T.C.
ALTINBAŞ UNIVERSITY
Institute of Graduate Studies
Information Technologies

STUDY AND EVALUATE A MECHANISM THAT IS ABLE TO HIGHLIGHT THE LEGAL DOCUMENTS

HUSAM ADEL ALI ALSHUWAILI

Master Thesis

Supervisor

Prof. Dr. Osman Nuri UCAN

Istanbul, 2020

STUDY AND EVALUATE A MECHANISM THAT IS ABLE TO HIGHLIGHT THE LEGAL DOCUMENTS

by
Husam Adel Ali Alshuwaili

Information Technologies

Submitted to the Institute of Graduate Studies in partial fulfillment of the requirements for the degree of Master of Science

ALTINBAŞ UNIVERSITY
2020

The thesis titled “STUDY AND EVALUATE A MECHANISM THAT IS ABLE TO HIGHLIGHT THE LEGAL DOCUMENTS” prepared and presented by “Husam Adel Ali Alshuwaili” was accepted as a Master of Science Thesis in Information Technologies.

Prof. Dr. Osman Nuri UCAN
Supervisor

Thesis Defense Jury Members:

Prof. Dr. Osman Nuri UCAN	School of Engineering and Natural Sciences, Altinbas University	_____
Asst. Prof. Dr. Abdullahi Abdu Ibrahim	School of Engineering and Natural Sciences, Altinbas University	_____
Assoc. Prof. Dr. Adil Deniz Duru	Faculty of Sport sciences, Marmara University	_____

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Approval Date of Institute of Graduate Studies:
____/____/____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Husam Adel Ali Alshuwaili

DEDICATION

First I would like to Allah Almighty for the power of mind , health, strength , guidance knowledge and skills to complete this study .This thesis is wholeheartedly dedicated to my beloved father, who have been my source of inspiration, he tells me that" every success in your life will be best gift for me". To my mother, who have been supporting me with the kind and pure love.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Asst. Prof. Osman N. UCAN for all support during my study. It’s great pleasure to express my deepest gratitude to my friends who have shared with me best moments during my study for the Master degree.

ABSTRACT

STUDY AND EVALUATE A MECHANISM THAT IS ABLE TO HIGHLIGHT THE LEGAL DOCUMENTS

Alshuwaili, Husam Adel Ali

M.Sc.,Information Technologies, Altinbas University ,

Supervisor: Prof. Dr. Osman Nuri UCAN

Date: 10/2020

Pages: 76

It is not uncommon that a legal case consists of hundreds to thou-sands of related documents, which a lawyer needs to scan through to understand a case and find useful information, potentially a shred of evidence. Therefore, many automated tools have emerged that are specifically designed to help in the Electronic Discovery process. The most basic ones enable more or less advanced full-text search, which is a we studied field called Information Retrieval. Another field, frequently applied in the legal domain, is the Artificial Intelligence, which has delivered numerous purposeful tools that are trying to substitute a lawyer in var tasks. The tool presented in this thesis belongs to this category the goal of this research is to study, design and evaluate a mechanism that is able to highlight legal documents with valuable information and filter out the irrelevant.

Keywords: Electronic Discovery , Artificial Intelligence, legal documents.

TABLE OF CONTENTS

Pages

ABSTRACT ..... **vii**
LIST OF TABLES ..... **ix**
LIST OF FIGURES ..... **x**
LIST OF ABBREVIATIONS ..... **xi**
1. INTRODUCTION **1**
2. RELATED WORK **4**
2.1 ARTIFICIAL INTELLIGENCE AND LAW 5
2.2 ELECTRONIC DISCOVERY 6
2.3 MEASURING RELEVANCE IN LEGAL DOMAIN 6
3. DATA CHARACTERISTICS **8**
3.1 DATASET INTRODUCTION 9
3.2 ANNOTATIONS 10
3.3 DATA STATISTICS 10
3.4 TYPES OF DOCUMENTS 15
4. NLP TOOL SELECTION **16**
4.1 NLP ALGORITHMS 16
4.1.1 Keywords 25
4.1.2 Named Entity Recognition 29
4.1.3 Relations 25
4.1.4 Topic Modeling 29
4.1.5 Document Category 25
4.2 INTRODUCTION OF THE TOOLS 18
4.2.1 Nltk: The Natural Language Toolkit 25
4.2.2 Google Cloud Natural Language 29
4.2.3 Watson Natural Language Understanding 25
4.2.4 Aylien 29
4.3 COMPARISON 21

