a term is subtracted or added to it depending to if ∂_i is positive or negative respectively. This means that after many optimization steps, some parameters will be close to zero, which can be helpful in applications such as feature selection, where parameter sparsity is desired. Note, that a Lasso penalty treats big and small parameter values the same.

A weight decay penalty, on the other hand, is fully differentiable which is optimal for gradient descent algorithms. Furthermore, since its derivative with respect to ∂_i is a linear function of ∂_i, bigger parameters face bigger penalty.

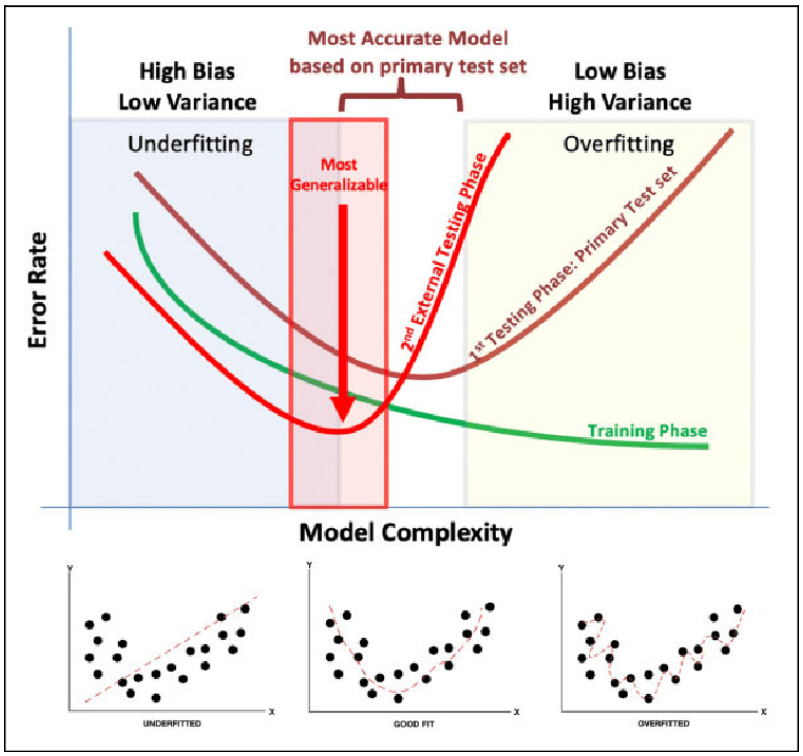


Figure 2.5. *The bias-variance tradeoff when fitting a model on a training set, validating its parameters in a validation set (here denoted with "Primary Test set" and finally evaluating its performance on a test set (here denoted with" External Test set") Source: [114]*

Dropout Layers

Dropout is a type of neural network layer, first introduced in [127], which randomly sets some parameters of the network equal to zero during training. This is inspired from

a voting method technique. Assume a committee of K estimators which averages output of the estimators y_i. It can be shown that if the errors of each estimator are uncorrelated between them then the expected error of the committee is the expected error of a single estimator divided by K.

Inspired by the previous discussion, let $\mathbf{M} = [m_1, \ldots, m_{|\theta|}]$, $m_i \in \text{Bernoulli}(p)$ be the masking vector that masks a subset of the network's parameters each forward+backward pass. The model, essentially minimizes the loss function $min_{\theta}\mathbb{E}_{\mathbf{m}}[\mathcal{L}(\mathbf{Y}, \mathbf{X}, \mathbf{m})]$ instead of the function $min_{\theta}\mathcal{L}(\mathbf{Y}, \mathbf{X})$. However, instead of having to learn K estimators as in our committee example, a network with dropouts is trained from a committee of thinned networks. This way it mimicks the effect of having multiple networks, that have been trained seperately, making a prediction.

During inference, after the training has ended, the dropout no longer deactivates any networks units. However, to alleviate the discrepancy between training with dropout and inferring without, the weights of the final network are multiplied by p[15]

Batch Normalization

A *Batch Normalization* layer simply normalizes the data over the batch-dimensions. This means that each feature of the data is projected to a N(0,1) normal distribution, after the mean and variance of each feature has been calculated over the batch dimensions. Formally, let the batch $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_B] \in \mathbb{R}^{B \times D}$ where B is the size of the batch and D is the dimension of each input sample. Then the transformed batch $\hat{\mathbf{X}} = [\hat{\mathbf{x}_1}, \ldots, \hat{\mathbf{x}_B}]$ is calculated as following:

$$\boldsymbol{\mu} = \frac{1}{B}\sum_{b=1}^{B}\mathbf{x}_b,\ \boldsymbol{\mu} \in \mathbb{R}^D$$

$$\boldsymbol{\sigma}^2 = \frac{1}{B}\sum_{b=1}^{B}(\mathbf{x}_b - \boldsymbol{\mu})^2,\ \boldsymbol{\sigma} \in \mathbb{R}^D$$

$$\mathbf{x}_b = \frac{\mathbf{x}_b - \boldsymbol{\mu}}{\boldsymbol{\sigma}}$$

This technique has been shown to work well in practise, but the theory behind why it's working is not so clear. Assume 2 network's nodes A, B that are connected by a Batch Normalization layer. The original paper [55] that introduced the technique claims it reduces the "*internal covariance shift*" of the distribution of layer's A outputs across different batches. Therefore layer B can learn its parameters easier, as the outputs from A don't constantly change distribution. Furthermore, layers normalized to N(0,1) can be more computationally efficient and robust to different hyperparameter values.

On the other hand, the authors of [122] claim the *internal covariance shift* is not reduced by *Batch normalization*, and in any case it doesn't seem to have any correlation with model performance. They further claim that the reason *Batch normalization* seems to help is because it smoothens the loss function leading to easier parameter optimization.

[15] To get a mathematical intuition on why this should be the case, the reader is referred to [127, 45]

Layer Normalization

Layer Normalization [8] is quite similar to *Batch Normalization* as they both calculate a mean and a variance. However, in *Layer Normalization* the mean and the variance are calculated over each dimension of the input, for each input independently from other inputs in the same batch - while in *Batch Normalization* the mean and variance are calculate for each feature separately across the batch dimension.

Layer Normalization, thus, offers a way to avoid the *internal covariance shift* without having to use batch size > 1. Furthermore, it offers a way of normalizing hidden states activations between successive steps of a recurrent neural network. Finally, a network using *Layer Normalization* has exactly the same behaviour during training and testing, as it doesn't need to store the mean and variance of each layer's activation from the training set in order to normalize samples in the test set.

2.7 Modern Machine Learning Networks

In this section, we describe the architectures of several neural network methods currently used in the literature, assessing their respective advantages and disadvantages.We introduce the *Recurrent Neural Network* architecture (RNN) and explain its usefulness in modeling sequential data. Then, we define the *Long Short-Term Memory* (LSTM) networks and explain how they overcome some problems faced by RNNs. The attention mechanisms, which can be used by the previous networks, will also be defined. Finally, we define the *Transformer* model and explain its computational efficiency.

2.7.1 Recurrent Neural Network (RNN)

Despite MLP's universal approximation property, they are not suitable for a class of problems. Consider the problem of the input being $\mathbf{x} = [\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(T_x)}] \in \mathbb{R}^{n \times T_x}$, i.e sequential data. Those data can be a time-series, a text, a video, etc. A human processes this type of data and answers questions related to them sequentially, while keeping in mind what information they have processed so far.

A Recurrent Neural Network (RNN) [9] is an attempt of modeling that behaviour using a neural network, where the already processed information from the previous inputs of the sequence is encoded into a *hidden state* vector[16] (see Figure 2.6).

Formally, let $\hat{\mathbf{y}}^{(i)}$ be the i-th output token generated by the RNN and $\mathbf{h}^{(i)}$ the i-th hidden state, then:

$$\mathbf{h}^{(i)} = g_1\Big(\mathbf{W}_{hh}\mathbf{h}^{(i-1)} + \mathbf{W}_{hx}\mathbf{x}^{(i)} + \mathbf{b}_h\Big) \tag{2.8}$$

$$\hat{\mathbf{y}}^{(i)} = g_2\Big(\mathbf{W}_{yh}\mathbf{h}^{(i)} + \mathbf{b}_y\Big) \tag{2.9}$$

Notice, the lack of superscripts for all the model's parameters. The same parameters are applied for each part of the sequential input, the only difference being in the current part of the sequential input $\mathbf{x}^{(i)}$ and the current *hidden state* $\mathbf{h}^{(i)}$.

[16]The term *context vector* is also common in the literature.

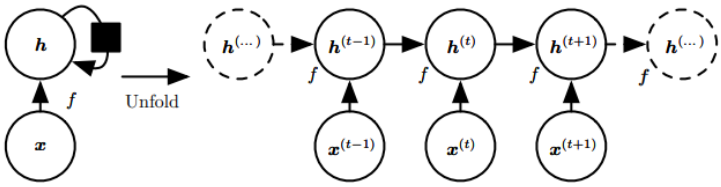


Figure 2.6. *A schema of an RNN with its outputs omitted. Left: A conceptual way of the direction of information across the network. Right: The computational graph of the network, if the sequence of computations is unfolded.*

Classifying the different problems that can be solved by an RNN

Assume that the input sequence has length T_x and the reference output sequence has length T_y. We can categorize the problems that can be solved by an RNN based on the relationship of those lengths, following many classic texts on the subject [45, 3]. As we will see (Figure 2.7) each problem requires a slight modification of the RNN architecture.

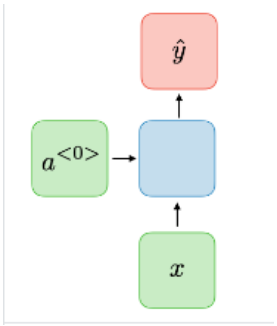
- $\mathbf{T_x = T_y = 1}$: The RNN is a simple 1-layered MLP network.
- $\mathbf{T_x = 1}$, $\mathbf{T_y > 1}$: The RNN acts as a decoder-only model where instead of the hidden vector, the predicted output is passed auto-regressively to the next step. This is typical in applications where x is already a contextually enriched vector[17]. A typical example would be predicting a text just from its starting word. The auto-regressive process terminates, after an end-of-input token is generated.
- $\mathbf{T_x > 1}$, $\mathbf{T_y = 1}$: The RNN passes the hidden state vector through each step. However, the output is not sequential as it is calculated only at the last step. Typical examples would include problems where the output is either a value (weather forecasting), or a class membership prediction vector (text classification, music sentiment classification).
- $\mathbf{T_x = T_y > 1}$: The RNN in each step calculates a part of the output as well ass passes the hidden state to the next step. Typical examples of this architecture would include syntax parsing, Named Entity Recognition, etc.

The Encoder-Decoder model

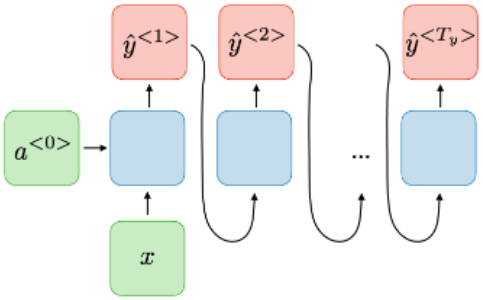
From our previous discussion we have left out the case of $\mathbf{T_x \neq T_y}$, $\mathbf{T_x > 1}$, $\mathbf{T_y > 1}$. This is an example of an encoder-decoder architecture[18], where both the length of the input as well as the length of the output can vary and thus be different. The model doesn't generate output just passes the hidden state vector (encoder model) until all the input has been captured in a *context vector*. Then the model acts like a regular RNN, generating output until an <end-of-input> token is generated (see Figure 2.8). Typical examples include: machine translation, text summarization, etc.

[17] This can be achieved by a pre-trained network, or as a result of an encoder-decoder architecture (see the $T_x \neq T_y$, $T_x > 1$, $T_y > 1$) case

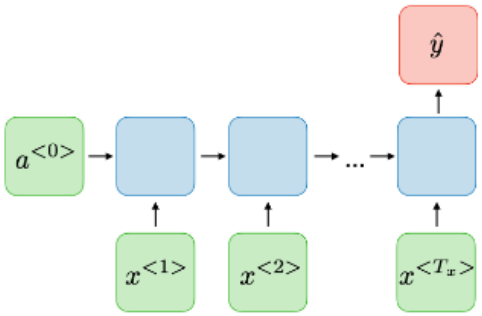
[18] The term *Sequence-to-Sequence* (Seq2Seq) is used interchangeably in the literature.



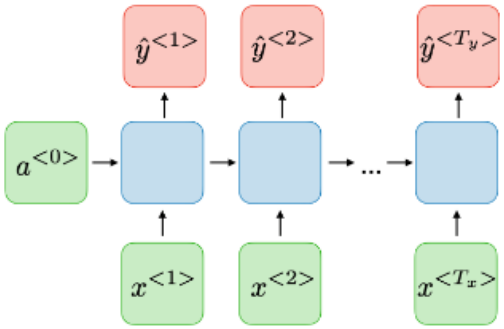
(a) $T_x = T_y = 1$. *The RNN acts as a single layer MLP network.*



(b) $T_x = 1$, $T_y > 1$ *Instead of the hidden vector, the predicted output is passed auto-regressively to the network each step.*



(c) $T_x > 1$, $T_y = 1$ *The output is calculated only after processing the last input. The hidden state is passed to the next step.*



(d) $T_x = T_y > 1$ *In each step, a part of the output is calculated and the hidden state is passed to the next step.*

Figure 2.7. *An illustration of different RNN architectures corresponding to different problems. The hidden vector here is denoted with "a". Source: [3]*

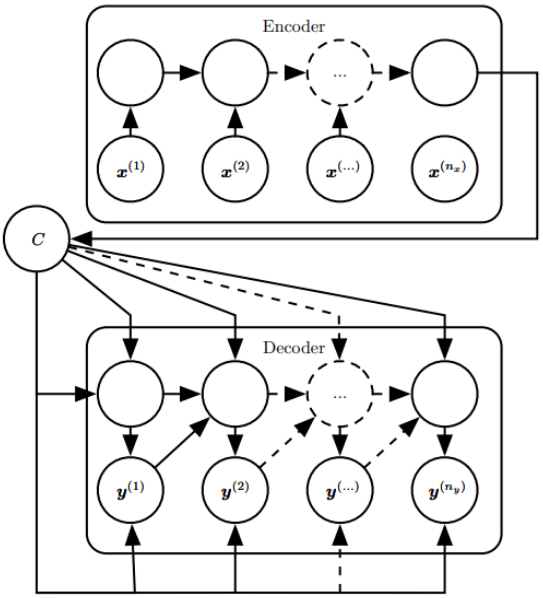


Figure 2.8. *A schema of an Encoder-Decoder RNN model where the input's context is captured the encoder model into a context vector c. The context vector informs the decoder model, as it auto-regressively generates the output sequence. Source: [45]*

Advantages and Disadvantages of RNNs

RNNs provide a way of sequentially modeling the representation of input data and sequentially generating the representation of the network's output. Therefore, there are suitable for most problems where the input our the desired output is sequential.

However there certain problems with the architecture that hinders the model's training and its ability to retain information from a long input.

- **Retaining the context from a long input:** The *hidden state* vector **h** as well as the *context* vector, due to their limited size, cannot encode sufficiently well all the information of the input sequence. The information at the start of the sequence may be used to generate part of the output's start (in the the $T_x = T_y > 1$) case, but it doesn't necessarily inform the output's last tokens because the hidden vector may have changed a lot by encoding information from the input's later tokens. Implicitly learning which tokens must be payed attention by the network leads as to the attention mechanism, that will be detailed in Section 2.7.3.

- **Unidirectionality:** The architecture described so far is unidirectional, i.e the input is parsed and the output is generated from left to right. This is not optimal since there are problems, for example machine translation, where the translation of a word depends not only on the preceding words but on the words that follow as well. The most common solution proposed to enable bidirectionality into the networks is to have 2 separate RNNs, one parsing left-to-right and the other right-to-left, parse the input and generate the corresponding *hidden/context* vectors which can be merged into one either by averaging or concatenating.

- **Vanishing/exploding gradients:** The RNNs use, like all the Deep Learning models we have introduced, the gradient descent algorithm for training, which involves the calculation of the gradient of the loss over the network's parameters. Due to the recurrent architecture of the network, the gradient tends to either vanish (i.e becomes extremely small) or explode (i.e becomes extremely large). Clipping the gradient can alleviate the exploding gradient problem, but solving the vanishing gradient problem requires a more complex network architecture as the one presented in Section 2.7.2.

2.7.2 Long Short Term Memory Network (LSTM)

Having introduced the RNN and some of the problems it faces, we will now introduce the *Long Short-Term Memory* (LSTM) [52] network architecture which is an expansion of the RNN architecture. LSTMs are designed to avoid the *vanishing gradient* problem by introducing a *cell memory* vector $\mathbf{c}^{(t)}$ which is propagated to the next state after it is modified by a tanh activation function. The tanh function's second derivative has nice properties that make vanishing gradients less propable.

Outside of the inner calculations that happen in a single step, the LSTM acts exactly like an RNN, propagating information recurrently to the next step and generating output every step if it is needed (Figure 2.9).

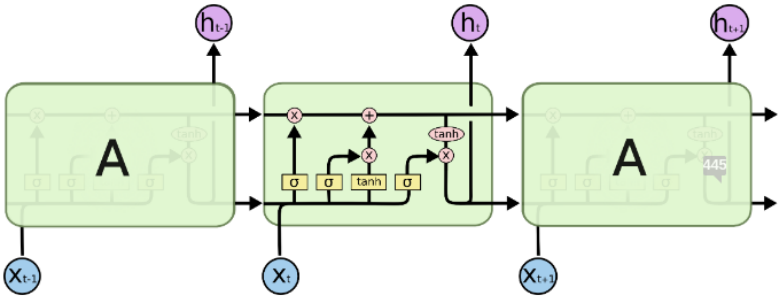


Figure 2.9. *A schema of an Long Short-Term Memory Network. Source: [100]*

Formally, the calculations that happen at the step t are the following,

$$\mathbf{f}^{(t)} = \sigma\left(\mathbf{W_f}\mathbf{x}^{(t)} + \mathbf{U_f}\mathbf{h}^{(t-1)} + \mathbf{b_f}\right) \tag{2.10}$$

$$\mathbf{i}^{(t)} = \sigma\left(\mathbf{W_i}\mathbf{x}^{(t)} + \mathbf{U_i}\mathbf{h}^{(t-1)} + \mathbf{b_i}\right) \tag{2.11}$$

$$\mathbf{o}^{(t)} = \sigma\left(\mathbf{W_o}\mathbf{x}^{(t)} + \mathbf{U_o}\mathbf{h}^{(t-1)} + \mathbf{b_o}\right) \tag{2.12}$$

$$\mathbf{u}^{(t)} = \tanh\left(\mathbf{W_u}\mathbf{x}^{(t)} + \mathbf{U_u}\mathbf{h}^{(t-1)} + \mathbf{b_o}\right) \tag{2.13}$$

$$\mathbf{c}^{(t)} = \mathbf{f}^{(t)} \odot \mathbf{c}^{(t-1)} + \mathbf{i}^{(t)} \odot \mathbf{u}^{(t)} \tag{2.14}$$

$$\mathbf{h}^{(t)} = \mathbf{o}^{(t)} \odot \tanh(\mathbf{c}^{(t)}) \tag{2.15}$$

where $\odot$ denotes pointwise vector multiplication, σ denotes the sigmoid activation function and tanh they hyperbolic tangent activation function (Section 2.4.2).

Equations (2.10,2.11,2.12) implement *gates* that the network uses to update its *cell memory* and *hidden state*. The **forget gate** (Equation 2.10) which information from the previous step's cell memory $\mathbf{c}^{(t-1)}$ will be forgotten in the current step. Similarly, the **input gate** (Equation: 2.11) learns which parts of the previous step's hidden vector $\mathbf{h}^{(t-1)}$ and the current input $\mathbf{x}^{(t)}$ will be stored into the current step's cell memory $\mathbf{c}^{(t-1)}$. However, the network still uses a hidden state and thus must learn via an **output gate** (Equation: 2.12) which information to propagate to the next step's hidden state, while keeping in mind the current memory cell vector.

LSTMs are a big improvement over classic RNN architectures, as they enable retaining context over longer sequences without facing the *vanish gradient* problem. However they still cannot retain context from a very long sequence, since the hidden state and cell memory vectors (which hold the context of a whole sequence) can only hold so much information. In section 2.7.3 we will see how this problem can be approached.

2.7.3 Attention Mechanism

The main problem of both RNN and LSTM architectures is that: a fixed-length *context* vector may not be enough to encode all the information from a long sequence of input, unless the *context* vector increases in size which is very taxing in terms of computational complexity. An attention mechanism for Encoder-Decoder Network architectures was introduced in [9], where the context vector $\mathbf{c}^{(i)}$ is calculated separately for each step i, each time paying different attention to parts of the input as those are encoded in the hidden states $\mathbf{h}^{(i)}$.

Formally (following the notation used in [23]), let $\mathbf{h}^{(i)}$, $\mathbf{s}^{(j)}$ denote the hidden states of the encoder and the decoder at the timestep i and j respectively. Then, the *context* vector $\mathbf{c}^{(j)}$ at decoding step j is informed by the encoder's hidden states as following:

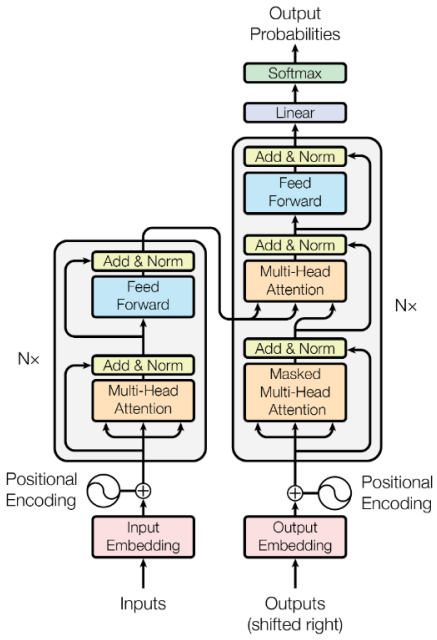
$$\mathbf{c}^{(j)} = \sum_{i=1}^{T} a_{ij}\mathbf{h}^{(i)} \tag{2.16}$$

$$a_{ij} = softmax\Big(a(\mathbf{s}^{(j-1)}, \mathbf{h}^{(i)})\Big) \tag{2.17}$$

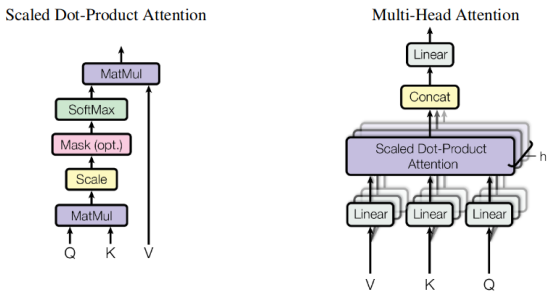
where a denotes an *alignment* function. Typical examples of alignment function include 1) **scaled dot product:** $a(\mathbf{m}, \mathbf{n}) = \frac{\mathbf{n}^T\mathbf{m}}{\sqrt{d_k}}$ which scales the dot product similarity function by the state vector's representation dimension length, 2) **general multiplicative:** $a(\mathbf{m}, \mathbf{n}) = \mathbf{v}_a^T\text{softmax}(\mathbf{n}^T\mathbf{W}\mathbf{m})$ and 3) **general additive:** $a(\mathbf{m}, \mathbf{n}) = \mathbf{v}_a^T\text{softmax}(\mathbf{W}_1\mathbf{n} + \mathbf{W}_2\mathbf{m})$, where $\mathbf{v}_a$, $\mathbf{W}_i$ are learnable parameters.

Notice that the alignment functions above, are differentiable. The same applies for the calculation of the attention scores a_{ij}. This implies that the attention scores are *softly* distributed over all possible timesteps, thereby this kind of attention mechanism is called **soft attention**. Furthermore, the attention takes into account all hidden states, therefore it's characterized as **global attention**[19].

[19]There are ways of applying *hard* attention and also calculating the *context* vector *locally*, but those are outside of the premise of our work.



(a) *The full architecture of an Encoder-Decoder model using the transformer module.*



(b) *Left: the scale dot-product attention function. Right:The Multi-head attention module*

Figure 2.10. *A schema of (a) the transformers encoder-decoder mode and (b) the multi-head attention module*

Self-Attention

What happens when the network's output is not a sequence? The attention mechanism we described above would not be able to separately attend to each of the decoder's *hidden state*, since there is only one. *Self-Attention*[20] is a method of employing attention when a decoder's hidden states are not available.

In essence, the model learns to attend to its own input without having any other information:

$$a(\mathbf{n}) = \mathbf{v}_a^T \text{softmax}(\mathbf{W}\mathbf{n}) \tag{2.18}$$

The Self-attention mechanism will be further elaborated on Section 2.7.4.

2.7.4 Transformer Network

The *Transformer* network was first introduced by (Vaswani, et al.) in 2017 [137], and since it has found extended success in most Machine Learning applications, such as Text Summarization [77], Video Understanding [14] and Machine Translation [153].

We will now describe formally the transformer module's architecture (Figure 2.10a). We assume the transformer network's input vector size is denoted with N and the hidden-representation size is denoted with d_{model}.

[20]The term intra-attention is also used, albeit less often.

Multi-Head Attention module

The transformer module uses a variation of the *self-attention* mechanism we defined in Section 2.7.3 (Figure 2.10b). Assuming inputs **K**, **V**, **Q**, which in the encoder of the transformer network are all equal to the previous layer's output - hence the use of *self-attention*, the scaled dot-product attention function (see Section 2.7.3) is applied:

$$\text{Attention}(\mathbf{K}, \mathbf{V}, \mathbf{Q}) = \text{softmax}\Big(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\Big)\mathbf{V} \tag{2.19}$$

The Multi-head attention module, performs the scaled dot-product attention function h times, after having previously projected the **K**, **V**, **Q** inputs from $\mathbb{R}^{N\times d_{\text{model}}}$ to $\mathbb{R}^{N\times h\times \frac{d_{\text{model}}}{h}}$. The result is h *attention heads* which are afterwards concatenated into a single matrix and then projected again to a $\mathbb{R}^{N\times d_{\text{model}}}$ matrix:

$$\mathbf{head}_i = \text{Attention}(\mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V, \mathbf{Q}\mathbf{W}_i^Q) \tag{2.20}$$

$$\text{MultiHeadAttention}(\mathbf{K}, \mathbf{V}, \mathbf{Q}) = [\mathbf{head}_1, \ldots, \mathbf{head}_h]\mathbf{W}^O \tag{2.21}$$

The resulting vector has the same shape as the output of a single self-attention module, however because of the multiple *attention heads* the network can attend to each input in multiple ways.

The Transformer Encoder-Decoder Model's architecture

The transformer network is not recurrent, therefore the whole input is passed as input to the network, at once. The input representation is constructed (for example using a Section 3.4.4 embedding) generating a $\mathbb{R}^{N\times d_{\text{model}}}$ vector, where N is the length of input and d_{model} is the dimension of the input representation. In order to encode positional information to each part of the input, we generate positional embedding vectors $\mathbf{PE} \in \mathbb{R}^{N\times d_{\text{model}}}$ as follows:

$$PE_{i,2j} = sin(i/10000^{2j/d_{\text{model}}})$$
$$PE_{i,2j+1} = cos(i/10000^{2j/d_{\text{model}}})$$

The rest of the encoder's architecture is straight forward: partial results are added and normalized (see Section 2.6.2), and the encoder's output is passed as input to the next encoder. The decoder acts similarly with the encoder module with some differences only:

- The Multi-Head Attention applied to the currently shifted output is masked so as the decoder attends only to tokens up to the token that have been until now generated.
- The decoder has a second Multi-Head Attention module which attends to the encoder stack's output.

Main differences with recurrent networks

The Transformer architecture has three main differences from the recurrent architectures we discussed in previous sections.

1. **Non-recurrency:** The Transformer network is not recurrent. This is helpful for various reason. Firstly, the vanishing gradient problem is avoided, since the gradient's maximum path length is not dependent on the length of the input sequence n, while in both RNN and LSTM it increases linearly with n. Secondly, the minimum number of sequential operations needed for a whole sequence to traverse the network is not dependent on the length of the input sequence, while in both RNN and LSTM it increases linearly with n. This enables parallelization of the computations, which are not possible in recurrent networks. Thirdly, the positional information is inserted directly into the input sequence, via the positional embeddings mechanism.

2. **Per-layer computational complexity:** Each layer of the transformer has $O(n^2 \cdot d)$ computational complexity, while recurrent networks have $O(n \cdot d^2)$. This means, that transformers are more efficient when the data dimension length is bigger than the length of the data.

3. **Fixed input-output length:** The architecture described above has the disadvantage of not being able to handle input and output sequences of arbitrary length. Instead, the maximum length of both input and output must be specified before the network is even trained. Therefore, encoding long texts using a Transformer Encoder-Decoder model may lead to splitting the input into multiple parts and creating separate representations that have no contextual information between them.

Overall, the advantages of the transformer over the recurrent networks have enabled training bigger models more efficiently. A further use of transformers that has been widely adopted in many Machine Learning applications is the creation of Transformer networks pretrained on very large amounts data, in a self-supervised fashion [29, 69]. Those networks can later be used in other applications with minimal or no fine-tuning. This provides a way of having rich contextual representations without having to pay the computational cost of training a very large neural network.

Chapter 3

Natural Language Processing

3.1 Introduction

Natural Language Processing (NLP)[1] is the scientific discipline that tries to formally understand *natural languages*. A *natural language* is a language that has evolved through its usage, without following any clear predefined rules. NLP makes use of research in the fields of Linguistics, Computer Science, Cognitive Science, and most recently Statistics and Artificial Intelligence, in order to produce a rich understanding of 1) how a language evolves and is used in theory and in practise and also 2) how to extract information from natural language texts.

Between the 1960s to 1990s the most prominent paradigm of Natural Language Processing was the *symbolic* one, where the process of formalizing natural languages was performed through the construction of formal rules that concerned the morphology, syntax and semantics of the language. This was based on the on a *realist* [2]linguistics paradigm which supported that the language's structure (syntax&semantics) are not derived by experience but are fixed in advance. Chomsky's quote [26] "One's ability to produce and recognize grammatical utterances is not based on notions of statistical approximation and the like" puts it succinctly.

From 1990s up to the present, the current paradigm of of NLP has changed. This change can be attributed to the revival of the *empiricist* school of linguistics, which claims that humans can learn language (syntax, semantics, etc.) by using their cognition. If humans can learn language by using general rules that can be distilled from their cognition, then the same applies for computers which can learn using statistics. That intuition gave rise to the statistical NLP paradigm that is most popular today. The recent advances in *Deep Learning* have been incorporated successfully in statistical NLP[3] and have been deployed in various NLP problems both in research and in industry.

However, in the recent years purely statistical methods for NLP have been criticised for

[1] Pre 2000's the terms *Language Technology*, *Language Engineering* were used interchangeably with Natural Language Processing

[2] The classic Statistical NLP textbook [83] by Manning and Schuetze make the case for the general distinction between *realist* and *empirical* linguistics. We find this naming makes many major philosophical assumptions behind the motivations of each paradigm, that are too general to be true. Therefore, we use them sparingly as both *realism* and *empiricism* in philosophy and linguistics represent vast amounts of literature.

[3] In the rest of our work the term NLP will be used exclusively for its statistical sub-discipline, since symbolic NLP is outside of our work's reach.

their lack of explainability. Therefore, an interest of hybrid models that use both neural networks for their rich representations and symbolic methods for their explainability, has grown with mixed results so far [48].

In the following sections of this chapter we will introduce practical applications of NLP, discuss how a text is preprocessed for NLP tasks and finally delve some of the most important topics in modern NLP; how to represent words and documents, and how to model language using them.

3.2 Applications

Natural Language can be identified everywhere around human activity. Furthermore, humans may need to automate some of their cognitive processing that happens using this language, using NLP. Therefore, the application landscape of modern NLP is vast both in terms of types of natural languages that are used, as well as in terms of what processing and what output is expected by the process. In this section, we will outline few of the modern NLP topics both in general domain language and in legal-domain language.

3.2.1 Applications to General-Purpose text

In this section we will list and briefly define NLP tasks for *general-purpose* text or tasks that are common in multiple domains. *General-purpose* text usually refers to text that is generated for and by the general public; this text should not contain a lot of specialized terminology. Figure 3.1 provides a map of most common NLP tasks.

- **Text Classification:** refers to the NLP task of predicting which class does an input text belong to. Common applications include news categorization, movie genre classification, etc.

- **Token-level classification:** includes tasks that require a separate classification for each token of the input sequence. Common tasks in this category include: a) **Named Entity Recognition** (*NER*) which is the task of identifying which tokens correspond to a named entity, and sometimes categorizes the token to an entity type, such as Organization, Location, etc, b) **Part-of Speech Tagging** (*POS*) which is the task grammaticaly tagging each token to a particular part of speech, such as verb, noun, etc, and c) **Syntactic Parsing**: where the syntactic dependencies between the input tokens are identified.

Several NLP tasks are thought to be more difficult, as solving them requires deeper understanding of the natural language[4]. Those tasks belong to the *Natural Language Understanding* (NLU) subset of NLP tasks.

- **Summarization:** is the task of generating a summary for a given input text. This supposes the NLP algorithm can understand the text and extract information from it.

[4]Of course which task is difficult is up to debate and surely is a function of our own historical biases.

- **Question Answering**: where the Question Answering system must answer a question, most of the times also given a text as input. This requires understanding of both the text and the question.

- **Sentiment Analysis**: is the task of extracting the sentiment of a text. The task is not trivial, since the use of irony, metaphors, or the simple use of a contrasting word such as "but" can flip the sentiment of a sentence.

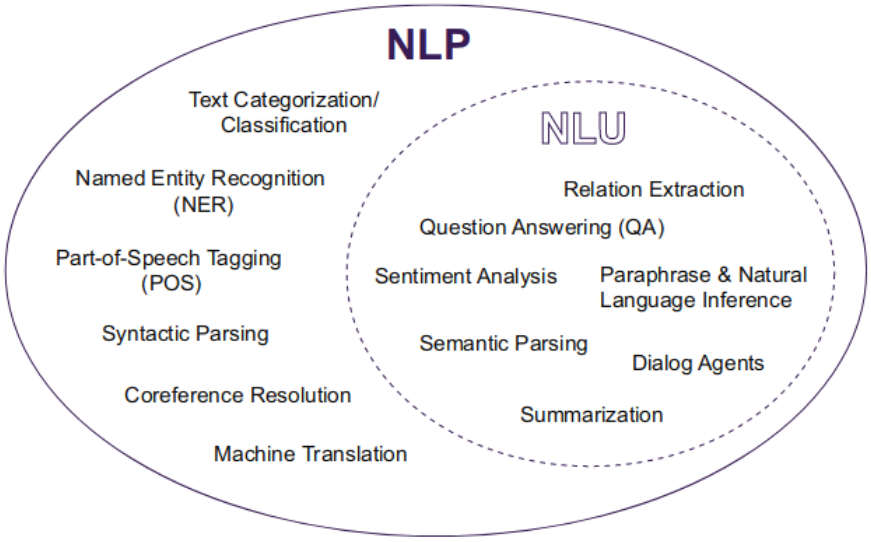


Figure 3.1. *A map of common tasks in modern NLP. Source: [48]*

3.2.2 Applications specific to Legal Text

The question of when and how to apply modern NLP methods to legal texts is recent one. Until recently, most legal NLP methods focused on building expert rule-based systems often representing legal-texts with complex domain-specific ontologies[5]. Recently, with the rise of big-data and end-to-end methods for NLP, there has been an increased interest in applying *Deep Learning* models to Legal Text analytics. Below, we will outline few of the tasks that those models are asked to solve:

- **Court Decision Prediction:** which is concerned with predicting the outcome of a judicial trial. [20] applies a hierarchical BERT architecture to predict the outcome and the specific violations as ruled by the *European Convention of Human Rights* (ECHR). Similarly, [24] employs deep learning models to predict the prison term, both in total and separately for each charge, for cases ruled by the China's Supreme People's Court. Applying large BERT models pre-trained on legal-texts has also been shown to increase predictive performance [85] on Romanian court rulings concerning bank disputes, as well as in classification tasks concerning European contracts, court rulings and laws [21].

[5]For further details, we refer the reader to K.Ashely's textbook [7] on AI and Legal Analytics.

- **Court Decision Summarization:** The summarization of judicial decisions is an emerging topic in legal NLP. The performance of various extractive and abstractive models is explored in [44]

- **Rationale Extraction:** which concerns the extraction of word/sentence/paragraphs that contain the rationale of a judicial decision texts. Researchers in [22] extract the rationale using a hierarchical variation of the BERT model where constraints placed for increasing the quality of extracted rationals are expressed through regularizers in the model's loss function.

- **Legal Question Answering:** is similar to general information-retrieval tasks. The model is asked to find legal text that is relevant to an input query. The approach of using neural networks with attention is explored in [60] using Vietnamese legal questions.

3.3 Text Preprocessing

Text preprocessing refers to the process transforming "raw" natural language to a format that is more easy to further process for a specific task.

- **Tokenization:** is the process by which the input text is segmented into separated tokens. Each token may correspond to an individual word (*word tokenization*) or part of a word -if subword tokenization is used; which tokenizes each infrequent word in multiple tokens corresponding to a character or a sequence of characters. Tokenization is important because *tokens* is the most fine-grained piece of information that an NLP model can analyze, and therefore the model can output information up to the token granularity.

- **Sentence Segmentation:** Several tasks require the input be segmented into its sentences. In many domains this is easy, since the "." (dot), the exclamation "!", or the "question mark" (?) symbol suffice. However, in many domains the *raw* input may be noisy; through the use of slang language, emoticons, etc, while the aforementioned symbols may be part of an abbreviation or a mathematical equation and therefore not signify the end of a sentence. In those cases, expert-defined rules and statistical methods can be used to improve performance of sentence segmentors.

- **Stopwords:** are words that are very common in a natural language and in certain applications may be ommited since they don't offer much (or even hurt) the model's performance, while they may also bias the evaluation results. Common stopwords include: 1) determiners ("a", "the", "my"), 2) coordinating conjunctions ("but", "neither", "for") and 3) prepositions ("upon", "behind", "after").

- **Lemmatization/Stemming:** is the process of replacing the words in the input sequence with their *lemmas* or *stem*, which in both cases is a morphological property of a word. This is useful for NLP applications that require limited vocabulary or do not need to differentiate between words of the same stem/with the same lemma.

- **Text Enriching:** information from Syntactic Parsing, Named Entity Recognition, or even task/domain-specific information can enrich the tokenized input text with further information useful for further NLP.

3.4 Word & Document Representations

3.4.1 Introduction

Words, sentences and documents are straight-forward concepts for humans, however a computer needs a way of representing a word, a sentence or a document in its memory. This representation, if it is to be beneficial for further analysis, may need to encapsulate the semantics of each word and even the semantics of the context the words appears in.

Human cognition already does so often by 1) drawing information from the *lemmatization* of each word, 2) *disambiguiating* the word's sense, 3) paying attention to each word's *connotation* or *sentiment*, etc[6].

In this section, we will introduce ways a computer can better represent a word, sentence or document via vectors or matrices.

3.4.2 Representation via Denotation

A simple way of thinking about representing a word is by ignoring the information provided by its context and simply denotating the word.

Vocabulary Indices

The most straight-forward way of denotating a word in a computer is assigning it to a unique number. Assuming a vocabulary of words $\mathbf{V}$, we can construct a mapping from each word to a vocabulary index (an integer): $f_v(w) : \mathbf{V} \rightarrow \mathbb{N}$. This can be easily implemented in a computer via a hash-map. Furthermore, it's a method that can express new words that are not in the dictionary, simply by mapping them to the next integer after the last one used.

Using this method to represent a sentence is simple. A sentence is a list of vocabulary indices that correspond to the word it contains in order. Likewise, a document can be represented as a list of sentences.

However, the method doesn't encode any semantic information to the words itself. The indexes are randomly assigned to each word and vocabulary indexes that are close in value may correspond to words with completely different meanings, connotations, etc. Similarly, words with similar meanings may not have similar vocabulary indices.

One-hot encoded embeddings

Attempting to distinguish the vocabulary index from each word's meaning, it is easy to devise the *one-hot encoding* method. In this representation method, each word still

[6]For a more extended analysis of how human cognition may infer information from lexical semantics, the reader is referred to the Jurafksy & Martin classical textbook on Speech and Language Processing [58]

corresponds to a vocabulary index but is instead represented as an one-hot vector at this specific index. Formally, let f_{oh} be the mapping function of this representation:

$$f_{\text{oh}}(w) : \mathbf{V} \rightarrow \{0, 1\}^{|\mathbf{V}|}$$
$$f_{\text{oh}}(w) = \mathbf{1}_{f_v(w)}$$

For example, if the word "language" is mapped to the vocabulary index $f_v(\text{language}) = 4$ and the vocabulary has 5 words, then it will be represented by the vector $[0, 0, 0, 1, 0]$ which is non-zero only at index 4.

Representing sentences can be just as straight-forward, by simply generating a $|\mathbf{V}|$-length vector which is non-zero and equal to one only in the positions that correspond to indexes of words they contain.

3.4.3 Frequency-based Representations

Merely denoting each word's occurrence (or absence) in a sentence is sub-optimal in terms of encoding as much of a sentence's meaning as possible, since it ignores how often the word occurs in the sentence and also how often the word occurs in other independent sentences. This observation leads us to *frequency-based* representations.

Count Vectorization

A *Count Vectorization* method treats a document as a Bag of Words (BoW).The document is represented as a $|\mathbf{V}|$ length vector which is non-zero and equal to the word's number of occurrences in the document, only at the indices that correspond to vocabulary indices of words found in the document. In other words, the ones of the one-hot encoded vectors are replaced by the count of each word.

Weighted Count Representations

Simply counting the number of occurrences in one document can be sub-optimal since there words that are unique to a document and, therefore not only their frequency inside a particular document must be taken into account, but also their (in)frequency in other documents.