4.3.1 Testing data: Mock legal case 25
4.3.2 Survey 29
4.3.3 Statistical comparison 25
4.3.4 Analytical comparison 29
5. CLASSIFICATION 49
5.1 ATTRIBUTES 49
5.1.1 Sentence meaning embedding 62
5.1.2 Keyword embedding 63
5.1.3 Entity embedding 62
5.1.4 Relation embedding 63
5.1.5 Category embedding 62
5.1.6 Structural characteristics 63
5.2 BOTTOM-LEVEL MODEL 51
5.3 LAW-CASE CLASSIFIER 53
5.4 SENTENCE RELEVANCE CLASSIFIER 55
5.4.1 General attributes 63
5.4.2 Case-specific attributes 62
5.5 DOCUMENT RELEVANCE CLASSIFIER 56
6. EVALUATION 70
6.1 DECISION POINT EVALUATION 56
6.1.1 Bottom-level estimator selection 63
6.1.2 Law-Case estimator selection 62
6.1.3 Layered architecture 62
6.1.4 Relations with types 62
6.2 PARAMETER TUNING 62
6.2.1 Bottom-level Model 63
6.2.2 Top-level Model 62
6.3 RESULTS 56
6.3.1 Evaluation of Law-Case classifier 63
6.3.2 Evaluation of Sentence classifier 63
6.3.3 Evaluation of Document classifier 62
7. CONCLUSION 68
REFERENCES 71

LIST OF TABLES

Pages

Table 3.1: Most frequent words in the dataset with and without stop words.....12

Table 4.1: NLP tools evaluated against the survey with strict string comparison.....27

Table 4.2: Jaccard index of the NLP tools evaluated against the survey with subset tolerance in string comparison.....28

Table 4.3: Jaccard index of the time-based entities extracted by Aylien 31

Table 6.1: The neural network parameter tuning on sentence meaning predicting the Law-Case classes.....62

LIST OF FIGURES

Pages

Figure 2.1: Simple example of Bayesian Network 4

Figure 3.1: Most frequent words in the dataset with and without stop words 12

Figure 5.1: Diagram of models 35

Figure 5.2: Distances among entity vectors in word2vec dictionar,,..38

Figure 5.3: Watson NLU relation encoding example in JSON.. ..39

Figure 5.4: Bottom-level model structure 43

Figure 5.5: Accuracies of bottom-layer models for prediction... ..45

Figure 5.6: Feature importance from Random Forest classification.. ..46

Figure 5.7: Importance of dimensions in the sentence meaning vectors 47

Figure 5.8: The first three components of PCA transformation of the sentence 49

Figure 5.9: Precision and recall of the general attributes.. ..50

Figure 5.10: Precision and recall of the general and case-specific at- tributes 51

Figure 5.11: Feature importance from Random tree classification of HOT 53

Figure 5.12: Heat map of accuracies for all values of parameters *w* and *h*.n Note that the values are shown in %. 54

Figure 6.1: Bottom-level estimators' accuracies for case-related or law- related classification of the sentence meaning 56

Figure 6.2: Estimator accuracies for the case-related or law-related classification of the sentence meaning. 58

Figure 6.3: Performance of all-in-one model (one layer) vs separate features (two layers). 59

Figure 6.4: Performance of a conventional embedding in one-hot vectors vs. 60

Figure 6.5: Parameter tuning of the Random Forest classifier for the case-related or law-related classification of documents. 63

Figure 6.6: Example of a sentence relevance in WARM document 65

Figure 6.7: Precision and recall of the binary relevance prediction of documents 66

1.INTRODUCTION

When lawyers, attorneys or paralegals obtain a legal case to work on, one of the first steps is the Electronic Discovery. They need to search, gather and clean, and organize legal documents in order to find pieces of evidence in legal case [1]. With increasing amount of information in the electronic form, the lawyers put more emphasis on an efficient process of Electronic Discovery

It is not uncommon that a legal case consists of hundreds to thou-sands of related documents, which a lawyer needs to scan through to understand a case and find useful information, potentially a shred of evidence. Therefore, many automated tools have emerged that are specifically designed to help in the Electronic Discovery process. The most basic ones enable more or less advanced full-text search, which is a we studied field called Information Retrieval. Another field, frequently applied in the legal domain, is the Artificial Intelligence, which has delivered numerous purposeful tools that are trying to substitute a lawyer in var tasks. The tool presented in this thesis belongs to this category the goal of this research is to study, design and evaluate a mechanism that is able to highlight legal documents with valuable information and filter out the irrelevant. Such tool could save the time spent on reading the documents and thus, increase the lawyers' efficiency. The tool is not expected to be 100% successful, as the relevance is a subjective factor and usually depends on a broad context, that is hard to acquire by the tool.

Nevertheless, the tools with at least a reasonable performance have proven to be helpful in the past [2], especially the low-precision & high-recall (systems (more in the section 1.3)

The thesis is structured in the following way. The first chapter discusses the previous work on artificial intelligence in the legal domain, similar tools, and understanding of the concept of relevance. The second chapter the working dataset is described, and the exploratory analysis provides some insight into the data and its characteristics. The third chapter presents research of the natural language processing algorithms and tools to eventually pick the most suited for our purpose. Next, the fourth chapter elaborates the attributes and estimators applied to predict the relevance. Finally, the sixth chapter shows some small experiments that helped assess intermediate decisions and evaluates the final models.

2. RELATED WORK

This chapter summarizes the previous work on the application of artificial intelligence in the legal domain. The first section describes a general role of artificial intelligence in law and how it evolved in time. Researchers in this field focused mainly on decision-making systems to support work of the jury; however, this research focuses more on the information retrieval and knowledge discovery in the legal domain, also known as the *Electronic Discovery*, which brings support for lawyers. Therefore, the second section looks more closely at this subfield by introducing state-of-the-art legal retrieval systems and underlines shared characteristics with the research presented here. The last section discusses an issue of relevance in the legal context and related studies.

2.1 ARTIFICIAL INTELLIGENCE AND LAW

The global publishing company *Springer* introduced a journal *Artificial Intelligence and Law* [1] in 1992 which contains 426 published articles (as of March 2018) describing theoretical and empirical studies in artificial intelligence (AI) from the legal perspective. Though the journal consists of articles concerned with AI as a new trend that needs to be bound by law, such as legal aspects of autonomous vehicles [3], robot sex [4] or responsibility of autonomous machines [5], it also contains studies of new innovative AI systems that perform automatic reason- ing and knowledge extraction in the legal domain. The collection of AI-law related articles provides enough information to create a brief overview of how this field has evolved.

The first legal search engines enabled retrieving results by first- order logic rules and fixed pieces of text [6]. In the early 1990s, the concept of full-text search has become widely used, and therefore, users could enjoy a retrieval by a plain natural language [6]. Shortly after, researchers started to focus on AI when creating a new legal retrieval system, such as legal text management system *Flexlaw* (1995) [7], integrated decision support system *DataLex WorkStations* (1995) [8], or system for heuristic retrieval of legal arguments *BankXX* (1996) [9]. As the machine learning was still quite nascent, the mentioned systems are a fusion of expert and AI systems with expert knowledge as the essential part of the system. A sign of such fusion is an attempt to formalize the *legal argument* ([10], [11] and most recently [12]) as an element between the *rules* (expert

systems) and *models* (AIsystems).

At the turn of the century, an attempt to conceptualize the legal do- main for the design of legal knowledge systems brought a number of ontologies, such as McCarty's language for legal discourse, Stamper's norma formalism, or Valente's functional ontology [13]. The ontolo- gies are categorizing the knowledge and breaking it into predicate relations, which express properties of objects, rules in the form of implications and second-order expressions adding more explanatory power. Unfortunately, they are not very widely applied, except for the Valente's Functional Ontology for Law (FOLaw) [14]. The conclusion of the research comparing the four main ontologies is the following: "None of the ontologies seems to have adequate provisions to specify legal procedures." [13].

Consequently, the formalistic implementations of the syllogistic model in legal domain were criticized ([15], [16]). Researchers real- ized that the deductions starting at axioms and following prescribed rules are limiting and introduce imprecise standards in the legal rea- soning. One of the possible solutions is to use relaxed formalisms, such as fuzzy logic [15] or neural networks [16]. Philipps et al. sug- gest that the usual perspective of rules applying linguistic entities (sentences, words, etc.) is not general and abstract enough to express the knowledge [16]. They compare the lawyer's work to the learning and producing results of the neural network. Examples of real-world applications following this scheme are the neural approach to legal reasoning system for family law in Australia [17], modeling the French Council of State decisions in NEUROLEX [18], or a decision support system in the field of insurance using a neural network [19].

Despite all the effort, lawyers were quite skeptical about AI helping in the legal domain, as can be seen in the article "AI in law practice? So far, not much" by Oskamp; Lauritsen (2002) [20]. The main problem seemed to be the lack of legal data, which could not have been applied to train and evaluate the AI models. This, however, changed during the first decade of the new century with the rise of the Internet. New collections of documents and structured databases enabled growth of a data exploration techniques: *data mining* was applied to find implicit information from multiple heterogeneous sources ("Classification System for Serial Criminal Patterns" [21]), *heuristic search* method was used in the case-based reasoning system AGATHA [22], and *text summarization* was applied in extracting the important parts of law from the XML corpus of judgments of the UK House of Lords [23].