Formally, let $D = \{d_1, \ldots, d_{|D|}\}$ be a collection of documents (i.e a collection of separate collections of words) and $tf(w, d)$, $idf(w, D)$ refer to the per-document term frequency and inverse document frequency metrics respectively. Then those can be formulated as following:

$$tf(w, d) = log(Count(w, d) + 1) \tag{3.1}$$
$$df(w, D) = \sum_{d \in D:\ w \in d} 1 \tag{3.2}$$
$$idf(w, D) = \frac{|D|}{df(w, D)} \tag{3.3}$$

where the log function is used to avoid integer overflow if the dataset is very large. This method allows us to formally represent both 1) how important is a word to a particular document, by using the tf(w,d) score, and 2) how infrequent it is in other documents, by using the idf(w,D) score. A score that takes both aforementioned scores into account is the **tf-idf** score, where the term-frequency score and the inverse-document-frequency are simply multiplied:

$$tf - idf(w, d, D) = tf(w, d) \cdot idf(w, D) \tag{3.4}$$

The tf-idf can straightforwardly replace the count used in the *Count Vectorizer* method, generating document vectors that take both in-document frequency, and frequency in seperate documents into account.

Sparse Representations & the Curse of Dimensionality

However, in most practical applications the Vocabulary size $|\mathbf{V}|$ is quite large, leading our previously discussed representation methods to generate *sparse* vectors. These vectors will be non zero on only a small percentage of their indices. This can be problematic for several reasons:

- **Memory Inefficiency:** The representations are very memory inefficient prohibiting or significantly slowing down any computational process between multiple vectors.
- **Curse of Dimensionality:** Most Machine Learning models and optimization methods cannot learn well when the number of dimensions in the input data is large. Adding more dimensions in the data requires exponential more training samples in order to not hinder the model's training.

If a representation method is to be used effectively in modern machine learning architectures, it must generate dense representations where more information is encoded in less dimensions.

3.4.4 Distributional Semantics & Continuous Word Representations

In this section, we will describe how modern *dense* representations came to be by leveraging the distributional semantics hypothesis. Furthermore, we will explain the most common of those methods in greater detail.

The Distributional Hypothesis

The *distributional hypothesis* states that the meaning of a word is related to its context, which can be directly expressed by the distribution of the words that this particular word is used in conjunction with. The hypothesis came into prominence in the 1950's from American linguists such as Zellig S. Harris who rejected the structuralist school of linguistics and its concepts of *signifier* and *signified*. Perhaps the most succinct description of the distributional hypothesis' main premise was given by the philosopher

L.Wittgenstein [142]: "For a large class of cases—though not for all—in which we employ the word "meaning" it can be defined thus: the meaning of a word is its use in the language."

In term of constructing a useful word representation, the *distributional hypothesis* is quite useful. If we assume that all the syntactical and semantic information of a word lies in its context, then by modeling the distribution of its context we can encode all the information a word contains.

Word2Vec

The *Word to Vector* (commonly abbreviated to *Word2Vec*) representation algorithm, introduced in [91], is a step towards using the distributional hypothesis to train a neural network in a *self-supervised* (Section 2.3) way. In fact *word2vec* is not a single algorithm as the original paper introduced two separate algorithms: **Continuous Bag of Word** (CBOW) and **Continuous Skip-gram** (CSG).

The authors redefine the representation construction problem: as a problem of finding optimal parameters $\mathbf{w}$, $\mathbf{c}$ such that a logistic regression could predict correctly for a target w and a context window L, if some words belong to w's L-window context. The positive examples include the L-nearest words to w in a single document, while negative examples can be drawn by selecting noise words from the vocabulary via their unigram probability. The loss function, if the target words are projected to $\mathbf{w}$ and positive/negative context words to $\mathbf{c}_+$, $\mathbf{c}_-$ respectively, takes the form:

$$\begin{aligned}\mathcal{L} &= -log\Big(\prod_{i=1}^{L_+} P(+|\mathbf{w}, \mathbf{c}_i^+) \prod_{i=1}^{L_-} P(+|\mathbf{w}, \mathbf{c}_i^-)\Big) \\ &= -log\Big(\sum_{i=1}^{L_+} \sigma(\mathbf{w} \cdot \mathbf{c}_i^+) + \sum_{i=1}^{L_+} \sigma(-\mathbf{w} \cdot \mathbf{c}_i^-))\Big)\end{aligned}$$

After training, one could use just the $\mathbf{w}$ vector representations, an average of $\mathbf{w}$, $\mathbf{c}$, or even set $\mathbf{w} = \mathbf{c}$ during training. Both CBOW and CSG minimize the loss function above, with their difference being on which is the algorithm's input

- **Continuous Skip-gram** predicts given a target word, whether the contexts words are indeed context words.
- **Continuous Bag of Words** predicts given a list of context words, whether a word is indeed a target word.

GloVe

The *Global Vector* for Word Representation (GloVe) [106] attempts to construct a single-layer linear projection model, like word2vec, however it also uses global corpus statistics. The intuition is that, using the corpus' co-occurrence matrix and ratios of co-occurrences can be beneficial. Specifically, if w_i, w_j are co-occur often while w_k is a word that co-occurs with either w_i or w_j, then the co-occurrence ratio $\frac{P(k \text{ in context of i})}{P(k \text{ in context of j})}$ would be large.

However, if w_k co-occurs with both or neither w_i, w_j then the ratio would be small. Therefore, this ratio can be used to train a similarity function, since in the first case we would like the representations to be close while in the second case the representations should not be close to each other.

Starting from this assumptions and making other necessary assumptions, the authors prove that the word vector representations can be found by minimizing an equation like the following:

$$\mathbf{w}_i^T \tilde{\mathbf{w}}_k + b_i + \tilde{b}_k - log(X_{i,k}) = 0$$

where $X_{i,k}$ denotes the number of co-occurrences of w_k in the context of w_i, $\mathbf{w}_i$ denotes the vector representation of word w_i and $\tilde{\mathbf{w}}_k$ the vector representation of context word w_k using separate vectors for context words. This equation can be minimized directly by using linear regression (Section 2.4.3).

FastText

FastText [19] is an extension of *word2vec*'s skipgram embedding algorithm. The authors notice that tokenizing each word into a separate token ignores the morphological structure of each word and may lead to words that were not in the training data not have a vector representation. Therefore, they propose a subword representation of a word where a word consists of a start and end token and all possible character n-grams can be found in the sentence. The word representation is constructed by averaging the embeddings of its subwords. The inclusion of the start and end tokens helps the model differentiate when a morpheme is used inside a word, as a suffix or a prefix.

This approach leads to representations that perform better in word analogy tasks and in word similarity tasks.

3.4.5 Contextualized Representations

The representation methods outlined so far generate **static** embeddings, that is embeddings that once the training-phase is finished remain the same for each word - regardless of the context inside which each word is used. The methods leveraged each word's context to generate the embeddings but during inference the context is ignored.

The problem discussed above is easier understood in the case of *polysemy*, where a word has multiple senses that depend on the context it's used. A *static representation* algorithm would learn just one (context-free) vector for the word and would not have any way of disambiguating between its senses.

Recent advances in *deep learning* architectures has motivated research on better ways to initialize parts of the network using other networks pre-trained on other tasks. Using a deep neural network as the pre-trained part of the network enables encoding of *contextual* information, along with high-level features that the network learns because of its depth.

CoVe

The *Context Vector*(CoVe) [87] representation approach is based on contextualizing words by extracting word vectors from a deep neural network model. Specifically, a bidirectional LSTM Encoder-Decoder model with attention (Section 2.7.2) is trained on a machine translation task. After the training is finished, the model's encoder is used to generate a sequence of hidden states that correspond to *contextual* representations for each word. Those representation are used as input to a different task-specific model.

The authors show that their approach benefits the performance of models trained for different tasks, such as Question Answering and Text Classification.

ELMo

The *Embeddings from Language Model* [107] representations, similarly to *CoVe*, generate *contextual* representations from a bidirectional LSTM with attention. Their approach differs, however, on 2 points:

- The ELMo is trained on the language modeling (see Section 3.5) task, which can be done in a semi-supervised manner - making use of both labeled and unlabeled data, thus taking advantage of very large unlabeled datasets.
- Furthermore, the input to the model is character-based not word-based. The input is convoluted via a Convolutional Layer, in order to construct cross-character representations.

Their approach improved the State of the Art in several Natural Language Understanding/Inference tasks.

GPT

OpenAI's GPT model [112] uses a transformer model acting as a decoder to solve the language-modeling problem. The transformer architecture can capture longer contexts than a conventional LSTM model, while also being more faster to train.

During fine-tuning the model to different tasks, the input can be restructured in order to fit input data's format that is required by each task (Figure 3.2)

The authors fine-tune the trained model on different tasks such as Question Answering, Semantic Similarity and Classification, achieving state of the art results in most datasets. However, the use of the Transformer's architecture decoder renders the representations generated by GPT *unidirectional*. This is because, in order to avoid biasing a model with the expected outputs, a Transformer decoder's self-attention module must mask the part of the input sequence that is to the right of the last input token it has generated.

BERT

The *Bidirectional Encoder Representations from Transformers* (BERT) [29] represents a paradigm shift in the representation learning literature, as it uses a Transformer En-

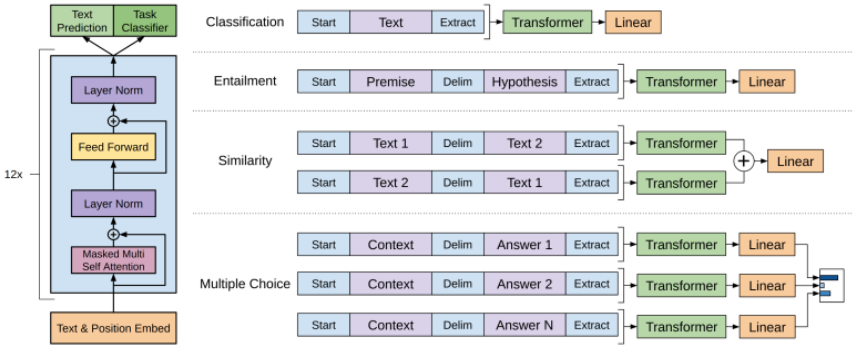


Figure 3.2. *Left: The Transformer [137] decoder architecture used for Language modeling in OpenAI's GPT [112]. Right: The input data transformation required in order to use GPT on different tasks. Source: [112]*

coder which can leverage the bidirectional context of each word. Figure 3.3 provides a direct comparison with previous models, highlighting BERT's ability of generating both contextualized and bi-directional representations.

The model is pre-trained on two tasks:

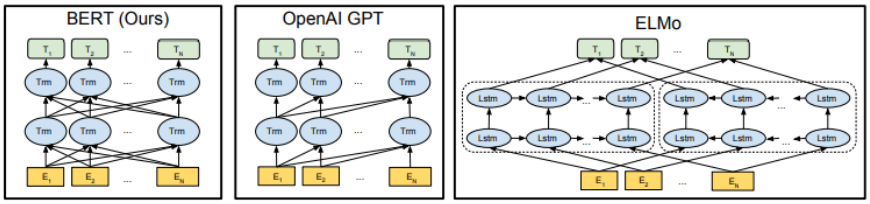


Figure 3.3. *A comparison of the way representations are generated by the neural network models we have described. Notice that only BERT learns bidirectional contextual representations directly. GPT [112] learns uni-directional representations, while ELMo [112] learns separate unidirectional contextual representations that merged together. Source: [29]*

1. **Masked Language Modeling:** This task replaces conventional uni-directional language modeling, which can be used to learn contextual representations encoding information only from one direction of the text. In this task, 15% of the input tokens are considered masked and are either: 1) replaced by the [MASK] token (80%), 2) replaced by a random token (10%) or 3) remain unchanged. The model is thus trained to predict the masked word. The reason masked words are not always replaced by the [MASK] token is to avoid big mismatches between pre-training and fine-tuning where the [MASK] token may not be present.

2. **Next Sentence Prediction:** In this task, the model is asked to predict whether as sentence B follows a sentence A in the text. To add negative learning examples, in 50% of the cases a random sentence is selected from whole corpus. Posing the problem as a binary classification task is useful, since if instead the next sentence was just masked and the model was asked to reproduce it, then this task would be significantly more under-constrained and thus less easy to learn from, while the

representations learned would include contextual information only from the first sentence.

The BERT model adopts a flexible input/output representation system (Figures: 3.4, 3.5):

- All words are tokenized using the WordPiece subword tokenizer. This minimizes out-of-vocabulary words and enables learning of morphological information. However it significantly increases the number of tokens in each sentence.
- A [CLS] token is inserted at the start of each sentence. This token can be used to for classification tasks, such as the Next Sentence Prediciton pre-training task..
- Seperation [SEP] tokens are inserted between segments that we wish to denote are separate. For example in the Next Sentence Prediction pre-training task, a separation token would be inserted between the two sentences. Furthermore,

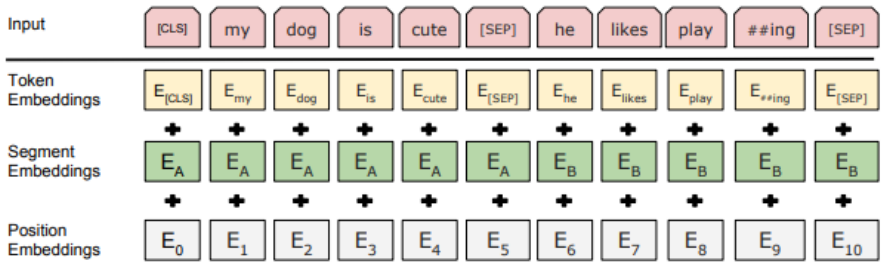


Figure 3.4. *BERT's input transformation system. Each word is tokenized using WordPiece's subword tokenizer, a [CLS] token is added at the start of each sentence, and [SEP] tokens seperate (question, answer) pairs. Source: [29]*

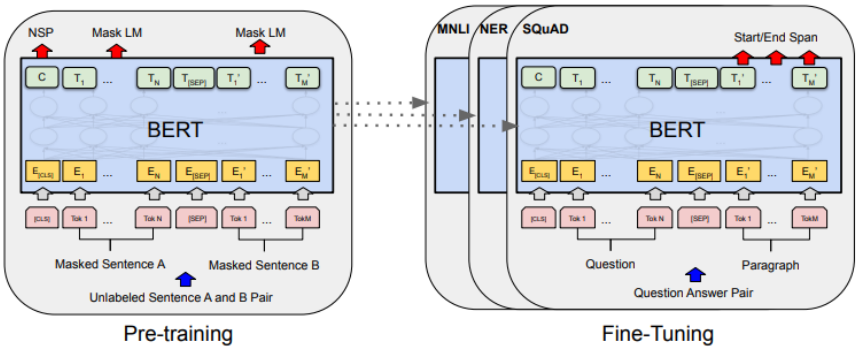


Figure 3.5. *BERT's pretraining and finetuning on a general two-sentence task. C denotes the contextual embedding of the [CLS] token, while T_i denotes the contextual embedding of the inputs i-th token. Source: [29]*

Beyond BERT

Several variations of BERT have been proposed which aim at either: 1) generating representations that are as good as BERT's but are smaller in terms of memory footprint

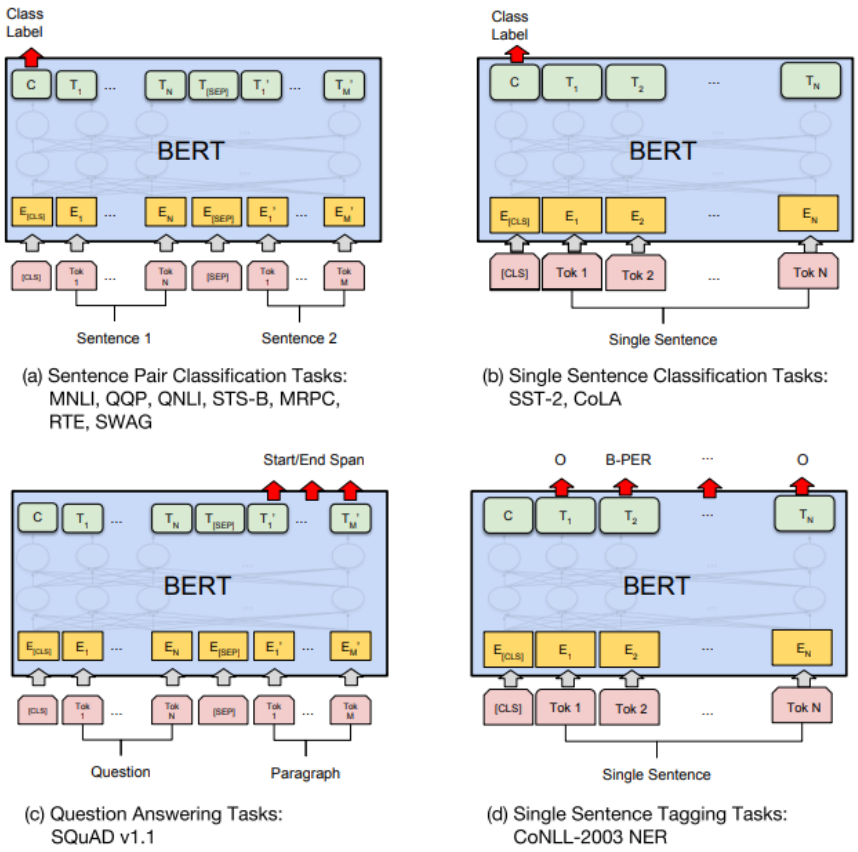


Figure 3.6. *BERT's finetuning on four different tasks. Note that BERT's input transformations are general enough to encode several tasks, with either one or many input sentences and with either a classification output or a sentence generation output. Source: [29]*

and require less computational power to generate, 2) changing the network's architecture in order to get more robust training and/or train on larger corpora, 3) generating representations for domain-specific texts, where the language may be domain-specific and therefore general-domain representations would not suffice.

DistilBERT [121] compresses BERT into a smaller and faster model, by training a smaller model to mimic the output distribution of BERT. The distilled model can have almost equal performance to BERT in several tasks.

RoBERTa [78] optimizes the original BERT's model training, by changing the Masked Language Modeling task to include different masks each time it encounters a sample, making slight changes to the NSP task, and selecting more robust hyperparameter values. BART [71] extends BERT's architecture by adding a decoder module, which enables arbitrary noising operations on the input pre-training data.

Pre-training from scratch or adapting pre-trained models on domain-specific data can be very useful for domain-specific tasks. Notable examples include: 1) LEGAL-BERT [21] where the model is trained on data from legislation, court cases, contracts 2) SCIBERT [11] where scientific publications from multiple domains are utilized for training 3) FinBERT [146] which pre-trains a BERT model on financial data and afterwards uses it to tackle the financial sentiment task.

3.5 Modeling Natural Language

Natural Language Modeling, most commonly abbreviated to LM, refers to the problem of modeling a language that has been *naturally* evolved through its use without any predefined rules. The modeling of a language can be thought as a probability problem where the model is tasked with finding the probability of a word-sequence occurring. Formally:

$$P(\mathbf{w}) = P(w_1, \ldots, w_T) = \prod_{i=1}^{T} P(w_i | w_1, \ldots w_{i-1}) \tag{3.5}$$

where the rightmost part is a direct use of the *chain rule of probability*.

A computer can, given a corpus C, directly model the probability $P(w_i|w_1, \ldots, w_{i-1})$ by the formula:

$$P(w_i|w_1, \ldots, w_{i-1}) = \frac{\text{COUNT}(w_i \text{ after } w_1 \ldots w_{i-1} \text{ in C})}{\text{COUNT}(w_1 \ldots w_{i-1} \text{ in C})} \tag{3.6}$$

It is easily seen that even with a large C some sentences which are not present in C, will be mapped to zero probability. This problem would be most evident in longer sequences of words. Furthermore, simple word interjections or omissions between words of a sentence, can result to a sentence having drastically different probability of occurring. In the next sections, we will discuss ways to mitigate the aforementioned problems.

3.5.1 N-gram

N-gram modeling is a direct application of equations 3.5,3.6 with the simplification of limited memory[7], that is a word occurring can be modeled as a function of itself and only a certain number of the words preceding it. Formally:

$$P(w_i|w_1, \ldots w_{i-1}) \approx P(w_i|w_{i-1}, \ldots w_{i-N}) \implies \tag{3.7}$$

$$P(w_1 \ldots w_T) = \prod_{i=1}^{T} P(w_i|w_{i-N} \ldots w_{i-1}) \tag{3.8}$$

This method enables the modeling of the probabilities for longer sequences, without needing to have extremely large amounts of data.

However, the N-gram memory simplification may produce wrong results when in order to predict the next most probable word - it is necessary to refer to textual context more than N-1 words away. Furthermore, the probability distribution produced by N-grams remains extremely sparse. Moreover, the memory required to store the probability distribution function grows exponentially with N, which is directly linked with the *curse of dimensionality* problem (see Section 3.4.3).

[7] This is often referred to as a *Markovian Assumption*.

3.5.2 Generalizing N-grams

The problem of unknown words

Unknown words is an important problem for a Language Model, since an N-gram method such as the one described in 3.7 cannot directly model the probability of a sentence containing a word that is not found in the vocabulary (OOV).

The most common solution to this, is inserting an *Unkown Word* token [UNK] into the vocabulary. However, the problem of modeling its probability remains. The simplest way to model is to already have a vocabulary, which can be used to replace all of corpus words that are not in it to the [UNK] token. The [UNK] token's probability in any N-gram can be then easily calculated as with every other token.

3.5.3 Smoothing

We can directly smoothen the probability density function in several straightforward ways. For example **add-k smoothing** modifies equation 3.6 by adding a constant factor into its numerator and the same factor multiplied by the cardinality of the N-gram token vocabulary:

$$P(w_i|w_1 \dots w_{i-1}) = \frac{\text{COUNT}(w_i \text{ after } w_1 \dots w_{i-N} \text{ in C}) + k}{\text{COUNT}(w_1 \dots w_{i-N} \text{ in C}) + |V|^N \cdot k} \tag{3.9}$$

3.5.4 Neural Network based Language Modeling

In order to avoid the computational complexity and sparsity problems that N-grams face, most Language Modeling nowadays uses Neural Network models. Usage of neural networks for language modeling was pioneered in the 1991 paper [90] by R.Miikkulainen and M.G.Dryer, however the method was popularized for larger neural networks in 2001 by Y.Bengio et al [13]. They implemented as simple 3-layered network where for predicting the i-th word of the output the N previous words were projected into their corresponding context vectors and passed through 2 layers of the network before being squashed into a word probability distribution by a softmax layer.

The intuition behind neural language modeling is simple. Due to the curse of dimensionality, learning a satisfactory language model directly from the words themselves and without utilizing their semantics is impossible. Therefore, probability of N-grams should not change by a lot when some of their tokens are replaced by syntactically or semantically similar tokens. This would require using word representations that can encapsulate this information, which is something that *deep* neural network achieve.

Modern Language Models leverage the rich information encoded in embedding that are generated by *deep* neural networks. As we have seen in Sections 2.7.1,2.7.2, Encoder-Decoder models such as RNN and LSTM can auto-regressively generate a sequence by predicting each time the next most probable token (i.e implementing a LM), will using rich representations of the tokens that preceded it. Finally, the breakthrough of the Transformer Encoder-Decoder (Section 2.7.4) model implements a Language Model with contextualized representations.

3.5.5 Evaluating a Language Model

Given a language model that can map each sequence of words $\mathbf{w}$ to a probability, how can we evaluate its performance on a test set that wasn't used during training? A most common way of to validate the model statistically using the perplexity function which for a single test-sample instance measures how probable (according to the LM) is the test-sample.

Let $\mathbf{TS} = \{\mathbf{t}_1, \ldots, \mathbf{t}_N\}$ be a set of test-samples where each test sample is a sequence of tokens $\mathbf{t}_i = w_1 \ldots w_{|t_i|}$. Extending the perplexity approach to a set of test-samples gives us:

$$PP(\mathbf{TS}) = 2^{-\frac{1}{N}\sum_{i=1}^{N} log_2 P(\mathbf{t}_i)}$$

where both the logarithm and the power function are used for numerical stabilization purposes. Lower complexity is desirable since it means that an unseen sequence corresponds to higher probability by the Language Model.

A perplexity score correlates both with the similarity of the train/test datasets, as well as with the inherent difficulty in learning the test dataset. Furthermore, it's corpus dependent meaning perplexities scores of models trained in different data cannot be compared directly.

In practise, perplexity is never used as the only automatic metric of measuring a Language Model's performance. Its main disadvantage are 1) penalizing the generation of samples not found in the training set, but which are equally valid 2) being overly sensitive to word reordering/interjections which generally don't change the semantics of a sentence that much. What perplexity can tell us is how well a model has fit the data given to it. Domain-specific automatic metrics for language generation, such as ROUGE [73], METEOR [10] and BLEU [104] are commonly used for evaluation generated language for a *specific task*.

Chapter 4

Automatic Legal Text Summarization

4.1 Introduction

Radev et al [110] defines a text summary as "(...) a text that is produced from one or more texts, that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually significantly less than that. Text here is used rather loosely and can refer to speech, multimedia documents, hypertext, etc". Due to manual summarization being complex and time consuming, from the 1950s significant research effort has been put into ATS focusing on summarizing, primarily, news articles and scientific journals, but recently on many other domains as well.

The intuition behind research in ATS is that the exponentially growing volume of digitized data is valuable only if it can be easily used by users directly or in downstream processes. Typical downstream processes of ATS systems currently in use are news page snippet generation in search engines, information retrieval and metadata generation.

In the legal domain, the information digitization has been typically slow. This can be attributed to the confidentiality of the texts used, the need to respect privacy laws and inertia which can be typical in traditional sectors, such as law. However, the ever increasing amount of digitized data in the law domain, the heterogeneity of the data's users and the data themselves has given rise to a need for ATS methods specialized for the law domain [36, 109, 39].

Up until the 1990s [136], the information retrieval from legal texts was conducted mostly manually, as automatic systems were just beginning to become commercially used. Consider the following three cases where ATS systems for legal court judgements can be useful: 1) Courts and organizations have specific teams of legal editors tasked to manually write a summary of a court's case. Automating part or all of the relevant process can allow legal editors to focus on other tasks of their job and contribute into more court judgements getting digitized with a summary. 2) Legal professionals and scholars have to read the entire judicial judgement to extract the main arguments of each side. Summaries that include these arguments can save them time, and allow for quantitative research in the field of legal arguments which wouldn't be possible by just manual summarization. 3) Lawyers, preparing their arguments for a case need to search for relevant past court judgements texts, selecting, through knowledge and experience, relevant passages, in order to acquire the in-depth context understanding they need.

Browsing through summarized versions of the judgments is intuitively easier and less time-consuming allowing them to focus on the main ideas and thus acquire a better understanding. Therefore, it is clear that ATS systems for legal texts can be very useful for many people working or researching in the legal domain, and that successful ATS systems that can meet the needs of diverse stakeholders; such as academics, lawyers, judges and simple citizens, need to be domain specific and adaptable.

In the following sections we will: 1) describe common datasets that are used in text summarization while also mentioning their disadvantages, 2) provide a review of the most common methods used in ATS, 3) discuss the automatic metrics used in evaluating ATS systems, as well as the criticism that has been applied to them, and 4) analyze the ATS methods developed specifically for the law domain, while mentioning the domain-specific challenges.

4.2 Datasets for General-Purpose Text Summarization

4.2.1 Introduction

In the following section we will introduce several text summarization datasets that are commonly used in the ATS literature. For each dataset we will mention its size, the type of text it contains and compare it with the other datasets that are commonly used. The datasets mainly come from the news-domain, which leads to several disadvantages such as the *extractive bias* and the *information layout bias* that we will expand on.

4.2.2 Common Datasets

In this section, we list some of the most common dataset used in Automatic Text Summarization research. We expand on each dataset's text domain and its properties concerning automatic summarization model training. The datasets are listed in chronological order.

The DUC, TAC conferences

The *Document Understanding Conference* (DUC, [102]) dataset consists of manually and automatically generated summaries of English newspaper and newswire articles, released as part the shared summarization task hosted at the DUC from 2001 through 2007. Each edition comes with different variations of the summarization task, including general/query-based summarization and single-document/multi-document summarization. From 2008 to this day, DUC's summarization task has been included in the Text Analysis Conference (TAC). The TAC datasets cover multi-document summarization, query-based summarization, update summarization and various other tasks.

Both datasets use texts from the news domain and, almost always, in the English language [33]. Although the datasets are useful for evaluation purposes, due to their limited size they're not typically used for training neural models [68]. In addition, as we

expand on in Section 4.5.3, the DUC datset contains human evaluation results for part of the texts' summaries.

The NYT news dataset

The *New York Times* (NYT, [120]) dataset contains over 650k articles from the New York Times magazine and summaries manually written by library scientists. It has been mostly used to train and evaluate extractive summarization models and phrase-importance predictors [70].

The CNN/Daily Mail corpus

The CNN/Daily Mail corpus [94] contains over 312k text-summary pairs of articles automatically scraped from the CNN and Daily Mail news websites. The corpus was produced by modifying an existing corpus used for passage-based question answering [49]. Each summary is manually created by the article's author and consist of the main bullet points accompanying the article's main text. The dataset has been use extensively on model training and evaluation for both extractive and abstractive ATS models.

The Wikisum dataset

Wikisum [75] is a multi-document summarization dataset containing summaries of 3.8M topics corresponding to a title of a Wikipedia article. The summary is the article's lead section while the cited articles (found in the Wikipedia article's references section) and the non- overlapping results of searching the topic on Google constitute the documents to be summarized. The dataset contains text that is significantly longer in size and its reference summaries contain more novel unigrams compared to traditional news datasets, such as CNN/Daily Mail and Gigaword, highlighting its importance on training abstractive models.

The Wikihow dataset

Wikihow is a large scale summarization dataset containing over 230k articles from the WikiHow knowledge base. Each article contains multi-step instructions explaining how to perform a certain task that can be solved in a series of steps. Each step has a one sentence summary and a more detailed explanation. The article summary consists of each step's one sentence summary while the rest of the text is the document to be summarized. The authors highlight: 1)the dataset's large proportion of novel unigrams in its reference summaries and 2)the small length of its reference summaries sentences (compared to each article's sentence length), as its major advantages over common datasets from the news domain.

The XSum dataset

The XSum dataset [96] consists of 226,711 BBC articles and their respective one-sentence summary. Each summary is written by the article's original author and it

serves as a preface to the article. This dataset is targeted for training and evaluating abstractive ATS systems, since simply extracting sentences from the input text would result in summaries considerably larger than the reference summaries. Furthermore, the authors support their claim of abstractiveness in their dataset by comparing the percentage of novel n-grams in the reference summaries that are not present in the source texts. Their comparisons show that XSum is more abstract than other news-domain datasets such as CNN/Daily Mail and NYT.

The Newsroom Dataset

The Newsroom dataset [46] consists of 1.3M articles from 38 major news publications and their respective summaries written by the article's author or the newsroom editor. The summary writing methodology varies from publication to publication, thus the dataset contains summaries that vary in terms of being more extractive or more abstractive. The authors proposed metrics for measuring the extractiveness and the abstractiveness of a summary and create subsets of their dataset into Newsroom-Abs, Newsroom-mixed, Newsroom-ext based on each subset's measure.

TIFU

The TIFU dataset [61] contains 120K posts scraped from Reddit's TIFU discussion forum where the users post personal mishaps of varying topics. Every post contains a relevant title and a summary (denoted by TLDR - an abbreviation for Too Long Didn't Read) written by the post's author. The TIFU-short version of the dataset uses the post's title as its summary, while the TIFU-long version uses the post's TLDR summary. The authors find that their dataset has 1)considerably higher abstractiveness and 2)the source text, in each example, has more uniform distribution of relevant (to the reference summary) information compared to traditional news-domain datasets.

4.2.3 Criticism of the Datasets

There has been extensive discussion on the datasets used for training and evaluating ATS systems. In this section we present some of the criticism found in the literature as summarized, mainly, in [70].

Summarization as an under-constrained task

The authors in [70] sample 100 random articles from the CNN/Daily Mail dataset and asked annotators to summarize them under two settings: 1) *constrained*: under which the annotators had to write a summary that answers a set of 3 questions relative to the article, and 2) *unconstrained*: under which they had to summarize what they thought to be important. In both settings, the annotators had to highlight the parts of the article they found useful in writing the summary and the sentences that were selected by at least n annotators were kept.

The results point to summarization being a moderately under-constrained task as 1) it is shown, that higher n results in significantly fewer sentences selected as important by at least n annotators and 2) annotators disagree more when they are asked to summarize under the unconstrained setting. However, the metric the authors used does not take into account the abstractive nature of manually written summaries.

Information layout bias

Datasets consisting of news articles usually follow the inverse pyramid pattern, namely, the closer one sentence is to the article's start, the more likely it is to be important in writing the article's summary. Thus, abstractive models trained on these datasets tend to be less abstract, since they can just repeat phrases from the beginning of the document [125, 96] and don't generalize well in more diverse datasets or datasets in other domains [57].

[70] measure the positional distribution of the sentences that annotators highlight as important for writing their summary of a randomly selected CNN/Daily Mail article, and find (see Fig.4.1) that, on average, almost 60 % of the important sentences are situated in the first third of the article.

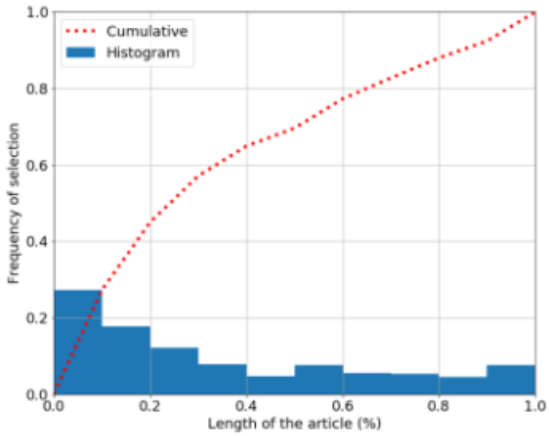


Figure 4.1. *Source: [70] Measuring how a sentence position in the article correlates with its probability of being highlighted by a human reviewer as important for writing a summary.*

Several dataset selection approaches have been proposed for reducing this bias. The TIFU [61] dataset attempts to alleviate this bias by choosing a discussion forum as the source of the summarization documents, since forum posts are more conversational and thus include less positional bias in their dispersion of important information . The Newsroom dataset [46] aims to reduce the information layout bias by including articles from various news publications thus including more diverse summarization strategies.

In conclusion, the inherent information layout bias in news articles datasets may be utilized to generate better summaries. However this approach may be too shallow for datasets in other domains, such as scientific articles, dialogue, forum posts, etc.

Extractive Bias

The extractive bias of a certain dataset is defined as the average n-gram overlap of each document, summary pair in the dataset. In essence, this measures how often a summary contains words and/or phrases not found in the document to be summarized. This bias varies between each dataset but, in general, more diverse datasets, datasets that include short summaries and datasets that contain documents from less formal topics exhibit less extractive bias in their reference summaries [61].

The extractive bias has been utilized in summarizing news articles in [125] where a pointer-generator network is trained to generate summaries by both using words from the whole network's vocabulary and copying words from the input text. However, it is not clear if the pointer-generator architecture can be of benefit in domains that do not exhibit extractive bias in their summaries. Furthermore, the copying mechanism may not be desired in cases where an abstractive summary is preferred.

4.3 Datasets for Legal Text Summarization

There exist several datasets for *Legal Text Summarization*, although due to the vast cross-country differences in the legal and specifically the judicial domain, those datasets may differ a lot even when they are written in the same language. For example, court judgement texts may contain or not contain headings indicative of the text's inner structure and may also have significant writing style differences that make thematic segmentation methods non-transferable to court judgements in other datasets [16]. Furthermore, there is not consistent use of already collected legal text summarization datasets, as every researcher trains and evaluates their methods in a dataset they collect for their specific research purposes, often enriched with non-standardized human annotations.

In this section, we list and describe several court-judgements summarization datasets.

Rechtspraak

The Rechtspraak dataset, provided by *Pandora Intelligence*[1], consists of 403,585 legal judgements from the "Rechtspraak" Dutch court. Each judgement text comes with the court's summary, the category label corresponding to the case and the court's verdict on the case.

BillSum

The Billsum [67] dataset contains 22,218 US Congressional and 1,237 Californian bills, both with their corresponding reference summaries. The goal dataset creators is to enable training and testing summarization models to different bill-related domains (Congressional & Californian in this case).

[1] https://www.pandoraintelligence.com/

Indian Supreme Court

The authors in [16] collect 17,347 judgements by the Supreme Court of India, from the years 1990-2018, along with their summaries created by *Westlaw* India[2]. The same authors in [17] collect 7,100 Supreme Court of India legal cases along with their headnotes that serve as short abstractive summaries. They further task human experts to segment the document into *Rhetorical Role* segments, and summarize each segment separately.

Multi-LexSum

The Multi-LexSum [126] dataset consists of 40,000 federal U.S large-scale civil rights lawsuit documents collected by the Civil Rights Litigation Clearinghouse (CRLC) organization. Each document contains several texts corresponding the complaint,motion,judicial opinion or settlement of the legal case, and thus collectively each text is very large in size. The dataset also contains abstractive summaries written by CRLC experts for 9,000 of the CLRC documents, with the summaries coming in different granularities: from tiny(25 words) to long(650 words). The input document size along with the expert-generated and curated summaries makes the Multi-LexSum dataset idea for evaluating Legal Multi-document Summarization methods on multiple summary granularities.

Canadian Courts

The dataset introduced in [145], contains 28,733 legal cases, that took place in Canadian courts, along with their corresponding human-generated summaries. A subset of the summaries and the main texts were classified by law-students into the following categories: issue, conclusion, reason, neither of the above.

US Board of Veterans' Appeals

The US Board of Veterans' Appeals legal cases dataset, introduced in [150], contains 35,000 veteran's appeal cases concerning whether the appellant was to be compensated for their service-connected Post-traumatic Stress Disorder (PTSD). Each case's summary is drafted by single judges or their staff attorneys. For a subset of the dataset, the authors provide extractive summaries, additional abstractive summaries and thematic classification data.

Biases in Legal Text Summarization Datasets

The literature in text summarization dataset biases has not been systematically extended to the domain of legal texts. That can be attributed to the lack of standardized datasets, and the differences between the legal text summarization datasets.

In [145], the inter-annotator agreement on the court judgement thematic segmentation task is found to be moderate. This points to the thematic tagging task being not under-constrained. The development of extensive annotation guidelines was explored in [144,

[2] http://www.westlawindia.com

126]. We know of no systematic analysis of the *position bias* or the *extractive bias* in the legal text summarization literature.

4.4 Automatic Summarization Methods

Automatic Summarization Methods can be classified by the type of summaries they generate, into three categories: 1) **extractive** methods; which create summaries by extracting the most important segments of a text, 2) **abstractive** methods; which generate a summary from scratch, without extracting segments from the text, and 3) **hybrid** methods; which combine the two previous approaches.

Furthermore, ATS methods may differ depending on the textual domain they are applied on, since both 1) the input text, 2) the evaluation metrics used measuring the summary's quality and 3) the type of summary required, is usually specific to the domain the ATS method is applied.

A different taxonomy of the ATS methods found in the literature, can be based on the number of input documents. **Single-document** ATS methods summarize a single document, with typical applications being the summarization of a news article, a book, an email, etc. **Multi-document** ATS methods generate a summary for multiple documents, with common applications including summarizing restaurant/hotel reviews, twitter thread/hashtags, etc. Several other classifications can be made, which are extensively detailed in [33]

In our work we will discuss exclusively the *Single-document* ATS methods, both *extractive* and *abstractive*, while also describing ATS methods specific for the legal-texts domain.

4.4.1 Extractive Summarization

Extractive Summarization methods generate a summary by extracting the most important sentences from the text[3]. The basic pipeline (Figure 4.2) of all extractive methods can be generalized as follows:

1. Each textual segment is preprocessed and a representation is constructed corresponding to it.

2. Through each segment's representation, the method assigns scores that correspond to each segment's probability of being extracted for a summary.

3. Using the aforementioned scores, a selection is made and post-processing transformations can enforce a summary constraint or make changes to its form (such as re-order the sentences).

Following [33, 97], we classify the extractive methods in the sections below.

[3]From here on, when we refer to sentences in this section we will not necessarily mean grammatical sentences, but sequences of tokens. Similarly with words we will generally refer to tokens.

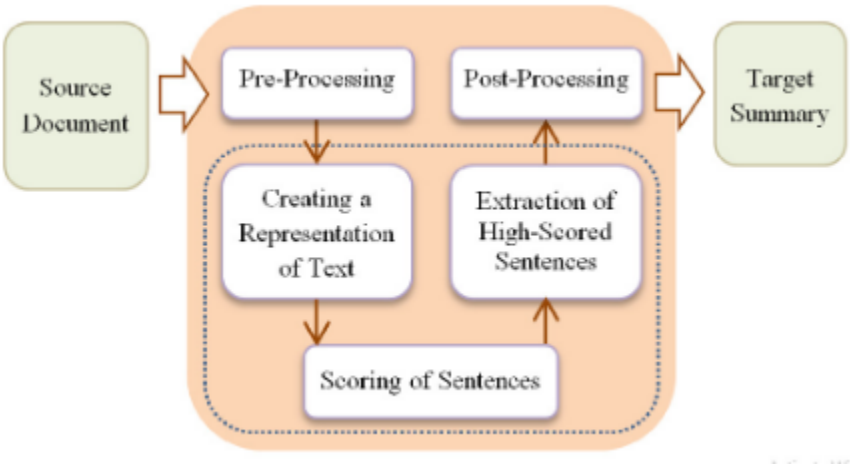


Figure 4.2. *A general workflow of an extractive ATS system. Source: [33]*

Linguistic Features based methods

The first approach on automating extractive summarization was proposed by Luhn in 1958 [80]. A list of words that are "significant" for a certain topic was compiled. Those words were modeled to be neither overly common, nor rare in the topic's corpus. Sentences that were eligible for extraction had to include "significant" words[4] no more than 4-words apart, and the sentence's significance score was the ratio of the square of the significant words it contained over its length.

Frequency based methods

Luhn's hard cut-off between significant words and insignificant words can be replaced by a statistical formula that measures how important that word is in the sentence. The tf-idf score (see Section 3.4.3) measures the word's importance in a sentence while also lowering the importance of words that are common across different documents. A similar to Luhn's sentence extraction criterion with modified sentence scores can be applied.