During the last few years, the automated tools helping in legal realm have become more

specialized, yet adequately tuned. These include automated tool for finding patents related to a particular topic [24], detection of tax evasion by networks of transactions with their corresponding audit observables [25], and finding and modeling contradictions in judicial statements [26] [27].

Some of the more theoretical researches has adopted *Bayesian net- works* for modeling legal arguments and eventually evidence ([28] and [29]). The *Bayesian network* is a directed acyclic graph with probability tables for each node [29].

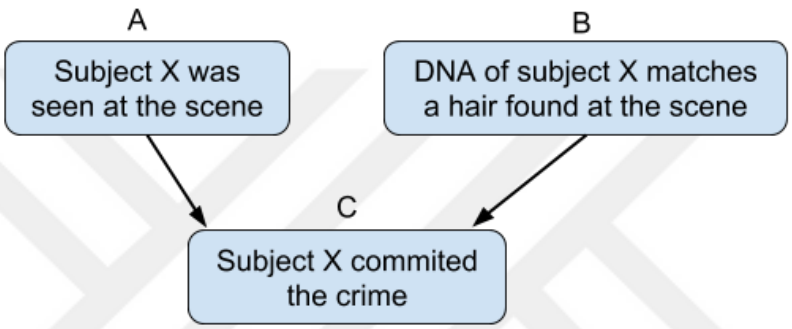


Figure 2.1: Simple example of Bayesian Network.

Each node holds a predicate, which is true with certain probability depending on the value of incoming nodes. For example, a table of the node *C* in the figure 1.1 would describe conditional probabilities of the events $P(C \mid A, B)$ for all possible combinations of values A and B [29]. The *Bayesian Network* enables to visualize the legal arguments and is able to work with uncertainty. The probabilities of unlikely events are especially of great importance since judges need to distinguish between coincidence and purpose.

2.2 ELECTRONIC DISCOVERY

The *Electronic Discovery*, also known as *E-discovery* or *Electronic Data Discovery* (EDD), is a process of searching, gathering, cleansing, and organizing of legal documents in order to find an evidence in a civil or criminal legal case [1]. The field of *Information Retrieval* (IR) is an essential part of the *E-Discovery*, as it provides efficient full-text search among the documents. Conrad mentions how the IR can be supple- mented by AI techniques in his work "E-Discovery

revisited: the need for artificial intelligence beyond information retrieval" [1]:

1. Intelligent relevance feedback
2. E-mail management
3. Social network analysis
4. Data mining and machine learning techniques
5. Anticipatory E-Discovery

The E-mail and Social network have become naturally a source of legal arguments with increasing amount of information stored in such media. Anticipatory E-Discovery is a new practice of corporates to prepare for a possible legal hold on some of their products or intellectual property.

The fourth category includes mainly the clustering and categorization of documents to organize them into separate groups. Research under *Xerox Research Center Europe* introduced a machine learning classifier *CategoriX*, which works with a fixed set of document categories and predicts document's category with probabilistic latent semantic analysis (PLSA) [30]. The mentioned technique is usually applied in the topic modeling of documents and is briefly explained in the section 4.1.4 as it is one of the working attributes of the system proposed here. Research of Noortwijk et al. presents the *Copac* system, which categorizes whole legal cases instead of documents [31]. More- over, they relaxed the set of categories and enabled users to define their own by selecting examples of positive and negative samples [31]. The primary goal of this research is the relevance prediction. How- ever, the categorization of documents is also part of this research, because the preparation for the relevance prediction is labeling doc- uments as case-related or law-related (5.3). The paper "Automatic categorization of case law" compares three algorithms for legal docu- ment classification: k-NN, decision tree algorithm, and a rule induction algorithm Ripper [32]. They conclude the decision tree and Ripper algorithms outperform the k-NN with Ripper being best by recall measure. For the categorization discussed in this research, a decision tree was applied for its satisfactory performance, simplicity, and trans- parentness. Moreover, the ensemble of decision trees – Random Forest– helped us to boost the performance.

The first enhancement of IR – Intelligent relevance feedback – best explains the workflow of the system proposed here. By feedback, Conrad means a consultation of the suggested relevance with the user, which might lead to recomputation [1]. This iterative process has proven to increase the accuracy of the predictions [1]. On the other hand, Zhao et al. show that more than one round of the feedback selection process is not helpful [33]. The proposed system provides a possibility of immediate feedback through relevance settings of individual documents. Each feedback causes an update of relevances within the scope of a particular legal case.

To compare the existing systems with the proposed, it is worth noting that most of them implement a slightly different use case. Their goal is to pick relevant documents out of massive storage of legal texts, in which documents might relate to multiple cases, whereas here the goal is to pick relevant documents out of a relatively small collection that is uploaded to the system by the lawyer. The proposed system is not allowed to share documents among cases nor look for information elsewhere than what is selected by the lawyer. This task is somewhat more difficult since the prediction model starts to learn almost from the beginning with each new case. Most of the similar studies focus either on the case or document retrieval from a global storage, hence the use case of the proposed system seems to be too unusual to be included in the recent public research.

2.3 MEASURING RELEVANCE IN LEGAL DOMAIN

Many studies are concerned with the concept of relevance in the le- gal domain, formalization of its subjective nature and its measuring and evaluation ([34], [35], [36]). The following discussion takes into consideration a human ability to decide the relevance.

When evaluating relevance decided by an automated system, the results should be always compared to the efficiency of humans. Results in IR and E-Discovery are hard to evaluate since there is no ground truth. Even among the humans annotators, the 100% agreement is simply not possible – the relevance is too subjective factor. Roitblat et al. conduct an experiment that compares efficiency of computer classification and manual review [2]. They pick 5000 documents from a collection of 1.6 mil documents (9.8% relevant) annotated by attorneys in the past for other purposes than research. Consequently, a new group of attorneys divided into two teams A and B were reviewing the documents. At the same time, two anonymous E-Discovery systems referred to as C and D were applied to classify all collection of 1.6 mil documents. Precision of the manual reviews (percentage of correctly relevant from all classified as relevant) was 20% and 18%

whereas the systems achieved precisions of 27% and 29%. Recall of the manual reviews (percentage of correctly relevant from all relevant) was 49% and 54% whereas the systems achieved recalls of 46% and 53% [2]. Note that the the recalls are quite comparable, whereas the precision is significantly higher for the classification systems. This research showed that the time invested in development of such systems is worthwhile. The present research will refer to these results in the evaluation chapter (7) to compare the precision and recall to both the human efficiency and state-of-the-art legal classification systems.

In ordinary IR, the precision is usually more significant measure than recall, since the user is often satisfied with the first few most relevant results. In the legal domain, both high precision and high recall are required, sometimes the recall is considered the dominant measure as missing a relevant information could be fatal [35]. Hogan et al. divide the E-Discovery systems to two groups according to the method for balancing precision/recall. First group is the CAHA systems (Computer Assisted Human Assessment), where humans assess and filter the result of the IR system [35]. The IR system returns a result with low precision but high recall and human selection rises the precision. The other systems are HACA (Human Aided Computer Assessment), in which a human provides an annotated samples of documents to improve models of the IR system [35]. Here, the HACA system is a fully automatic system with high precision and high recall. This research presents a system that falls into the latter group.

Opijnen and Santos in their study "On the concept of relevance in legal information retrieval" breaks down the relevance into six concepts, which are necessary to ensure relevant result [34]. Whereas they understand the relevance in context of the IR, Hogat et al. in their study "The centrality of user modeling to high recall with high precision search" relates the relevance to a supervised classification system, which is closer to the system presented here [37].

As the definition of relevance is entirely derived from the user's requirements, Hogan et al. present a User Modeling (UM) approach to achieve high performance in search tasks [37]. The four main concepts of UM are:

6. Use Case – determines the goals of the user and their preference towards precision and recall.

7. Scope – defines the relative amount of concepts that are under- stood by the user as relevant. The scope defines a threshold that tell which concepts are core and which are peripheral.

8. *Nuance* – is a measure determining how relevant are concepts within a document, considering lexical relationships.

9. *Linguistic Variability* – determines the variability of expressions of the concept. The expressions can be lexical or syntactic.

3. DATA CHARACTERISTICS

One of the most important preconditions for building an efficient model is to understand the input data characteristics. People developed numerous models based on text classification that are supposed to classify text of any domain, which is, however, at the expense of the efficiency. The specialized models always have potential to outperform the general one. The real contribution of this work is the exploration of characteristics in the legal data and its application on the process of building a classification model. This chapter will introduce our working dataset, its statistics, and characteristics.

3.1 DATASET INTRODUCTION

The end users of the Legato system are mostly the U.S. lawyers, which influences our choice of dataset. The U.S. legal system is quite specific with its own laws, constitution, trial process and most importantly for us – the terminology. Also, we have to take into consideration specifics in the structure of the documents: medical records, police records, affidavits, emergency call transcripts, etc. We do not claim that our application will not work for other than U.S. legal documents; however, the model might manage to recognize the structure and terminology of the U.S. documents more accurately.