Document Centroid Based Methods

The MEAD software [111] used multiple documents to calculate the average tf-idf score of each word, thereby creating a *centroid* tf-idf score. Each sentence's importance was calculated by adding the tf-idf score of each of its words that exceeded a tf-idf score threshold.

Similarly for single-document summarization, a centroid-based approach consists of finding the average tf-idf sentence representation in a text and afterwards selecting the sentences that are most similar to the centroid (usually the cosine distance function is used). This approach, although successful, fails to model inter-sentence importance as each sentence is compared only to the *centroid* sentence.

[4] Later ATS literature referred to those words as "topic words"

Graph based methods

Graph-based approaches attempt to bypass the problems faced by the centroid sentence approaches, by constructing a graph of the sentences where each node denotes the similarity between each sentence. Sentences whose similarity score is below a certain threshold can be thought as non-adjacent.

A straight-forward way of modeling each sentence's importance using the PageRank [103] algorithm was proposed independently in the *LexRank* algorithm [34], and the *TextRank* algorithm [89]. LexRank was proposed for multi-document summarization while TextRank was applied to single-document sentence or keyword extraction. They have further minor differences in the text preprocessing and the parameterization of the algorithm.

The basic intuition was to apply the PageRank [103] algorithm over the sentence-graph:

$$p(u) = \frac{d}{N} + (1-d) \sum_{v \in \text{adj}[u]} \frac{\text{sim}(u,v)}{\sum_{z \in \text{adj[v]}} sim(z,v)} p(v) \tag{4.1}$$

Each sentence's score measures its *centrality*, that is its importance in the cluster of sentences that are similar to it. This enables having multiple sentence clusters with important sentences that are different between each cluster, whereas previous centroid-based algorithms had just a centroid sentence as a template.

The scores are updated iteratively until the algorithm converges. The convergence is guaranteed by properties of *stochastic matrices* in a *Markov Chain*. More specifically, a vectorized version of equation 4.1 is:

$$\mathbf{p} = \Big(d\mathbf{U} + (1-d)\mathbf{S}\Big)^T \mathbf{p} = \mathbf{A}^T \mathbf{p} \tag{4.2}$$

where $\mathbf{p}$ is the centrality scores vector, $\mathbf{S}$ the cross-sentence similarity matrix and $\mathbf{U}$ is a matrix whose every element is equal to 1/N.

$\mathbf{A}$ corresponds to a transition matrix of a Markovian chain. For $\mathbf{p}$ to converge into a stationary distribution, it suffices that $\mathbf{A}$ is such that the Markovian chain is *irreducible*[5] and *aperiodic*[6]. By inserting the "dumping factor" d the convergence is guaranteed.

There are many variations of the TextRank/PageRank algorithm. One particularly important in query-based summarization is the Biased LexRank algorithm [101]. It modifies equation 4.1 by increasing the dumping factor for sentences that are most relevant to the query and similarly decreases it for not relevant sentences:

$$p(u) = d\frac{rel(u,q)}{\sum_{z \in \text{Corpus}} rel(z,q)} + (1-d) \sum_{v \in \text{adj}[u]} \frac{\text{sim}(u,v)}{\sum_{z \in \text{adj[v]}} sim(z,v)} p(v) \tag{4.3}$$

[5] $\forall i,j \; \exists n : \mathbf{A}^n(i,j) \neq 0$. This means that every state is reachable by another state. The inclusion of the term $d\mathbf{U}$ guarantees that.

[6] $\text{gcd}\{n : \mathbf{A}^n(i,i) > 0\} = 1, \; \forall i$.

Relevance functions that can be used include a word-frequency function:

$$rel(u, q) = \sum_{w \in q} log(tf(w, u) + 1) \cdot log(tf(w, q) + 1) \cdot idf(w)$$

or in the case of vector embeddings vector similarity functions, such as the cosine distance function :

$$rel(\mathbf{u}, \mathbf{q}) = \frac{\mathbf{u} \cdot \mathbf{q}}{||\mathbf{u}|| \cdot ||\mathbf{q}||}$$

Matrix Decomposition based methods

Those methods attempt to model the topics found in each text and then extract sentences that are closest to each topic. This method is called *Latent Semantic Analysis* (LSA) [128] and its based on the *Singular Value Decomposition* of the documents sentence-token matrix.

Formally, let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be the sentence-token matrix where m is the size of the document's vocabulary and n is the number of its sentences. Typically, each sentence-token element corresponds to its tf-idf score, but other sentence-token representation methods can be applied. SVD decomposes the matrix into:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{4.4}$$

where $\mathbf{U} \in \mathbb{R}^{m \times n}$, $\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$, $\mathbf{V}^T \in \mathbb{C}^{n \times n}$ with $\mathbf{\Sigma}$ being a diagonal non-negative matrix. Diagonal matrix $\mathbf{\Sigma} = \{\sigma_1^2, \ldots, \sigma_n^2\}$ captures the importance of each topic in the document, while each row of matrix $\boldsymbol{V}^T$ corresponds to the topic-based representation of each sentence. Therefore, each row i in matrix $\mathbf{D} = \mathbf{\Sigma}\mathbf{V}^T$ corresponds to the significance to of topic i in every sentence of the document. In order to select only the most important topics in modeling the document (similarly with applying a similarity threshold in previous methods) the matrices can be pruned as in Figure 4.3

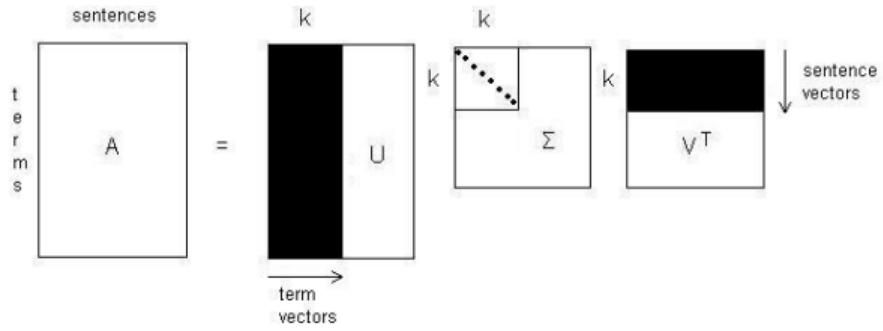


Figure 4.3. *A schematic of the LSA method using SVD on the term(token)-sentence matrix. Only the k-most important topics are taken into consideration, by pruning the corresponding rows and columns as shown. Source: [128]*

Having decomposed the token-sentences matrix, there are several ways of extracting the most important sentences:

- A straightforward approach is to select one sentence for each topic; the sentence that is closest to the topic as modeled in each row of $\mathbf{V}^T$.

- Another approach, is to take into consideration the topic's significance as well as all the topics in each sentence by calculating for each sentence the score function:

$$s_k = \sqrt{\sum_{j=1}^{k} v_{k,j}^2 \sigma_j^2} \tag{4.5}$$

  This way the most important sentences would be those that are most similar to significant topics.

Deep Learning based methods

We now turn our attention to *deep learning* methods for extractive summarization. Those methods utilize the rich representations generated by deep neural networks in order to select which text segments should be extracted for a summary.

Recurrent neural networks have been used in order to select which sentence will be extracted. [25] use a Convolutional Neural Network to encode each sentence separately and afterwards use those the sequence of those encodings as input into an LSTM network (Figure 4.4)

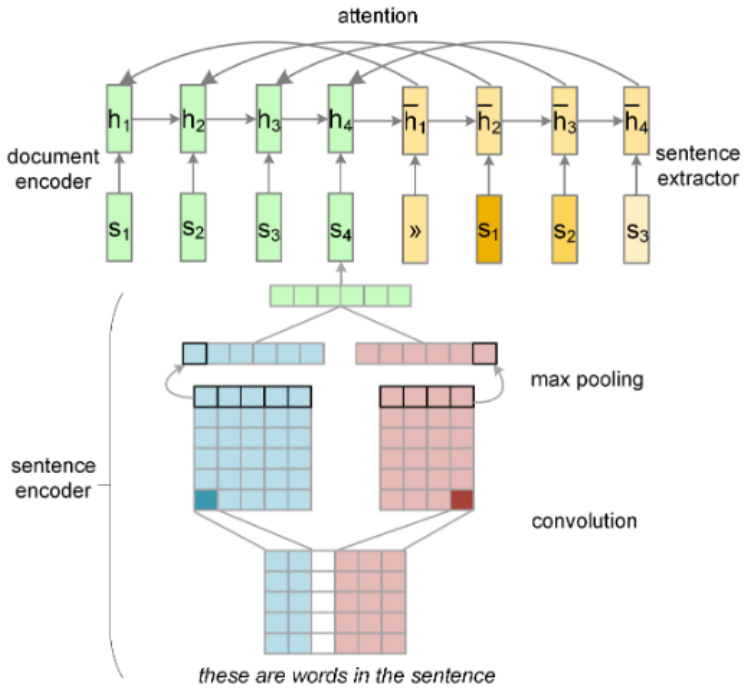


Figure 4.4. *Combining a convolutional sentence encoder with an LSTM document encoder. Source: [25]*

In [77] the BERT bidirectional-transformer network is used for predicting whether a sentence should be extracted or not. Bert enables use of *contextual* sentence embeddings that are useful in cross-sentence tasks. A vanilla BERT encoder cannot generate sen-

tence level representations, as its output vectors are grounded to token representations, including [CLS] token's representation which can be used as a representation for the whole document. The architecture is modified (Figure 4.5) by interchanging the segment embeddings between sentences and by inserting a [CLS] token between each sentence. Those [CLS] token's BERT embeddings will afterwards be used for the binary classification of the extraction label.

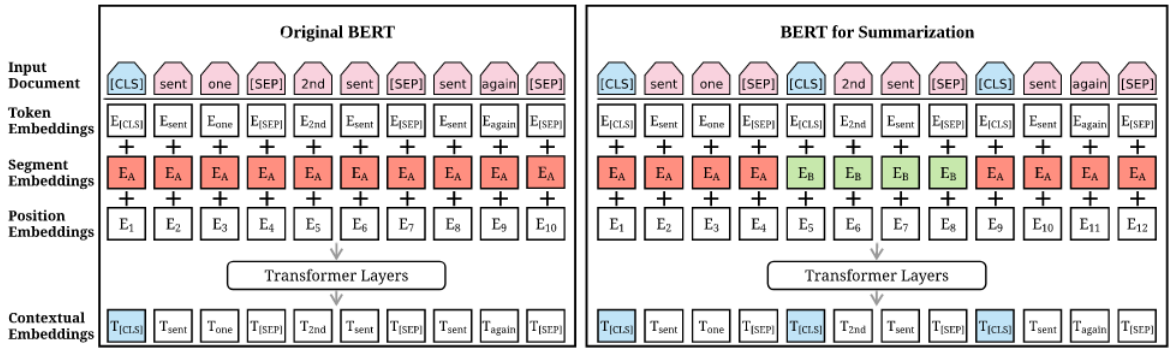


Figure 4.5. *Adapting the original BERT encoder (left) to produce sentence level tokens for extractive summarization (right). Source: [77]*

4.4.2 Abstractive Summarization

Abstractive ATS methods generate summaries without needing to copy words or segments from the document. This adds increased flexibility in the summary generation process, as the generated summary doesn't necessarily have to use words only found in the input document. Furthermore, it enables generating shorter but more succinct summaries than those generated by extractive ATS methods which extract sentences that may include overlapping information.

However, abstractive ATS systems require a deeper understanding of the input text in to produce a good summary. Deeper understanding of the input text is usually achieved using neural network *deep representations*, while deeper understanding during summary generation requires a sophisticated enough *language model*.

In the following sections we will introduce modern approaches to abstractive text summarization utilizing *deep neural networks*. We focus on *deep neural* methods, since they are the first to reach satisfactory results in abstractive summary generation.

Deep Neural Network methods

In [27] recurrent neural networks with attention are utilized in order to generate abstractive summaries. [94] insert additional POS, NER tags in each word embedding to better capture *key-words*, and furthermore introduce a "switch" mechanism that can decide in the case of *Out-of-Vocabulary* tokens whether to generate each new token by using the decoder's output or by copying the correspondent word from the input text. The copying mechanism is further explored in [125], where the generated word probability distribution is instead calculated by mixing the word probability distribution of the

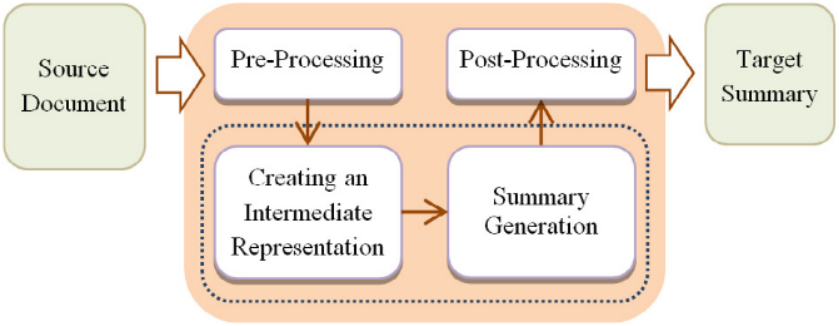


Figure 4.6. *A general schema of an abstractive ATS system. Source: [33]*

decoder output with the word probability distribution of copying the word from the text. They further introduce a *coverage* regularization mechanism that ensures the attention weights are activated evenly throughtout the text.

The BERT architecture was first used for abstractive summarization in [77] where a BERT encoder pretrained on extractive summarization was trained in conjunction with a BERT decoder to produce abstractive summaries. BERT's architecture is extended in [71] by the insertion of a dedicated decoder module which enables better pretraining on language generation tasks such as summarization. The PEGASUS pretrained encoder is utilized in [129] where the problem of training using the cross entropy loss on only one reference summary is mitigated by contrastive learning on different "silver" summaries. Similarly, [79] use contrastive learning on a distribution of silver summaries so that the probability of generating each candidate summary is related to their quality, as judged by the model.

4.5 Evaluation Metrics for Summarization

4.5.1 Introduction

Several Metrics have been proposed for the evaluation of Automatic Text Summarization models. We can categorize those metrics to 1) *automatic* metrics; which don't require any human evaluation of the summaries, and 2) *human* evaluation metrics; which require manual evaluation by a human. Automatic metrics enable faster and less expensive evaluation of large amount of summaries, compared to human evaluation metrics. However, as we will expand on in Section 4.5.3, those automatic metrics may not necessarily agree with human judgement. Therefore, ATS methods are, almost always, also evaluated by humans.

4.5.2 Automatic Evaluation Metrics

In [40], the automated evaluation metrics are categorized into intrinsic metrics, which measure the summary's quality based on a reference summary, and into extrinsic metrics, which measure the summary's quality on the basis of the summary's usefulness in solving other tasks, such as Question Answering and Information Retrieval.

Intrinsic summarization metrics aim to measure the lexical or semantic overlap between the generated summary and the reference summary, as a proxy for evaluating the summary's informativeness and quality. Informativeness refers to the coverage the generated summary provides over the reference summary or the original document and is measured mainly using lexical overlap metrics such as ROUGE [73], BLEU [104] and METEOR [10]. Quality refers to desired properties of the generated summary, such as grammaticality, redundancy, coherence and readability. The degree to which these qualities, as measured by human reviewers, are correlated to lexical overlap metrics is still debated in the literature [99, 35] and researchers in automatic summarization often include human evaluation of their summaries to measure these qualities.

BLEU

BLEU (*Bilingual Evaluation Understudy*) [104] is a language-agnostic evaluation metric designed for evaluating machine translation. BLEU is a precision-based N-gram overlap metric, which means it calculates the N-gram overlap between reference and candidate translation normalized by the number of N-grams present in the candidate translation. More formally, the BLEU-N score of a set of candidate translations C and a reference translation S is given by:

$$BLEU - N(C,S) = \frac{\sum_{c \in C} \sum_{g \in GRAM(N,c)} count_{clip}(g,S)}{\sum_{c \in C} \sum_{g \in GRAM(N,c)} count(g,S)}$$

where GRAM(N, c) denotes the set of N-grams in text c, and $count_{clip}(g,S)$ the number of occurrences of the N-gram g clipped by the number of its occurrences in the reference translation S, in order to avoid awarding high scores to candidate translations that repeat the same "reasonable" N-gram. The authors also introduce a brevity penalty to encourage matching candidate-reference translation length.

ROUGE

ROUGE (*Recall Oriented Understudy for Gisting Evaluation*,[73]) is a software package that was developed specifically for evaluating automatically generated summaries. It includes a variety of lexical-overlap evaluation metrics, each aiming to measure different aspects of the summarizer's performance. It has been widely adopted as the most-prominent automatic evaluation metric in machine text summarization [1, 40]. Below we define each ROUGE variants, as described in the original paper [73], and explain their differences:

- **ROUGE-N**: measures the N-gram overlap between the candidate summary and a set of reference summaries, namely, if we define RS to be the set of reference summaries, and GRAM(N,S) as the set of all N-grams in a candidate summary S:

$$ROUGE - N(RF,S) = \frac{\sum_{ref\_sum \in RF} \sum_{g \in GRAM(N,ref\_sum)} count_{matching}(g)}{\sum_{ref\_sum \in RF} \sum_{g \in GRAM(N,ref\_sum)} count(g)}$$

which favours candidate summaries with common N-grams across multiple reference summaries, as the denominator normalizes the nominator's sum over all possible reference summaries N-grams. In the case of a single reference summary, the ROUGE-N definition is simplified to:

$$ROUGE - N(ref\_sum, S) = \frac{\sum_{g \in GRAM(N, ref\_sum)} count_{matching}(g)}{\sum_{g \in GRAM(N, ref\_sum)} count(g)}$$

The metric is recall oriented because the percentage of overlapping N-grams is calculated over the N-grams found in the reference summaries.

- **ROUGE-S**: measures the Skip 2-gram overlap between a reference summary X and a candidate summary Y normalized by the number of skip 2-grams in the reference/-candidate summary, depending on if recall or precision is measured. In essence, skip 2-grams find pairs of words that are common in both reference & candidate summary and appear in the same order in both of them, while also allowing for gaps of a predefined maximum length d_{skip}. If d_{skip} is set equal to zero, then ROUGE-S is equivalent to ROUGE-2. If d_{skip} is increased more common pairs of words are captured which may lead to spurious matching.

- **ROUGE-SU***: is an generalized extension of the ROUGE-S metric. It allows for arbitrary values of d_{skip} (e.g ROUGE-SU4 uses $d_{skip} = 4$), and also captures unigram overlap (which can be achieved by adding to both reference and candidate summary a dummy word-token every $d_{skip} + 1$ words.

- **ROUGE-L**: measures the length of the longest common subsequence of words found in both generated and reference summary. A subsequence of words is defined as a sequence of words which can be found in the original sequence in the exact relative order they appear in the subsequence. The LCS score is normalized by the candidate summary's length, when measuring recall or the reference summary's length when measuring precision accordingly. More formally,

$$LCS_R(ref\_sum, cand\_sum) = \frac{LCS(ref\_sum, cand\_sum)}{|cand\_sum|}$$

$$LCS_P(ref\_sum, cand\_sum) = \frac{LCS(ref\_sum, cand\_sum)}{|ref\_sum|}$$

In order to define ROUGE-L for whole summaries, we define each sentence in both candidate and reference summary to be a separate sequence of words. The ROUGE-L score of a candidate summary C and a reference summary R is defined:

$$ROUGE - L(R, C) = \frac{\sum_{r \in R} LCS_{\cup}(r, C)}{|R|}$$

where the nominator is divided by number of words in the reference summary in order to measure recall, and $LCS_{\cup}$ denotes the union-Longest Common Subsequence

which, more formally, is defined as:

$$LCS_{\cup}(r, C) = |\cup_{c \in C} \{largest\_subsequence(r, c)\}|$$

This approach has obvious advantages over ROUGE-N as it 1) allows for measuring fluency in the candidate summary and 2) matching lexical overlap on a sentence level basis and not strictly in consecutive word order. However LCS-score does not change whether or not there are common subsequences of smaller length than LCS and penalizes simple changes in the order of the words which may not change the meaning of the text.

- **ROUGE-W**: generalizes the ROUGE-L metric by assigning different credit to each LCS depending on how consecutive are its words, this way penalizing LCS with many non-consecutive words. The Weighted LCS is calculated using a Dynamic Programming table that stores at each pair of word indices i,j (iterating over reference and candidate summary respectively) the length of consecutive matches at position i,j and a Dynamic Programming table that calculates the Weighted LCS up to the indices i,j awarding bigger scores to indexes pairs that correspond to bigger length of consecutive matches.

ROUGE-1|2|L have become the standard methods to evaluate an automatic summarization system's performance in informativeness and fluency [86]. However, it is often found to be very weakly correlated with a human reviewer's judgement of those aspects [35]. There have been attempts to extend ROUGE metrics beyond simple lexical matching, which we describe below. However, they are not widely used in the literature, although they claim higher correlation to human judgements.

- **ROUGE-WE**: [98] extends the original ROUGE metrics beyond exact N-gram matching. It uses soft semantic matching, by calculating the similarity of the Word2Vec [91] vectors of the reference and candidate N-grams. In order to reduce the number of OOV N-grams, which would be more for large values of N and on ROUGE-SU^*, n-grams Word2Vec representations are composed by the element-wise product of their word's Word2Vec vectors.

- **ROUGE-2.0**: [41] offers improvements over the original ROUGE metrics, adding synonym matching, stop-words removal, and topic-oriented evaluation based on POS tagging.

- **ROUGE-G**: is a graph-based approach that aims to offer joint lexical and semantic similarity evaluation. The authors utilize WordNet [38] on which a topic-sensitive version of PageRank is run in order to create representations for every set of word *senses*. The vectors semantic similarity is combined with the original ROUGE's lexical similarity to calculate the final evaluation score.

METEOR

METEOR (*Metric for Evaluation of Translation with Explicit ORdering*) [10] is an automatic machine translation evaluation metric introduced as an improvement of the BLEU metric, that we described in section 4.5.2, based on lexical alignment. The authors measure harmonic means, as they deem BLEU's brevity penalty insufficient, and pre-align the uni-grams of the reference and the candidate translation using the Porter Stemmer Algorithm and synonimity. Then, the regular lexical unigram-overlap metrics are calculated, as in BLEU and ROUGE, however here the harmonic mean (F-score) is used. In addition, the authors aim to award longer n-gram overlap by penalizing over-fragmented alignments: the candidate translation is split into chunks of consecutive uni-grams aligned with the corresponding consecutive uni-grams in the reference translation.

BERTScore

BertScore [149] is an automatic evaluation metric used to evaluate performance in text generation tasks, such as machine translation, summarization and image captioning. The metric proposes computing the similarity of the contextual embeddings of the reference and the candidate text. The metric, by not using N-gram matching such as those found in conventional text generation evaluation metrics, aims to offer a more robust evaluation system. More formally, let $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$, $\mathbf{y} = \langle y_1, \ldots, y_n \rangle$ be the tokenized reference and candidate text accordingly. The tokenized text is contextually embedded by a pretrained BERT model:

$$\mathbf{X} = BERT(\mathbf{x}),\ \mathbf{Y} = BERT(\mathbf{y})$$

where $\mathbf{X} = \langle \mathbf{X_1}, \ldots \mathbf{X_D} \rangle, \mathbf{Y} = \langle \mathbf{Y_1}, \ldots \mathbf{Y_W} \rangle \in R^{W \times W}$ with D being equal to the BERT's model hidden size and D to the model's context window. Therefore, soft-matching each one of the D token's contextualized embeddings in $\mathbf{X}$ with its most similar in $\mathbf{Y}$ (and reversely), using the cosine vector similarity function, grants us the following metrics

$$BERTScore_R(\mathbf{X}, \mathbf{Y}) = \frac{1}{|\mathbf{x}|} \sum_{x_i \in \mathbf{x}} \max_{y_j \in \mathbf{y}} \frac{\mathbf{X_i}^T \mathbf{Y_j}}{|\mathbf{X_i}||\mathbf{Y_j}|}$$

$$BERTScore_P(\mathbf{X}, \mathbf{Y}) = \frac{1}{|\mathbf{y}|} \sum_{y_j \in \mathbf{y}} \max_{x_i \in \mathbf{x}} \frac{\mathbf{X_i}^T \mathbf{Y_j}}{|\mathbf{X_i}||\mathbf{Y_j}|}$$

Frequent words might dilute the results as they are rarely indicative of cross-text similarity. Therefore, the authors also normalize by each token's idf score and apply plus-one smoothing to get an idf value for OOV words. The authors offer an evaluation framework that is language-agnostic and doesn't require much fine-tuning, but comes with serious computational costs which is severely increased if there is no pretrained BERT model in the language used.

4.5.3 Criticism of the Automatic Evaluation Metrics

As we described in the introductory section 4.5.1, automatic evaluation of a summary may be conducted using various criteria, ranging from informativeness to cohesiveness to factuality and coherence. This is very difficult to be achieved with metrics, as the lexical-overlap metrics defined in Section 4.5.2, not designed for this purpose. In the rest of this section, we shall give a brief overview of the main points of criticism, towards the most commonly used summarization evaluation metrics, as found in the recent literature.

Weak correlation with human evaluations:

ROUGE metrics were initially evaluated on the DUC dataset by checking their correlation with human judgement. Human judgement is usually calculated, following [42], as the average human score on coherence, fluency, consistency and relevance. However, [108] has shown that the DUC dataset contains relatively lower scored human evaluations and that most of the inter-metric disagreement happens on higher human-rated summaries Fig. 4.7. More recent work [15], on the more widely used *CNN/Daily Mail* dataset finds that the inter-metric disagreement can also take place in lower scoring ranges.

ROUGE 's correlation with human judgement on the CNN/Daily Mail dataset is investigated in [70], finding weak correlation for summaries generated by extractive and abstractive ATS systems. [35], recently made a comprehensive study on 12 common evaluation metrics for automatic summarization comparing different model architectures trained on CNN/Daily Mail. Fluency and coherence scores are found to be very weakly correlated with most metrics. Relevance shows weak or moderate correlation with most metrics which is expected since most metrics are based on token overlap.

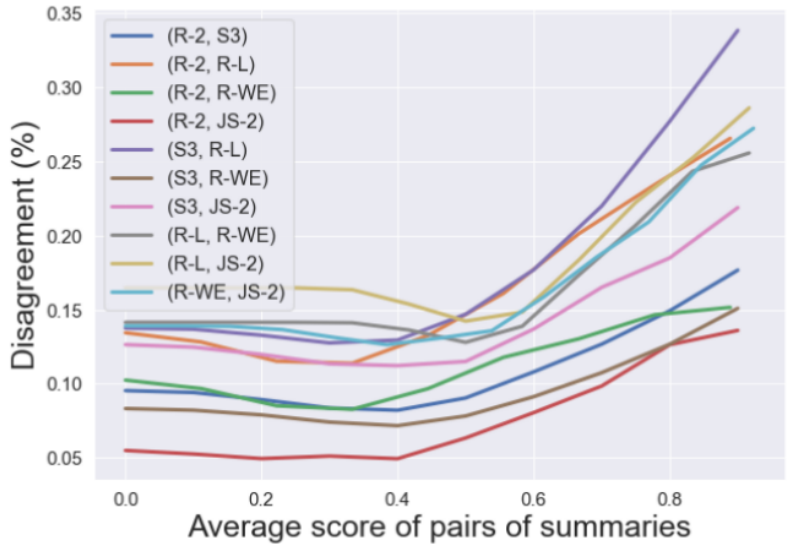


Figure 4.7. *Disagreement of various evaluation metrics in different scoring ranges. Source: [108]*

Evaluating faithfulness/factuality

The extent to which automatic abstractive summarization systems "hallucinate" facts is explored in [86]. It is shown that most summaries generated by abstractive summarization systems contain "hallucinations" (i.e inferences using information mostly extrinsic to the main input text) and thus are unfaithful. However only a small percent of unfaithful summaries are factual, and in fact, faithfulness and factuality are weakly correlated with the ROUGE-N, ROUGE-L, BERTScore metrics, pointing to the metrics being inadequate to evaluate summaries under those aspects.

[42] investigate the correlation of various automatic evaluation metrics with intrinsic summary properties such as Consistency and fluency. A moderate negative correlation is shown between the novel N-gram evaluation metrics and human-evaluated Consistency and, thus, faithfulness may be at odds with summaries generated by abstractive systems, which are prone to include more novel N-grams in their summaries.

4.5.4 Human Evaluation Metrics

Automatic metrics can be useful for measuring the informativeness of a summary, however as we saw in section 4.5.3, they can be mediocre indicators of a summary's quality. Therefore, it is not uncommon to have human annotators score the quality of an automatically generated summary. The human evaluation data can be used to improve the metrics used to evaluate summarization tasks [35] and provide a more robust framework of evaluating summaries. The human studies are usually conducted via Amazon's Mechanical Turk. Generally, recent literature that employs human annotators uses the following criteria [70]:

- **Relevance**: The degree to which a summary has captured the important content from the source.
- **Consistency**: Consistency measures the factual alignment between the summary and the source text.
- **Fluency**: The degree to which the summary contains individually fluent/high quality sentences.
- **Coherence**: Coherence measures the degree to which the source text's main ideas are meaningfully organized into different sentences

Due to the recent research interest in abstract ATS systems, there has been increased concern on the factuality and faithfulness of the generated summaries [86, 42]. **Faithfulness** is defined as the degree to which a summary contains only information contained in the main text, which is very similar to the *consistency* metric. **Factuality** is defined the degree to which the statements found in the summary are true; thus a summary may be factual but not faithful to the main text. Abstractive summaries tend to be unfaithful to the source text and there is active research interest in quantifying the percentage of valid summary "hallucinations" [86].

4.6 Summarizing Legal Texts

4.6.1 Domain Informed Preprocessing

Legal Texts - and in our case; court judgements - can be highly structured texts often divided in thematic segments. Typically, there exists an *introduction* and a *conclusion* segment which include information about the court's composition, the date and place the court's judgement were made, the names of the defendant/appellant, etc. Furthermore, there exists a *context* segment where information specific to the case are displayed as well the ruling of a lower court if that exists. In many cases this segment is composed by the arguments of each side. Finally, the court's judgement is analyzed in a *judgement* section.

Various preprocessing pipelines can be found in the literature. [43] normalize the text by replacing legal abbreviations with their full form. In [4] information from the header (introduction) section were removed as its information was considered irrelevant for the summarization task. Text-specific entities and dates in [109] are extracted using PoS tagging.

4.6.2 Further Difficulties

Legal texts exhibit various differences with typical texts used in training and evaluating ATS systems.

- Legal texts are have bigger length than texts from most common text summarization datasets, that very often come from the news-articles domain. This can significantly increase the computational cost of pre-processing the data, as well as training neural networks. Moreover, increased length may lead to the need of segmenting the input text creating separate segment-representations which fail to capture inter-segment relationships .
- The use of legal terminology is very common in legal texts, whereas common text summarization datasets contain little to no legal terminology. Therefore, pre-trained models on a non-legal domain may need significant fine-tuning in legal-domain texts to ensure good model performance.

4.6.3 Related Work

Feature-based methods

The starting point of applying modern NLP/Machine learning methods on summarizing legal texts was based on feature-based approaches[7]. The LetSum [36] software application segments the text into thematic segments corresponding to: Introduction, Context, Juridical Analysis and Conclusion, based on domain-specific linguistic features and relative position in the text. Afterwards, each sentence is scored according to the thematic segment it belongs using relative position information and tf-idf scores. Sentences

[7] Combining linguistic and frequential features as expanded in Section 4.4.1

are selected until a category is represented by a pre-specified percentage of extracted sentences.

In [39] the use of in-text citations in Federal Court of Australia cases is explored for both citations in the text and citations of the text. Each citation's importance is calculated using a centrality algorithm based on textual similarity.

The CaseSummarizer [109] software application uses an average tf-idf word score as a sentence score where each sentence's score is adjusted using domain-specific features such as the number of dates it contains, the number of named entity references it contains and the position of the sentence.

Graph-based methods

In [62] a sentence-graph is constructed where each directed edge weight corresponds to content embedding of the first node to the second, and key-sentences in each connected component are selected by their key-word strength, while complimentary sentences that explain facts, proofs, or rules following directed links to the key-sentence.

The performance of unsupervised extractive algorithms such as TextRank for plain-English summarization of contracts is explored in [84], where it is shown that extractive algorithms do not perform well because of the linguistic differences and abstractiveness differences between the reference summaries and the input legal text.

Machine Learning methods

Methods mentioned so far require significant amount of expert-level knowledge to determine the domain-specific features that are used in the algorithms. In [47], House of the Lord judgements were manually labeled with Rhetorical Structure features[8] and Relevance scores for each sentence. Afterwards, a Machine Learning model was trained on predicting Rhetorical structure features and relevance scores using automatically extracted lexical features.

Rhetorical Structure tagging is further explored in [123] where the rhetorical structure labels are predicted using *Conditional Random Fields* CRFs trained on linguistic and NER features. Each sentence is ranked according to a K-mixture model using tf, idf scores and sentences are extracted based on their rhetorical structure label and according to pre-specified percentages for each label.

In [147] a Naive Bayes Classifier is trained on labeled Canadian court cases data. The features used include: 1) relative position, 2) HTML emphasis tag features (found in the original text, 3) legal genre which is identified using lexical features and 4) tf-idf sentence scores.

[8] Rhetorical Structure theory is a computational theory of in-text discourse. It models the text's coherence by assigning tags to each textual segment. Each tag corresponds to the rhetorical role this segment (*nucleus*) plays in the text, and is further linked (*relation*) to other text segments (*satellites*). More information is available at https://www.sfu.ca/rst/01intro/intro.html

Deep Learning methods

Recent approaches to legal text summarization make the case of using *deep neural networks*, as they can generate rich representations of the data and, thus, capture semantic, syntactic textual information with greater precision than other methods.

[150] train a CNN classifier to predict the outcome of an appeal to the US Board of Veterans. Sentences that were found to be highly predictive of the case outcome were selected as candidate sentences. The sentences were further classified into a thematic segment, and one sentence from each segment was selected to be in the summary. They find that the predictive quality of a sentence concerning the case's output is not necessarily correlated with its informativeness in a summary.

In [4], an extractive summarization labeled dataset is created by labeling sentences in the text as sentences as extractable, if they are similar to the case's abstractive reference summary. Afterwards a CNN and an LSTM model are trained to classify the sentences.

The summarization problem in [145] is posed as a Legal Arguments Text-mining problem, where using a manually labeled dataset, a neural network is tasked with classifying each sentence to either: 1) *Legal Issue*, 2) *Conclusion*, 3) *Reason*, 4) *Non-IRC* - which is reserved for sentences that do not belong to either of the three aforementioned categories. In follow-up work [144], the same authors attempt to label the whole dataset using sentence representation similarity between sentences in the main the text and labeled sentences in the reference summary.

The *deep learning* methods mentioned so far correspond to extractive ATS models. In [37] apart from extractive neural models, the performance of deep neural networks in abstractive summarization of legal court cases is explored. They find the abstractive summaries generated by attentive LSTMs and Seq2Seq Transformer networks are similar in fluency with human-generated summaries. However in some cases,they may be completely irrelevant to the input text and mention nonfactual information. Similar results are reported in [81].

Chapter 5

Proposed Methods

In this chapter, we firstly describe the data collection protocol we followed for constructing our dataset. Then, we compare our dataset with other text summarization datasets on token-level statistics and evaluate the suitability of extractive summarization methods using *Extractive Fragment* analysis [46]. Finally, we propose several ATS methods for summarizing the court decisions in our dataset and define the corresponding pre/post processing pipelines.

5.1 Constructing a Dataset for Greek court decision summarization

5.1.1 Motivation

The digitization of rulings made by Greek courts is still in its growing stage, since there are no available APIs to serve the needs of law practitioners and scholars. Most Greek courts, in fact, do not offer any digital copy of their rulings, and those that do - doing so in the form of HTML pages, making querying and searching more difficult while also requiring significant text pre-processing.

However, law practitioners in Greece have to spend many hours every week to browse through many past court judgements, selecting those that are relevant to their case and reading them through. A dataset that can be made available through a REST API can reduce the time spent on searching for relevant judgements, and also be used to train ATS models that can generate summaries for judgements that do not come with one, which can also assist law practitioners in gaining in-depth understanding of a court-judgement faster.

We make the dataset collected for the purposes of our thesis available open-source[1], with the hope it encourages further research into the area of Legal Text Summarization for low-resource languages such as Greek.

[1] https://github.com/DominusTea/LegalSum-Dataset/releases/tag/v1.0.0

5.1.2 Data Crawler

For the aforementioned reasons, a web crawling script was developed using Python's SCRAPY[2] framework. Upon surveying the Greek courts' websites, two courts were found to have digitized part of their rulings making them applicable for our dataset; the Greek Court of Cassation[3] (Άρειος Πάγος), and the Greek Council of State[4] (Συμβούλιο της Επικρατείας). Two separate crawlers were developed, one for each separate court that offered some of its rulings in digital format. The data-crawlers collected the court's decision main text in both cases and the decision summaries for the AreiosPagos rulings, since the summaries where not available for the rulings of the Greek Council of State. Furthermore, court decision metadata; such as date, type of court, category tags, were collected or inferred when possible.

For the rest of our work, we used only the AreiosPagos dataset, since having reference summaries was crucial for evaluating our methods and training the methods that use Neural Networks. Furthermore, a large number of the older Council of States judgements contained transcription errors such as: missing words, missing sentences, randomly interjected escape characters. Fixing those errors when possible or automatically identifying and excluding erroneous documents was not feasible as about half or the documents would have to be omitted and significant amount of time would have to be spent into fixing the rest, therefore we leave it as future work.

5.1.3 Data Analysis

In this section we expand on the data exclusion protocol we followed for the dataset creation and discuss the results of the data analysis we conducted on the resulting dataset by comparing its token-level length statistics and extractive fragment coverage and density with other text summarization datasets.

Data Collection Protocol

The data crawler successfully scraped 7,607 samples of judicial decision & summary pairs, issued by the Areios Pagos court from 1990 to 2018. Along with the main text and summary, for each decision we collect additional metadata that correspond to the order number and type of court case, the year the decision was issued, the division of the court that issued it. The metadata are presented in Table 5.1.

We check for duplicate entries, which could be a result of either a bug in our scrapping software or the Areios Pagos' website, by ensuring that each dataset entry corresponds to a unique url. No duplicates were found fitting the previous criteria.

We conduct further duplicate-entries checks by implementing a string-matching deduplication process on each pair of decisions texts. In order to avoid checking lexical similarity between each pair of texts which would have computational complexity of $\mathbb{O}(n * (n-1))/2$, we check only pairs of decision that belong to the same case category.

[2] https://scrapy.org/

[3] http://www.areiospagos.gr/

[4] http://www.adjustice.gr/

AreiosPagos Court Judgements captured Metadata

Metadata	Data type	Inferred	Description
Case category	String		The general category that each court case is classified to by the Areios Pagos court's legal editors. Each case belongs to one category.
Case tags	String		The category tags that correspond to each court case, as classified to by the Areios Pagos court's legal editors. Each case may have multiple tags.
Court division	String	✓	The specific court division and its type (e.g Penal, Civil, etc.) of the Areios Pagos court that issued the court decision.
Issue Year	Integer	✓	The year that the court issued its decision.
Court's case identifier	String	✓	The identifier given by the court for the particular case. It is unique among cases judged by the same court division, but not across them.
Source URL	String	✓	The URL to the original Areios Pagos HTML webpage from which the text, summary and metadata were sourced.

Table 5.1. *The metadata collected for our AreiosPagos court judgements dataset. The metadata that were automatically inferred by us using the judgements main text are labeled with a ✓in the Inferred column.*

In order to further reduce the computational cost, for each decision we check only the 20 closest to it chronologically. The document similarity is calculated via the *fuzzywuzzy*[5] fuzzy string matching python library.

The result of this analysis, points to many categories having some documents with duplicate texts. We manually investigate each category and remove the duplicate entries complying to the following deletion criteria:

- Both text and summary are duplicates. Differences in dates, names of the judges, lawyers and appellants are ignored.

- Almost duplicate texts with very different summaries are not deleted. Upon manual inspection, those texts had several sentences that led to the case summary being very different. Therefore, we consider them "hard cases" and keep them in the dataset.

This process removed 1,208 samples from our dataset. We further remove samples that have null summary (-29 samples). The final dataset contains 6,370 samples. A general schema of the dataset creation protocol we followed can be seen in Figure 5.1.

[5] https://github.com/seatgeek/fuzzywuzzy

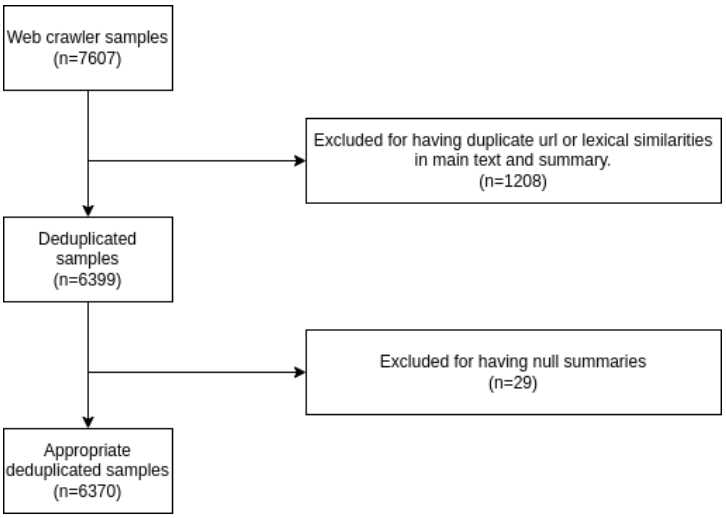


Figure 5.1. *A schema of the dataset creation protocol we followed. n refers to the number of samples that remain/are excluded at each step.*

Data Exploration

In this section we provide information about basic token-level statistics of our dataset and compare it will other text summarization datasets.

Our data is organized into 504 unique categories. The data are *over-dispersed* over the categories labels as the average category frequency is 0.198% with standard deviation equal to 0.5549. Furthermore, the category labels correspond to quite different with each other court cases, indicating high diversity in our dataset. The 10 most frequent case category labels are highlighted in Figure 5.2.