Unfortunately, we have not acquired enough training data from the users of Legato, mainly because the legal cases are ongoing and the documents are confidential, and thus cannot be included in the research. For the same reason, it is hard to find authentic public trial documents on the Internet. We have found a solution thanks to the *American Mock Trial Association* [1] and *Classroom Law Project* [2]. Both associations create legal cases for educational purposes, which then serve as a material for the mock trial competitions. All their legal cases are fictional, though some of them were inspired by real-world situations. Still, any resemblance to actual persons and events is purely coincidental. All 20 cases, used as a training set, were downloaded from two mentioned associations. Not to be biased with one legal area, cases of diverse legal fields are included:

2x School environment

educational malpractice or indifference

1x **Bullying**

intentional infliction of emotional distress

2x Criminal law

assault, battery or murder

3x Inmate partner violence

domestic violence by a current or former partner

3x First Amendment

breach of freedom of speech, religion, press etc.

4x Child neglect

child injury or death caused by parents, baby-sitter or teacher

2x **Sex crime**

sexual harassment and intentional disease transmissions

3x **Tort law**

fault leading to suffer or harm, without breaching any contract

All mock cases have a similar structure: there are approx. 5-8 testimonies of the involved people called affidavits, and the rest of the documents are emails, medical records, police reports, emergency call transcripts, description of the past similar cases etc. The affidavits are up to 10 pages long, usually describing the issue in depth, whereas the other documents are generally one or two pages long. All extra information specific to the mock trials, explanations of the competitions and summaries were excluded. Some extra headers and footers that contain metadata were also removed not to confuse the model training.

During the model building, one needs to take into account that the real cases might be more or less different from the mock cases. For instance, the affidavits are not expected to appear in the real cases as much as in the mock cases. Also, one can expect more emails and papers of unknown structure in real cases. Generally, we found two major shifts:

1. the real cases will most likely contain shorter documents

2. the real cases will contain more unrelated documents This research presents a solution for both issues:

3. The length of a document should be an insignificant param- eter when determining the document's relevancy. Therefore, the following research focuses is on sentences rather than on a document as a whole. In other words, the relevance of a docu- ment is determined by exploring relevances of the individual sentences. As a result, the model is slightly simplified, but the cross-sentence information is not completely ignored, as will be seen in the further chapters.

4. The second issue is caused by the lack of motivation to add unrelated documents during the creation of the mock cases. Still, we can find some documents that are related to the case but are not relevant, e.g., resume of a defendant. A straight- forward solution is to add some irrelevant documents from other cases to simulate the balance in a real situation. However, this adaptation probably will not be necessary as the models need balanced classes for efficient learning, and therefore, a dominance of irrelevant documents would only bias the classification.

3.2 ANNOTATIONS

For the purpose of evaluation of the proposed system, two annotators performed labeling of both the sentences and documents of the dataset. The annotation process was concluded in the Legato system by two annotators. Each document was annotated by one of them and corrected by the other. They assigned every document a label HOT, WARM or COLD according to the relevance. The HOT label signifies strong relevance as it contains essential information needed for defense or accusation at the court. The WARM label means that the lawyer should see the document because it might contain valuable in- formation about the case. The COLD label signifies that the document does not contain any relevant information.

Besides the documents, the parts of the text were labeled as well. Annotators were highlighting passages from the text, which could span multiple sentences. The further processing marks a sentence as relevant, if and only if there is an intersection with some passage annotated as relevant. Only HOT and WARM documents contain relevant parts of the text.

3.3 DATA STATISTICS

As mentioned earlier, the dataset contains 20 cases, which together cover 263 documents. That is 13.15 documents per case on average. The documents contain 234,798 words in total, which means the average number of words per document is 893. Counting only unique words, the vocabulary has a size of 14,518 words which is approx. 1.18 % of English vocabulary according to the *Oxford English Dictionary* [38].

Table 3.1: Most frequent words in the dataset with and without stop words.

All words		**Without stop words**	
word	**count**	**word**	**count**
the	10825	school	855
to	6632	would	683
I	6274	one	566
and	5912	time	497
of	5479	1	476
a	4896	said	451
in	3603	told	420
that	3499	could	409
was	2846	get	391
for	1942	like	386

The left part of the figure 3.1 presents counts of the most frequent words in the documents. It is not a surprise that all top ten words are common stop words of English. The stop words are usually removed before any further processing of the text; however, they might carry valuable information too. For instance, recognition of the author or formality level can be determined based on the stop words usage. One can notice that the usage of stop words is not different from the general English.

The right side of the table shows most frequent words after the stop words removal. Here, the frequencies are clearly influenced by the cases from the school environment. Next, the words "said" and "told" indicate that the documents often contain reporting of events and conversations. The other words again fit more or less in the general English. The word frequencies show that no strange artifacts in the text bias the vocabulary.