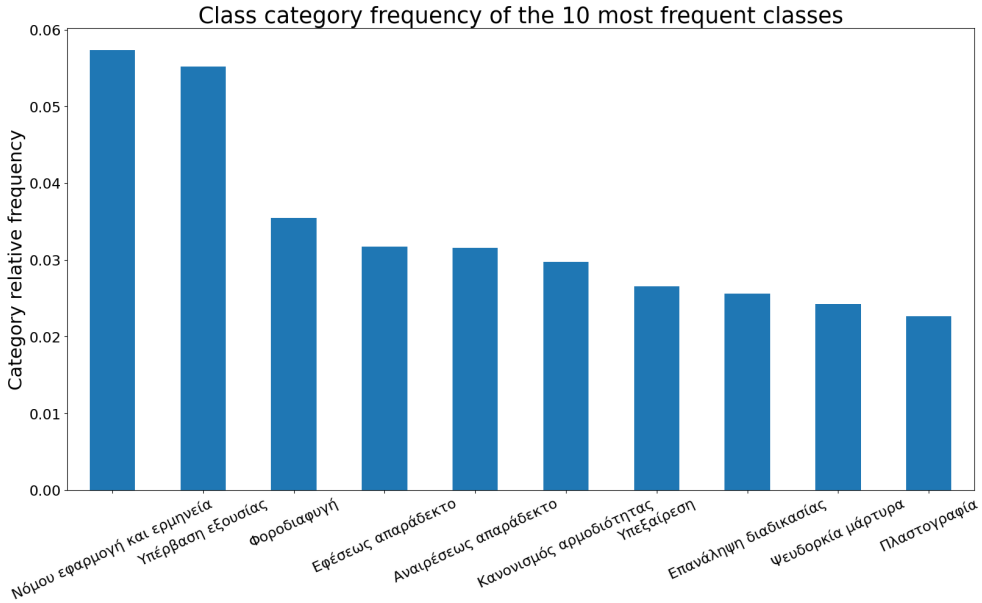


Figure 5.2. *The 10 most frequent categories in our AreiosPagos dataset. The x-axis corresponds to the category labels. The y-axis corresponds to the absolute frequency of each category.*

We further explore lexical properties of our dataset by calculating statistics concern-

Dataset Comparison on token-level length statistics

Statistical Property	AreiosPagos	Rechtspraak	Newsroom	CNN-DailyMail
#Documents	6,399	403,585	1,321,995	311,672
Avg.tokens/doc	2,398.6	2,341.5	658.6	766.0
Avg.sent/doc	51.49	140.6	-	29.74
Avg.tokens/sent	46.5	16.6	-	25.75
#Summaries	6,370	403,585	1,321,995	311,672
Avg.tokens/sum	88.1	62.1	26.7	25.75
Avg.sent/sum	5.36	3.41	-	3.72
Avg.tokens/sent	17.8	-	-	-

Table 5.2. *Statistical properties of text summarization dataset using average ratios of token-level lengths for document and summaries and their sentences. AreiosPagos and Rechtspraak are legal court-cases text datasets, while Newsroom and CNN-DailyMail are news-domain summarization datasets. Results on the Newsroom dataset are reported from [46]. Results on the CNN-DailyMail, Rechtspraak datasets are reported from [81]. The upper part of the table, presents statistics on the judgements' main texts , while the lower part presents the statistics on the judgement summaries.*

ing the average length in tokens, the average number of sentences and the average token in every sentence. A comparison of our dataset with other text summarization datasets is shown in Table 5.2. Our court-decisions dataset contains longer document and summaries, both in terms of tokens and in terms of sentences. Furthermore, sentences in the main texts are also significantly longer than sentences in news-domain datasets.

In order to analyze the similarities between each court-decision text and its corresponding summary, we replicate the *Extractive Fragment* analysis found in [46] and compare our dataset with news-domain datasets. Each reference summary is divided into segments such that each segment corresponds to the longest possible segment of consecutive words found both in the reference summary and the main text.

Let T, S be a text, summary pair and $\mathcal{F}(A,)$ be the set of the corresponding extractive fragments. The *Extractive Fragment Coverage* measures the percentage of words in the summary that are also extractive fragments; that is they can also be found in the main text.

$$\mathrm{C} = \frac{1}{|S|} \sum_{f \in \mathcal{F}(A,S)} |f| \tag{5.1}$$

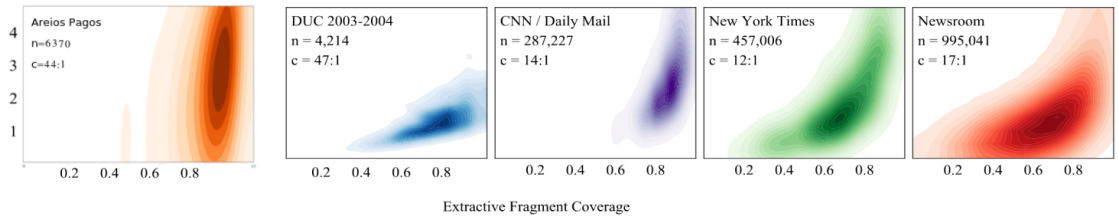
In order to measure how extractible is a summary from a text, the *Extractive Fragment Density* metric is defined which attributes higher density scores to texts that have longer extractive fragments:

$$\mathrm{D} = \frac{1}{|S|} \sum_{f \in \mathcal{F}(A,S)} |f|^2 \tag{5.2}$$

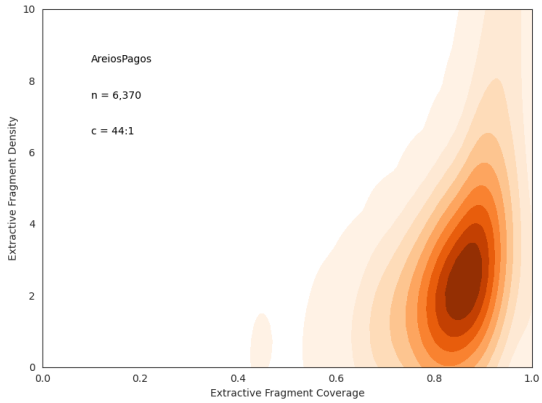
Our dataset's results compared to other datasets are illustrated in Figure 5.3. Our dataset is rather homogeneous in terms of coverage, as over 80% of the words in the reference summary can be found in the text. In terms of density, our dataset is most similar to the CNN/Daily Mail dataset, having high density score which means that the reference summaries can be modeled by a fewer amounts of extractions than the extractions needed

for other datasets. We also find considerable variance in the *density* axis, indicating that some judgements may be summarized by less sentence extractions than others. Overall, the aforementioned remarks imply that extractive summarization methods may generate useful summaries, provided lexical overlap is a good indicator of a candidate summary's quality.

However, our dataset exhibits high compression scores, as the average length ratio of text to summary is 44; which is 3-4 times bigger than the compression score of news-domain datasets. This indicates that summarizing court judgement texts can be more computationally expensive compared to news-domain datasets, if the whole text is used as input. Furthermore, pre-trained neural-network methods that have a fixed input size constraint, may need to be retrained with a bigger input size cap, or used as they are but with their inputs truncated or condensed.



(a) *A comparison of the Extractive Fragment Coverage-Extractive Fragment Density relationship for the AreiosPagos dataset compared to other text summarization datasets. Leftmost: the AreiosPagos dataset (ours). Right: News-domain datasets as reported in [46]. The AreiosPagos dataset is homogeneous Coverage as most words in the summary appear in the main text. In terms of Extractive Fragment Density, the AreiosPagos dataset shows more variance than the other datasets, while having generally high scores.*



(b) *The full range of the kernel density estimate plot of the AreiosPagos dataset*

Figure 5.3. *Plots of the Extractive Fragment Density/Coverage for various summarization datasets. Data observations are plotted using a kernel density estimate method. n denotes the number of documents in the dataset, and c refers to the compression ratio of the main text's length over the summary's length.*

5.2 Proposed Automatic Summarization Methods

In this section we describe the models we propose for tackling the problem of summarizing Greek judicial decisions.

5.2.1 Extractive Summarization Models

LexRank

We propose and evaluate several versions of the LexRank [34] extractive summarization algorithm. More specifically, we evaluate different similarity functions used to measure cross-sentence similarity.[6] Let s_1, s_2 be two sentences. We evaluate the following LexRank similarity function & sentence representation combinations:

- BoW with Common Words sentence similarity: Cross-sentence similarity is defined as the number of words found in both sentences normalized by the sum of both sentences' lengths. Formally:

$$sim_{cw}(s_1, s_2) = \frac{|s_1 \cap s_2|}{log(|s_1|) + log(|s_2|)}$$

  This essentially implements the similarity function used in the TextRank algorithm [89], aside from some minor changes in the parameterization of the power method used to converge to the LexRank sentence scores.

- Tf-idf BoW with cosine sentence similarity: This is the cross-sentence similarity metric used in the LexRank [34] paper, in which, the cross-sentence similarity is given by the cosine distance of the tf-idf BoW sentence vectors. Formally,

$$sim_{cos\_tfidf}(s_1, s_2) = 1 - \frac{\textbf{Tf-Idf}(s_1) \cdot \textbf{Tf-Idf}(s_2)}{\|\textbf{Tf-Idf}(s_1)\| \times \|\textbf{Tf-Idf}(s_2)\|}$$

  where the **Tf-Idf** vectors are calculated in a BoW fashion, by the sum of the one-hot vectors of each word in the sentence weighted by the word's idf score:

$$\text{Tf-Idf}(s) = \sum_{w \in s} 1_{index(w)} idf(w)$$

  where index(x) $\in [1, N_{vocab}]$ and $1_i \in [0, 1]^{N_{vocab}}$ denotes the indicator function and is non zero only at the position of the index i.

- Idf modified BoW Word2Vec with cosine sentence similarity : We construct the sentence vector by averaging its words unigram Word2Vec vectors (which are precalculated using the *Gensim* [115] library). The cross sentence similarity is calculate using the cosine distance vector metric. Formally:

$$sim_{w2v}(s_1, s_2) = 1 - \frac{\textbf{W2V}(s_1) \cdot \textbf{W2V}(s_2)}{\|\textbf{W2V}(s_1)\| \times \|\textbf{W2V}(s_2)\|}$$

[6] We modify and extend the LexRank implementation [82] found on *Github*, by adding more sentence similarity metrics and domain-specific preprocessing steps that we will elaborate on in section 5.3.

where $\mathbf{W2V}(s) = \sum_{w \in s} \mathbf{word2vec}(w)/|s|$ and each w2v word vector is scaled by the word's idf value. We use the train+validation subsets of the AreiosPagos dataset and the full Council of State dataset for training the Word2Vec representations[7].

Biased LexRank

We implement the biased LexRank algorithm [101] by extending our default LexRank implementation (based on [82]). As mentioned in section 4.4.1, the biased version of the algorithm changes the way the damping factor is distributed to each sentence, from attributing it uniformly to biasing it according to a prior belief on the importance of each sentence. Here, this is achieved by utilizing the semantic similarity of each sentence with the judgement's tags, as described in 5.1.2. The semantic similarity of each sentence with the judgement tags is calculated using the common words sentence similarity function.

5.2.2 Abstractive Summarization Models

We implement a standard Encoder-Decoder framework for abstractive summarization based on the model proposed in [77]. Both the Encoder and the Decoder are multi-layer bidirectional Transformer models. The Encoder is initialized using the Greek BERT's [69] weights. Part of the Greek BERT's training data was the Greek part of the European Parliament Proceedings Parallel Corpus, a domain which can be similar with our legal judgements domain. The model was implemented using *Huggingface's* transformer library [143].

5.3 Proposed Preprocessing & Post-processing Pipelines

5.3.1 The Preprocessing pipeline

Extractive Summarization Preprocessing

The extractive summarization models we have described in section 5.2.1, require each input text to be segmented to its sentences and each sentence to be tokenized into separate words. Additionally, due to the Greek language's additional complexity (e.g word declension) further pre-processing steps, such as stop-word removal and word lemmatization, must be made in order to avoid a rapid increase of the vocabulary size. Specifically, we utilize the spacy library [53, 28] to implement the following preprocessing pipeline:

- The sentences are separated from each other, paying special attention to domain-specific acronyms that could potentially lead a punctuation sentence segmentor to end a sentence prematurely.
- Each sentence is tokenized into separate words using spacy's dependency parser.
- Stop-words are removed and tokens are lemmatized.

[7] Including the Greek Council of State may have been a poor choice, as upon later inspection it became evident that a large part of its document contained text of poor quality. For more information see Section 5.1.2.

- If token vectors are required by the algorithm, the pipeline returns the vectors found in the spacy language module used.

Abstractive Summarization Preprocessing

The abstractive summarization model is not compatible with the pipeline we described above. This is due to several reasons:

- Since the pretrained greek-BERT representations are utilized, we are not able to use our pipeline's tokenizer since we would have to ensure an exact match between the tokens the model was trained with and the models produced by our pipeline's tokenizer. In addition, the tokens greek-BERT was trained on are produced by a *subword* tokenizer, which splits words not found in its vocabulary to known word parts found in the vocabulary (such as prefixes and suffixes).

- The model has a practically fixed vocabulary size, since increasing it would result into a large increase in computational complexity during training & inference. Therefore, learning BERT representations for new tokens would not be cost-efficient.

Consequently, we utilize greek-BERT's default subword tokenizer which:

- Splits the input, both judgement & judgement summary, into tokens. When a word is not found in the tokenizer's vocabulary, then it is split into known subwords.

- Input is truncated and/or padded to accommodate the model's hidden size constraint of 512 tokens.

- Each token is encoded into the corresponding token id, an integer ranging from 1 to N_{vocab}.

- The input is moved to the GPU (if it is available) using PyTorch's [105] CUDA API.

- We experiment with text reordering; moving the last part of the text which contains the court's decision, to the start of each input string. This way the court's decision will never be truncated due to the model's 512 maximum input size.

- We further experiment with removing duplicate whitespace tokens, as well as the beginning and the ending of each text which correspond to general information about the date of the trial, the location it took place, the names of the judges, the appellants and the lawyers. The aim is to fit much important data as possible to the 512-tokens input.

- We experiment with including the category tags, that correspond with each court case, at the start of every input. The main text is separated by the tags using a special separator token "[SEP]".

5.3.2 The Post-processing pipeline

In the case of extractive summarization, the sentences extracted from the text are concatenated, in the order they appear in the input text, to form the generated summary. An exact match between our generated extractive summaries and the reference summaries is not possible, since the latter are abstractive in nature. To ensure a fair comparison we constrain our generated extractive summaries to be at three times the length of our reference summaries.

In the case of abstractive summarization, the model output token ids that are decoded into words using the model's tokenizer. Furthermore, special tokens which the tokenizer adds by default are omitted from the final summary.

Chapter 6

Experimental Results

In this chapter, we analyze the results of the automatic evaluation of our methods using the ROUGE metrics. In addition to automatic evaluation, we also provide the results of human evaluation of the generated summaries by legal-experts, measure the correlation of the human evaluation scores with the automatic ROUGE metrics and evaluate the inter-evaluator agreement both on their evaluation scores and their human-generated summaries.

6.1 Automated Evaluation

6.1.1 Motivation

Manual evaluation of court judgement summaries is a time-consuming process that requires from the evaluators to be legal-experts and be focused enough to apply their specialized knowledge in reading and understanding the court judgement's main text and, afterwards, evaluating summaries based on the text they read. Therefore, in order to evaluate the whole test subset of our dataset, we resort to *automatic evaluation*. We choose to use the ROUGE lexical-overlap metrics, as is the norm in most ATS research literature.

6.1.2 Automated Evaluation Pipeline

The evaluation of the generated summaries is conducted using the ROUGE [73] automatic metric. We modify a python re-implementation[1] of the original Perl ROUGE script, by adding options for stemming greek words, and improving the tokenization and sentence segmentation on our dataset. We insert options of removing greek stopwords and/or stemming every word. Figure 6.1 presents a schema of the automatic evaluation pipeline.

[1] https://github.com/Diego999/py-rouge

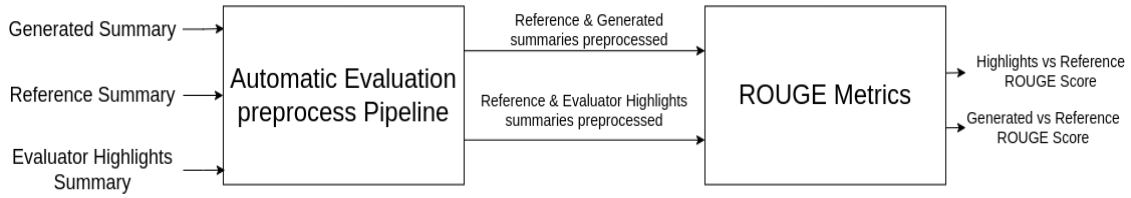


Figure 6.1. *A schema of the automatic evaluation pipeline. Both reference, evaluator highlight-summaries and generated summaries are preprocessed. The ROUGE metrics are used to compare generated and evaluator-highlight summaries versus the reference summaries.*

6.1.3 Automated Evaluation Results

The results of our methods using the ROUGE automatic evaluation metric, after stemming and stopword removal, are shown in Table 6.1. The extractive summarization methods extracted the most important sentences until three times the length of the reference summary was reached. The abstractive summarization methods generated summaries of arbitrary length, as they have learned when to end a summary during the training phase.

In the extractive LexRank methods, we tested different cross-sentence similarity metrics finding the tf-idf score to be best. We attribute the Word2Vec sentence representations poor results, to the bad quality of the Council of State dataset texts used for training the Word2Vec representations (see Section 5.1.2 for more information).

The biased LexRank method, was found to be worse than regular LexRank pointing to the category tags being less relevant for sentence extraction. However, biased LexRank outperformed the random sentences baseline.

For abstractive methods, we tested different prompt engineering methods, with the aim of including as much relevant information to the 512 tokens of the input. We check the following methods which were described in Section 5.3.1: 1) Rearranging the input text so that the court case's result is always included in the start of the text and is never truncated due to the maximum input size limitation, 2) Removing general text that is irrelevant to the case, such as date, name of appellants and judges, etc., 3) Including the case's category tags in the input, 4) Halving the input document using our LexRank algorithm.

We find that text reordering and removal offer slight increase in performance. Including the category tags resulted in greatly increased performance, which can be attributed to the tags offering contextual information for the rest of the input text. Furthermore, halving the input text using LexRank decreases the model's performance. This may be explained by the halved text becoming very different from the texts that were used to pre-train the BERT model, as removing half of the document can significantly alter its coherence and also remove important information.

Method	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-L	ROUGE-W
$\text{LexRank}_{\text{tf-idf}}$	71.46	**42.90**	**23.78**	**17.29**	**8.35**
$\text{LexRank}_{\text{com}}$	**71.51**	42.10	22.02	15.09	7.09
$\text{LexRank}_{\text{w2v}}$	69.63	39.63	19.69	12.93	6.05
Biased Lexrank	67.73	41.05	22.06	15.50	7.33
Random Sentence	70.64	40.26	19.77	13.41	6.28
BERT	62.90	38.52	20.64	14.37	5.28
BERT(RE)	62.80	38.51	20.39	14.19	5.21
BERT(RE+RM)	62.08	38.83	21.26	14.42	5.28
BERT(RE+RM+C)	**64.24**	**40.40**	**22.27**	**15.34**	**5.64**
BERT(RE+RM+LR)	62.01	37.89	20.32	13.71	4.99
BERT(RE+RM+LR+C)	63.98	39.85	21.90	15.33	**5.64**

Table 6.1. *Automatic evaluation results presented in two segments of the table corresponding to a) Automatic extractive summarizers, b) automatic abstractive summarizers. The extractive methods extract sentences until they reach three times the length of reference summary. In the abstractive models we modify the inputs and label the models accordingly; RE:the text is rearranged so the case result is always included and at the start of input, RM: unnecessary parts of the text are removed, C: the case's category tags are included at the start of the input, LR: the input document if halved using* $\text{LexRank}_{\text{tf-idf}}$*. The ROUGE scores are F1 scores given in percentages (%) form. The ROUGE-L/W scores are reported without stopword removal for the BERT methods. The best performing automatic method in each category is in* ***bold****.*

6.2 Human Evaluation

6.2.1 Motivation

The Automatic Evaluation metrics we used in Section 6.1 allow for fast evaluation of large number of summaries, without any human supervision. However, as those metrics factor in only lexical overlaps, their scores may not necessarily be indicative of a summary's quality, as we have seen in Section 4.5.3. In order to get a more precise evaluation of our methods' performance, as well as study the correlation between the automatic metrics and human judgement, we need to carry out a human evaluation study. The human evaluation study was conducted through our web application interface which will be further explained in Section 7.4.

6.2.2 Human Evaluation Pipeline

Human evaluation of court judgement summaries in our dataset is a challenging task, as in order to ensure high quality standards in evaluation, we have to limit our human evaluator pool exclusively to people with at least undergraduate level experience in the legal domain. Furthermore, our human evaluators have to devote significant amount of time in reading the court judgement texts and evaluating their summaries, actively using their legal domain knowledge and abilities. Therefore, only six (6) human evaluators actually responded to our study's call for participants. Details about our respondents'

age and legal domain knowledge can be found in Section 6.2.3. We leave as feature work the replication of our study's results using a bigger sample of legal experts.

In order to ensure the survey's estimated completion time was reasonable, we selected court judgements that had length less than the average judgement length in the test dataset. The resulting average completion time was 42.5 minutes (std: 3.30).

Specifically, the evaluators were instructed to:

1. Read the court's judgement text and highlight the sentences they believe are important in creating a summary.

2. Evaluate abstractive summaries of the judgement.

3. Evaluate extractive summaries of the judgement.

For the evaluation metrics, we modify the metrics introduced in [70] to our court decisions domain:

- **Relevance:** The degree to which a summary has captured the important content from the judicial decision.

- **Consistency:** The factual alignment between the summary and the judicial decision's main text.

- **Fluency:** The degree to which the summary contains individually fluent/high quality sentences.

- **Coherence:** Coherence measures the degree to which the main ideas of the judicial decision summary are meaningfully organized into different sentences

In the case of extractive summaries, the evaluators are asked to evaluate the summaries only on the *Relevance* metric, since the others are not applicable to extractive summaries. The abstractive summaries generated by the BERT model are in lower case form and contain no diacritics. Therefore, in order to avoid biasing the human evaluators, we lowercase and remove the diacritics from the reference summaries as-well.

We construct extractive summaries using the human evaluators' highlights from each text. The human evaluators' highlights are used to extract from each text the corresponding segments and construct extractive summaries, which after truncated to match the length of the extractive summaries generated by our Extractive Summarization methods, can be directly compared them.

All evaluators are assigned the same five (5) court judgements and their human evaluation scores are analyzed for inter-evaluator agreement and their correlation with ROUGE metrics is measured. Figure 6.2 provides an outline of the human evaluation pipeline.

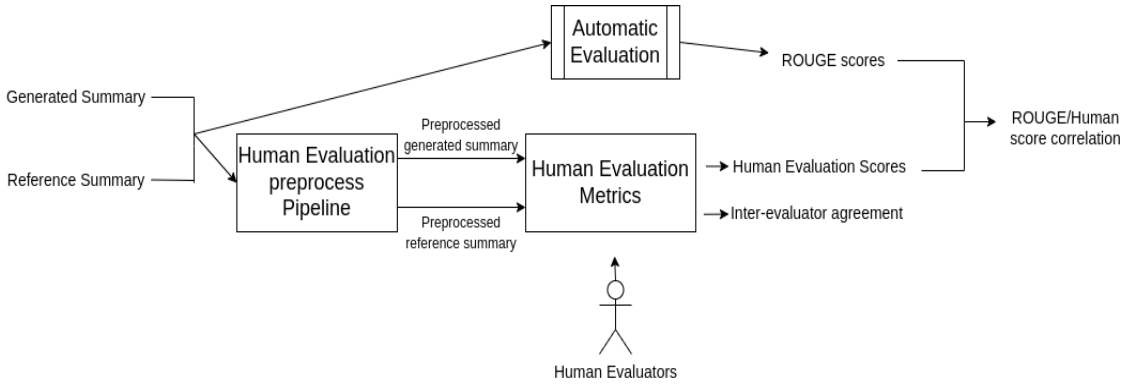


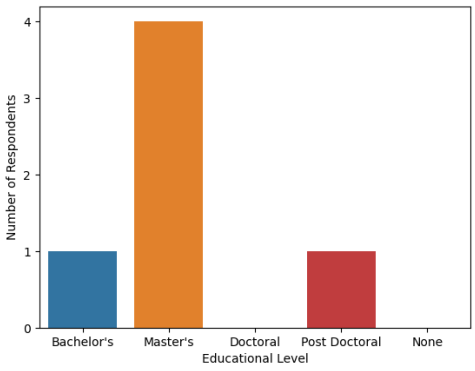
Figure 6.2. *A schema of the human evaluation pipeline. Reference summaries are preprocessed to match the BERT generated summaries. The human evaluation metrics are used to measure inter*

6.2.3 Study participant information

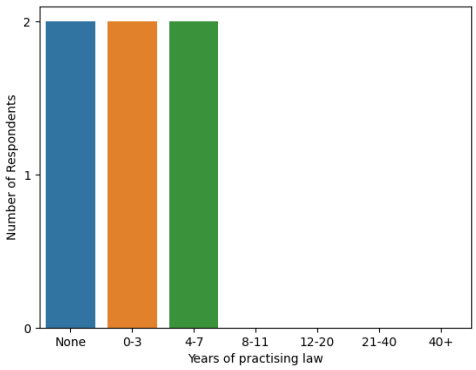
We first present our study's participants information about their legal-domain education knowledge, familiarity with practising law and reading court judgements as well as their usefulness estimation for a Court Judgements ATS system.

Our evaluator sample appears to have very good knowledge of the legal-domain, as most of them are currently enrolled or have obtained a Master's degree in a law-related domain (Figure 6.3a). Furthermore, our study participants have good familiarity with the law-domain as 4/6 have practised law with 2/6 practising law for over 4 years (Table 6.3b).

Reading court judgements is a familiar task for most of our study's participants, as shown in Figure 6.4a. All, except one of the participants spend over 2 hours weekly reading court judgements. One participant even spends 8-16 hours weekly. This supports our opinion that a good performing ATS system for court judgements would be beneficial for law practitioners. This position is further supported by our study participants' own usefulness estimation of such app if it was to be created (Figure 6.4b). All but one of the legal-experts that participated in our study rate the usefulness of such an application as "8-10" in the Likert scale [72].

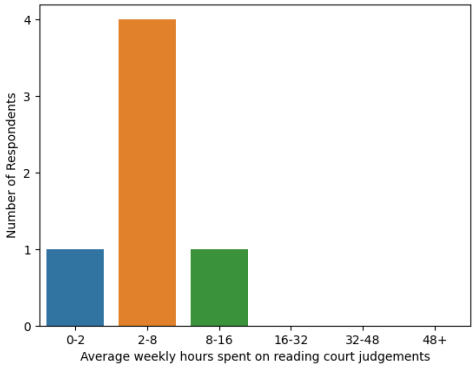


(a) *Educational level of each participant in our study. Y-axis corresponds to number of participants having a particular education level.*

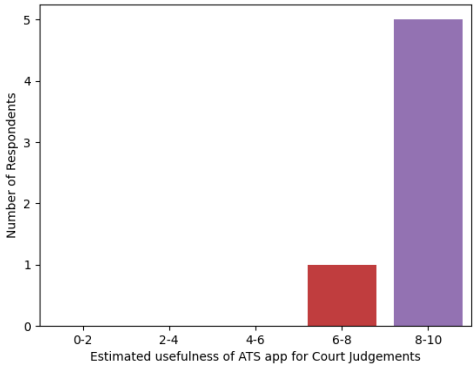


(b) *Years of practising law of each participant in our study. Y-axis corresponds to number of participants belonging to a particular practising years bin.*

Figure 6.3. *Our study participants' familiarity with the law-domain by: (a) their law-domain educational level, (b) years of practising law.*



(a) *Average weekly hours spent on reading court judgements by our study's participants. Y-axis corresponds to number of participants.*



(b) *Estimated usefulness, by our study's participants, of an ATS application for court judgements.*

Figure 6.4. *Answers by our study's participants corresponding to (a) average weekly hours spent on reading court judgements, (b) estimated usefulness of an court judgements ATS application.*

6.2.4 Human evaluation metric results

In order to evaluate our methods' performance, we ask the human evaluators to evaluate our abstractive summaries, and the reference summaries in terms of *Relevance*, *Consistency*, *Coherence* and *Fluency*. The evaluators also evaluate our LexRank and Biased LexRank extractive summaries in terms of *Relevance*. The aggregate results are presented in Table 6.2

Summary	Relevance	Fluency	Coherence	Consistency
Reference	3.9	3.7	3.7	3.7
BERT(generated)	1.9	3.1	3.3	1.8
$\text{Lexrank}_{\text{tf-idf}}$	2.9	-	-	-
Biased LexRank	3.0	-	-	-

Table 6.2. *Human evaluation results, on the modified human evaluation metrics using a 1-5 Likert scale. The first section of the table compares our BERT abstractive summarization method with reference summaries. The second section of the table compares human evaluated Relevance score of the summaries generated by the vanilla LexRank and the Biased LexRank algorithms.*

We note that in terms of fluency and coherence, our *abstractive* model has similar but lower performance to the reference summaries. That indicates that the generated text can be read easily and is internally coherent and similar, in those regards, to the reference summaries. However, our model under-performs compared to the reference summaries, in terms of relevance and consistency. This means that compared to a reference summary, it fails to capture the relevant information from the judicial judgement and also may be factual inconsistent with it by referencing information not found in the original text. Reference summaries appear to be much better at capturing the relevant context of the court decision and are more factually consistent with it.

It is important to note that the reference summaries have surprisingly mediocre scores in the fluency and coherence metrics, and what probably enabled the model to have comparable scores in those metrics is the pre-training phase. Furthermore, the relevance and consistency scores of reference summaries are above average but not perfect, indicating the need for better curated datasets and standardized practises in manual summary writing in the Greek court judgements domain.

The extractive summaries performed relatively well, as the relevance score is close to average and to the *abstractive* reference summary's relevance score. However, a straightforward comparison between those methods is not sensible as *extractive* summaries are very different from *abstractive* summaries. The vanilla LexRank with tf-idf similarity function and the biased LexRank had no statistical important difference in terms of Relevance score.

6.2.5 Correlation of human evaluation & automatic metrics

In order to asses the performance of the ROUGE automatic metrics in evaluating summaries, we measure the Pearson correlation of each ROUGE metric score with each human evaluation metric score (Table 6.3). This analysis can serve as a way of finding which ROUGE metrics can substitute which human evaluation metrics when the latter are not easily available. We average the fluency and coherence metrics creating a metric for *internal* readability. Similarly, we average the relevance and the consistency metrics creating a metric for *external* summary factuality and relevance.

Metrics	Relevance	Fluency	Coherence	Consistency	Internal	External	Average
ROUGE-1	0.0390	0.2869	**0.3050**	0.1560	0.3167	0.0906	0.2037
ROUGE-2	**0.2786**	**0.7645**	0.2653	**0.3133**	**0.4264**	**0.2953**	**0.3609**
ROUGE-3	0.0622	0.6393	-0.0300	-0.0280	0.1686	0.0475	0.1080
ROUGE-L	0.0159	0.6644	0.2569	-0.0880	0.3933	0.0476	0.2204
ROUGE-W	-0.2172	0.6508	0.1037	-0.1782	0.2733	-0.2012	0.03605

Table 6.3. *Pearson's correlation of human evaluation scores and ROUGE metrics F1-scores. For each human evaluation metric, the most correlated automatic metric is highlighted in* ***bold*** *while the less correlated is underlined.*

We find that the internal readability metrics; *Fluency* and *Coherence* are moderately correlated with ROUGE metrics. Specifically, *Fluency* is positively correlated highly with all ROUGE metrics, with the highest correlation being with the ROUGE-2 metric and the ROUGE-L metric. *Coherence* has moderate positive correlation with ROUGE-2. This is expected, as those metrics measure large lexical overlap with large (common) sequences of words, and thus a summary that scores high on those metrics is expected to have fluent and coherent sentences.

The external metrics show less correlation with the ROUGE-metrics, which is expected as *Relevance* and *Consistency* are not properties of a summary that can be sufficiently measured by lexical overlaps. In both cases, the ROUGE-2 metric seems to correlate higher. However, we note the need of developing new metrics for court judgement text summarization that correlate better with human judgement in terms of the text's relevance and consistency.

We note the existence of metrics, such as ROUGE-3 and ROUGE-W seem to offer little in terms of human evaluation prediction capacity as they show very small positive or even negative correlation with human evaluation.

6.2.6 Human Evaluators Highlights Analysis

In order to further assess our extractive methods, we analyze the highlights the human evaluators extracted from each text. In table 6.4 we present the automatic evaluation scores of both the original highlight summaries and the highlight summaries truncated to three times the length of the reference summary, similarly to the extractive summaries that were generated by our methods for the automatic evaluation in Section 6.1.

We note that the evaluators summaries are, in average, 6.4 times the size of the reference summary which is significantly larger than the size constraint of 3.0 times the reference summary we set for our extractive summarizers. This indicates that legal experts prefer longer extractive summaries.

In terms of ROUGE score, the evaluators' highlights summaries score higher than our extractive summarization methods. Considering the mediocre human metric scores of our extractive methods, the ROUGE scores seem able to capture the quality of an extractive summary as they assign the higher score to a legal-expert summary and to a automatically generated extractive summary that legal experts rate as mediocre.

Considering the mediocre human metric scores of our extractive methods, and the

fact that the legal-expert generated extractive summaries score higher in terms of ROUGE metrics, than automatically generated extractive summaries, we can conclude that the ROUGE metric can be useful in assessing an extractive summary's quality. However, we note that when ROUGE metric scores are close, as is the case for LexRank and Biased LexRank, the ROUGE metrics may not align with human judgement. In our human evaluation survey, the extractive methods have similar *Relevance* scores (Table 6.2), while having small but noticeable differences in terms of ROUGE score.

We also compare, using the ROUGE metrics, the human-generated highlight summaries with the automatically extracted summaries, considering the first to be reference extractive summaries (Table 6.5). We find that the vanilla LexRank method clearly outperforms the Biased LexRank method. However, taking into consideration that the legal experts assign similar relevance score to those methods (Table 6.2), we note that small or even moderate differences in terms of ROUGE score do not necessarily translate to differences in terms of human judgement. This may be explained by the fact that extractive summarization is an under-constrained task and extractive summaries which show great lexical overlap with a reference summary, thus having high ROUGE scores, may not be the only type of summaries that perform well in terms of human judgement.

Human/Auto Summaries	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-L	ROUGE-W	sum/doc	sum/ref
Eval.Highlights	64.56	40.72	23.21	13.93	6.56	0.170	6.43
Eval.Highlights(capped)	69.44	42.17	24.09	14.93	7.14	0.088	2.54
$\text{Lexrank}_{\text{tf-idf}}$	61.45	36.90	19.52	12.15	6.41	0.079	3.00
Biased LexRank	55.57	35.41	18.68	14.58	7.63	0.079	3.00

Table 6.4. *Average length statistics and ROUGE F1-scores of the extractive summaries generated using the human evaluators' highlights and the automatically generated extractive summaries versus the reference abstractive summaries. The second row represents the evaluators' highlights summaries truncated to three times the length of the reference summary, matching the extractive summaries in Table 6.1. The last two columns present the token-level length statistics of the summaries compared to the court's judgement main text and reference summary respectively.*

Auto Summaries (vs Evaluator highlights)	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-L	ROUGE-W
$\text{Lexrank}_{\text{tf-idf}}$	80.24	46.66	24.86	16.14	7.26
Biased LexRank	74.23	42.49	21.81	16.84	8.00

Table 6.5. *ROUGE metric comparison of automatic extractive summarization methods using the human evalutors' highlights summaries as reference. We report the average ROUGE-F1 score, over all evaluators and all court judgement summaries in our human evaluation study.*

Summary Type	Relevance	Fluency	Coherence	Consistency	Internal	External	Average
Abstractive	0.6405	-0.0215	0.0709	0.6400	0.0260	0.6754	0.4332
Extractive	0.4250	-	-	-	-	-	-

Table 6.6. *Krippendorff's alpha agreement metric on each human evaluation metric for each summary type. The Internal metric is the average of Fluency and Coherence metrics. The External metric is the average of Relevance and Consistency metrics. In the abstractive summaries category, we include both reference and generated abstractive summaries as human evaluators were evaluated both in the same way and in a randomized order without knowing if any of the summaries was written by legal experts.*

6.2.7 Human Evaluators Agreement

In Table 6.6, we measure the inter-evaluator(/annotator) agreement of human metrics using the Krippendorff's alpha metric for interval variables[2]. Measuring the agreement can be helpful as a way of quantifying which human metrics are well-defined and, thus, human evaluators give similar scores. It can also be used to find human metrics which are ambiguously defined or naturally more subjective, and thus inter-evaluator agreement is low. Those results may be used to inform our interpretation of human evaluator metric results downstream. Furthermore, low inter-evaluator agreement results can lead to more thorough metric definitions for the low-agreement metrics.

We find that the human evaluators systematically agree on the *external* metrics: *Relevance* and *Consistency*. Their evaluations seem to be more unreliable in terms of the *internal* metrics, especially *fluency*. These findings are similar to [35] and indicate that the task of the task of evaluating the summary's inclusion or not of all the relevant information from the main text as well as its factual consistency with it is more objective than the task of evaluating a summary's fluency and inner coherence, which can be more subjective due to differences in personal reading/writing style.

In order to measure the agreement on the highlights each evaluator had extracted from the court's decision text, we calculate for each text the average pairwise highlight agreement between each pair of evaluators. Let N_{evals} be the number of evaluators and $H^{(i)}$ the set of the N_{evals} highlight sets collected for text i.

$$H^{(i)}_{\text{avg}} = \frac{1}{N_{\text{evals}}(N_{\text{evals}}-1)} \sum_{H^{(i)}} \sum_{H^{(j)} \neq H^{(i)}} ||H^{(i)} \cap H^{(j)}|| / ||H^{(i)} \cup H^{(j)}|| \tag{6.1}$$

The results in Table 6.7 show large differences of highlighting style between the evaluators. Furthermore, there is large variance in the highlights agreement between each question, which may be attributed to the different highlighting style of each evaluator and also qualitative differences in the texts. This result further supports the position that extractive summarization is an under-constrained task, as each evaluator has a different approach in generating an extractive summary. This remark however, as we saw in Table 6.6, does not imply that human evaluation of extractive summaries is under-constrained, as manually generating an extractive summary is quite different from evaluating an automatically constructed one.

q1	q2	q3	q4	q5	average
0.2619	0.0908	0.1531	0.2556	0.3957	0.2314± 0.141

Table 6.7. *Average pairwise highlight agreement on each question over all human evaluators. The pairwise agreement is calculated as the ratio between the intersection and the union of the two sets of highlights.*

[2]We use the implementation in https://pypi.org/project/krippendorff/

Chapter 7

Web Application

7.1 Introduction

For the purposes of disseminating the data collected for our court judgement dataset and conducting a human evaluation study on automatic summarization methods for court judgements, we developed a web application using the **JHipster** framework[1]. Our application has built-in support for automatic building, testing and deployment.

In the following sections, we first describe the technology stack used for developing our application, both client-side and server-side. Then, we outline the entities in our database schema and explain the RESTful API developed for the dissemination of our dataset's documents. Finally, we expand on our application's *Survey Page*, by explaining in detail every page of the survey and giving a general outline of the whole process.

7.2 Technology Stack

In this section, we describe the technology stack of the frameworks we utilized for developing our web application. The stack for the most part, follows the stack of the automatically generated template web application using JHipster, with our stack including support for Survey generation.

7.2.1 Client-side stack

Application Development

We use **Angular** as our main web-application development framework for *Typescript*. In order to develop responsive front-end interfaces, we utilize the **Bootstrap** CSS framework. The internalization throughout the client-side stack takes place using the **i18n** internalization framework.

Development Workflow

We use **npm** as the Javascript package manager for our application. In order to have quick and optimized building times for our development server we utilize the **Webpack** module bundler.

[1] https://www.jhipster.tech/ is a code-generation framework for web-development.

7.2.2 Server-side stack

The server-side application is built and run using the **Maven** project management software. The application is configured using **Spring Boot**. We use MySQL for our RDBMS system. The server-side application is a complete **Spring** application, using *Spring* to create a REST MVC application, utilizing **Spring Security** for authentication and access-control and **Spring Data JPA** for the JPA based data access layers.

7.2.3 Testing

We utilize **JUnit5** and **Jest** for Unit and UI tests respectively. The *Angular* code is tested using the **Cypress** framework.

7.2.4 Version Control

We use **git** as our version control software. Distributed version control is achieved by hosting our code in **GitHub**. Our database schema changes are version-controlled using the **liquibase** framework.

7.2.5 Survey Page

We use the **SurveyJS** framework in order to generate and display our dynamic survey for each participant in the study. The framework is supported in *Angular* with a powerfull API that enables dynamic survey customization before, during and after the survey's completion.

In our case, the survey is dynamically and independently generated for each participant in the study, inserting the questions that correspond to that particular participant and localizing the study to the participant's web client currently selected language. After the survey is completed, the participant's answers are stored in our database.

7.3 Data Model & API

One of the main purposes of our web application is disseminating the dataset of court decision documents we have collected. To that end, we develop a data model that structures our data and we make the structured data available through a RESTful API.

7.3.1 Data Model

In order to develop a database relationship schema that is compatible with *JHipster*, we construct a *JHipster Domain Language* (JDL) data model[2]. This way, our application can automatically: 1) update the database and the liquibase changeset, 2) create a JPA entity, a Spring Data JPA Repository and a Spring MVC REST Controller, for the server-side of our application, 3) create Angular component, router and service corresponding

[2] https://start.jhipster.tech/jdl-studio/

to the entity as well as HTML views for the visualization of the entity data, 4) generate integration and performance tests.

We structure the data in the JDL data model that is displayed in Figure 7.1.

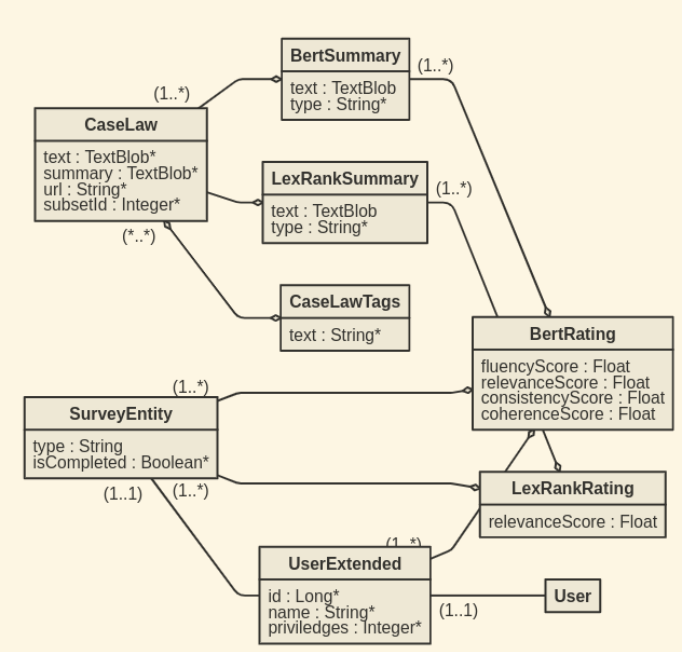


Figure 7.1. *A JDL schema of the entity fields and the cross-entity relationships in our database.*

Each Survey Entity is uniquely identified with the User it corresponds to. Each Survey Entity is related to multiple BERT or LexRank summary ratings, with each BERT/LexRank summary rating entity being related - using a Many-to-One relationship - to a BERT summary or a LexRank summary, respectively. Each summary entity is related Many-to-One to a case-law entity, that corresponds to a judicial judgement text, its corresponding reference summary, as well as metadata such as the classification tags corresponding to the judgement and the url to the webpage from which the judgement was scrapped.

Our database schema allows for surveys independently customized to each legal expert, by including different judgements and summaries to be evaluated.

7.3.2 API

Having defined the database model, we generated RESTful API that corresponds to CRUD operations on the database entities. We utilize **SwaggerUI** to expose the API to the users, by generating a web-interface which consists of the API complete documentation as well as an interface for sending and displaying the results of an API request. The API requests are secured through *JSON Web Tokens* (JWT) using the *Spring Security* access-control framework. Our documentation web-interface page, automatically authenticates the API requests, by including the JWT token to the requests under the hood, thus enabling for secure but simple use of the API through the web interface.

Figure 7.2 includes an example of the web interface for creating or updating an al-

ready existent BERT summary entity. Figure 7.3 includes a screen-capture of our API's documentation interface and Figure 7.4 presents the graphical interface of sending an API request through our documentation's interface page.

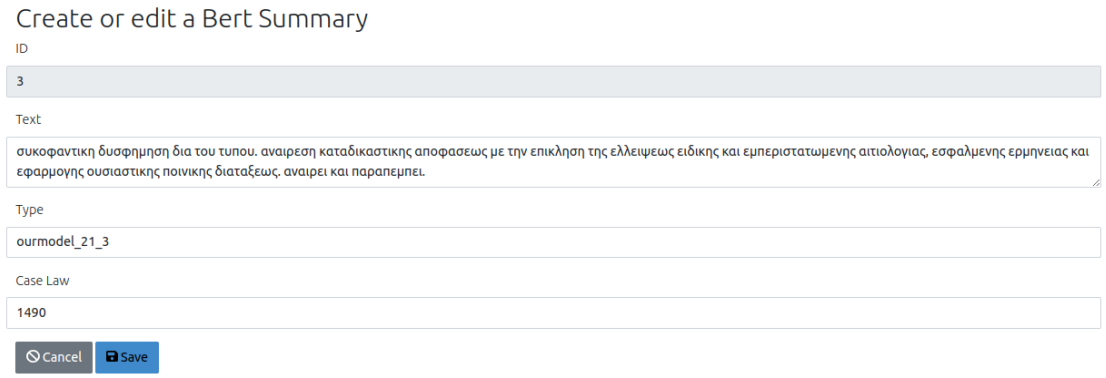


Figure 7.2. *Creation/Update interface for the Bert Summary entity through our applications web interface.*

Figure 7.3. *Our REST API's documentation interface. Screen-capture displays the interface for two entities of our database.*

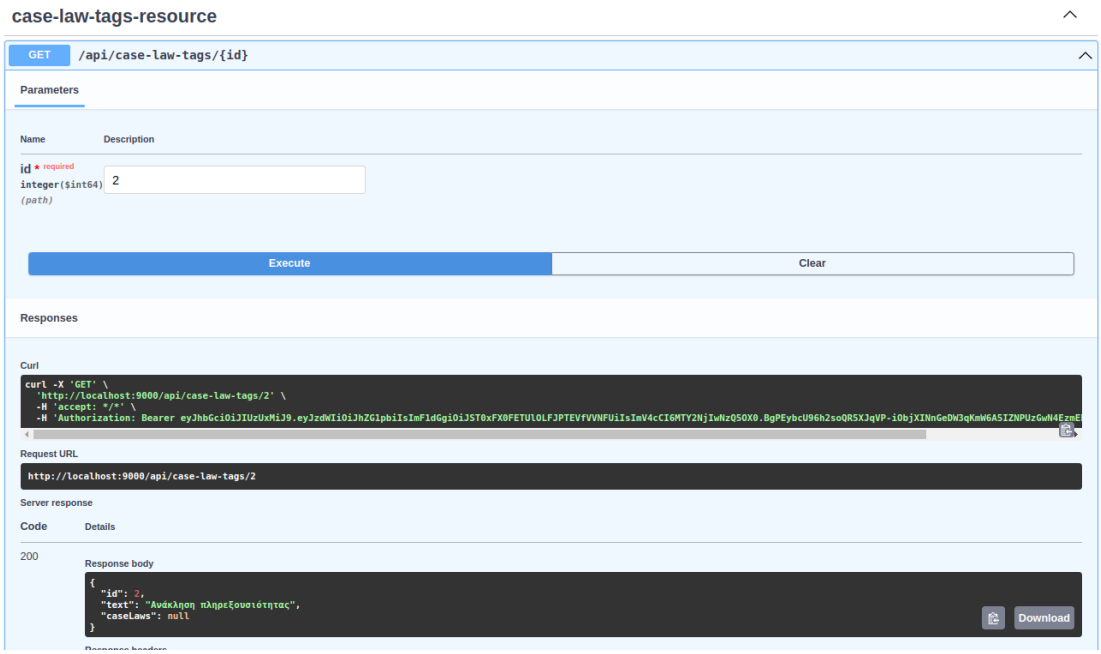


Figure 7.4. *Screen-capture of sending a REST API request and displaying its result through our documentation page's interface.*

The API endpoint for sending HTTP RESTful requests to our web-application for a particular entity is "http://WEB_SERVER_IP:PORT/api/ENTITY_NAME", where WEB_-SERVER_IP, PORT denote, respectively, the IP that hosts the web application's server and the PORT through which the web server listens to HTTP requests. The CRUD operations supported by our web-application are implemented as GET, POST, PUT, DELETE, PATCH HTTP requests that we will expand on in the following subsections.

GET-Requests

Get requests are supported in the following endpoint formats:

- **BASE_URL/api/ENTITY_NAME/id**: where the entity item with the specified id is returned.

- **BASE_URL/api/ENTITY_NAME/count[?parameter.filterFunc=value]***: where the number of entity items is returned. This query can be passed with several parameter values to limit the entities that will count towards the returned count value.

- **BASE_URL/api/ENTITY_NAME[?parameter.filterFunc=value]*[?page=pv][?size=sv][?sort=sv]**: where the entity items that match the parameter values specified are returned. The items can be returned in a specified sorting order. Furthermore, the output response is paginated and the request can specify the page index and size that will be returned.

DELETE-Requests

Delete request endpoints take the form of **BASE_URL/api/ENTITY_NAME/id** and correspond to the specified entity's deletion.

POST/PUT/PATCH-Requests

Those request endpoints follow the format: **BASE_URL/api/ENTITY_NAME/id** and correspond to an entity item's creation/update/partial update, respectively. The POST, PUT requests must contain the full item's schema in the request body, while the PATCH request can contain only part of the item's schema as it corresponds to partial update.

Request Parameters

As mentioned previously, our GET requests can specify parameter values that the returned entity items' attributes must have. The parameters available through our REST API are the following:

- **greaterThan/greaterThanOrEqual/lessThan/lessThanOrEqual:** which are applicable only for integer entity attributes and implement the corresponding mathematical comparison operators. The parameter value is an integer.
- **contains/notContains:** which apply only to string entity attributes and restrict the response output only to entities whose attribute string value contains/does not contain a substring specified by the parameter value. The parameter value must be of string type.
- **equals/notEquals:** which are applicable to all entity attribute types and implement the equality/inequality mathematical operator. The parameter value has the same type as the entity attribute that is compared with.
- **in/notIn:** which are applicable to all entity attribute types and implement the list membership operation. The parameter value is an array whose items have the same type as the entity attribute that this parameter applies to.
- **specified:** which applies to all entity attribute types and controls whether an entity attribute must be specified in order that the entity is included in the response output. The parameter value is boolean.
- **distinct:** which takes a boolean parameter value and controls whether duplicates are allowed in the response output or not.

User Resource

The user resource endpoint **BASE_URL/api/admin/user[/login_id]** is available only to authenticated users with administrator privileges. Through this endpoint, the administrators have access to CRUD operations for the User entities.

- **GET requests:** Return the user specified by the login_id parameter, or all users in a paginated form, if this parameter is not used. In the second case, the page-index, page-size and sorting criteria can be specified as request parameters, similarly to the Entity GET-requests.
- **DELETE requests:** The user with the specified login_id is deleted.
- **POST/PUT requests:** correspond to the insertion/update/partial update of an User entity. The request body must include the full schema of the inserted/updated entity item.

User JWT-controller Resource

The user jwt-controller resource authentication endpoint (**BASE_URL/api/authenticate**) serves POST requests that include in their body: the username, the password and a "rememmemberMe" flag. The request returns a jwt token to the user, if the password corresponds to the specified username. The jwt token is used for user authentication through all other API requests.

Account Resource

This resource includes the register endpoint (**BASE_URL/api/register**) which serves only POST requests. The request's body must include values for all attributes of the User entity's schema. The authenticate endpoint (**BASE_URL/api/authenticate**) serves only GET requests which include the JWT token in the request body and returns whether that JWT corresponds to an authenticated user or not. Finally, the user activation endpoint (**BASE_URL/api/authenticate**) serves GET requests that contain a confirmation-key (sent to new users via their registered email) parameter and returns whether a user can be activated using this key.

The following endpoints are available only to authenticated users:

- **BASE_URL/api/account:** which serves both GET and POST requests and returns/updates the User's account information.
- **BASE_URL/api/reset-password/[init|finish]:** which serves POST request that either initialize the password reset process by sending the corresponding password-reset email containing the password-reset key, or reset the password using the password-reset key and and the new password, respectively.

The web application, at the time of writing of our thesis, is currently hosted at `http://62.217.82.149:8080`. User access can be granted through request.

7.4 Survey Page

For the human evaluation study, we develop a web interface using the SurveyJS[3] Javascript framework for dynamic survey creation. The web-interface page consists of

[3] `https://surveyjs.io`

two parts: 1) **Part 1**; where the structure of the survey is explained, an answering guide is presented and the participants answer questions about their legal-domain knowledge. 2) **Part 2**; where the participants extract highlights from each judicial judgement text and evaluate extractive and abstractive summaries corresponding to each text. A flowchart schema of the survey page web-interface is presented in Figure 7.5.

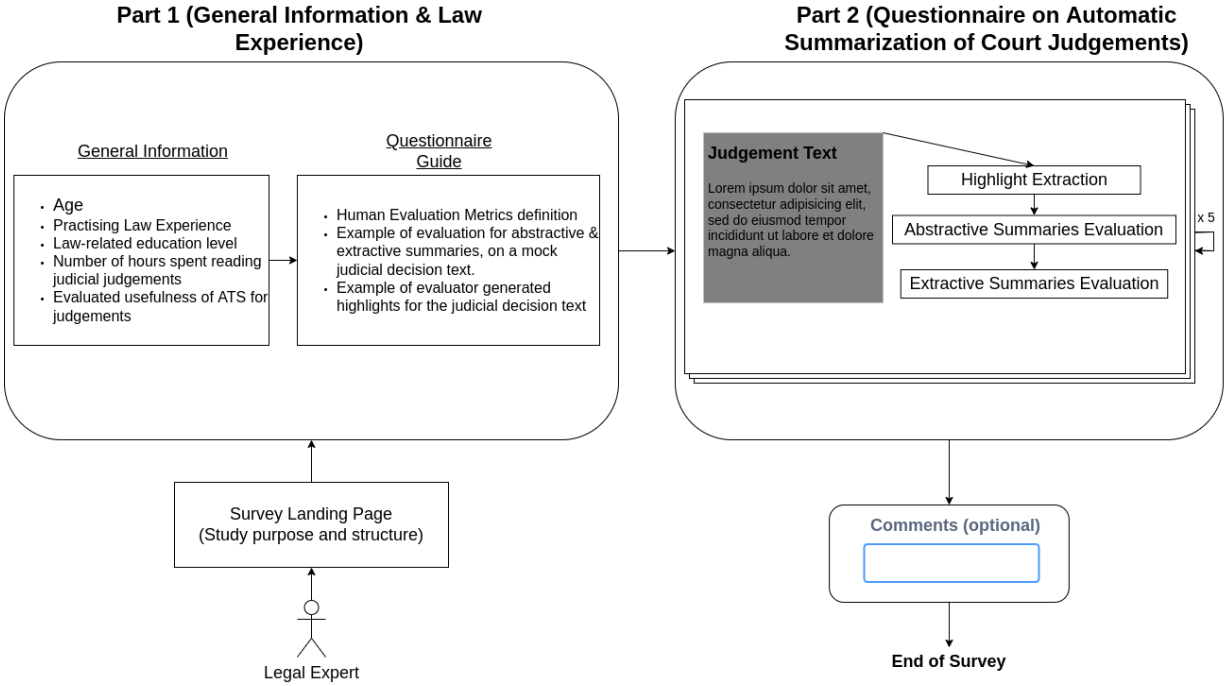


Figure 7.5. *A flowchart schema of the survey page web-interface.*

7.4.1 Part 1: Participants General Information & Questionnaire Guide

In this section of the survey, the survey structure and purpose is explained, the participant is given a guide on how the questions will be formatted and how to answer them. Finally the participant answers general information questions about their law-domain knowledge in terms of educational level and the hours spent reading judicial decisions.

Survey Purpose & Structure

First, the legal-expert human evaluator is presented with the survey's page landing page. In this page, the purpose of conducting is explained and the structure of the survey is outlined. Furthermore, the participant is given instructions of how to change the survey's language should they desire it; the survey page currently supports Greek and English. The landing page can be seen in Figure 7.6.

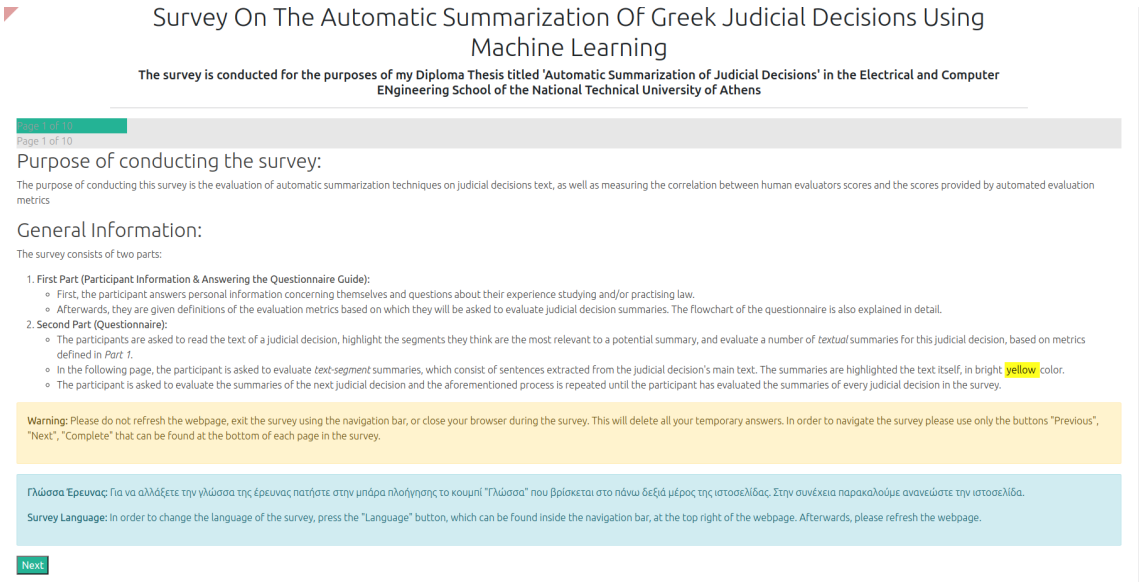


Figure 7.6. *A screenshot of survey's landing page. This page explains the purpose of the survey, and also outlines the survey's structure to the participant.*

Participant General Information & Legal-Domain knowledge

In the next page, the participant is asked to answer questions relating to general information about them; such as their age, questions relating to their legal-domain educational level and their time spent reading court decision texts. Finally, the participants evaluate the usefulness of an automatic summarization app for court judgement texts. The corresponding web-page can be seen in Figure 7.7.

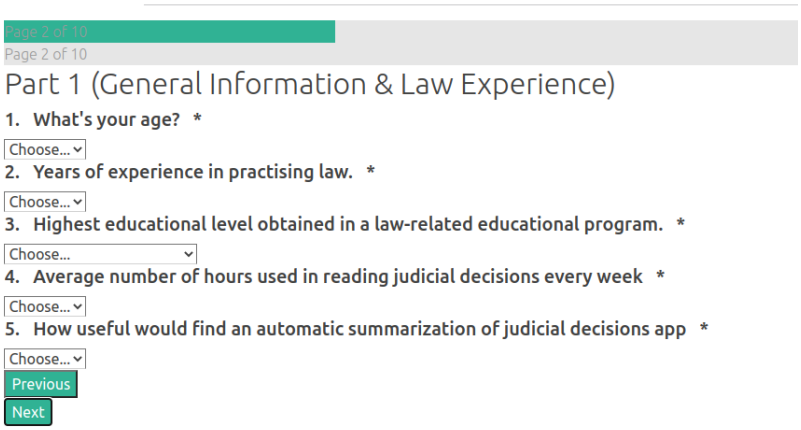


Figure 7.7. *A screenshot of survey's general & legal knowledge questionnaire page.*

Questionnaire Guide

In the following page, the participant is presented with a guide that explains the questionnaire question's format and how the participant must answer each question. First, the metrics that the participant must use, are defined. Afterwards, a mock court judgement text is given to the participant in order to familiarize them with the segment

highlight task they will have to complete in each court judgement text. Finally, the participant is presented with mock abstractive & extractive summaries of the text and a metric evaluation table, so they understand their format in the rest of the survey. In Figure 7.8, we provide two screen-captures of the aforementioned page.

7.4.2 Part 2: Questionnaire on Automatic Summarization of Court Judgements

In this part of the survey, the participant evaluates extractive and abstractive summaries of the judicial judgement texts. In the case of abstractive summaries, the participant is presented -in random order - the reference summary of the court's judgement and the summary generated by our abstractive summarization method. In the case of extractive summaries, the participant is presented with the extractive summaries generated by our LexRank$_{\text{tf-idf}}$ and Biased LexRank methods.

Highlight Extraction & Abstractive Summaries Evaluation

The participant is given the court judgement's main text and is asked to extract the segments they assess to be relevant to a potential summary of the judgement. The participant selects each segment separately which registers as a highlight after the corresponding "highlight" button is pressed. Afterwards, the participant evaluates abstractive summaries of the judicial judgement. Those consist of a reference summary; generated by the court's legal editors, and an automatically generated summary; generated by our abstractive summarization method. The aforementioned process can be seen in Figure 7.9.

Extractive Summaries Evaluation

In the evaluation of extractive summaries, the human evaluators were presented with the main text of the court decision with the extracted summaries highlighted in bright yellow color (Figure 7.11). After reading each extractive summary separately, the participant rates it and proceeds to the next extractive summary for the same judgement text.

End of Survey

After completing the survey, the human evaluators are presented with an optional text prompt for adding any comment they have on the survey and the questions asked (Figure 7.10).

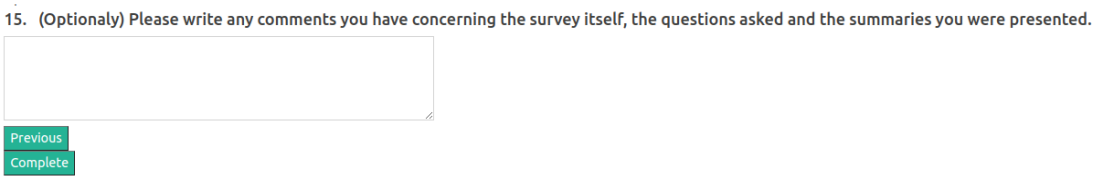
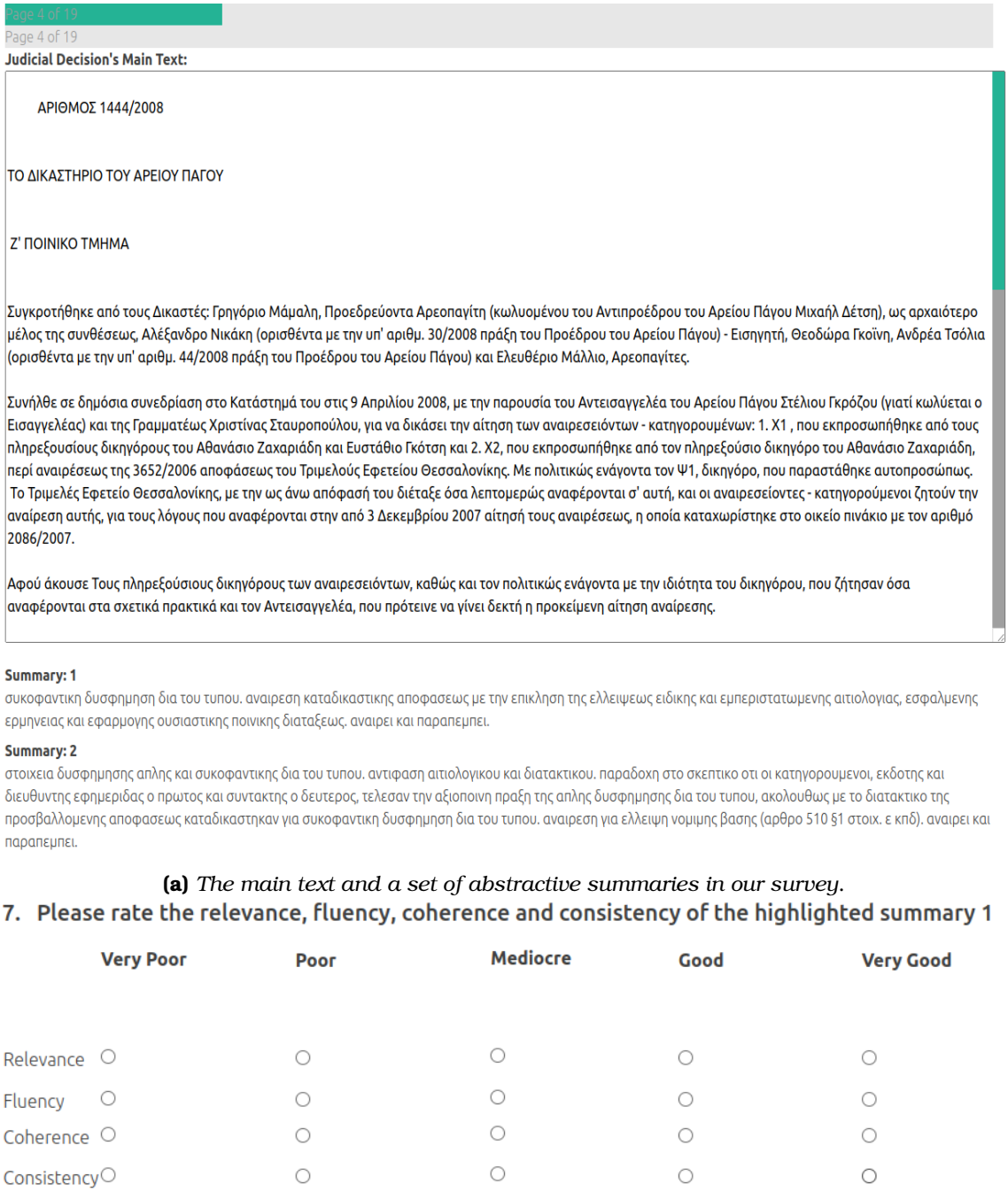


Figure 7.10. *A screenshot of optional text prompt for comments that is given to the evaluators after completing the survey.*

(a) *First part of the questionnaire guide page.*

(b) *The evaluation metrics interface for abstractive summarization in our survey's guide page.*

Figure 7.8. *Two screen-captures of the questionnaire guide page.*



(a) *The main text and a set of abstractive summaries in our survey.*

(b) *The evaluation metrics interface for abstractive summarization in our survey.*

Figure 7.9. *Two screen-captures of the abstractive summarization evaluation in our survey.*

ΤΟ ΔΙΚΑΣΤΗΡΙΟ ΤΟΥ ΑΡΕΙΟΥ ΠΑΓΟΥ

Σ' ΠΟΙΝΙΚΟ ΤΜΗΜΑ

Συγκροτήθηκε από τους Δικαστές: Γρηγόριο Μάμαλη, Προεδρεύοντα Αρεοπαγίτη (κωλυομένου του Αντιπροέδρου του Αρείου Πάγου Μιχαήλ Δέτση), ως αρχαιότερο μέλος της συνθέσεως, Αλέξανδρο Νικάκη (ορισθέντα με την υπ' αριθμ. 30/2008 πράξη του Προέδρου του Αρείου Πάγου) - Εισηγητή, Θεοδώρα Γκοΐνη, Ανδρέα Τσόλια (ορισθέντα με την υπ' αριθμ. 44/2008 πράξη του Προέδρου του Αρείου Πάγου) και Ελευθέριο Μάλλιο, Αρεοπαγίτες.

Σ υνήλθε σε δημόσια συνεδρίαση στο Κατάστημά του στις 9 Απριλίου 2008, με την παρουσία του Αντεισαγγελέα του Αρείου Πάγου Στέλιου Γκρόζου (γιατί κωλύεται ο Εισαγγελέας) και της Γραμματέως Χριστίνας Σταυροπούλου, για να δικάσει την αίτηση των αναιρεσειόντων - κατηγορουμένων: 1. Χ1 , που εκπροσωπήθηκε από τους πληρεξουσίους δικηγόρους του Αθανάσιο Ζαχαριάδη και Ευστάθιο Γκότση και 2. Χ2, που εκπροσωπήθηκε από τον πληρεξούσιο δικηγόρο του Αθανάσιο Ζαχαριάδη, περί αναιρέσεως της 3652/2006 αποφάσεως του Τριμελούς Εφετείου Θεσσαλονίκης. Με πολιτικώς ενάγοντα τον Ψ1, δικηγόρο, που παραστάθηκε αυτοπροσώπως.
Το Τριμελές Εφετείο Θεσσαλονίκης, με την ως άνω απόφασή του διέταξε όσα λεπτομερώς αναφέρονται σ' αυτή, και οι αναιρεσείοντες - κατηγορούμενοι ζητούν την αναίρεση αυτής, για τους λόγους που αναφέρονται στην από 3 Δεκεμβρίου 2007 αίτησή τους αναιρέσεως, η οποία καταχωρίστηκε στο οικείο πινάκιο με τον αριθμό 2086/2007.

Α φού άκουσε Τους πληρεξούσιους δικηγόρους των αναιρεσειόντων, καθώς και τον πολιτικώς ενάγοντα με την ιδιότητα του δικηγόρου, που ζήτησαν όσα αναφέρονται στα σχετικά πρακτικά και τον Αντεισαγγελέα, που πρότεινε να γίνει δεκτή η προκείμενη αίτηση αναίρεσης.

ΣΚΕΦΘΗΚΕ ΣΥΜΦΩΝΑ ΜΕ ΤΟ ΝΟΜΟ
Η αξιόποινη πράξη της δυσφημήσεως περιλαμβάνει, σύμφωνα με το άρθρο 362 Π.Κ., αντικειμενικώς με τον υπό του δράστη ισχυρισμό ενώπιον τρίτου ή διάδοση με οποιοδήποτε τρόπο για κάποιον άλλον γεγονότος, δυναμένου να βλάψει την τιμή ή την υπόληψη αυτού υποκειμενικώς δε τη γνώση του δράστη, ότι το ισχυριζόμενο ή διαδιδόμενο γεγονός είναι κατάλληλο να βλάψει την τιμή ή την υπόληψη άλλου και τη θέληση όπως ισχυρισθεί ενώπιον τρίτου ή διαδώσει το τοιούτο βλαπτικό γεγονός. Εξάλλου, για τη στοιχειοθέτηση της υπό του άρθρου 363 του ιδίου κώδικα προβλεπομένης αξιόποινης πράξεως της συκοφαντικής δυσφημήσεως απαιτείται, επί πλέον των αναφερθέντων στοιχείων, όπως το ως άνω γεγονός, το οποίο ισχυρίσθηκε ή διέδωσε ο δράστης, είναι ψευδές και αυτός να τελεί σε γνώση της αναληθείας του. Ο ισχυρισμός ή η διάδοση του δυσφημιστικού γεγονότος, μπορεί να γίνει και δια του τύπου, οπότε υπάρχει για τους υπευθύνους του εντύμου έγκλημα απλής ή συκοφαντικής δυσφήμησης δια του τύπου, το οποίο, μετά την κατάργηση με το άρθρο μόνο του Ν. 2243/1994 (που ισχύει από της 30.10.1994) όλων των ειδικών περί τύπου διατάξεων, συντελείται από τις ίδιες ακριβώς προϋποθέσεις που απαιτούνται για την απλή και συκοφαντική δυσφήμηση. Στην προκειμένη περίπτωση, με την προσβαλλόμενη απόφαση, όπως εξ αυτής προκύπτει, το Τριμελές Εφετείο Θεσσαλονίκης κήρυξε ενόχους τους αναιρεσείοντες, για το ότι "στη ..... στις .... ισχυρίστηκαν για άλλον εν γνώσει τους ψευδές γεγονός που μπορούσε να βλάψει την τιμή και υπόληψή του. Σ υγκεκριμένα ο μεν Χ1 ως εκδότης και διευθυντής της εφημερίδας ..... - ..., που κυκλοφόρησε στη ...., ο δε Χ2, ως συντάκτης της εν λόγω εφημερίδας καταχώρισαν άρθρο στο φύλλο της ....., στο οποίο αναφέρονταν ότι "υπάρχουν πολλά ερωτηματικά για την πρώην διοίκηση του Δήμου ...., καθώς κάθηκαν 330.000 ευρώ για τα σχολεία της ....., η προηγούμενη διοίκηση του Δήμου ..... εισέπραξε τα χρήματα προκειμένου να καλύψει τις λειτουργικές ανάγκες των σχολείων της περιοχής, πλην όμως δεν το έπραξε για το έτος 2002". Με τον τρόπο αυτό άφηναν να γίνει δεκτό με τη λογική ότι ο εγκαλών Ψ1, ως Δήμαρχος ....., το διάστημα από 1.1.1998 έως 31.12.2002 δεν διέθεσε το ποσό των 330.000 ευρώ για τις σχολικές ανάγκες της περιφέρειας του Δήμου .... το έτος 2002, μολονότι αυτά είχαν εισπραχθεί για το λόγο αυτό. Τα όσα δε υποστήριξαν στο προαναφερθέν άρθρο και υπέπεσαν στην αντίληψη του αναγνωστικού κοινού της εν λόγω εφημερίδας ήταν εν γνώσει τους ψευδή, καθώς η αλήθεια ήταν ότι ο εγκαλών με την προαναφερθείσα ιδιότητα του διέθεσε όλα τα κονδύλια που δόθηκαν στο Δήμο ..... για τις σχολικές ανάγκες της περιφέρειας του, χωρίς να αφήσει μέρος αυτών αδιάθετο και μπορούσαν να επιφέρουν μείωση στην τιμή και υπόληψη του εγκαλούντος, καθώς ήταν αντίθετα στην ευπρέπεια και ηθική". Εξάλλου, το Εφετείο αιτιολογώντας την καταδικαστική του κρίση, διέλαβε στο σκεπτικό της αποφάσεως ότι από τα εκτιθέμενα σ' αυτό πραγματικά περιστατικά πλήρως αποδείχθηκαν τα κατά το νόμο στοιχεία για την στοιχειοθέτηση της αντικειμενικής και υποκειμενικής υπόστασης επί αποδιδόμενης στους κατηγορουμένους πράξης της απλής δυσφήμησης δια του τύπου και πρέπει να κηρυχθούν ένοχοι κατά το διατακτικό. Ε τσι όμως δεν προκύπτει σαφώς και ορισμένως για ποία αξιόποινη πράξη κατεδίκασε το εφετείο τους αναιρεσείοντες, δηλαδή γι' αυτή της απλής δυσφημήσεως ή για εκείνη της συκοφαντικής δυσφημήσεως.
Συνεπώς καθίσταται ανέφικτος ο ακυρωτικός έλεγχος, αν στην προκειμένη περίπτωση τα υπό του δικάσαντος δικαστηρίου γενόμενα δεκτά περιστατικά, κατά την περί πραγμάτων ανέλεγκτη κρίση του, υπήχθησαν ορθώς ή όχι στο νόμο και έτσι η προσβαλλόμενη απόφαση, λόγω της ασάφειας αυτής, στερείται νόμιμης βάσης και υπέπεσε στην πλημμέλεια του άρθρου 510 παρ. 1 στοιχ. Ε' ΚΠΔ. Επομένως ο από τη διάταξη αυτή δεύτερος λόγος της αναιρέσεως είναι βάσιμος και γι' αυτό πρέπει να γίνει δεκτή η κρινόμενη αίτηση, να αναιρεθεί εξ ολοκλήρου η ως άνω απόφαση και να παραπεμφθεί (άρθρο 519 ΚΠΔ) η υπόθεση για νέα συζήτηση στο ίδιο δικαστήριο, συγκροτούμενο από άλλους δικαστές, εκτός από εκείνους που δίκασαν προηγουμένως.

ΓΙΑ ΤΟΥΣ ΛΟΓΟΥΣ ΑΥΤΟΥΣ
Αναιρεί την 3652/2006 απόφαση του Τριμελούς Εφετείου Θεσσαλονίκης. Και
Παραπέμπει την υπόθεση για νέα συζήτηση στο ίδιο δικαστήριο, το οποίο θα συγκροτηθεί από άλλους δικαστές, εκτός από εκείνους που δίκασαν προηγουμένως.
Κρίθηκε και αποφασίσθηκε στην Αθήνα στις 30 Απριλίου 2008. Και
Δημοσιεύθηκε στην Αθήνα, σε δημόσια συνεδρίαση στο ακροατήριό του, στις 2 Ιουνίου 2008.

Ο ΠΡΟΕΔΡΕΥΩΝ Η ΓΡΑΜΜΑΤΕΑΣ

Figure 7.11. *A screenshot of the extractive summarization evaluation in our survey.*

Chapter 8

Conclusions

8.1 Discussion

Automatic Text Summarization (ATS) is an active research field, where various methods are used to automatically shorten a text, while preserving the most important or relevant information within the original content. ATS methods can be beneficial in cases where manually reading the whole text is very time consuming. Furthermore, ATS methods can be useful for downstream tasks such as, web-page snippet generation for search engines, screenplay/book summarization that enables easier metadata tagging and semantic linking/searching, generating a consensus summary of a business by summarizing a number of individual reviews, etc.

In the legal domain, the need for robust and reliable ATS systems is large. Law practitioners, judges and scholars have to manually search for statures and caselaws that are relevant to their work. Summarizing legal texts is hard task; as those texts are often long and contain legal terminology. In the case of summarizing court rulings, the job is often outsourced to specialized legal editors.

In this work, we experiment with different methods of automatically summarizing judgements from Greek courts. Because of the lack of any Greek court-judgement dataset, we developed web-crawling scripts in order to construct two datasets of Greek court judgements; 1) The AreiosPagos dataset, which contains judgements, and their corresponding reference summaries and category tags, from the Greek Court of Cassation, 2) The STE dataset, which contains judgements and corresponding metadata from the Greek Council of State, but doesn't contain any reference summaries. We compare the AreiosPagos dataset with several other text-summarization datasets using metrics common in the relevant literature.

We developed an *extractive summarization* system based on the LexRank algorithm, that extracts the important sentences from the judgement's text. We compare several variations of the sentence similarity function used by the system.

We also developed an *abstractive summarization* system using an Encoder-Decoder model based on the BERT architecture. The model uses weights pre-trained on greek legal tasks using typical BERT self-supervised tasks, that were recently open-sourced by researchers [69]. The model is fine-tuned on our AreiosPagos dataset.

Our ATS systems were automatically evaluated using the ROUGE metrics. We find

that our extractive summarization methods outperform the random sentence baseline, but there is still room for improvement. For the abstractive summarization methods, we find that domain-specific text preprocessing, that removes redundant information from the text and incorporates case-specific category descriptions, improves the model's performance.

We further evaluated our systems with the help of legal experts in terms of the *relevance, consistency, coherence, fluency* of the generated summaries. The results of the extractive summarizers look promising, as they manage to capture some of the relevant passages in the court judgement texts. The abstractive summarizers produce relatively fluent and coherent summaries which, however, fail to be factually consistent with the judgement's main text or capture all the relevant context information that a summary must have. The large amount of time that our human evaluators spend every week reading past court judgements indicates that research in legal ATS systems that are better at capturing all the relevant passages in a summary and are more factually grounded, could be beneficial for the Legal practitioners in Greece.

8.2 Future Work

Our work can be extended into three different ways: 1) By studying how various different neural network architectures can benefit the generated summaries' quality, 2) study the human evaluation performance on neural network architectures that take actively aim to produce factually consistent summaries 3) By developing domain-specific automatic evaluation metrics and preprocessing.

Different Neural Network architectures that can be explored include:

- Hierarchical Transformer Networks: which can bypass the transformer's quadratic complexity, by first generating segment-level representations and afterwards merging them into a document-level representation. The document-level representations can be constructed either naively by concatenation or averaging, or by further Transformer transformation as in [152]. This approach may greatly improve maximum input size limit that exists in the BERT architecture.

- Different Attention Mechanism: which reduce the quadratic complexity of the original self-attention layer. Those include the Longformer [12] and the Reformer [65] architectures. However, those models require pre-training from scratch, which was outside of the score of our work.

In terms of studying abstractive summarization models that apply factuality constraints during training, our work can be extended by:

- Incorporating negative summary samples during training with contrastive learning, as in [76]. This can help the model to avoid being unfaithful to the input text. In the court-judgements domain, negative summaries can be generated by replacing case-law citations with random ones.

- Jointly learning to summarize while also learning to generate and answer questions relating to the input court judgement text. This has been shown to produce more faithful abstractive summaries [139, 95].

In terms of future work on domain-specific evaluation metrics and preprocessing, we would like to experiment with:

- Conducting a large-scale human evaluation study where the evaluators are asked to highlight important sentences from a court judgement. This dataset can be used to measure the correlation between human evaluation and automatic metrics, while also serve as a way of constructing better *extractive summarization* systems.
- Developing domain/language-specific methods for textual segmentation and tagging in Greek court-judgements. Tagging via mining legal arguments [145, 144] or extracting a judgement's rationale [22] seem promising. This information can be used downstream to inform both *extractive* and *abstractive* summarization systems.

Finally, our work can be extended by using the "Συμβούλιο της Επικρατείας" Court Judgement dataset we collected, for further training and evaluation of our methods. This would, however, require development of 1) denoising processes that would clear the data from transcription noise and 2) quality evaluation processes that asses the fidelity of the texts after the denoising process.