The following statistics explore the *keywords* of the collection, i.e., words that occur more frequently than usual. The method used is explained in the section 4.1.1. During the scoring, the Reuters articles from *NLTK* library served as the reference corpus. The keywords extracted from all documents are: city, school, know, got, joey, just, police, princess, time, person, really, like, and child. Again, the school environment plays an important role.

Out of the 263 documents, 107 are HOT, 85 WARM and 71 COLD, which confirms our idea about the dominance of relevant documents. Interestingly, the length of a document in sentences correlates to its relevance with 76 sentences in average in HOT documents, 53 in WARM and 23 in COLD. This might be explained by the dominance of affi- davits, which are usually relevant and people there explain incidents in long sentences. The most common words in HOT documents are the reporting words, such as "said", "told" or "would". In most common words of WARM documents appears the "police", "time" or "new". COLD documents, on the other hand, contain a lot of numbers ("1" is, in fact, the most common one) and then "metro", "officer" or "city".

3.4 TYPES OF DOCUMENTS

When looking at the documents, one can recognize some common features, which groups them into distinct classes. The testimonies are one large group of documents, yet they are closer to medical records, bills, call transcripts, police records, etc. than to the rest of the documents. The other distinctive group consists of documents that do not relate to the case itself but rather explain the law, policies, and past legal cases. No people, location, dates or other named entities appear in these documents. However, it is not true, that they are not relevant to the case. Usually, they consist of some passage from law code that relates to the incidents discussed in the case, which is likely a relevant piece of information for a lawyer.

In this research, the first group of documents directly involved in the case will be referred to as the *case-related* documents, and the second group will be referred to as the law-related documents. The precise definition and classification are elaborated in the section 5.3. Besides the mentioned two groups, the dataset contains few other documents that do not seem to fit in either of them. Nevertheless, none of them accounts for a group large enough, and therefore, they are considered as outliers and assigned to one of the existing groups. These include pictures, tables, or newspapers.

4. NLP TOOL SELECTION

The amount of text in legal documents can vary greatly. In the presented dataset, most of the documents stretch across one page, though the average number of pages is six and some documents contain up to ten pages. The typical approach of text processing is to convert it into a document-term matrix that contains the counts of term occurrences in each document. Unfortunately, the problem with this approach is that the long documents are represented by a vector, which *information gain* is rather low. High *information gain* implies that the numerical representation captures the general idea, the purpose of the document, and characteristics of similarity/dissimilarity with other documents. To address the issue, the following research represents a document with specific features extracted by one or more NLP tools rather than plain words of the document. One can look for the most interesting pieces of information in the text, such as people, locations, dates, crime- related words or email addresses. Nowadays, there are numerous services providing extraction of such features. This chapter presents some of them and collects results from testing on one of the mock legal cases. Simultaneously, a survey on text processing was conducted, in which participants labeled features from the text in a similar way as the NLP tools. Consequently, the results of the NLP services and a survey are compared to choose the most suitable for further research. It is important to note that the field of NLP feature extraction is a well-studied topic and the tools produce very stable results, probably using somewhat similar algorithms (which are described in the following section). Even though we could dedicate part of this work to developing well-suited algorithms for NLP feature extraction, we decided to take advantage of the existing tools and put the focus on processing and application of the features to the high-quality data mining and machine learning algorithms.

4.1 NLP ALGORITHMS

First, let us introduce the features and well-known algorithms for their extraction from text in general. Note that the further mentioned tools might use modified or different techniques for extraction of the features.

4.1.1 Keywords

Keywords are people, places, words, or ideas that are understood as important in the given context. In the presented research, the con- text is a document, and thus the keyword is expected to reflect the real semantic essence of the document. Then, the *keyness* is a quality measure of the importance in the given text. [39, ch. 4] We do not restrict the keyword to be one word because most of the tools are able to extract multi-word keywords, also called *key phrases*. By combining words together, the phrases usually gain a new meaning that cannot be inferred from the individual words. Therefore, if we assumed only one-word keywords, we would never find the keyness hidden in the phrases.

Two factors drive the process of identifying keywords. First, the more often a word occurs in the document, the more likely it is a keyword. And second, the more often a word occurs generally in a speech, the less likely it is a keyword of any document. [39, ch. 4] The second factor ensures that words we use very often in the speech, such as prepositions, conjunctions or the most common nouns, are not considered as keywords, even though their occurrence is frequent. Common implementation of the idea is called the *TF-IDF* algo- rithm [40, ch. 6]. The weight of a term is defined by its *term frequency* and *inverse document frequency*. The term frequency $\mathrm{tf}_{t,d}$ is simply a number of occurrences of the term t in the document d. The *inverse document frequency* is defined as

$$\mathrm{idf}_t = log \frac{\mathrm{N}}{\mathrm{df}_t} \tag{4.1}$$

where N is the total number of documents and df_t is a *document frequency*, which is equal to the number of documents containing term t. It is crucial to take a logarithm of the expression because the *inverted document frequency* increases dramatically with enlarging collection of documents, and at the same time, it needs to be comparable with the *term frequency*, which is rather small.