Appendices

Appendix A

Web application API

We make the API generated by the the *Springdoc-openapi* library for our application available through this link. However for brevity reasons, here we report only the part that corresponds to our entity's schema.


```
{
   "schemas":{
      "BertRating":{
         "type":"object",
         "properties":{
            "id":{
               "type":"integer",
               "format":"int64"
            },
            "fluencyScore":{
               "maximum":10,
               "exclusiveMaximum":false,
               "minimum":0,
               "exclusiveMinimum":false,
               "type":"number",
               "format":"float"
            },
            "relevanceScore":{
               "maximum":10,
               "exclusiveMaximum":false,
               "minimum":0,
               "exclusiveMinimum":false,
               "type":"number",
               "format":"float"
            },
            "consistencyScore":{
               "maximum":10,
               "exclusiveMaximum":false,
               "minimum":0,
               "exclusiveMinimum":false,
               "type":"number",
               "format":"float"
            },
            "coherenceScore":{
               "maximum":10,
               "exclusiveMaximum":false,
               "minimum":0,
               "exclusiveMinimum":false,
               "type":"number",
               "format":"float"
            },
            "surveyEntity":{
               "$ref":"#/components/schemas/SurveyEntity"
            },
            "bertSummary":{
               "$ref":"#/components/schemas/BertSummary"
            },
            "userExtended":{
               "$ref":"#/components/schemas/UserExtended"
            }
         }
      },
      "BertSummary":{
         "required":[
            "type"
         ],
         "type":"object",
```

```
        "properties":{
            "id":{
                "type":"integer",
                "format":"int64"
            },
            "text":{
                "type":"string"
            },
            "type":{
                "type":"string"
            }
        }
    },
    "SurveyEntity":{
        "required":[
            "isCompleted"
        ],
        "type":"object",
        "properties":{
            "id":{
                "type":"integer",
                "format":"int64"
            },
            "type":{
                "type":"string"
            },
            "isCompleted":{
                "type":"boolean"
            }
        }
    },
    "User":{
        "required":[
            "activated",
            "login"
        ],
        "type":"object",
        "properties":{
            "id":{
                "type":"integer",
                "format":"int64"
            },
            "login":{
                "maxLength":50,
                "minLength":1,
                "pattern":"^(?>[a-zA-Z0-9!$&*+=?^_`{|}~.-]+@[a-zA-Z0-9-]+(?:\\.[a-zA-Z0-9-]+)*)|(?>[_.@A-Za-z0-9-]+)$",
                "type":"string"
            },
            "firstName":{
                "maxLength":50,
                "minLength":0,
                "type":"string"
            },
            "lastName":{
                "maxLength":50,
                "minLength":0,
                "type":"string"
            },
            "email":{
                "maxLength":254,
                "minLength":5,
                "type":"string"
            },
            "activated":{
                "type":"boolean"
            },
            "langKey":{
                "maxLength":10,
                "minLength":2,
                "type":"string"
            },
            "imageUrl":{
                "maxLength":256,
                "minLength":0,
                "type":"string"
            },
            "resetDate":{
                "type":"string",
                "format":"date-time"
            }
        }
    },
```

```
"UserExtended":{
    "required":[
        "id",
        "name",
        "priviledges"
    ],
    "type":"object",
    "properties":{
        "id":{
            "type":"integer",
            "format":"int64"
        },
        "name":{
            "maxLength":2147483647,
            "minLength":1,
            "type":"string"
        },
        "priviledges":{
            "maximum":2,
            "minimum":0,
            "type":"integer",
            "format":"int32"
        },
        "hasExtendedUser":{
            "$ref":"#/components/schemas/User"
        },
        "bertratings":{
            "uniqueItems":true,
            "type":"array",
            "items":{
                "$ref":"#/components/schemas/BertRating"
            }
        },
        "isRespondent":{
            "$ref":"#/components/schemas/SurveyEntity"
        }
    }
},
"CaseLaw":{
    "required":[
        "subsetId",
        "url"
    ],
    "type":"object",
    "properties":{
        "id":{
            "type":"integer",
            "format":"int64"
        },
        "text":{
            "type":"string"
        },
        "summary":{
            "type":"string"
        },
        "url":{
            "type":"string"
        },
        "subsetId":{
            "maximum":2,
            "minimum":0,
            "type":"integer",
            "format":"int32"
        }
    }
},
"LexRankSummary":{
    "required":[
        "type"
    ],
    "type":"object",
    "properties":{
        "id":{
            "type":"integer",
            "format":"int64"
        },
        "text":{
            "type":"string"
        },
        "type":{
            "type":"string"
        },
```

```
            "caseLaw":{
                "$ref":"#/components/schemas/CaseLaw"
            }
        }
    },
    "CaseLawTags":{
        "required":[
            "text"
        ],
        "type":"object",
        "properties":{
            "id":{
                "type":"integer",
                "format":"int64"
            },
            "text":{
                "type":"string"
            },
            "caseLaws":{
                "uniqueItems":true,
                "type":"array",
                "items":{
                    "$ref":"#/components/schemas/CaseLaw"
                }
            }
        }
    },
    "AdminUserDTO":{
        "required":[
            "login"
        ],
        "type":"object",
        "properties":{
            "id":{
                "type":"integer",
                "format":"int64"
            },
            "login":{
                "maxLength":50,
                "minLength":1,
                "pattern":"^(?>[a-zA-Z0-9!$&*+=?^_`{|}~.-]+@[a-zA-Z0-9-]+(?:\\.[a-zA-Z0-9-]+)*)|(?>[_.@A-Za-z0-9-]+)$",
                "type":"string"
            },
            "firstName":{
                "maxLength":50,
                "minLength":0,
                "type":"string"
            },
            "lastName":{
                "maxLength":50,
                "minLength":0,
                "type":"string"
            },
            "email":{
                "maxLength":254,
                "minLength":5,
                "type":"string"
            },
            "imageUrl":{
                "maxLength":256,
                "minLength":0,
                "type":"string"
            },
            "activated":{
                "type":"boolean"
            },
            "langKey":{
                "maxLength":10,
                "minLength":2,
                "type":"string"
            },
            "createdBy":{
                "type":"string"
            },
            "createdDate":{
                "type":"string",
                "format":"date-time"
            },
            "lastModifiedBy":{
                "type":"string"
            },
            "lastModifiedDate":{
```

```
            "type":"string",
            "format":"date-time"
        },
        "authorities":{
            "uniqueItems":true,
            "type":"array",
            "items":{
                "type":"string"
            }
        }
    }
},
"ManagedUserVM":{
    "required":[
        "login"
    ],
    "type":"object",
    "properties":{
        "id":{
            "type":"integer",
            "format":"int64"
        },
        "login":{
            "maxLength":50,
            "minLength":1,
            "pattern":"^(?>[a-zA-Z0-9!$&*+=?^_`{|}~.-]+@[a-zA-Z0-9-]+(?:\\.[a-zA-Z0-9-]+)*)|(?>[_.@A-Za-z0-9-]+)$",
            "type":"string"
        },
        "firstName":{
            "maxLength":50,
            "minLength":0,
            "type":"string"
        },
        "lastName":{
            "maxLength":50,
            "minLength":0,
            "type":"string"
        },
        "email":{
            "maxLength":254,
            "minLength":5,
            "type":"string"
        },
        "imageUrl":{
            "maxLength":256,
            "minLength":0,
            "type":"string"
        },
        "activated":{
            "type":"boolean"
        },
        "langKey":{
            "maxLength":10,
            "minLength":2,
            "type":"string"
        },
        "createdBy":{
            "type":"string"
        },
        "createdDate":{
            "type":"string",
            "format":"date-time"
        },
        "lastModifiedBy":{
            "type":"string"
        },
        "lastModifiedDate":{
            "type":"string",
            "format":"date-time"
        },
        "authorities":{
            "uniqueItems":true,
            "type":"array",
            "items":{
                "type":"string"
            }
        },
        "password":{
            "maxLength":100,
            "minLength":4,
            "type":"string"
        }
```

```
            }
        },
        "LoginVM":{
            "required":[
                "password",
                "username"
            ],
            "type":"object",
            "properties":{
                "username":{
                    "maxLength":50,
                    "minLength":1,
                    "type":"string"
                },
                "password":{
                    "maxLength":100,
                    "minLength":4,
                    "type":"string"
                },
                "rememberMe":{
                    "type":"boolean"
                }
            }
        },
        "JWTToken":{
            "type":"object",
            "properties":{
                "id_token":{
                    "type":"string"
                }
            }
        },
        "KeyAndPasswordVM":{
            "type":"object",
            "properties":{
                "key":{
                    "type":"string"
                },
                "newPassword":{
                    "type":"string"
                }
            }
        },
        "PasswordChangeDTO":{
            "type":"object",
            "properties":{
                "currentPassword":{
                    "type":"string"
                },
                "newPassword":{
                    "type":"string"
                }
            }
        },
        "UserDTO":{
            "type":"object",
            "properties":{
                "id":{
                    "type":"integer",
                    "format":"int64"
                },
                "login":{
                    "type":"string"
                }
            }
        }
    }
}
```

Appendix B

Model Hyperparameters

B.1 LexRank Summarizer

For the LexRank summarizer we set the cross-sentence similarity threshold to 0.03. The PageRank submodule of the LexRank algorithm uses a dumping factor = 0.85

B.2 BERT Encoder-Decoder Summarization Model

B.2.1 Architecture

Our model's encoder uses the pre-trained weights from the Greek (base-uncased) BERT model [69][1]. Our model also uses Greek BERT's *WordPiece* subword tokenizer.

B.2.2 Training

The models are trained for 3/4 epochs with learning rate of 5e-5 with a linear learning rate schedule. The optimiser selected is *HuggingFace's* implementation of the AdamW algorithm, using the default hyperparameters of $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-8$. During both training and validation steps, the batch size is set to 2. During inference, we set the minimum summary length equal to 8, generated using n=5 beams, applied bi-gram blocking and applied a exponential length reward of 1.3.

[1] https://huggingface.co/nlpaueb/bert-base-greek-uncased-v1

Appendix C

Summaries Examples

In this appendix, we present examples of generated and reference summaries, along with the court decision text, from the test subset of our dataset.

C.1 Abstractive Summaries

Reference summaries are presented exactly as they were shown to human evaluators; uncased and with the diacritics removed, in order to match the output format of the generated summaries.

C.1.1 Test case - 1

- **Reference Summary:** στοιχεια δυσφημησης απλης και συκοφαντικης δια του τυπου. αντιφαση αιτιολογικου και διατακτικου. παραδοχη στο σκεπτικο οτι οι κατηγορουμενοι, εκδοτης και διευθυντης εφημεριδας ο πρωτος και συντακτης ο δευτερος, τελεσαν την αξιοποινη πραξη της απλης δυσφημησης δια του τυπου, ακολουθως με το διατακτικο της προσβαλλομενης αποφασεως καταδικαστηκαν για συκοφαντικη δυσφημηση δια του τυπου. αναιρεση για ελλειψη νομιμης βασης (αρθρο 510 §1 στοιχ. ε κπδ). αναιρει και παραπεμπει.

- **Generated Summary:** συκοφαντικη δυσφημηση δια του τυπου. αναιρεση καταδικαστικης αποφασεως με την επικληση της ελλειψεως ειδικης και εμπεριστατωμενης αιτιολογιας, εσφαλμενης ερμηνειας και εφαρμογης ουσιαστικης ποινικης διαταξεως. αναιρει και παραπεμπει.

ΑΡΙΘΜΟΣ 1444/2008
ΤΟ ΔΙΚΑΣΤΗΡΙΟ ΤΟΥ ΑΡΕΙΟΥ ΠΑΓΟΥ
Ζ΄ ΠΟΙΝΙΚΟ ΤΜΗΜΑ
Συγκροτήθηκε από τους Δικαστές: Γρηγόριο Μάμαλη, Προεδρεύοντα Αρεοπαγίτη (κωλυομένου του Αντιπροέδρου του Αρείου Πάγου Μιχαήλ Δέτση), ως αρχαιότερο μέλος της συνθέσεως, Αλέξανδρο Νικάκη (ορισθέντα με την υπ΄ αριθμ. 30/2008 πράξη του Προέδρου του Αρείου Πάγου) - Εισηγητή, Θεοδώρα Γκοΐνη, Ανδρέα Τσόλια (ορισθέντα με την υπ΄ αριθμ. 44/2008 πράξη του Προέδρου του Αρείου Πάγου) και Ελευθέριο Μάλλιο, Αρεοπαγίτες.
Συνήλθε σε δημόσια συνεδρίαση στο Κατάστημά του στις 9 Απριλίου 2008, με την παρουσία του Αντεισαγγελέα του Αρείου Πάγου Στέλιου Γκρόζου (γιατί κωλύεται ο Εισαγγελέας) και της Γραμματέως Χριστίνας Σταυροπούλου, για να δικάσει την αίτηση των αναιρεσειόντων - κατηγορουμένων: 1. Χ1 , που εκπροσωπήθηκε από τους πληρεξουσίους δικηγόρους του Αθανάσιο Ζαχαριάδη και Ευστάθιο Γκότση και 2. Χ2, που εκπροσωπήθηκε από τον πληρεξούσιο δικηγόρο του Αθανάσιο Ζαχαριάδη, περί αναιρέσεως της 3652/2006 αποφάσεως του Τριμελούς Εφετείου Θεσσαλονίκης. Με πολιτικώς ενάγοντα τον Ψ1, δικηγόρο, που παραστάθηκε αυτοπροσώπως. Το Τριμελές Εφετείο Θεσσαλονίκης, με την ως άνω απόφασή του διέταξε όσα λεπτομερώς αναφέρονται σ΄ αυτή, και οι αναιρεσείοντες - κατηγορούμενοι ζητούν την αναίρεση αυτής, για τους λόγους που αναφέρονται στην από 3 Δεκεμβρίου 2007 αίτησή τους αναιρέσεως, η οποία καταχωρίστηκε στο οικείο πινάκιο με τον αριθμό 2086/2007.
Αφού άκουσε Τους πληρεξούσιους δικηγόρους των αναιρεσειόντων, καθώς και τον πολιτικώς ενάγοντα με την ιδιότητα του δικηγόρου, που ζήτησαν όσα αναφέρονται στα σχετικά πρακτικά και τον Αντεισαγγελέα, που πρότεινε να γίνει δεκτή η προκείμενη αίτηση αναίρεσης.
ΣΚΕΦΘΗΚΕ ΣΥΜΦΩΝΑ ΜΕ ΤΟ ΝΟΜΟ Η αξιόποινη πράξη της δυσφημήσεως περιλαμβάνει, σύμφωνα με το άρθρο 362 Π.Κ., αντικειμενικώς με τον υπό του δράστη ισχυρισμό ενώπιον τρίτου ή διάδοση με οποιοδήποτε τρόπο για κάποιον άλλον γεγονότος, δυναμένου να βλάψει την τιμή ή την υπόληψη αυτού υποκειμενικώς δε τη γνώση του δράστη, ότι το ισχυριζόμενο ή διαδιδόμενο γεγονός είναι κατάλληλο να βλάψει την τιμή ή την υπόληψη άλλου και τη θέληση όπως ισχυρισθεί ενώπιον τρίτου ή διαδώσει το τοιούτο βλαπτικό γεγονός. Εξάλλου, για τη στοιχειοθέτηση της υπό του άρθρου 363 του ιδίου κώδικα προβλεπομένης αξιόποινης πράξεως της συκοφαντικής δυσφημήσεως απαιτείται, επί πλέον των αναφερθέντων στοιχείων, όπως το ως άνω γεγονός, το οποίο ισχυρίσθηκε ή διέδωσε ο δράστης, είναι ψευδές και αυτός να τελεί σε γνώση της αναληθείας του. Ο ισχυρισμός ή η διάδοση του δυσφημιστικού γεγονότος, μπορεί να γίνει και δια του τύπου, οπότε υπάρχει για τους υπευθύνους του εντίμου έγκλημα απλής ή συκοφαντικής δυσφήμησης δια του τύπου, το οποίο, μετά την κατάργηση με το άρθρο μόνο του Ν. 2243/1994 (που ισχύει από της 30.10.1994) όλων των ειδικών περί τύπου διατάξεων, συντελείται από τις ίδιες ακριβώς προϋποθέσεις που απαιτούνται για την απλή και συκοφαντική δυσφήμηση. Στην προκειμένη περίπτωση, με την προσβαλλόμενη απόφαση, όπως εξ αυτής προκύπτει, το Τριμελές Εφετείο Θεσσαλονίκης κήρυξε ενόχους τους αναιρεσείοντες, για το ότι ¨στη ..... στις .... ισχυρίστηκαν για άλλον εν γνώσει τους ψευδές γεγονός που μπορούσε να βλάψει την τιμή και υπόληψή του. Συγκεκριμένα ο μεν Χ1 ως εκδότης και διευθυντής της εφημερίδας ..... - ..., που κυκλοφόρησε στη ...., ο δε Χ2, ως συντάκτης της εν λόγω εφημερίδας καταχώρισαν άρθρο στο φύλλο της ....., στο οποίο αναφέρονταν ότι ΰπάρχουν πολλά ερωτηματικά για την πρώην διοίκηση του Δήμου ...., καθώς χάθηκαν 330.000 ευρώ για τα σχολεία της ....., η προηγούμενη διοίκηση του Δήμου ..... εισέπραξε τα χρήματα προκειμένου να καλύψει τις λειτουργικές ανάγκες των σχολείων της περιοχής, πλην όμως δεν το έπραξε για το έτος 2002¨. Με τον τρόπο αυτό άφηναν να γίνει δεκτό με τη λογική ότι ο εγκαλών Ψ1, ως Δήμαρχος ....., το διάστημα από 1.1.1998 έως 31.12.2002 δεν διέθεσε το ποσό των 330.000 ευρώ για τις σχολικές ανάγκες της περιφέρειας του Δήμου .... το έτος 2002, μολονότι αυτά είχαν εισπραχθεί για το λόγο αυτό. Τα όσα δε υποστήριξαν στο προαναφερθέν άρθρο και υπέπεσαν στην αντίληψη του αναγνωστικού κοινού της εν λόγω εφημερίδας ήταν εν γνώσει τους ψευδή, καθώς η αλήθεια ήταν ότι ο εγκαλών με την προαναφερθείσα ιδιότητα του διέθεσε όλα τα κονδύλια που δόθηκαν στο Δήμο ..... για τις σχολικές ανάγκες της περιφέρειας του, χωρίς να αφήσει μέρος αυτών αδιάθετο και μπορούσαν να επιφέρουν μείωση στην τιμή και υπόληψη του εγκαλούντος, καθώς ήταν αντίθετα στην ευπρέπεια και ηθική¨. Εξάλλου, το Εφετείο αιτιολογώντας την καταδικαστική του κρίση, διέλαβε στο σκεπτικό της αποφάσεως ότι από τα εκτιθέμενα σ΄ αυτό πραγματικά περιστατικά πλήρως αποδείχθηκαν τα κατά το νόμο στοιχεία για την στοιχειοθέτηση της αντικειμενικής και υποκειμενικής υπόστασης επί αποδιδόμενης στους κατηγορουμένους πράξης της απλής δυσφήμησης δια του τύπου και πρέπει να κηρυχθούν ένοχοι κατά το διατακτικό. Ετσι όμως δεν προκύπτει σαφώς και ορισμένως για ποία αξιόποινη πράξη κατεδίκασε το εφετείο τους αναιρεσείοντες, δηλαδή γι΄ αυτή της απλής δυσφημήσεως ή για εκείνη της συκοφαντικής δυσφημήσεως. Συνεπώς καθίσταται ανέφικτος ο ακυρωτικός έλεγχος, αν στην προκειμένη περίπτωση τα υπό του δικάσαντος δικαστηρίου γενόμενα δεκτά περιστατικά, κατά την περί πραγμάτων ανέλεγκτη κρίση του, υπήχθησαν ορθώς ή όχι στο νόμο και έτσι η προσβαλλόμενη απόφαση, λόγω της ασάφειας αυτής, στερείται νόμιμης βάσης και υπέπεσε στην πλημμέλεια του άρθρου 510 παρ. 1 στοιχ. Ε΄ ΚΠΔ. Επομένως ο από τη διάταξη αυτή δεύτερος λόγος της αναιρέσεως είναι βάσιμος και γι΄ αυτό πρέπει να γίνει δεκτή η κρινόμενη αίτηση, να αναιρεθεί εξ ολοκλήρου η ως άνω απόφαση και να παραπεμφθεί (άρθρο 519 ΚΠΔ) η υπόθεση για νέα συζήτηση στο ίδιο δικαστήριο, συγκροτούμενο από άλλους δικαστές, εκτός από εκείνους που δίκασαν προηγουμένως.
ΓΙΑ ΤΟΥΣ ΛΟΓΟΥΣ ΑΥΤΟΥΣ Αναιρεί την 3652/2006 απόφαση του Τριμελούς Εφετείου Θεσσαλονίκης. Και Παραπέμπει την υπόθεση για νέα συζήτηση στο ίδιο δικαστήριο, το οποίο θα συγκροτηθεί από άλλους δικαστές, εκτός από εκείνους που δίκασαν προηγουμένως. Κρίθηκε και αποφασίσθηκε στην Αθήνα στις 30 Απριλίου 2008. Και Δημοσιεύθηκε στην Αθήνα, σε δημόσια συνεδρίαση στο ακροατήριό του, στις 2 Ιουνίου 2008.
Ο ΠΡΟΕΔΡΕΥΩΝ Η ΓΡΑΜΜΑΤΕΑΣ

C.1.2 Test case - 3

- **Reference Summary:** απορριπτει τις αιτησεις αναιρεσεως ως απαραδεκτες, διοτι στρεφονται κατα βουλευματος που δεν επιτρεπεται αναιρεση και η δευτερη για τον επιπροσθετο λογο οτι ασκηθηκε με δηλωση στον εισαγγελεα του αρειου παγου, χωρις να συντρεχει νομιμη περιπτωση. .

- **Generated Summary:** απορριπτεται η αιτηση αναιρεσεως, ως ανυποστηρικτη, λογω μη παραστασης του κλητευθεντος αναιρεσειοντος.

ΑΡΙΘΜΟΣ 1345/2008
ΤΟ ΔΙΚΑΣΤΗΡΙΟ ΤΟΥ ΑΡΕΙΟΥ ΠΑΓΟΥ
ΣΤ ΠΟΙΝΙΚΟ ΤΜΗΜΑ - ΣΕ ΣΥΜΒΟΥΛΙΟ
Συγκροτήθηκε από τους Δικαστές: Γεώργιο Σαραντινό, Αντιπρόεδρο Αρείου Πάγου, Βασίλειο Λυκούδη και Ανδρέα Τσόλια - Εισηγητή, Αρεοπαγίτες. Με την παρουσία και του Αντεισαγγελέα του Αρείου Πάγου Βασιλείου Μαρκή (γιατί κωλύεται ο Εισαγγελέας) και της Γραμματέως Πελαγίας Λόζιου .
Συνήλθε σε Συμβούλιο στο Κατάστημά του στις 4 Δεκεμβρίου 2007, προκειμένου να αποφανθεί για την αίτηση της αναιρεσείουσας - κατηγορουμένης Χ1, που δεν παραστάθηκε στο συμβούλιο, περί αναιρέσεως του υπ' αριθμ. 275/2007 βουλεύματος του Συμβουλίου Εφετών Αθηνών. Το Συμβούλιο Εφετών Αθηνών, με το ως άνω βούλευμά του διέταξε όσα λεπτομερώς αναφέρονται σ' αυτό, και η αναιρεσείουσα - κατηγορούμενη ζητεί τώρα την αναίρεση του βουλεύματος τούτου, για τους λόγους που αναφέρονται στις από 3 και 8 Μαρτίου 2007 αιτήσεις της αναιρέσεως, οι οποίες καταχωρίστηκαν στο οικείο πινάκιο με τον αριθμό 426/2007. Έπειτα ο Αντεισαγγελέας του Αρείου Πάγου Βασίλειος Μαρκής εισήγαγε για κρίση στο Συμβούλιο τη σχετική δικογραφία με την πρόταση του Αντεισαγγελέα του Αρείου Πάγου Στέλιου Γκρόζου με αριθμό 274/29.6.2007, στην οποία αναφέρονται τα ακόλουθα: Εισάγω, σύμφωνα με το άρθρο 476 παρ.1 Κ.Π.Δ.: α) την υπάριθ. 64/8-3-2007 αίτηση αναιρέσεως της κατηγορουμένης Χ1, η οποία ασκήθηκε στο όνομα και για λογαριασμό της από τον δικηγόρο Αθηνών Ιωάννη Βαρουτά, δυνάμει της από 5-3-2007 προσαρτημένης στην αίτηση και νομίμως θεωρημένης εξουσιοδοτήσεως και στρέφεται κατά του υπάριθ. 275/2007 βουλεύματος του Συμβουλίου Εφετών Αθηνών και β) την από 3-3-2007 ταυτόσημη αίτηση της ίδιας αναιρεσείουσας Χ1 προς τον Εισαγγελέα του Αρείου Πάγου, για αναίρεση επίσης του ανωτέρω υπάριθ. 275/2007 βουλεύματος του Συμβουλίου Εφετών Αθηνών, εκθέτω δε τα ακόλουθα: Κατά τη διάταξη του άρθρου 463 Κ.Π.Δ. ένδικο μέσο μπορεί να ασκήσει μόνο εκείνος που ο νόμος του δίνει ρητά αυτό το δικαίωμα, κατά δε το άρθρο 476 παρ.1 του ίδιου Κώδικα, το ένδικο μέσο απορρίπτεται ως απαράδεκτο, εκτός των άλλων περιπτώσεων που αναφέρονται στη διάταξη αυτή και όταν ασκήθηκε εναντίον βουλεύματος, για το οποίο δεν προβλέπεται. Εξάλλου από τη διάταξη του άρθρου 476 παρ.2 του αυτού ως άνω Κώδικα, που πριν από την τροποποίησή της με το άρθρο 38 Ν. 3160/2003 όριζε ότι "κατά της απόφασης ή του βουλεύματος που απορρίπτει το ένδικο μέσο ως απαράδεκτο επιτρέπεται μόνο αναίρεση" και μετά την εν λόγω τροποποίηση απαλείφθηκε η φράση "ή του βουλεύματος", προκύπτει ότι δεν προβλέπεται πλέον αναίρεση κατά του βουλεύματος που απορρίπτει το ένδικο μέσο ως απαράδεκτο (ΑΠ 776/2006, ΑΠ 297/2006). Πρέπει στο σημείο αυτό να τονισθεί ότι η ανωτέρω διάταξη του άρθρου 476 παρ.2 Κ.Π.Δ., που, μετά την αντικατάστασή της με το άρθρο 38 Ν. 3160/2003, περιορίζει το δικαίωμα ασκήσεως αναιρέσεως εναντίον βουλεύματος που απορρίπτει την έφεση ως απαράδεκτη, συμπορεύεται με τα άρθρα 4 παρ.1 και 20 παρ.1 του Συντάγματος, αφού ούτε άνιση ρύθμιση περιέχει, που να συνεπάγεται δυσμενή μεταχείριση ορισμένων διαδίκων, ούτε στερεί τον διάδικο από το δικαίωμα παροχής έννομης προστασίας από τα δικαστήρια, δοθέντος μάλιστα και του ότι ο κοινός νομοθέτης δεν υποχρεούται από το Σύνταγμα να θεσπίζει ένδικα μέσα κατά των αποφάσεων και, ως εκ τούτου, ο διάδικος μπορεί και οφείλει να υπολογίζει, κατά την εκδίκαση της υποθέσεώς του, ότι είναι ενδεχόμενο η μέλλουσα να εκδοθεί απόφαση να μην υπόκειται σε ένδικα μέσα, παρά το γεγονός ότι αυτά επιτρέπονταν κατά την έναρξη και κατά τη διάρκεια της δίκης (ή της ποινικής διώξεως). Όπου κρίθηκε αναγκαία η καθιέρωση ενδίκου μέσου υπήρξε ρητή αποτύπωση της βουλήσεως του συντακτικού νομοθέτη με τις ειδικές προβλέψεις των άρθρων 95 παρ.1 β'(αναίρεση τελεσιδίκων αποφάσεων των διοικητικών δικαστηρίων για υπέρβαση εξουσίας ή παράβαση νόμου) και 96 παρ.2 Συντάγματος (έφεση στο αρμόδιο τακτικό δικαστήριο κατά των αποφάσεων αστυνομικών αρχών και αρχών αγροτικής ασφάλειας). Δεν είναι δε αντίθετες οι ανωτέρω διατάξεις ούτε και προς το άρθρο 6 παρ.1 της ΕΣΔΑ, που κατοχυρώνει την αρχή της δίκαιης δίκης, από την οποία δεν συνάγεται υποχρέωση του εθνικού νομοθέτη για καθιέρωση ενδίκων μέσων, αλλά ούτε και στο άρθρο 26 του Συντάγματος, το οποίο καθιερώνει την αρχή της διακρίσεως των εξουσιών (ΑΠ 1486/2005). Στην προκειμένη περίπτωση από τα έγγραφα της δικογραφίας, τα οποία επιτρεπτώς επισκοπεί ο Άρειος Πάγος για την έρευνα του παραδεκτού της αιτήσεως αναιρέσεως, προκύπτουν τα εξής: Το Τριμελές Πλημμελειοδικείο Αθηνών με την υπάριθ. 8969/7-2-2005 απόφασή του κήρυξε ένοχη την αναιρεσείουσα κατηγορουμένη Χ1 για τις πράξεις της παραβάσεως του Ν. 2971/2001 και του Ν. 1337/1983 και της επέβαλε συνολική ποινή φυλακίσεως ενός (1) έτους και τριών (3) μηνών. Κατά της αποφάσεως αυτής η αναιρεσείουσα άσκησε έφεση, η οποία κηρύχθηκε απαράδεκτη με το προσβαλλόμενο υπάριθ. 275/2007 βούλευμα του Συμβουλίου Εφετών Αθηνών. Κατά του ως άνω όμως βουλεύματος δεν επιτρέπεται, σύμφωνα με τα προεκτεθέντα, να ασκηθεί αναίρεση. Με τα δεδομένα αυτά καθίσταται φανερό ότι η αναιρεσείουσα άσκησε αναίρεση κατά βουλεύματος, εναντίον του οποίου δεν επιτρέπεται να ασκηθεί τέτοιο ένδικο μέσο, δηλαδή άσκησε μη επιτρεπόμενο σάυτήν ένδικο μέσο. Επομένως οι κρινόμενες αιτήσεις αναιρέσεως πρέπει, κατέφαρμογή της διατάξεως του άρθρου 476 παρ.1 Κ.Π.Δ., να απορριφθούν ως απαράδεκτες και να καταδικασθεί η αναιρεσείουσα στα δικαστικά έξοδα. Ανεξάρτητα από αυτά η από 3-3-2007 αίτηση αναιρέσεως, που ασκήθηκε με δήλωση στον Εισαγγελέα του Αρείου Πάγου, πρέπει να απορριφθεί ως απαράδεκτη και για τον εξής επιπρόσθετο λόγο: Από τις διατάξεις των άρθρων 473 παρ.2 και 474 παρ.1 Κ.Π.Δ. προκύπτει ότι, κατά γενική αρχή, η αίτηση αναιρέσεως ασκείται με δήλωση του δικαιουμένου διαδίκου ενώπιον των οριζομένων από την τελευταία οργάνων, στα οποία δεν περιλαμβάνεται και ο Εισαγγελέας του Αρείου Πάγου. Η κατεξαίρεση άσκηση αναιρέσεως με δήλωση που επιδίδεται στον Εισαγγελέα του Αρείου Πάγου μπορεί να γίνει μόνο εναντίον καταδικαστικής αποφάσεως, όχι δε και εναντίον οποιασδήποτε άλλης, η οποία δεν έχει αυτόν τον χαρακτήρα (ΑΠ 578/2005, ΑΠ 295/2001), όπως είναι και η απόφαση, με την οποία απορρίπτεται, η έφεση ως απαράδεκτη, πράγμα που συμβαίνει στην κρινόμενη υπόθεση.
ΓΙΑ ΤΟΥΣ ΛΟΓΟΥΣ ΑΥΤΟΥΣ Προτείνω: Α) Να απορριφθούν ως απαράδεκτες: α) η υπάριθ. 64/8-3-2007 αίτηση αναιρέσεως της Χ1 κατά του υπάριθ. 275/2007 βουλεύματος του Συμβουλίου Εφετών Αθηνών και β) η από 3-3-2007 ταυτόσημη αίτηση της ίδιας αναιρεσείουσας Χ1 προς τον Εισαγγελέα του Αρείου Πάγου, για αναίρεση επίσης του ανωτέρω υπάριθ. 275/07 βουλεύματος του Συμβουλίου Εφετών Αθηνών. Και Β) Να καταδικασθεί η αναιρεσείουσα στα δικαστικά έξοδα. Αθήνα, 9-5-2007
Ο Αντεισαγγελέας του Αρείου Πάγου Στέλιος Κ. Γκρόζος Αφού άκουσε τον Αντεισαγγελέα, που αναφέρθηκε στην παραπάνω εισαγγελική πρόταση και έπειτα αποχώρησε, ΣΚΕΦΘΗΚΕ ΣΥΜΦΩΝΑ ΜΕ ΤΟ ΝΟΜΟ ΕΠΕΙΔΗ, σύμφωνα με τη διάταξη του άρθρου 463 εδ. α' του Κώδικα Ποινικής Δικονομίας "ένδικο μέσο μπορεί να ασκήσει μόνο εκείνος, που ο νόμος του δίνει ρητά αυτό το δικαίωμα". Επομένως, αν με κάποια διάταξη καθορίζονται ορισμένα πρόσωπα, ως δικαιούμενα, εξ αντιδιαστολής προκύπτει ότι δεν δικαιούνται και άλλα, μη μνημονευόμενα πρόσωπα. Περαιτέρω, κατά το άρθρο 476 παρ. 1 του ίδιου Κώδικα, το ένδικο μέσο απορρίπτεται ως απαράδεκτο, εκτός των άλλων περιπτώσεων που αναφέρονται στη διάταξη αυτή και όταν ασκήθηκε εναντίον βουλεύματος, για το οποίο δεν προβλέπεται. Εξάλλου, από τη διάταξη του άρθρου 476 παρ. 2 του αυτού ως άνω Κώδικα, που πριν από την τροποποίησή της με το άρθρο 38 του Ν. 3160/2003 όριζε ότι "κατά της απόφασης ή του βουλεύματος που απορρίπτει το ένδικο μέσο ως απαράδεκτο επιτρέπεται μόνο αναίρεση" και μετά την εν λόγω τροποποίηση απαλείφθηκε η φράση "ή του βουλεύματος", προκύπτει ότι δεν προβλέπεται πλέον αναίρεση κατά του βουλεύματος που απορρίπτει το ένδικο μέσο ως απαράδεκτο. Στο σημείο αυτό πρέπει να λεχθεί ότι η παραπάνω διάταξη του άρθρου 476 παρ. 2 Κ.Π.Δ., που, όπως σήμερα ισχύει, περιορίζει το δικαίωμα ασκήσεως αναιρέσεως κατά του βουλεύματος που απορρίπτει την έφεση ως απαράδεκτη, δεν αντίκειται στα άρθρα παρ. 1 και 20 παρ. 1 του Συντάγματος, αφού δεν περιέχει άνιση ρύθμιση, που να συνεπάγεται δυσμενή μεταχείριση κάποιων διαδίκων, αλλ' ούτε στερεί τον διάδικο από το δικαίωμα παροχής έννομης προστασίας από τα δικαστήρια, αφού μάλιστα και ο κοινός νομοθέτης δεν υποχρεούται από το Σύνταγμα να θεσπίζει ένδικα μέσα κατά των αποφάσεων και, ως εκ τούτου, ο διάδικος μπορεί και οφείλει να υπολογίζει, κατά την εκδίκαση της υποθέσεώς του, ότι είναι ενδεχόμενο, η απόφαση που πρόκειται να εκδοθεί να μη προσβάλλεται με ένδικα μέσα, παρά το γεγονός ότι αυτά επιτρέπονταν κατά την έναρξη και κατά τη διάρκεια της δίκης (ή της ποινικής διώξεως). Όπου κρίθηκε αναγκαία η θέσπιση ένδικου μέσου, υπήρξε ρητή η αποτύπωση της βουλήσεως του συντακτικού νομοθέτη με τις ειδικές προβλέψεις των άρθρων 95 παρ. 1 β' (αναίρεση των τελεσίδικων αποφάσεων των διοικητικών δικαστηρίων για υπέρβαση εξουσίας ή παράβαση νόμου) και 96 παρ. 2 του Συντάγματος (έφεση στο αρμόδιο τακτικό δικαστήριο κατά των αποφάσεων αστυνομικών αρχών και αρχών αγροτικής ασφάλειας). Δεν είναι δε αντίθετες οι ανωτέρω διατάξεις ούτε και προς το άρθρο 6 παρ. 1 της ΕΣΔΑ, που κατοχυρώνει την αρχή της δίκαιης δίκης, από την οποία δεν συνάγεται υποχρέωση του εθνικού νομοθέτη για καθιέρωση ένδικων μέσων, αλλά ούτε και στο άρθρο 26 του συντάγματος, το οποίο καθιερώνει την αρχή της διακρίσεως των εξουσιών. Τέλος, από τις διατάξεις των άρθρων 473 παρ. 2 και 474 παρ. 1 του Κώδικα Ποινικής Δικονομίας συνάγεται ότι, κατά γενική αρχή, η αίτηση αναιρέσεως ασκείται με δήλωση του δικαιουμένου διαδίκου ενώπιον των οργάνων που ορίζονται από την τελευταία ως άνω διάταξη, μεταξύ των οποίων δεν περιλαμβάνεται και ο Εισαγγελεύς του Αρείου Πάγου. Η κατ' εξαίρεση άσκηση αναιρέσεως με δήλωση που επιδίδεται στον εισαγγελέα του Αρείου Πάγου μπορεί να γίνει μόνο εναντίον καταδικαστικής αποφάσεως, όχι δε και εναντίον οποιασδήποτε άλλης, η οποία δεν έχει αυτόν τον χαρακτήρα, όπως είναι και η απόφαση, με την οποία απορρίπτεται η έφεση ως απαράδεκτη. Εν προκειμένω, από τα έγγραφα της δικογραφίας, τα οποία επιτρεπτώς επισκοπούνται από τον Άρειο Πάγο, για την έρευνα του παραδεκτού της αιτήσεως αναιρέσεως, προκύπτουν τα ακόλουθα: Το Τριμελές Πλημμελειοδικείο Αθηνών, με την 8969/7.2.2005 απόφασή του, κήρυξε ένοχη την αναιρεσείουσα κατηγορουμένη Χ1 για τις πράξεις της παραβάσεως του Ν. 2971/2001 και του Ν. 1337/1983 και της επέβαλε συνολική ποινή φυλακίσεως ενός έτους και τριών μηνών. Κατά της αποφάσεως αυτής, η αναιρεσείουσα άσκησε έφεση, η οποία κηρύχθηκε απαράδεκτη, με το 275/2007 βούλευμα του Συμβουλίου Εφετών Αθηνών, κατά του οποίου η κατηγορουμένη άσκησε τις κρινόμενες αιτήσεις αναιρέσεως. Σύμφωνα, όμως με όσα αναφέρονται παραπάνω στη μείζονα σκέψη, κατά του ως άνω βουλεύματος δεν επιτρέπεται να ασκηθεί το ένδικο μέσο της αναιρέσεως και επομένως, οι κρινόμενες αιτήσεις αναιρέσεως, που συνεκδικάζονται, λόγω της μεταξύ τους πρόδηλης συνάφειας, πρέπει, κατ' εφαρμογή της διατάξεως του άρθρου 476 παρ. 1 του Κώδικα Ποινικής δικονομίας, να απορριφθούν ως απαράδεκτες και να καταδικασθεί η αναιρεσείουσα στα δικαστικά έξοδα (άρθρο 583 παρ. 1 ΚΠΔ). Ανεξάρτητα από αυτά, η από 3 Μαρτίου 2007 αίτηση αναιρέσεως, η οποία ασκήθηκε με δήλωση προς τον Εισαγγελέα του Αρείου Πάγου, πρέπει να απορριφθεί ως απαράδεκτη, αφού σύμφωνα με τα ανωτέρω εκτιθέμενα, άσκηση αναιρέσεως με δήλωση, επιδιδόμενη στον Εισαγγελέα του Αρείου Πάγου μπορεί να γίνει μόνο εναντίον καταδικαστικής αποφάσεως, περίπτωση που δεν συντρέχει εν προκειμένω.
ΓΙΑ ΤΟΥΣ ΛΟΓΟΥΣ ΑΥΤΟΥΣ ΑΠΟΡΡΙΠΤΕΙ α) την 64/8.3.2007 αίτηση αναιρέσεως της Χ1 κατά του 275/2007 βουλεύματος του Συμβουλίου Εφετών Αθηνών και β) την από 3.3.2007 ταυτόσημη αίτηση της ίδιας αναιρεσείουσας Χ1 που ασκήθηκε με δήλωσή της προς τον εισαγγελέα του Αρείου Πάγου για αναίρεση επίσης του ανωτέρω 275/2007 βουλεύματος του Συμβουλίου Εφετών Αθηνών. Και ΚΑΤΑΔΙΚΑΖΕΙ την αναιρεσείουσα στα δικαστικά έξοδα, τα οποία ανέρχονται στο ποσό των διακοσίων είκοσι ευρώ (220€). Κρίθηκε και αποφασίστηκε στην Αθήνα στις 14 Μαρτίου 2008. Και, Εκδόθηκε στην Αθήνα στις 20 Μαΐου 2008.
Ο ΑΝΤΙΠΡΟΕΔΡΟΣ Η ΓΡΑΜΜΑΤΕΑΣ

C.2 Extractive Summaries

C.2.1 Test case - 1

LexRank algorithm

ΤΟ ΔΙΚΑΣΤΗΡΙΟ ΤΟΥ ΑΡΕΙΟΥ ΠΑΓΟΥ

Ζ' ΠΟΙΝΙΚΟ ΤΜΗΜΑ

Συγκροτήθηκε από τους Δικαστές: Γρηγόριο Μάμαλη, Προεδρεύοντα Αρεοπαγίτη (κωλυομένου του Αντιπροέδρου του Αρείου Πάγου Μιχαήλ Δέτση), ως αρχαιότερο μέλος της συνθέσεως, Αλέξανδρο Νικάκη (ορισθέντα με την υπ' αριθμ. 30/2008 πράξη του Προέδρου του Αρείου Πάγου) - Εισηγητή, Θεοδώρα Γκοϊνη, Ανδρέα Τσόλια (ορισθέντα με την υπ' αριθμ. 44/2008 πράξη του Προέδρου του Αρείου Πάγου) και Ελευθέριο Μάλλιο, Αρεοπαγίτες.

Συνήλθε σε δημόσια συνεδρίαση στο Κατάστημά του στις 9 Απριλίου 2008, με την παρουσία του Αντεισαγγελέα του Αρείου Πάγου Στέλιου Γκρόζου (γιατί κωλύεται ο Εισαγγελέας) και της Γραμματέως Χριστίνας Σταυροπούλου, για να δικάσει την αίτηση των αναιρεσειόντων - κατηγορουμένων: 1. Χ1 , που εκπροσωπήθηκε από τους πληρεξουσίους δικηγόρους του Αθανάσιο Ζαχαριάδη και Ευστάθιο Γκότση και 2. Χ2, που εκπροσωπήθηκε από τον πληρεξούσιο δικηγόρο του Αθανάσιο Ζαχαριάδη, περί αναιρέσεως της 3652/2006 αποφάσεως του Τριμελούς Εφετείου Θεσσαλονίκης. Με πολιτικώς ενάγοντα τον Ψ1, δικηγόρο, που παραστάθηκε αυτοπροσώπως.
Το Τριμελές Εφετείο Θεσσαλονίκης, με την ως άνω απόφασή του διέταξε όσα λεπτομερώς αναφέρονται σ' αυτή, και οι αναιρεσείοντες - κατηγορούμενοι ζητούν την αναίρεση αυτής, για τους λόγους που αναφέρονται στην από 3 Δεκεμβρίου 2007 αίτησή τους αναιρέσεως, η οποία καταχωρίστηκε στο οικείο πινάκιο με τον αριθμό 2086/2007.

Αφού άκουσε Τους πληρεξούσιους δικηγόρους των αναιρεσειόντων, καθώς και τον πολιτικώς ενάγοντα με την ιδιότητα του δικηγόρου, που ζήτησαν όσα αναφέρονται στα σχετικά πρακτικά και τον Αντεισαγγελέα, που πρότεινε να γίνει δεκτή η προκείμενη αίτηση αναίρεσης.

ΣΚΕΦΘΗΚΕ ΣΥΜΦΩΝΑ ΜΕ ΤΟ ΝΟΜΟ
Η αξιόποινη πράξη της δυσφημήσεως περιλαμβάνει, σύμφωνα με το άρθρο 362 Π.Κ., αντικειμενικώς με τον υπό του δράστη ισχυρισμό ενώπιον τρίτου ή διάδοση με οποιοδήποτε τρόπο για κάποιον άλλον γεγονότος, δυναμένου να βλάψει την τιμή ή την υπόληψη αυτού υποκειμενικώς δε τη γνώση του δράστη, ότι το ισχυριζόμενο ή διαδιδόμενο γεγονός είναι κατάλληλο να βλάψει την τιμή ή την υπόληψη άλλου και τη θέληση όπως ισχυρισθεί ενώπιον τρίτου ή διαδώσει το τοιούτο βλαπτικό γεγονός. Εξάλλου, για τη στοιχειοθέτηση της υπό του άρθρου 363 του ιδίου κώδικα προβλεπομένης αξιόποινης πράξεως της συκοφαντικής δυσφημήσεως απαιτείται, επί πλέον των αναφερθέντων στοιχείων, όπως το ως άνω γεγονός, το οποίο ισχυρίσθηκε ή διέδωσε ο δράστης, είναι ψευδές και αυτός να τελεί σε γνώση της αναληθείας του. Ο ισχυρισμός ή η διάδοση του δυσφημιστικού γεγονότος, μπορεί να γίνει και δια του τύπου, οπότε υπάρχει για τους υπευθύνους του εντίμου έγκλημα απλής ή συκοφαντικής δυσφήμησης δια του τύπου, το οποίο, μετά την κατάργηση με το άρθρο μόνο του Ν. 2243/1994 (που ισχύει από της 30.10.1994) όλων των ειδικών περί τύπου διατάξεων, συντελείται από τις ίδιες ακριβώς προϋποθέσεις που απαιτούνται για την απλή και συκοφαντική δυσφήμηση. Στην προκειμένη περίπτωση, με την προσβαλλόμενη απόφαση, όπως εξ αυτής προκύπτει, το Τριμελές Εφετείο Θεσσαλονίκης κήρυξε ενόχους τους αναιρεσείοντες, για το ότι "στη ..... στις .... ισχυρίστηκαν για άλλον εν γνώσει τους ψευδές γεγονός που μπορούσε να βλάψει την τιμή και υπόληψή του. Συγκεκριμένα ο μεν Χ1 ως εκδότης και διευθυντής της εφημερίδας ..... - ..., που κυκλοφόρησε στη ...., ο δε Χ2, ως συντάκτης της εν λόγω εφημερίδας καταχώρισαν άρθρο στο φύλλο της ....., στο οποίο αναφέρονταν ότι "υπάρχουν πολλά ερωτηματικά για την πρώην διοίκηση του Δήμου ...., καθώς χάθηκαν 330.000 ευρώ για τα σχολεία της ....., η προηγούμενη διοίκηση του Δήμου ..... εισέπραξε τα χρήματα προκειμένου να καλύψει τις λειτουργικές ανάγκες των σχολείων της περιοχής, πλην όμως δεν το έπραξε για το έτος 2002". Με τον τρόπο αυτό άφηναν να γίνει δεκτό με τη λογική ότι ο εγκαλών Ψ1, ως Δήμαρχος ....., το διάστημα από 1.1.1998 έως 31.12.2002 δεν διέθεσε το ποσό των 330.000 ευρώ για τις σχολικές ανάγκες της περιφέρειας του Δήμου .... το έτος 2002, μολονότι αυτά είχαν εισπραχθεί για το λόγο αυτό. Τα όσα δε υποστήριξαν στο προαναφερθέν άρθρο και υπέπεσαν στην αντίληψη του αναγνωστικού κοινού της εν λόγω εφημερίδας ήταν εν γνώσει τους ψευδή, καθώς η αλήθεια ήταν ότι ο εγκαλών με την προαναφερθείσα ιδιότητα του διέθεσε όλα τα κονδύλια που δόθηκαν στο Δήμο ..... για τις σχολικές ανάγκες της περιφέρειας του, χωρίς να αφήσει μέρος αυτών αδιάθετο και μπορούσαν να επιφέρουν μείωση στην τιμή και υπόληψη του εγκαλούντος, καθώς ήταν αντίθετα στην ευπρέπεια και ηθική". Εξάλλου, το Εφετείο αιτιολογώντας την καταδικαστική του κρίση, διέλαβε στο σκεπτικό της αποφάσεως ότι από τα εκτιθέμενα σ' αυτό πραγματικά περιστατικά πλήρως αποδείχθηκαν τα κατά το νόμο στοιχεία για την στοιχειοθέτηση της αντικειμενικής και υποκειμενικής υπόστασης επί αποδιδόμενης στους κατηγορουμένους πράξης της απλής δυσφήμησης δια του τύπου και πρέπει να κηρυχθούν ένοχοι κατά το διατακτικό. Έτσι όμως δεν προκύπτει σαφώς και ορισμένως για ποία αξιόποινη πράξη κατεδίκασε το εφετείο τους αναιρεσείοντες, δηλαδή γι' αυτή της απλής δυσφημήσεως ή για εκείνη της συκοφαντικής δυσφημήσεως.
Συνεπώς καθίσταται ανέφικτος ο ακυρωτικός έλεγχος, αν στην προκειμένη περίπτωση τα υπό του δικάσαντος δικαστηρίου γενόμενα δεκτά περιστατικά, κατά την περί πραγμάτων ανέλεγκτη κρίση του, υπήχθησαν ορθώς ή όχι στο νόμο και έτσι η προσβαλλόμενη απόφαση, λόγω της ασάφειας αυτής, στερείται νόμιμης βάσης και υπέπεσε στην πλημμέλεια του άρθρου 510 παρ. 1 στοιχ. Ε' ΚΠΔ. Επομένως ο από τη διάταξη αυτή δεύτερος λόγος της αναιρέσεως είναι βάσιμος και γι' αυτό πρέπει να γίνει δεκτή η κρινόμενη αίτηση, να αναιρεθεί εξ ολοκλήρου η ως άνω απόφαση και να παραπεμφθεί (άρθρο 519 ΚΠΔ) η υπόθεση για νέα συζήτηση στο ίδιο δικαστήριο, συγκροτούμενο από άλλους δικαστές, εκτός από εκείνους που δίκασαν προηγουμένως.

ΓΙΑ ΤΟΥΣ ΛΟΓΟΥΣ ΑΥΤΟΥΣ
Αναιρεί την 3652/2006 απόφαση του Τριμελούς Εφετείου Θεσσαλονίκης. Και
Παραπέμπει την υπόθεση για νέα συζήτηση στο ίδιο δικαστήριο, το οποίο θα συγκροτηθεί από άλλους δικαστές, εκτός από εκείνους που δίκασαν προηγουμένως.
Κρίθηκε και αποφασίσθηκε στην Αθήνα στις 30 Απριλίου 2008. Και
Δημοσιεύθηκε στην Αθήνα, σε δημόσια συνεδρίαση στο ακροατήριό του, στις 2 Ιουνίου 2008.

Figure C.1. *LexRank algorithm on test case 1.*

Biased LexRank algorithm

ΤΟ ΔΙΚΑΣΤΗΡΙΟ ΤΟΥ ΑΡΕΙΟΥ ΠΑΓΟΥ

Ζ' ΠΟΙΝΙΚΟ ΤΜΗΜΑ

Συγκροτήθηκε από τους Δικαστές: Γρηγόριο Μάμαλη, Προεδρεύοντα Αρεοπαγίτη (κωλυομένου του Αντιπροέδρου του Αρείου Πάγου Μιχαήλ Δέτση), ως αρχαιότερο μέλος της συνθέσεως, Αλέξανδρο Νικάκη (ορισθέντα με την υπ' αριθμ. 30/2008 πράξη του Προέδρου του Αρείου Πάγου) - Εισηγητή, Θεοδώρα Γκοΐνη, Ανδρέα Τσόλια (ορισθέντα με την υπ' αριθμ. 44/2008 πράξη του Προέδρου του Αρείου Πάγου) και Ελευθέριο Μάλλιο, Αρεοπαγίτες.

Συνήλθε σε δημόσια συνεδρίαση στο Κατάστημά του στις 9 Απριλίου 2008, με την παρουσία του Αντεισαγγελέα του Αρείου Πάγου Στέλιου Γκρόζου (γιατί κωλύεται ο Εισαγγελέας) και της Γραμματέως Χριστίνας Σταυροπούλου, για να δικάσει την αίτηση των αναιρεσειόντων - κατηγορουμένων: 1. Χ1 , που εκπροσωπήθηκε από τους πληρεξουσίους δικηγόρους του Αθανάσιο Ζαχαριάδη και Ευστάθιο Γκότση και 2. Χ2, που εκπροσωπήθηκε από τον πληρεξούσιο δικηγόρο του Αθανάσιο Ζαχαριάδη, περί αναιρέσεως της 3652/2006 αποφάσεως του Τριμελούς Εφετείου Θεσσαλονίκης. Με πολιτικώς ενάγοντα τον Ψ1, δικηγόρο, που παραστάθηκε αυτοπροσώπως.
Το Τριμελές Εφετείο Θεσσαλονίκης, με την ως άνω απόφασή του διέταξε όσα λεπτομερώς αναφέρονται σ' αυτή, και οι αναιρεσείοντες - κατηγορούμενοι ζητούν την αναίρεση αυτής, για τους λόγους που αναφέρονται στην από 3 Δεκεμβρίου 2007 αίτησή τους αναιρέσεως, η οποία καταχωρίστηκε στο οικείο πινάκιο με τον αριθμό 2 086/2007.

Αφού άκουσε Τους πληρεξούσιους δικηγόρους των αναιρεσειόντων, καθώς και τον πολιτικώς ενάγοντα με την ιδιότητα του δικηγόρου, που ζήτησαν όσα αναφέρονται στα σχετικά πρακτικά και τον Αντεισαγγελέα, που πρότεινε να γίνει δεκτή η προκείμενη αίτηση αναίρεσης.

ΣΚΕΦΘΗΚΕ ΣΥΜΦΩΝΑ ΜΕ ΤΟ ΝΟΜΟ
Η αξιόποινη πράξη της δυσφημήσεως περιλαμβάνει, σύμφωνα με το άρθρο 362 Π.Κ., αντικειμενικώς με τον υπό του δράστη ισχυρισμό ενώπιον τρίτου ή διάδοση με οποιοδήποτε τρόπο για κάποιον άλλον γεγονότος, δυναμένου να βλάψει την τιμή ή την υπόληψη αυτού υποκειμενικώς δε τη γνώση του δράστη, ότι το ισχυριζόμενο ή διαδιδόμενο γεγονός είναι κατάλληλο να βλάψει την τιμή ή την υπόληψη άλλου και τη θέληση όπως ισχυρισθεί ενώπιον τρίτου ή διαδώσει το τοιούτο βλαπτικό γεγονός. Εξάλλου, για τη στοιχειοθέτηση της υπό του άρθρου 363 του ιδίου κώδικα προβλεπομένης αξιόποινης πράξεως της συκοφαντικής δυσφημήσεως απαιτείται, επί πλέον των αναφερθέντων στοιχείων, όπως το ως άνω γεγονός, το οποίο ισχυρίσθηκε ή διέδωσε ο δράστης, είναι ψευδές και αυτός να τελεί σε γνώση της αναληθείας του. Ο ισχυρισμός ή η διάδοση του δυσφημιστικού γεγονότος, μπορεί να γίνει και δια του τύπου, οπότε υπάρχει για τους υπευθύνους του εντίμου έγκλημα απλής ή συκοφαντικής δυσφήμησης δια του τύπου, το οποίο, μετά την κατάργηση με το άρθρο μόνο του Ν. 2243/1994 (που ισχύει από της 30.10.1994) όλων των ειδικών περί τύπου διατάξεων, συντελείται από τις ίδιες ακριβώς προϋποθέσεις που απαιτούνται για την απλή και συκοφαντική δυσφήμηση. Στην προκειμένη περίπτωση, με την προσβαλλόμενη απόφαση, όπως εξ αυτής προκύπτει, το Τριμελές Εφετείο Θεσσαλονίκης κήρυξε ενόχους τους αναιρεσείοντες, για το ότι "στη ..... στις .... ισχυρίστηκαν για άλλον εν γνώσει τους ψευδές γεγονός που μπορούσε να βλάψει την τιμή και υπόληψή του. Συγκεκριμένα ο μεν Χ1 ως εκδότης και διευθυντής της εφημερίδας ..... - ..., που κυκλοφόρησε στη ...., ο δε Χ2, ως συντάκτης της εν λόγω εφημερίδας καταχώρισαν άρθρο στο φύλλο της ....., στο οποίο αναφέρονταν ότι "υπάρχουν πολλά ερωτηματικά για την πρώην διοίκηση του Δήμου ...., καθώς χάθηκαν 330.000 ευρώ για τα σχολεία της ....., η προηγούμενη διοίκηση του Δήμου ..... εισέπραξε τα χρήματα προκειμένου να καλύψει τις λειτουργικές ανάγκες των σχολείων της περιοχής, πλην όμως δεν το έπραξε για το έτος 2002". Με τον τρόπο αυτό άφηναν να γίνει δεκτό με τη λογική ότι ο εγκαλών Ψ1, ως Δήμαρχος ....., το διάστημα από 1.1.1998 έως 31.12.2002 δεν διέθεσε το ποσό των 330.000 ευρώ για τις σχολικές ανάγκες της περιφέρειας του Δήμου .... το έτος 2002, μολονότι αυτά είχαν εισπραχθεί για το λόγο αυτό. Τα όσα δε υποστήριξαν στο προαναφερθέν άρθρο και υπέπεσαν στην αντίληψη του αναγνωστικού κοινού της εν λόγω εφημερίδας ήταν εν γνώσει τους ψευδή, καθώς η αλήθεια ήταν ότι ο εγκαλών με την προαναφερθείσα ιδιότητα του διέθεσε όλα τα κονδύλια που δόθηκαν στο Δήμο ..... για τις σχολικές ανάγκες της περιφέρειας του, χωρίς να αφήσει μέρος αυτών αδιάθετο και μπορούσαν να επιφέρουν μείωση στην τιμή και υπόληψη του εγκαλούντος, καθώς ήταν αντίθετα στην ευπρέπεια και ηθική". Εξάλλου, το Εφετείο αιτιολογώντας την καταδικαστική του κρίση, διέλαβε στο σκεπτικό της αποφάσεως ότι από τα εκτιθέμενα σ' αυτό πραγματικά περιστατικά πλήρως αποδείχθηκαν τα κατά το νόμο στοιχεία για την στοιχειοθέτηση της αντικειμενικής και υποκειμενικής υπόστασης επί αποδιδόμενης στους κατηγορουμένους πράξης της απλής δυσφήμησης δια του τύπου και πρέπει να κηρυχθούν ένοχοι κατά το διατακτικό. Έτσι όμως δεν προκύπτει σαφώς και ορισμένως για ποία αξιόποινη πράξη κατεδίκασε το εφετείο τους αναιρεσείοντες, δηλαδή γι' αυτή της απλής δυσφημήσεως ή για εκείνη της συκοφαντικής δυσφημήσεως.
Συνεπώς καθίσταται ανέφικτος ο ακυρωτικός έλεγχος, αν στην προκειμένη περίπτωση τα υπό του δικάσαντος δικαστηρίου γενόμενα δεκτά περιστατικά, κατά την περί πραγμάτων ανέλεγκτη κρίση του, υπήχθησαν ορθώς ή όχι στο νόμο και έτσι η προσβαλλόμενη απόφαση, λόγω της ασάφειας αυτής, στερείται νόμιμης βάσης και υπέπεσε στην πλημμέλεια του άρθρου 510 παρ. 1 στοιχ. Ε' ΚΠΔ. Επομένως ο από τη διάταξη αυτή δεύτερος λόγος της αναιρέσεως είναι βάσιμος και γι' αυτό πρέπει να γίνει δεκτή η κρινόμενη αίτηση, να αναιρεθεί εξ ολοκλήρου η ως άνω απόφαση και να παραπεμφθεί (άρθρο 519 ΚΠΔ) η υπόθεση για νέα συζήτηση στο ίδιο δικαστήριο, συγκροτούμενο από άλλους δικαστές, εκτός από εκείνους που δίκασαν προηγουμένως.

ΓΙΑ ΤΟΥΣ ΛΟΓΟΥΣ ΑΥΤΟΥΣ
Αναιρεί την 3652/2006 απόφαση του Τριμελούς Εφετείου Θεσσαλονίκης. Και
Παραπέμπει την υπόθεση για νέα συζήτηση στο ίδιο δικαστήριο, το οποίο θα συγκροτηθεί από άλλους δικαστές, εκτός από εκείνους που δίκασαν προηγουμένως.
Κρίθηκε και αποφασίσθηκε στην Αθήνα στις 30 Απριλίου 2008. Και
Δημοσιεύθηκε στην Αθήνα, σε δημόσια συνεδρίαση στο ακροατήριό του, στις 2 Ιουνίου 2008.

Figure C.2. *Biased LexRank algorithm on test case 1.*

C.2.2 Test case - 3

LexRank algorithm

ΤΟ ΔΙΚΑΣΤΗΡΙΟ ΤΟΥ ΑΡΕΙΟΥ ΠΑΓΟΥ

ΣΤ' ΠΟΙΝΙΚΟ ΤΜΗΜΑ - ΣΕ ΣΥΜΒΟΥΛΙΟ

Συγκροτήθηκε από τους Δικαστές: Γεώργιο Σαραντινό, Αντιπρόεδρο Αρείου Πάγου, Βασίλειο Λυκούδη και Ανδρέα Τσόλια - Εισηγητή, Αρεοπαγίτες.
Με την παρουσία και του Αντεισαγγελέα του Αρείου Πάγου Βασιλείου Μαρκή (γιατί κωλύεται ο Εισαγγελέας) και της Γραμματέως Πελαγίας Λόζιου .

Συνήλθε σε Συμβούλιο στο Κατάστημά του στις 4 Δεκεμβρίου 2007, προκειμένου να αποφανθεί για την αίτηση της αναιρεσείουσας - κατηγορουμένης Χ1, που δεν παραστάθηκε στο συμβούλιο, περί αναιρέσεως του υπ' αριθμ. 275/2007 βουλεύματος του Συμβουλίου Εφετών Αθηνών. Το Συμβούλιο Εφετών Αθηνών, με το ως άνω βούλευμά του διέταξε όσα λεπτομερώς αναφέρονται σ' αυτό, και η αναιρεσείουσα - κατηγορούμενη ζητεί τώρα την αναίρεση του βουλεύματος τούτου, για τους λόγους που αναφέρονται στις από 3 και 8 Μαρτίου 2007 αιτήσεις της αναιρέσεως, οι οποίες καταχωρίστηκαν στο οικείο πινάκιο με τον αριθμό 426/2007.
Έπειτα ο Αντεισαγγελέας του Αρείου Πάγου Βασίλειος Μαρκής εισήγαγε για κρίση στο Συμβούλιο τη σχετική δικογραφία με την πρόταση του Αντεισαγγελέα του Αρείου Πάγου Στέλιου Γκρόζου με αριθμό 274/29.6.2007, στην οποία αναφέρονται τα ακόλουθα:
Εισάγω, σύμφωνα με το άρθρο 476 παρ.1 Κ.Π.Δ.: α) την υπ'αριθ. 64/8-3-2007 αίτηση αναιρέσεως της κατηγορουμένης Χ1, η οποία ασκήθηκε στο όνομα και για λογαριασμό της από τον δικηγόρο Αθηνών Ιωάννη Βαρουτά, δυνάμει της από 5-3-2007 προσαρτημένης στην αίτηση και νομίμως θεωρημένης εξουσιοδοτήσεως και στρέφεται κατά του υπ'αριθ. 275/2007 βουλεύματος του Συμβουλίου Εφετών Αθηνών και β) την από 3-3-2007 ταυτόσημη αίτηση της ίδιας αναιρεσείουσας Χ1 προς τον Εισαγγελέα του Αρείου Πάγου, για αναίρεση επίσης του ανωτέρω υπ'αριθ. 275/2007 βουλεύματος του Συμβουλίου Εφετών Αθηνών, εκθέτω δε τα ακόλουθα:
Κατά τη διάταξη του άρθρου 463 Κ.Π.Δ. ένδικο μέσο μπορεί να ασκήσει μόνο εκείνος που ο νόμος του δίνει ρητά αυτό το δικαίωμα, κατά δε το άρθρο 476 παρ.1 του ίδιου Κώδικα, το ένδικο μέσο απορρίπτεται ως απαράδεκτο, εκτός των άλλων περιπτώσεων που αναφέρονται στη διάταξη αυτή και όταν ασκήθηκε εναντίον βουλεύματος, για το οποίο δεν προβλέπεται. Εξάλλου από τη διάταξη του άρθρου 476 παρ.2 του αυτού ως άνω Κώδικα, που πριν από την τροποποίησή της με το άρθρο 38 Ν. 3160/2003 όριζε ότι "κατά της απόφασης ή του βουλεύματος που απορρίπτει το ένδικο μέσο ως απαράδεκτο επιτρέπεται μόνο αναίρεση" και μετά την εν λόγω τροποποίηση απαλείφθηκε η φράση "ή του βουλεύματος", προκύπτει ότι δεν προβλέπεται πλέον αναίρεση κατά του βουλεύματος που απορρίπτει το ένδικο μέσο ως απαράδεκτο (ΑΠ 776/2006, ΑΠ 297/2006). Πρέπει στο σημείο αυτό να τονισθεί ότι η ανωτέρω διάταξη του άρθρου 476 παρ.2 Κ.Π.Δ., που, μετά την αντικατάστασή της με το άρθρο 38 Ν. 3160/2003, περιορίζει το δικαίωμα ασκήσεως αναιρέσεως εναντίον βουλεύματος που απορρίπτει την έφεση ως απαράδεκτη, συμπορεύεται με τα άρθρα 4 παρ.1 και 20 παρ.1 του Συντάγματος, αφού ούτε άνιση ρύθμιση περιέχει, που να συνεπάγεται δυσμενή μεταχείριση ορισμένων διαδίκων, ούτε στερεί τον διάδικο από το δικαίωμα παροχής έννομης προστασίας από τα δικαστήρια, δοθέντος μάλιστα και του ότι ο κοινός νομοθέτης δεν υποχρεούται από το Σύνταγμα να θεσπίζει ένδικα μέσα κατά των αποφάσεων και, ως εκ τούτου, ο διάδικος μπορεί και οφείλει να υπολογίζει, κατά την εκδίκαση της υποθέσεώς του, ότι είναι ενδεχόμενο η μέλλουσα να εκδοθεί απόφαση να μην υπόκειται σε ένδικα μέσα, παρά το γεγονός ότι αυτά επιτρέπονταν κατά την έναρξη και κατά τη διάρκεια της δίκης (ή της ποινικής διώξεως). Όπου κρίθηκε αναγκαία η καθιέρωση ενδίκου μέσου υπήρξε ρητή αποτύπωση της βουλήσεως του συντακτικού νομοθέτη με τις ειδικές προβλέψεις των άρθρων 95 παρ.1 β'(αναίρεση τελεσιδίκων αποφάσεων των διοικητικών δικαστηρίων για υπέρβαση εξουσίας ή παράβαση νόμου) και 96 παρ.2 Συντάγματος (έφεση στο αρμόδιο τακτικό δικαστήριο κατά των αποφάσεων αστυνομικών αρχών και αρχών αγροτικής ασφάλειας). Δεν είναι δε αντίθετες οι ανωτέρω διατάξεις ούτε και προς το άρθρο 6 παρ.1 της ΕΣΔΑ, που κατοχυρώνει την αρχή της δίκαιης δίκης, από την οποία δεν συνάγεται υποχρέωση του εθνικού νομοθέτη για καθιέρωση ενδίκων μέσων, αλλά ούτε και στο άρθρο 26 του Συντάγματος, το οποίο καθιερώνει την αρχή της διακρίσεως των εξουσιών (ΑΠ 1486/2005).
Στην προκειμένη περίπτωση από τα έγγραφα της δικογραφίας, τα οποία επιτρεπτώς επισκοπεί ο Άρειος Πάγος για την έρευνα του παραδεκτού της αιτήσεως αναιρέσεως, προκύπτουν τα εξής: Το Τριμελές Πλημμελειοδικείο Αθηνών με την υπ'αριθ. 8969/7-2-2005 απόφασή του κήρυξε ένοχη την αναιρεσείουσα κατηγορουμένη Χ1 για τις πράξεις της παραβάσεως του Ν. 2971/2001 και του Ν. 1337/1983 και της επέβαλε συνολική ποινή φυλακίσεως ενός (1) έτους και τριών (3) μηνών. Κατά της αποφάσεως αυτής η αναιρεσείουσα άσκησε έφεση, η οποία κηρύχθηκε απαράδεκτη με το προσβαλλόμενο υπ'αριθ. 275/2007 βούλευμα του Συμβουλίου Εφετών Αθηνών. Κατά του ως άνω όμως βουλεύματος δεν επιτρέπεται, σύμφωνα με τα προεκτεθέντα, να ασκηθεί αναίρεση.
Με τα δεδομένα αυτά καθίσταται φανερό ότι η αναιρεσείουσα άσκησε αναίρεση κατά βουλεύματος, εναντίον του οποίου δεν επιτρέπεται να ασκηθεί τέτοιο ένδικο μέσο, δηλαδή άσκησε μη επιτρεπόμενο σ'αυτήν ένδικο μέσο. Επομένως οι κρινόμενες αιτήσεις αναιρέσεως πρέπει, κατ'εφαρμογή της διατάξεως του άρθρου 476 παρ.1 Κ.Π.Δ., να απορριφθούν ως απαράδεκτες και να καταδικασθεί η αναιρεσείουσα στα δικαστικά έξοδα.
Ανεξάρτητα από αυτά η από 3-3-2007 αίτηση αναιρέσεως, που ασκήθηκε με δήλωση στον Εισαγγελέα του Αρείου Πάγου, πρέπει να απορριφθεί ως απαράδεκτη και για τον εξής επιπρόσθετο λόγο: Από τις διατάξεις των άρθρων 473 παρ.2 και 474 παρ.1 Κ.Π.Δ. προκύπτει ότι, κατά γενική αρχή, η αίτηση αναιρέσεως ασκείται με δήλωση του δικαιουμένου διαδίκου ενώπιον των οριζομένων από την τελευταία οργάνων, στα οποία δεν περιλαμβάνεται και ο Εισαγγελέας του Αρείου Πάγου. Η κατεξαίρεση άσκηση αναιρέσεως με δήλωση που επιδίδεται στον Εισαγγελέα του Αρείου Πάγου μπορεί να γίνει μόνο εναντίον καταδικαστικής αποφάσεως, όχι δε και εναντίον οποιασδήποτε άλλης, η οποία δεν έχει αυτόν τον χαρακτήρα (ΑΠ 578/2005, ΑΠ 295/2001), όπως είναι και η απόφαση, με την οποία απορρίπτεται, η έφεση ως απαράδεκτη, πράγμα που συμβαίνει στην κρινόμενη υπόθεση.

Figure C.3. *LexRank algorithm on test case 3.*

ΓΙΑ ΤΟΥΣ ΛΟΓΟΥΣ ΑΥΤΟΥΣ

Προτείνω:

Α) Να απορριφθούν ως απαράδεκτες: α) η υπ'αριθ. 64/8-3-2007 αίτηση αναιρέσεως της Χ1 κατά του υπ'αριθ. 275/2007 βουλεύματος του Συμβουλίου Εφετών Αθηνών και β) η από 3-3-2007 ταυτόσημη αίτηση της ίδιας αναιρεσείουσας Χ1 προς τον Εισαγγελέα του Αρείου Πάγου, για αναίρεση επίσης του ανωτέρω υπ'αριθ. 275/07 βουλεύματος του Συμβουλίου Εφετών Αθηνών. Και

Β) Να καταδικασθεί η αναιρεσείουσα στα δικαστικά έξοδα.

Αθήνα, 9-5-2007

Ο Αντεισαγγελέας του Αρείου Πάγου Στέλιος Κ. Γκρόζος

Αφού άκουσε τον Αντεισαγγελέα, που αναφέρθηκε στην παραπάνω εισαγγελική πρόταση και έπειτα αποχώρησε,

ΣΚΕΦΘΗΚΕ ΣΥΜΦΩΝΑ ΜΕ ΤΟ ΝΟΜΟ

ΕΠΕΙΔΗ, σύμφωνα με τη διάταξη του άρθρου 463 εδ. α' του Κώδικα Ποινικής Δικονομίας "ένδικο μέσο μπορεί να ασκήσει μόνο εκείνος, που ο νόμος του δίνει ρητά αυτό το δικαίωμα". Επομένως, αν με κάποια διάταξη καθορίζονται ορισμένα πρόσωπα, ως δικαιούμενα, εξ αντιδιαστολής προκύπτει ότι δεν δικαιούνται και άλλα, μη μνημονευόμενα πρόσωπα. Περαιτέρω, κατά το άρθρο 476 παρ. 1 του ίδιου Κώδικα, το ένδικο μέσο απορρίπτεται ως απαράδεκτο, εκτός των άλλων περιπτώσεων που αναφέρονται στη διάταξη αυτή και όταν ασκήθηκε εναντίον βουλεύματος, για το οποίο δεν προβλέπεται. Εξάλλου, από τη διάταξη του άρθρου 476 παρ. 2 του αυτού ως άνω Κώδικα, που πριν από την τροποποίησή της με το άρθρο 38 του Ν. 3160/2003 όριζε ότι "κατά της απόφασης ή του βουλεύματος που απορρίπτει το ένδικο μέσο ως απαράδεκτο επιτρέπεται μόνο αναίρεση" και μετά την εν λόγω τροποποίηση απαλείφθηκε η φράση "ή του βουλεύματος", προκύπτει ότι δεν προβλέπεται πλέον αναίρεση κατά του βουλεύματος που απορρίπτει το ένδικο μέσο ως απαράδεκτο. Στο σημείο αυτό πρέπει να λεχθεί ότι η παραπάνω διάταξη του άρθρου 476 παρ. 2 Κ.Π.Δ., που, όπως σήμερα ισχύει, περιορίζει το δικαίωμα ασκήσεως αναιρέσεως κατά του βουλεύματος που απορρίπτει την έφεση ως απαράδεκτη, δεν αντίκειται στα άρθρα παρ. 1 και 20 παρ. 1 του Συντάγματος, αφού δεν περιέχει άνιση ρύθμιση, που να συνεπάγεται δυσμενή μεταχείριση κάποιων διαδίκων, αλλ' ούτε στερεί τον διάδικο από το δικαίωμα παροχής έννομης προστασίας από τα δικαστήρια, αφού μάλιστα και ο κοινός νομοθέτης δεν υποχρεούται από το Σύνταγμα να θεσπίζει ένδικα μέσα κατά των αποφάσεων και, ως εκ τούτου, ο διάδικος μπορεί και οφείλει να υπολογίζει, κατά την εκδίκαση της υποθέσεώς του, ότι είναι ενδεχόμενο, η απόφαση που πρόκειται να εκδοθεί να μη προσβάλλεται με ένδικα μέσα, παρά το γεγονός ότι αυτά επιτρέπονταν κατά την έναρξη και κατά τη διάρκεια της δίκης (ή της ποινικής διώξεως). Όπου κρίθηκε αναγκαία η θέσπιση ένδικου μέσου, υπήρξε ρητή η αποτύπωση της βουλήσεως του συντακτικού νομοθέτη με τις ειδικές προβλέψεις των άρθρων 95 παρ. 1 β' (αναίρεση των τελεσίδικων αποφάσεων των διοικητικών δικαστηρίων για υπέρβαση εξουσίας ή παράβαση νόμου) και 96 παρ. 2 του Συντάγματος (έφεση στο αρμόδιο τακτικό δικαστήριο κατά των αποφάσεων αστυνομικών αρχών και αρχών αγροτικής ασφάλειας). Δεν είναι δε αντίθετες οι ανωτέρω διατάξεις ούτε και προς το άρθρο 6 παρ. 1 της ΕΣΔΑ, που κατοχυρώνει την αρχή της δίκαιης δίκης, από την οποία δεν συνάγεται υποχρέωση του εθνικού νομοθέτη για καθιέρωση ένδικων μέσων, αλλά ούτε και στο άρθρο 26 του συντάγματος, το οποίο καθιερώνει την αρχή της διακρίσεως των εξουσιών. Τέλος, από τις διατάξεις των άρθρων 473 παρ. 2 και 474 παρ. 1 του Κώδικα Ποινικής Δικονομίας συνάγεται ότι, κατά γενική αρχή, η αίτηση αναιρέσεως ασκείται με δήλωση του δικαιουμένου διαδίκου ενώπιον των οργάνων που ορίζονται από την τελευταία ως άνω διάταξη, μεταξύ των οποίων δεν περιλαμβάνεται και ο Εισαγγελεύς του Αρείου Πάγου. Η κατ' εξαίρεση άσκηση αναιρέσεως με δήλωση που επιδίδεται στον εισαγγελέα του Αρείου Πάγου μπορεί να γίνει μόνο εναντίον καταδικαστικής αποφάσεως, όχι δε και εναντίον οποιασδήποτε άλλης, η οποία δεν έχει αυτόν τον χαρακτήρα, όπως είναι και η απόφαση, με την οποία απορρίπτεται η έφεση ως απαράδεκτη.

Εν προκειμένω, από τα έγγραφα της δικογραφίας, τα οποία επιτρεπτώς επισκοπούνται από τον Άρειο Πάγο, για την έρευνα του παραδεκτού της αιτήσεως αναιρέσεως, προκύπτουν τα ακόλουθα: Το Τριμελές Πλημμελειοδικείο Αθηνών, με την 8969/7.2.2005 απόφασή του, κήρυξε ένοχη την αναιρεσείουσα κατηγορουμένη Χ1 για τις πράξεις της παραβάσεως του Ν. 2971/2001 και του Ν. 1337/1983 και της επέβαλε συνολική ποινή φυλακίσεως ενός έτους και τριών μηνών. Κατά της αποφάσεως αυτής, η αναιρεσείουσα άσκησε έφεση, η οποία κηρύχθηκε απαράδεκτη, με το 275/2007 βούλευμα του Συμβουλίου Εφετών Αθηνών, κατά του οποίου η κατηγορουμένη άσκησε τις κρινόμενες αιτήσεις αναιρέσεως. Σύμφωνα, όμως με όσα αναφέρονται παραπάνω στη μείζονα σκέψη, κατά του ως άνω βουλεύματος δεν επιτρέπεται να ασκηθεί το ένδικο μέσο της αναιρέσεως και επομένως, οι κρινόμενες αιτήσεις αναιρέσεως, που συνεκδικάζονται, λόγω της μεταξύ τους πρόδηλης συνάφειας, πρέπει, κατ' εφαρμογή της διατάξεως του άρθρου 476 παρ. 1 του Κώδικα Ποινικής δικονομίας, να απορριφθούν ως απαράδεκτες και να καταδικασθεί η αναιρεσείουσα στα δικαστικά έξοδα (άρθρο 583 παρ. 1 ΚΠΔ). Ανεξάρτητα από αυτά, η από 3 Μαρτίου 2007 αίτηση αναιρέσεως, η οποία ασκήθηκε με δήλωση προς τον Εισαγγελέα του Αρείου Πάγου, πρέπει να απορριφθεί ως απαράδεκτη, αφού σύμφωνα με τα ανωτέρω εκτιθέμενα, άσκηση αναιρέσεως με δήλωση, επιδιδόμενη στον Εισαγγελέα του Αρείου Πάγου μπορεί να γίνει μόνο εναντίον καταδικαστικής αποφάσεως, περίπτωση που δεν συντρέχει εν προκειμένω.

ΓΙΑ ΤΟΥΣ ΛΟΓΟΥΣ ΑΥΤΟΥΣ ΑΠΟΡΡΙΠΤΕΙ α) την 64/8.3.2007 αίτηση αναιρέσεως της Χ1 κατά του 275/2007 βουλεύματος του Συμβουλίου Εφετών Αθηνών και β) την από 3.3.2007 ταυτόσημη αίτηση της ίδιας αναιρεσείουσας Χ1 που ασκήθηκε με δήλωσή της προς τον εισαγγελέα του Αρείου Πάγου για αναίρεση επίσης του ανωτέρω 275/2007 βουλεύματος του Συμβουλίου Εφετών Αθηνών. Και ΚΑΤΑΔΙΚΑΖΕΙ την αναιρεσείουσα στα δικαστικά έξοδα, τα οποία ανέρχονται στο ποσό των διακοσίων είκοσι ευρώ (220€).

Κρίθηκε και αποφασίστηκε στην Αθήνα στις 14 Μαρτίου 2008. Και,

Εκδόθηκε στην Αθήνα στις 20 Μαΐου 2008.

Figure C.4. *(Cont.) LexRank algorithm on test case 3.*

Biased LexRank algorithm

ΤΟ ΔΙΚΑΣΤΗΡΙΟ ΤΟΥ ΑΡΕΙΟΥ ΠΑΓΟΥ

ΣΤ' ΠΟΙΝΙΚΟ ΤΜΗΜΑ - ΣΕ ΣΥΜΒΟΥΛΙΟ

Συγκροτήθηκε από τους Δικαστές: Γεώργιο Σαραντινό, Αντιπρόεδρο Αρείου Πάγου, Βασίλειο Λυκούδη και Ανδρέα Τσόλια - Εισηγητή, Αρεοπαγίτες.
Με την παρουσία και του Αντεισαγγελέα του Αρείου Πάγου Βασιλείου Μαρκή (γιατί κωλύεται ο Εισαγγελέας) και της Γραμματέως Πελαγίας Λόζιου .

Συνήλθε σε Συμβούλιο στο Κατάστημά του στις 4 Δεκεμβρίου 2007, προκειμένου να αποφανθεί για την αίτηση της αναιρεσείουσας - κατηγορουμένης Χ1, που δεν παραστάθηκε στο συμβούλιο, περί αναιρέσεως του υπ' αριθμ. 275/2007 βουλεύματος του Συμβουλίου Εφετών Αθηνών. Το Συμβούλιο Εφετών Αθηνών, με το ως άνω βούλευμά του διέταξε όσα λεπτομερώς αναφέρονται σ' αυτό, και η αναιρεσείουσα - κατηγορούμενη ζητεί τώρα την αναίρεση του βουλεύματος τούτου, για τους λόγους που αναφέρονται στις από 3 και 8 Μαρτίου 2007 αιτήσεις της αναιρέσεως, οι οποίες καταχωρίστηκαν στο οικείο πινάκιο με τον αριθμό 426/2007.
Έπειτα ο Αντεισαγγελέας του Αρείου Πάγου Βασίλειος Μαρκής εισήγαγε για κρίση στο Συμβούλιο τη σχετική δικογραφία με την πρόταση του Αντεισαγγελέα του Αρείου Πάγου Στέλιου Γκρόζου με αριθμό 274/29.6.2007, στην οποία αναφέρονται τα ακόλουθα:
Εισάγω, σύμφωνα με το άρθρο 476 παρ.1 Κ.Π.Δ.: α) την υπ'αριθ. 64/8-3-2007 αίτηση αναιρέσεως της κατηγορουμένης Χ1, η οποία ασκήθηκε στο όνομα και για λογαριασμό της από τον δικηγόρο Αθηνών Ιωάννη Βαρουτά, δυνάμει της από 5-3-2007 προσαρτημένης στην αίτηση και νομίμως θεωρημένης εξουσιοδοτήσεως και στρέφεται κατά του υπ'αριθ. 275/2007 βουλεύματος του Συμβουλίου Εφετών Αθηνών και β) την από 3-3-2007 ταυτόσημη αίτηση της ίδιας αναιρεσείουσας Χ1 προς τον Εισαγγελέα του Αρείου Πάγου, για αναίρεση επίσης του ανωτέρω υπ'αριθ. 275/2007 βουλεύματος του Συμβουλίου Εφετών Αθηνών, εκθέτω δε τα ακόλουθα:
Κατά τη διάταξη του άρθρου 463 Κ.Π.Δ. ένδικο μέσο μπορεί να ασκήσει μόνο εκείνος που ο νόμος του δίνει ρητά αυτό το δικαίωμα, κατά δε το άρθρο 476 παρ.1 του ίδιου Κώδικα, το ένδικο μέσο απορρίπτεται ως απαράδεκτο, εκτός των άλλων περιπτώσεων που αναφέρονται στη διάταξη αυτή και όταν ασκήθηκε εναντίον βουλεύματος, για το οποίο δεν προβλέπεται. Εξάλλου από τη διάταξη του άρθρου 476 παρ.2 του αυτού ως άνω Κώδικα, που πριν από την τροποποίησή της με το άρθρο 38 Ν. 3160/2003 όριζε ότι "κατά της απόφασης ή του βουλεύματος που απορρίπτει το ένδικο μέσο ως απαράδεκτο επιτρέπεται μόνο αναίρεση" και μετά την εν λόγω τροποποίηση απαλείφθηκε η φράση "ή του βουλεύματος", προκύπτει ότι δεν προβλέπεται πλέον αναίρεση κατά του βουλεύματος που απορρίπτει το ένδικο μέσο ως απαράδεκτο (ΑΠ 776/2006, ΑΠ 297/2006). Πρέπει στο σημείο αυτό να τονισθεί ότι η ανωτέρω διάταξη του άρθρου 476 παρ.2 Κ.Π.Δ., που, μετά την αντικατάστασή της με το άρθρο 38 Ν. 3160/2003, περιορίζει το δικαίωμα ασκήσεως αναιρέσεως εναντίον βουλεύματος που απορρίπτει την έφεση ως απαράδεκτη, συμπορεύεται με τα άρθρα 4 παρ.1 και 20 παρ.1 του Συντάγματος, αφού ούτε άνιση ρύθμιση περιέχει, που να συνεπάγεται δυσμενή μεταχείριση ορισμένων διαδίκων, ούτε στερεί τον διάδικο από το δικαίωμα παροχής έννομης προστασίας από τα δικαστήρια, δοθέντος μάλιστα και του ότι ο κοινός νομοθέτης δεν υποχρεούται από το Σύνταγμα να θεσπίζει ένδικα μέσα κατά των αποφάσεων και, ως εκ τούτου, ο διάδικος μπορεί και οφείλει να υπολογίζει, κατά την εκδίκαση της υποθέσεώς του, ότι είναι ενδεχόμενο η μέλλουσα να εκδοθεί απόφαση να μην υπόκειται σε ένδικα μέσα, παρά το γεγονός ότι αυτά επιτρέπονταν κατά την έναρξη και κατά τη διάρκεια της δίκης (ή της ποινικής διώξεως). Όπου κρίθηκε αναγκαία η καθιέρωση ενδίκου μέσου υπήρξε ρητή αποτύπωση της βουλήσεως του συντακτικού νομοθέτη με τις ειδικές προβλέψεις των άρθρων 95 παρ.1 β'(αναίρεση τελεσιδίκων αποφάσεων των διοικητικών δικαστηρίων για υπέρβαση εξουσίας ή παράβαση νόμου) και 96 παρ.2 Συντάγματος (έφεση στο αρμόδιο τακτικό δικαστήριο κατά των αποφάσεων αστυνομικών αρχών και αρχών αγροτικής ασφάλειας). Δεν είναι δε αντίθετες οι ανωτέρω διατάξεις ούτε και προς το άρθρο 6 παρ.1 της ΕΣΔΑ, που κατοχυρώνει την αρχή της δίκαιης δίκης, από την οποία δεν συνάγεται υποχρέωση του εθνικού νομοθέτη για καθιέρωση ενδίκων μέσων, αλλά ούτε και στο άρθρο 26 του Συντάγματος, το οποίο καθιερώνει την αρχή της διακρίσεως των εξουσιών (ΑΠ 1486/2005).
Σ την προκειμένη περίπτωση από τα έγγραφα της δικογραφίας, τα οποία επιτρεπτώς επισκοπεί ο Άρειος Πάγος για την έρευνα του παραδεκτού της αιτήσεως αναιρέσεως, προκύπτουν τα εξής: Το Τριμελές Πλημμελειοδικείο Αθηνών με την υπ'αριθ. 8969/7-2-2005 απόφασή του κήρυξε ένοχη την αναιρεσείουσα κατηγορουμένη Χ1 για τις πράξεις της παραβάσεως του Ν. 2971/2001 και του Ν. 1337/1983 και της επέβαλε συνολική ποινή φυλακίσεως ενός (1) έτους και τριών (3) μηνών. Κατά της αποφάσεως αυτής η αναιρεσείουσα άσκησε έφεση, η οποία κηρύχθηκε απαράδεκτη με το προσβαλλόμενο υπ'αριθ. 275/2007 βούλευμα του Συμβουλίου Εφετών Αθηνών. Κατά του ως άνω όμως βουλεύματος δεν επιτρέπεται, σύμφωνα με τα προεκτεθέντα, να ασκηθεί αναίρεση.
Με τα δεδομένα αυτά καθίσταται φανερό ότι η αναιρεσείουσα άσκησε αναίρεση κατά βουλεύματος, εναντίον του οποίου δεν επιτρέπεται να ασκηθεί τέτοιο ένδικο μέσο, δηλαδή άσκησε μη επιτρεπόμενο σ'αυτήν ένδικο μέσο. Επομένως οι κρινόμενες αιτήσεις αναιρέσεως πρέπει, κατ'εφαρμογή της διατάξεως του άρθρου 476 παρ.1 Κ.Π.Δ., να απορριφθούν ως απαράδεκτες και να καταδικασθεί η αναιρεσείουσα στα δικαστικά έξοδα.
Ανεξάρτητα από αυτά η από 3-3-2007 αίτηση αναιρέσεως, που ασκήθηκε με δήλωση στον Εισαγγελέα του Αρείου Πάγου, πρέπει να απορριφθεί ως απαράδεκτη και για τον εξής επιπρόσθετο λόγο: Από τις διατάξεις των άρθρων 473 παρ.2 και 474 παρ.1 Κ.Π.Δ. προκύπτει ότι, κατά γενική αρχή, η αίτηση αναιρέσεως ασκείται με δήλωση του δικαιουμένου διαδίκου ενώπιον των οριζομένων από την τελευταία οργάνων, στα οποία δεν περιλαμβάνεται και ο Εισαγγελέας του Αρείου Πάγου. Η κατεξαίρεση άσκηση αναιρέσεως με δήλωση που επιδίδεται στον Εισαγγελέα του Αρείου Πάγου μπορεί να γίνει μόνο εναντίον καταδικαστικής αποφάσεως, όχι δε και εναντίον οποιασδήποτε άλλης, η οποία δεν έχει αυτόν τον χαρακτήρα (ΑΠ 578/2005, ΑΠ 295/2001), όπως είναι και η απόφαση, με την οποία απορρίπτεται, η έφεση ως απαράδεκτη, πράγμα που συμβαίνει στην κρινόμενη υπόθεση.

Figure C.5. *Biased LexRank algorithm on test case 3.*

ΓΙΑ ΤΟΥΣ ΛΟΓΟΥΣ ΑΥΤΟΥΣ

Προτείνω:

Α) Να απορριφθούν ως απαράδεκτες: α) η υπ'αριθ. 64/8-3-2007 αίτηση αναιρέσεως της Χ1 κατά του υπ'αριθ. 275/2007 βουλεύματος του Συμβουλίου Εφετών Αθηνών και β) η από 3-3-2007 ταυτόσημη αίτηση της ίδιας αναιρεσείουσας Χ1 προς τον Εισαγγελέα του Αρείου Πάγου, για αναίρεση επίσης του ανωτέρω υπ'αριθ. 275/07 βουλεύματος του Συμβουλίου Εφετών Αθηνών. Και Β) Να καταδικασθεί η αναιρεσείουσα στα δικαστικά έξοδα.

Αθήνα, 9-5-2007

Ο Αντεισαγγελέας του Αρείου Πάγου Στέλιος Κ. Γκρόζος

Αφού άκουσε τον Αντεισαγγελέα, που αναφέρθηκε στην παραπάνω εισαγγελική πρόταση και έπειτα αποχώρησε,

ΣΚΕΦΘΗΚΕ ΣΥΜΦΩΝΑ ΜΕ ΤΟ ΝΟΜΟ

ΕΠΕΙΔΗ, σύμφωνα με τη διάταξη του άρθρου 463 εδ. α' του Κώδικα Ποινικής Δικονομίας "ένδικο μέσο μπορεί να ασκήσει μόνο εκείνος, που ο νόμος του δίνει ρητά αυτό το δικαίωμα". Επομένως, αν με κάποια διάταξη καθορίζονται ορισμένα πρόσωπα, ως δικαιούμενα, εξ αντιδιαστολής προκύπτει ότι δεν δικαιούνται και άλλα, μη μνημονευόμενα πρόσωπα. Περαιτέρω, κατά το άρθρο 476 παρ. 1 του ίδιου Κώδικα, το ένδικο μέσο απορρίπτεται ως απαράδεκτο, εκτός των άλλων περιπτώσεων που αναφέρονται στη διάταξη αυτή και όταν ασκήθηκε εναντίον βουλεύματος, για το οποίο δεν προβλέπεται. Εξάλλου, από τη διάταξη του άρθρου 476 παρ. 2 του αυτού ως άνω Κώδικα, που πριν από την τροποποίησή της με το άρθρο 38 του Ν. 3160/2003 όριζε ότι "κατά της απόφασης ή του βουλεύματος που απορρίπτει το ένδικο μέσο ως απαράδεκτο επιτρέπεται μόνο αναίρεση" και μετά την εν λόγω τροποποίηση απαλείφθηκε η φράση "ή του βουλεύματος", προκύπτει ότι δεν προβλέπεται πλέον αναίρεση κατά του βουλεύματος που απορρίπτει το ένδικο μέσο ως απαράδεκτο. Στο σημείο αυτό πρέπει να λεχθεί ότι η παραπάνω διάταξη του άρθρου 476 παρ. 2 Κ.Π.Δ., που, όπως σήμερα ισχύει, περιορίζει το δικαίωμα ασκήσεως αναιρέσεως κατά του βουλεύματος που απορρίπτει την έφεση ως απαράδεκτη, δεν αντίκειται στα άρθρα παρ. 1 και 20 παρ. 1 του Συντάγματος, αφού δεν περιέχει άνιση ρύθμιση, που να συνεπάγεται δυσμενή μεταχείριση κάποιων διαδίκων, αλλ' ούτε στερεί τον διάδικο από το δικαίωμα παροχής έννομης προστασίας από τα δικαστήρια, αφού μάλιστα και ο κοινός νομοθέτης δεν υποχρεούται από το Σύνταγμα να θεσπίζει ένδικα μέσα κατά των αποφάσεων και, ως εκ τούτου, ο διάδικος μπορεί και οφείλει να υπολογίζει, κατά την εκδίκαση της υποθέσεώς του, ότι είναι ενδεχόμενο, η απόφαση που πρόκειται να εκδοθεί να μη προσβάλλεται με ένδικα μέσα, παρά το γεγονός ότι αυτά επιτρέπονταν κατά την έναρξη και κατά τη διάρκεια της δίκης (ή της ποινικής διώξεως). Όπου κρίθηκε αναγκαία η θέσπιση ένδικου μέσου, υπήρξε ρητή η αποτύπωση της βουλήσεως του συντακτικού νομοθέτη με τις ειδικές προβλέψεις των άρθρων 95 παρ. 1 β' (αναίρεση των τελεσίδικων αποφάσεων των διοικητικών δικαστηρίων για υπέρβαση εξουσίας ή παράβαση νόμου) και 96 παρ. 2 του Συντάγματος (έφεση στο αρμόδιο τακτικό δικαστήριο κατά των αποφάσεων αστυνομικών αρχών και αρχών αγροτικής ασφάλειας). Δεν είναι δε αντίθετες οι ανωτέρω διατάξεις ούτε και προς το άρθρο 6 παρ. 1 της ΕΣΔΑ, που κατοχυρώνει την αρχή της δίκαιης δίκης, από την οποία δεν συνάγεται υποχρέωση του εθνικού νομοθέτη για καθιέρωση ένδικων μέσων, αλλά ούτε και στο άρθρο 26 του συντάγματος, το οποίο καθιερώνει την αρχή της διακρίσεως των εξουσιών. Τέλος, από τις διατάξεις των άρθρων 473 παρ. 2 και 474 παρ. 1 του Κώδικα Ποινικής Δικονομίας συνάγεται ότι, κατά γενική αρχή, η αίτηση αναιρέσεως ασκείται με δήλωση του δικαιουμένου διαδίκου ενώπιον των οργάνων που ορίζονται από την τελευταία ως άνω διάταξη, μεταξύ των οποίων δεν περιλαμβάνεται και ο Εισαγγελεύς του Αρείου Πάγου. Η κατ' εξαίρεση άσκηση αναιρέσεως με δήλωση που επιδίδεται στον εισαγγελέα του Αρείου Πάγου μπορεί να γίνει μόνο εναντίον καταδικαστικής αποφάσεως, όχι δε και εναντίον οποιασδήποτε άλλης, η οποία δεν έχει αυτόν τον χαρακτήρα, όπως είναι και η απόφαση, με την οποία απορρίπτεται η έφεση ως απαράδεκτη.

Εν προκειμένω, από τα έγγραφα της δικογραφίας, τα οποία επιτρεπτώς επισκοπούνται από τον Άρειο Πάγο, για την έρευνα του παραδεκτού της αιτήσεως αναιρέσεως, προκύπτουν τα ακόλουθα: Το Τριμελές Πλημμελειοδικείο Αθηνών, με την 8969/7.2.2005 απόφασή του, κήρυξε ένοχη την αναιρεσείουσα κατηγορουμένη Χ1 για τις πράξεις της παραβάσεως του Ν. 2971/2001 και του Ν. 1337/1983 και της επέβαλε συνολική ποινή φυλακίσεως ενός έτους και τριών μηνών. Κατά της αποφάσεως αυτής, η αναιρεσείουσα άσκησε έφεση, η οποία κηρύχθηκε απαράδεκτη, με το 275/2007 βούλευμα του Συμβουλίου Εφετών Αθηνών, κατά του οποίου η κατηγορουμένη άσκησε τις κρινόμενες αιτήσεις αναιρέσεως. Σύμφωνα, όμως με όσα αναφέρονται παραπάνω στη μείζονα σκέψη, κατά του ως άνω βουλεύματος δεν επιτρέπεται να ασκηθεί το ένδικο μέσο της αναιρέσεως και επομένως, οι κρινόμενες αιτήσεις αναιρέσεως, που συνεκδικάζονται, λόγω της μεταξύ τους πρόδηλης συνάφειας, πρέπει, κατ' εφαρμογή της διατάξεως του άρθρου 476 παρ. 1 του Κώδικα Ποινικής δικονομίας, να απορριφθούν ως απαράδεκτες και να καταδικασθεί η αναιρεσείουσα στα δικαστικά έξοδα (άρθρο 583 παρ. 1 ΚΠΔ). Ανεξάρτητα από αυτά, η από 3 Μαρτίου 2007 αίτηση αναιρέσεως, η οποία ασκήθηκε με δήλωση προς τον Εισαγγελέα του Αρείου Πάγου, πρέπει να απορριφθεί ως απαράδεκτη, αφού σύμφωνα με τα ανωτέρω εκτιθέμενα, άσκηση αναιρέσεως με δήλωση, επιδιδόμενη στον Εισαγγελέα του Αρείου Πάγου μπορεί να γίνει μόνο εναντίον καταδικαστικής αποφάσεως, περίπτωση που δεν συντρέχει εν προκειμένω.

ΓΙΑ ΤΟΥΣ ΛΟΓΟΥΣ ΑΥΤΟΥΣ ΑΠΟΡΡΙΠΤΕΙ α) την 64/8.3.2007 αίτηση αναιρέσεως της Χ1 κατά του 275/2007 βουλεύματος του Συμβουλίου Εφετών Αθηνών και β) την από 3.3.2007 ταυτόσημη αίτηση της ίδιας αναιρεσείουσας Χ1 που ασκήθηκε με δήλωσή της προς τον εισαγγελέα του Αρείου Πάγου για αναίρεση επίσης του ανωτέρω 275/2007 βουλεύματος του Συμβουλίου Εφετών Αθηνών. Και ΚΑΤΑΔΙΚΑΖΕΙ την αναιρεσείουσα στα δικαστικά έξοδα, τα οποία ανέρχονται στο ποσό των διακοσίων είκοσι ευρώ (220€).

Κρίθηκε και αποφασίστηκε στην Αθήνα στις 14 Μαρτίου 2008. Και,

Εκδόθηκε στην Αθήνα στις 20 Μαΐου 2008.

Figure C.6. *(Cont.) Biased LexRank algorithm on test case 3.*

Bibliography

[1] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, "Text summarization techniques: A brief survey," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 10, 2017. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2017.081052

[2] S. Almog, *The Origins of the Law in Homer*, ser. Law & Literature. Berlin, Germany: De Gruyter, Mar. 2022. [Online]. Available: https://doi.org/10.1515/9783110766110

[3] A. Amidi and S. Amidi, "Recurrent neural networks cheatsheet star," accessed Sep. 30,2022. [Online]. Available: https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

[4] D. Anand and R. Wagh, "Effective deep learning approaches for summarization of legal texts," *Journal of King Saud University - Computer and Information Sciences*, Dec 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157819301259

[5] I. Angelidis, I. Chalkidis, C. Nikolaou, P. Soursos, and M. Koubarakis, "Nomothesia: A linked data platform for greek legislation," 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:203576149

[6] M. Ankur Kumar and K. Sanjay, "Study and analysis of mycin expert system," *International Journal of Engineering and Computer Science*, vol. 4, no. 10, 2016. [Online]. Available: http://www.ijecs.in/index.php/ijecs/article/view/3009

[7] K. D. Ashley, *Artificial intelligence and legal analytics*. Cambridge, England: Cambridge University Press, Jul. 2017. [Online]. Available: https://doi.org/10.1017/9781316761380

[8] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv:1607.06450 [cs, stat]*, Jul 2016. [Online]. Available: http://arxiv.org/abs/1607.06450

[9] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1409.0473

[10] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Association for Computational Linguistics, Jun 2005, p. 65–72. [Online]. Available: https://www.aclweb.org/anthology/W05-0909

[11] I. Beltagy, K. Lo, and A. Cohan, "Scibert: A pretrained language model for scientific text," *arXiv:1903.10676 [cs]*, Sep 2019. [Online]. Available: http://arxiv.org/abs/1903.10676

[12] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv:2004.05150*, 2020. [Online]. Available: https://arxiv.org/abs/2004.05150

[13] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *The Journal of Machine Learning Research*, vol. 3, no. 3, p. 1137–1155, Nov 2003.

[14] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" *arXiv:2102.05095 [cs]*, Jun 2021. [Online]. Available: http://arxiv.org/abs/2102.05095

[15] M. Bhandari, P. Gour, A. Ashfaq, and P. Liu, "Metrics also disagree in the low scoring range: Revisiting summarization evaluation metrics," *arXiv:2011.04096 [cs]*, Nov 2020. [Online]. Available: http://arxiv.org/abs/2011.04096

[16] P. Bhattacharya, K. Hiware, S. Rajgaria, N. Pochhi, K. Ghosh, and S. Ghosh, "A comparative study of summarization algorithms applied to legal case judgments," in *Advances in Information Retrieval*, ser. Lecture Notes in Computer Science, L. Azzopardi, B. Stein, N. Fuhr, P. Mayr, C. Hauff, and D. Hiemstra, Eds. Cham: Springer International Publishing, 2019, p. 413–428.

[17] P. Bhattacharya, S. Poddar, K. Rudra, K. Ghosh, and S. Ghosh, "Incorporating domain knowledge for extractive summarization of legal case documents," in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*. São Paulo Brazil: ACM, Jun 2021, p. 22–31. [Online]. Available: https://dl.acm.org/doi/10.1145/3462757.3466092

[18] C. M. Bishop, *Pattern Recognition and Machine Learning*, ser. Information Science and Statistics. Berlin, Heidelberg: Springer-Verlag, 2006.

[19] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, p. 135–146, 2017.

[20] I. Chalkidis, I. Androutsopoulos, and N. Aletras, "Neural legal judgment prediction in english," in *Proceedings of the 57th Annual Meeting of the*

Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, Jul 2019, p. 4317–4323. [Online]. Available: https://aclanthology.org/P19-1424

[21] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "Legal-bert: The muppets straight out of law school," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov 2020, p. 2898–2904. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.261

[22] I. Chalkidis, M. Fergadiotis, D. Tsarapatsanis, N. Aletras, I. Androutsopoulos, and P. Malakasiotis, "Paragraph-level rationale extraction through regularization: A case study on european court of human rights cases," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, Jun 2021, p. 226–241. [Online]. Available: https://aclanthology.org/2021.naacl-main.22

[23] S. Chaudhari, V. Mithal, G. Polatkan, and R. Ramanath, "An attentive survey of attention models," *ACM Transactions on Intelligent Systems and Technology*, vol. 12, no. 5, pp. 53:1–53:32, Jul 2021.

[24] H. Chen, D. Cai, W. Dai, Z. Dai, and Y. Ding, "Charge-based prison term prediction with deep gating network," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Aug 2019, p. 6362–6367. [Online]. Available: https://aclanthology.org/D19-1667

[25] J. Cheng and M. Lapata, "Neural summarization by extracting sentences and words," *arXiv:1603.07252 [cs]*, Jun 2016. [Online]. Available: http://arxiv.org/abs/1603.07252

[26] N. Chomsky, *Syntactic structures*, 13th ed., ser. Janua Linguarum. Series Minor. Studia Memoriae Nicolai van Wijk Dedicata, C. H. van Schooneveld, Ed. Berlin, Germany: De Gruyter Mouton, Jan. 1978.

[27] S. Chopra, M. Auli, and A. M. Rush, "Abstractive sentence summarization with attentive recurrent neural networks," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, Jun 2016, p. 93–98. [Online]. Available: https://www.aclweb.org/anthology/N16-1012

[28] I. Daras, M. Gogoulos, and P. Louridas, "Google summer of code 2018 project - spacy now speaks greek," https://github.com/eellak/gsoc2018-spacy, 2018, accessed Oct. 28,2022.

[29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun 2019, p. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

[30] H. W. Dong, W. Y. Hsiao, L. C. Yang, and Y. H. Yang, "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," *arXiv:1709.06298 [cs, eess, stat]*, Nov 2017. [Online]. Available: http://arxiv.org/abs/1709.06298

[31] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *The Journal of Machine Learning Research*, vol. 12, p. 2121–2159, Apr 2011.

[32] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2001.

[33] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert Systems with Applications*, vol. 165, no. 113679, Mar 2021.

[34] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of Artificial Intelligence Research*, vol. 22, p. 457–479, Dec 2004, arXiv: 1109.2128.

[35] A. R. Fabbri, W. Kryściński, B. McCann, C. Xiong, R. Socher, and D. Radev, "SummEval: Re-evaluating Summarization Evaluation," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 391–409, Apr 2021. [Online]. Available: https://doi.org/10.1162/tacl_a_00373

[36] A. Farzindar and G. Lapalme, "Letsum, an automatic legal text summarizing system," *Jurix*, pp. 11–18, Jan 2004.

[37] D. Feijo and V. Moreira, "Summarizing legal rulings: Comparative experiments," in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*. Varna, Bulgaria: INCOMA Ltd., Sep 2019, p. 313–322. [Online]. Available: https://aclanthology.org/R19-1036

[38] C. Fellbaum, *WordNet*. John Wiley Sons, Ltd, 2012. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781405198431.wbeal1285

[39] F. Galgani, P. Compton, and A. Hoffmann, "Citation based summarisation of legal texts," in *Proceedings of the 12th Pacific Rim international conference on Trends in Artificial Intelligence*, ser. PRICAI'12. Berlin, Heidelberg: Springer-Verlag, Jun 2012, p. 40–52. [Online]. Available: https://doi.org/10.1007/978-3-642-32695-0_6

[40] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: a survey," *Artificial Intelligence Review*, vol. 47, no. 1, p. 1–66, Jan 2017. [Online]. Available: https://doi.org/10.1007/s10462-016-9475-9

[41] K. Ganesan, "Rouge 2.0: Updated and improved measures for evaluation of summarization tasks," *arXiv:1803.01937 [cs]*, Mar 2018. [Online]. Available: http://arxiv.org/abs/1803.01937

[42] S. Gehrmann, Y. Deng, and A. Rush, "Bottom-up abstractive summarization," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Oct 2018, p. 4098–4109. [Online]. Available: https://www.aclweb.org/anthology/D18-1443

[43] S. Ghosh, M. Dutta, and T. Das, "Indian legal text summarization: A text normalisation-based approach," *arXiv:2206.06238 [cs]*, Sep 2022. [Online]. Available: http://arxiv.org/abs/2206.06238

[44] I. Glaser, S. Moser, and F. Matthes, "Summarization of german court rulings," in *Proceedings of the Natural Legal Language Processing Workshop 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Aug 2021, p. 180–189. [Online]. Available: https://aclanthology.org/2021.nllp-1.19

[45] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[46] M. Grusky, M. Naaman, and Y. Artzi, "Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies," *arXiv:1804.11283 [cs]*, May 2020. [Online]. Available: http://arxiv.org/abs/1804.11283

[47] B. Hachey and C. Grover, "Extractive summarisation of legal texts," *Artificial Intelligence and Law*, vol. 14, no. 4, p. 305–345, Dec 2006.

[48] K. Hamilton, A. Nayak, B. Božić, and L. Longo, "Is neuro-symbolic ai meeting its promise in natural language processing? a structured review," *arXiv:2202.12205 [cs]*, Jun 2022. [Online]. Available: http://arxiv.org/abs/2202.12205

[49] K. M. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," *arXiv:1506.03340 [cs]*, Nov 2015. [Online]. Available: http://arxiv.org/abs/1506.03340

[50] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning. lecture 6." [Online]. Available: https://cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

[51] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, p. 6840–6851. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html

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

[53] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, "spaCy: Industrial-strength Natural Language Processing in Python," Tech. Rep., 2020.

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

[55] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv:1502.03167 [cs]*, Mar 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[56] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, p. 583–589, Aug 2021. [Online]. Available: https://doi.org/10.1038/s41586-021-03819-2

[57] T. Jung, D. Kang, L. Mentch, and E. Hovy, "Earlier isn't always better: Sub-aspect analysis on corpus and system biases in summarization," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Nov 2019, p. 3324–3335. [Online]. Available: https://www.aclweb.org/anthology/D19-1327

[58] D. Jurafsky and J. H. Martin, *Speech and Language Processing (3rd Edition - January 2022 Draft)*. USA: Prentice-Hall, Inc., 2022.

[59] S. Jürgen, "Who invented backpropagation?" accessed Sep. 12, 2022. [Online]. Available: https://people.idsia.ch/~juergen/who-invented-backpropagation.html

[60] P. M. Kien, H.-T. Nguyen, N. X. Bach, V. Tran, M. L. Nguyen, and T. M. Phuong, "Answering legal questions by learning neural attentive text representation," in *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Sep 2020, p. 988–998. [Online]. Available: https://aclanthology.org/2020.coling-main.86

[61] B. Kim, H. Kim, and G. Kim, "Abstractive summarization of reddit posts with multi-level memory networks," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Jun 2019, p. 2519–2531. [Online]. Available: https://www.aclweb.org/anthology/N19-1260

[62] M.-Y. Kim, Y. Xu, and R. Goebel, "Summarization of legal texts with high cohesion and automatic compression rate," in *New Frontiers in Artificial Intelligence*, ser. Lecture Notes in Computer Science, Y. Motomura, A. Butler, and D. Bekki, Eds. Berlin, Heidelberg: Springer, 2013, p. 190–204.

[63] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, Jan 2017. [Online]. Available: http://arxiv.org/abs/1412.6980

[64] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv:1312.6114 [cs, stat]*, May 2014. [Online]. Available: http://arxiv.org/abs/1312.6114

[65] N. Kitaev, L. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," *CoRR*, vol. abs/2001.04451, 2020. [Online]. Available: https://arxiv.org/abs/2001.04451

[66] M. Koniaris, G. Papastefanatos, and I. Anagnostopoulos, "Solon: A holistic approach for modelling, managing and mining legal sources," *Algorithms*, vol. 11, no. 12, 2018. [Online]. Available: https://www.mdpi.com/1999-4893/11/12/196

[67] A. Kornilova and V. Eidelman, "Billsum: A corpus for automatic summarization of us legislation," in *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, 2019, p. 48–56, arXiv:1910.00523 [cs]. [Online]. Available: http://arxiv.org/abs/1910.00523

[68] M. Koupaee and W. Y. Wang, "Wikihow: A large scale text summarization dataset," *ArXiv*, 2018. [Online]. Available: https://arxiv.org/pdf/1810.09305

[69] J. Koutsikakis, I. Chalkidis, P. Malakasiotis, and I. Androutsopoulos, "Greek-bert: The greeks visiting sesame street," in *11th Hellenic Conference on Artificial Intelligence*, ser. SETN 2020. New York, NY, USA: Association for Computing Machinery, 2020, p. 110–117. [Online]. Available: https://doi.org/10.1145/3411408.3411440

[70] W. Kryscinski, N. S. Keskar, B. McCann, C. Xiong, and R. Socher, "Neural text summarization: A critical evaluation," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Aug 2019, p. 540–551. [Online]. Available: https://aclanthology.org/D19-1051

[71] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Apr 2020, p. 7871–7880. [Online]. Available: https://aclanthology.org/2020.acl-main.703

[72] R. Likert, "A technique for the measurement of attitudes," *Archives of Psychology*, vol. 140, no. 22, pp. 5–55, 1932.

[73] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. Association for Computational Linguistics, Jul 2004, p. 74–81. [Online]. Available: https://www.aclweb.org/anthology/W04-1013

[74] N. Liu, J. Wang, and Y. Gong, "Deep self-organizing map for visual classification," in *2015 International Joint Conference on Neural Networks (IJCNN)*, Jul 2015, p. 1–6.

[75] P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, and N. M. Shazeer, "Generating wikipedia by summarizing long sequences," *ICLR*, 2018.

[76] W. Liu, H. Wu, W. Mu, Z. Li, T. Chen, and D. Nie, "Co2sum: Contrastive learning for factual-consistent abstractive summarization," *ArXiv*, vol. abs/2112.01147, 2021.

[77] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Nov 2019, p. 3730–3740. [Online]. Available: https://www.aclweb.org/anthology/D19-1387

[78] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv:1907.11692 [cs]*, Jul 2019. [Online]. Available: http://arxiv.org/abs/1907.11692

[79] Y. Liu, P. Liu, D. Radev, and G. Neubig, "Brio: Bringing order to abstractive summarization," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, Feb 2022, p. 2890–2903. [Online]. Available: https://aclanthology.org/2022.acl-long.207

[80] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development*, vol. 2, no. 2, p. 159–165, Dec 1958.

[81] N. v. d. Luijtgaarden, "Automatic summarization of legal text," Master's thesis, Utrecht University, Aug 2019, accessed Aug. 27, 2022. [Online]. Available: https://studenttheses.uu.nl/handle/20.500.12932/34261

[82] S. Luka, S. Alexey, and github/karambolishe, "lexrank," https://github.com/crabcamp/lexrank, 2018, accessed Oct. 28,2022.

[83] C. D. Manning and H. Schutze, *Foundations of statistical natural language processing*, ser. The MIT Press. London, England: MIT Press, May 1999.

[84] L. Manor and J. J. Li, "Plain english summarization of contracts," *arXiv:1906.00424 [cs]*, Jun 2019. [Online]. Available: http://arxiv.org/abs/1906.00424

[85] M. Masala, R. C. A. Iacob, A. S. Uban, M. Cidota, H. Velicu, T. Rebedea, and M. Popescu, "jurbert: A romanian bert model for legal judgement prediction," in *Proceedings of the Natural Legal Language Processing Workshop 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Aug 2021, p. 86–94. [Online]. Available: https://aclanthology.org/2021.nllp-1.8

[86] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald, "On faithfulness and factuality in abstractive summarization," *arXiv:2005.00661 [cs]*, May 2020. [Online]. Available: http://arxiv.org/abs/2005.00661

[87] B. McCann, J. Bradbury, C. Xiong, and R. Socher, "Learned in translation: Contextualized word vectors," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6297–6308.

[88] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, p. 115–133, Dec 1943.

[89] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Jul 2004, p. 404–411. [Online]. Available: https://www.aclweb.org/anthology/W04-3252

[90] R. Miikkulainen and M. G. Dyer, "Natural language processingwith modular neural networks and distributed lexicon," in *Cognitive Science*, 1991, pp. 343,399.

[91] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv:1301.3781 [cs]*, Sep 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[92] M. Minsky and S. A. Papert, *Perceptrons: An Introduction to Computational Geometry*. MIT press, 1969.

[93] M. Mitchell, "Why ai is harder than we think," in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. GECCO '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 3. [Online]. Available: https://doi.org/10.1145/3449639.3465421

[94] R. Nallapati, B. Zhou, C. N. d. Santos, C. Gulcehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence rnns and beyond," *arXiv:1602.06023 [cs]*, Aug 2016. [Online]. Available: http://arxiv.org/abs/1602.06023

[95] F. Nan, C. Nogueira dos Santos, H. Zhu, P. Ng, K. McKeown, R. Nallapati, D. Zhang, Z. Wang, A. O. Arnold, and B. Xiang, "Improving factual consistency of abstractive summarization via question answering," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th*

International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Online: Association for Computational Linguistics, Aug. 2021, pp. 6881–6894. [Online]. Available: https://aclanthology.org/2021.acl-long.536

[96] S. Narayan, S. B. Cohen, and M. Lapata, "Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Oct 2018, p. 1797–1807. [Online]. Available: https://www.aclweb.org/anthology/D18-1206

[97] A. Nenkova and K. McKeown, *A Survey of Text Summarization Techniques*. Boston, MA: Springer US, 2012, p. 43–76. [Online]. Available: http://link.springer.com/10.1007/978-1-4614-3223-4_3

[98] J.-P. Ng and V. Abrecht, "Better summarization evaluation with word embeddings for rouge," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Jun 2015, p. 1925–1930. [Online]. Available: https://aclanthology.org/D15-1222

[99] J. Novikova, O. Dušek, A. Cercas Curry, and V. Rieser, "Why we need new evaluation metrics for nlg," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Sep 2017, p. 2241–2252. [Online]. Available: https://www.aclweb.org/anthology/D17-1238

[100] C. Olah, "Understanding lstm networks," Aug 2015, accessed Sep. 29, 2022. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[101] J. Otterbacher, G. Erkan, and D. R. Radev, "Biased lexrank: Passage retrieval using random walks with question-based priors," *Information Processing Management*, vol. 45, no. 1, p. 42–54, Jan 2009. [Online]. Available: https://doi.org/10.1016/j.ipm.2008.06.004

[102] P. Over, H. Dang, and D. Harman, "Duc in context," *Information Processing and Management*, vol. 43, no. 6, p. 1506–1520, Aug 2007. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=50955

[103] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Technical Report 1999-66, November 1999. [Online]. Available: http://ilpubs.stanford.edu:8090/422/

[104] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Jul 2002, p. 311–318. [Online]. Available: https://www.aclweb.org/anthology/P02-1040

[105] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[106] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Jul 2014, p. 1532–1543. [Online]. Available: https://aclanthology.org/D14-1162

[107] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Mar 2018, p. 2227–2237. [Online]. Available: https://aclanthology.org/N18-1202

[108] M. Peyrard, "Studying summarization evaluation metrics in the appropriate scoring range," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Jul 2019, p. 5093–5100. [Online]. Available: https://www.aclweb.org/anthology/P19-1502

[109] S. Polsley, P. Jhunjhunwala, and R. Huang, "Casesummarizer: A system for automated summarization of legal texts," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*. Osaka, Japan: The COLING 2016 Organizing Committee, Sep 2016, p. 258–262. [Online]. Available: https://aclanthology.org/C16-2054

[110] D. R. Radev, E. Hovy, and K. McKeown, "Introduction to the special issue on summarization," *Comput. Linguist.*, vol. 28, no. 4, p. 399–408, dec 2002. [Online]. Available: https://doi.org/10.1162/089120102762671927

[111] D. R. Radev, H. Jing, and M. Budzikowska, "Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies," in *NAACL-ANLP 2000 Workshop: Automatic Summarization*, 2000. [Online]. Available: https://aclanthology.org/W00-0403

[112] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018, accessed Sep. 23,2022. [Online]. Available: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

[113] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," *arXiv:2102.12092 [cs]*, 2021. [Online]. Available: https://doi.org/10.48550/arxiv.2102.12092

[114] H. Rashidi, N. Tran, E. Betts, L. Howell, and R. Green, "Artificial intelligence and machine learning in pathology: The present landscape of supervised methods," *Academic Pathology*, vol. 6, 09 2019. [Online]. Available: https://doi.org/10.1177/2374289519873088

[115] R. Rehurek and P. Sojka, "Gensim–python framework for vector space modelling," *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, vol. 3, no. 2, 2011.

[116] F. Rosenblatt, "The perceptron. a perceiving and recognizing automaton." Cornell Aeronautical Laboratory, Bufallo, New York, Tech. Rep. 85-460-1, 1957.

[117] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Washington, DC: Spartan Books, 1962.

[118] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 60886088, p. 533–536, Oct 1986.

[119] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2010.

[120] E. Sandhaus, "The new york times annotated corpus." [Online]. Available: https://catalog.ldc.upenn.edu/LDC2008T19

[121] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv:1910.01108 [cs]*, Feb 2020. [Online]. Available: http://arxiv.org/abs/1910.01108

[122] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?" *arXiv:1805.11604 [cs, stat]*, Apr 2019. [Online]. Available: http://arxiv.org/abs/1805.11604

[123] M. Saravanan and B. Ravindran, "Identification of rhetorical roles for segmentation and summarization of a legal judgment," *Artificial Intelligence and Law*, vol. 18, no. 1, p. 45–76, Mar 2010.

[124] J. Schmidt, M. R. G. Marques, S. Botti, and M. A. L. Marques, "Recent advances and applications of machine learning in solid-state materials science," *npj Computational Materials*, vol. 5, no. 11, p. 1–36, Aug 2019.

[125] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," *arXiv:1704.04368 [cs]*, Apr 2017. [Online]. Available: http://arxiv.org/abs/1704.04368

[126] Z. Shen, K. Lo, L. Yu, N. Dahlberg, M. Schlanger, and D. Downey, "Multi-lexsum: Real-world summaries of civil rights lawsuits at multiple granularities," *arXiv:2206.10883 [cs]*, Jun 2022. [Online]. Available: http://arxiv.org/abs/2206.10883

[127] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, p. 1929–1958, 2014, accessed Sep. 19,2022.

[128] J. Steinberger and K. Jezek, "Using latent semantic analysis in text summarization and summary evaluation," in *Proceedings of the 7th International Conference ISIM*, Jan 2004.

[129] S. Sun and W. Li, "Alleviating exposure bias via contrastive learning for abstractive text summarization," *arXiv:2108.11846 [cs]*, Aug 2021. [Online]. Available: http://arxiv.org/abs/2108.11846

[130] R. Susskind, *Online Courts and the Future of Justice*. Oxford University Press, Nov. 2019.

[131] Βλάχου-Βλαχοπούλου, Μαγδαληνή-Χριστίνα, "Οι Πηγές του Δικαίου," Master's thesis, ΠΜΣ Δημόσιο Δίκαιο, Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών, 2019, accessed Oct. 27, 2022. [Online]. Available: https://pergamos.lib.uoa.gr/uoa/dl/frontend/el/browse/2887267

[132] Φ.Α.Χάγιεκ, *Το σύνταγμα της ελευθερίας*. Εκδόσεις Καστανιώτη, 2008.

[133] P. Tiersma, "The creation, structure, and interpretation of the legal text," accessed Sep. 11,2022. [Online]. Available: http://www.languageandlaw.org/LEGALTEXT.HTM

[134] P. M. Tiersma, *PARCHMENT, PAPER, PIXELS. Law and the Technologies of Communication*. Chicago, London: The University of Chicago Press, 2010.

[135] A. Turing, "I.—computing machinery and intelligence," *Mind*, vol. LIX, no. 236, p. 433–460, Oct 1950.

[136] H. Turtle, "Text retrieval in the legal world," *Artificial Intelligence and Law*, vol. 3, no. 1, p. 5–54, Mar 1995.

[137] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," Jun 2017. [Online]. Available: https://arxiv.org/abs/1706.03762v5

[138] J. Waldron, "The rule of law and the importance of procedure," *Nomos*, vol. 50, p. 3–31, 2011.

[139] A. Wang, K. Cho, and M. Lewis, "Asking and answering questions to evaluate the factual consistency of summaries," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 5008–5020. [Online]. Available: https://aclanthology.org/2020.acl-main.450

[140] J. Weizenbaum, "Eliza—a computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9, no. 1, p. 36–45, Jan 1966.

[141] J. Whitley, "Cretan laws and cretan literacy," *American Journal of Archaeology*, vol. 101, no. 4, pp. 635–661, Oct. 1997. [Online]. Available: https://doi.org/10.2307/506828

[142] L. Wittgenstein, *Philosophical Investigations*, 3rd ed. London, England: Blackwell, Apr. 1969, pp. 42–43.

[143] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6

[144] H. Xu, J. Savelka, and K. D. Ashley, "Toward summarizing case decisions via extracting argument issues, reasons, and conclusions," in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*. New York, NY, USA: Association for Computing Machinery, Mar 2021, p. 250–254. [Online]. Available: https://doi.org/10.1145/3462757.3466098

[145] H. Xu, J. Šavelka, and K. D. Ashley, "Using argument mining for legal text summarization," in *JURIX*, vol. 334, 2020, pp. 184–193. [Online]. Available: https://doi.org/10.3233/FAIA200862

[146] Y. Yang, M. C. S. UY, and A. Huang, "Finbert: A pretrained language model for financial communications," *arXiv:2006.08097 [cs]*, Jul 2020. [Online]. Available: http://arxiv.org/abs/2006.08097

[147] M. Yousfi-Monod, A. Farzindar, and G. Lapalme, "Supervised machine learning for summarizing legal documents," in *Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science, A. Farzindar and V. Kešelj, Eds. Berlin, Heidelberg: Springer, 2010, p. 51–62.

[148] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *arXiv:1906.05113 [cs, eess]*, Apr 2020. [Online]. Available: http://arxiv.org/abs/1906.05113

[149] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," Sep 2019. [Online]. Available: https://openreview.net/forum?id=SkeHuCVFDr

[150] L. Zhong, Z. Zhong, Z. Zhao, S. Wang, K. D. Ashley, and M. Grabmair, "Automatic summarization of legal decisions using iterative masking of predictive sentences," in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, ser. ICAIL '19. New York, NY, USA: Association for Computing Machinery, Jun 2019, p. 163–172. [Online]. Available: https://doi.org/10.1145/3322640.3326728

[151] Z.-H. Zhou and J. Feng, "Deep forest," *National Science Review*, vol. 6, pp. 74 – 86, Jan 2019. [Online]. Available: https://doi.org/10.1093/nsr/nwy108

[152] C. Zhu, R. Xu, M. Zeng, and X. Huang, "A hierarchical network for abstractive meeting summarization with cross-domain pretraining," *arXiv:2004.02016 [cs]*, Sep 2020, arXiv: 2004.02016. [Online]. Available: http://arxiv.org/abs/2004.02016

[153] J. Zhu, Y. Xia, L. Wu, D. He, T. Qin, W. Zhou, H. Li, and T. Y. Liu, "Incorporating bert into neural machine translation," *arXiv:2002.06823 [cs]*, Feb 2020. [Online]. Available: http://arxiv.org/abs/2002.06823

List of Abbreviations

AI	Artificial Intelligence
ATS	Automatic Text Summarization
BoW	Bag of Words
BERT	Bidirectional Encoder Representations from Transformers
CBoW	Continuous Bag of Words
ML	Machine Learning
MSE	Mean Squared Error
LCS	Longest Common Subsequence
LSTM	Long Short-Term Memory
NER	Named Entity Recognition
NLP	Natural Language Processing
POS	Part Of Speach
OOV	Out Of Vocabulary