The tf-idf weight is computed as follows [40, ch. 6]:

$$\text{tf-idf}_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t$$

The *term frequency* ensures higher weight for more frequent terms in the document and *inverse document frequency* ensures higher weight for more obscure words. Now, $\text{tf-idf}_{t,d}$ can be a good approximation of the keyness measure and is often used in the keyword extraction. The last step is

to establish a threshold value that distinguishes keywords and non-keywords by comparing it to the tf-idf values of the terms.

4.1.2 Named Entity Recognition

To extract valuable information from the text, it is necessary to include *named entities* (NEs). They are entities that can be referred to with a proper name [41, ch. 21]. In the speech, NEs are always noun phrases representing entities such as people, places, things, organizations, temporal or numerical expressions or events.

Let us pinpoint how important the named entities are in the con- text of legal cases. Consider a typical use case of our application: a lawyer is searching through hundreds of emails with different senders and receivers. The name of the victim, place, and date of the crime are known. If the lawyer is forced to search through the documents man- ually, they will filter the emails by occurrences of the aforementioned words in the text. All of the words are named entities, which can be extracted from each document automatically and eventually lead the algorithm to increase the relevance of documents accordingly. Thanks to the named entities, the algorithm performs the classification in the same manner as the lawyer would do, which is the goal. The academic approach to find a *named entity* is often based on the

statistical sequence model [41, ch. 21]. An entity type and boundaries are found with one pass over the text according to a set of rules. The rules include *Part-of-Speech Tagging*[1] , which helps to find the noun phrases, and therefore, the boundaries. Next, the *word shapes* are ex- amined, which are simple string representations that generalize a group of expressions. For instance, replacing any letter for X and any digit for d, one can describe a unique word shape dd XX, which rep- resents hours of the day written in US format, such as 12 am or 03 PM. Other rules include prefix matching or dictionary of predefined named entities.

In comparison to the academic approach, the commercial approach is based on the combination of updated dictionaries, rules and super- vised machine learning [42]. In the world of Big Data, it has become easier to collect large dictionaries of given names, family names, organizations or geological places (gazetteers). The named entity recognition is performed by multiple passes over the text data. First, the high-precision but low-recall rules are applied, then more entities are added by substring search of the previous entities. In the next phase, the entities are consulted

with the dictionaries and only then, the statistical sequence model is applied considering all the features from the previous stages [41, ch. 21]. Further in the thesis and the application, the following set of *named entities* is considered:

- Person
- Organization
- Location
- Time
- Email Address
- Crime
- Health Condition

4.1.3 Relations

Once the extraction of the *named entities* is done, one can start looking for the relationships between them. In the legal environment, we can illustrate the added value of a relation by a testimony, in which a witness tells about both a person and a location; however, we are interested in the document only if there is a relation between these two.

Most of the *relations extraction* algorithms are focused only on bi- nary relations. Multi-entity relations would be very expensive to ex- tract. Often, a representation of the Resource Description Framework (RDF)[43] is applied to define the relation as a triple

<subject, predicate, object>

where the *subject* is an entity that performs an activity specified in the *predicate* with respect to the *object*. In the presented context, the *predicate* is the type of the relation; it can be, for example, "employed- by", "built", "son-of" or "cheated-on".

The following groups of algorithms can be applied to extract the relations: **hand-written patterns**, **supervised machine learning**, **semi- supervised**, and **unsupervised**. We will introduce the first two of them.

Hand-written patterns

The work "Automatic acquisition of hyponyms from large text corpora" by Hearst [44] describes the first idea of using patterns to extract relations. The entity names are ignored, and only a set of patterns recognizes a relation. The patterns are based on the lexicology (POS tagging), the syntax (syntax tree) and the type of the entity (person, location, etc.). For example, the pattern [41, sec. 21.2]

$$NP_0 \text{ such as } NP_1\{, NP_2..., (\text{and}|\text{or})NP_i\}, i \geq 1$$

is able to catch the relation hyponym for many occurrences. For in- stance, a sentence

... by any poison, such as Cyanide, Arsenic or Ricin ... would generate relations

- <poison, hyponym of, Cyanide>,
- <poison, hyponym of, Arsenic>, and
- <poison, hyponym of, Ricin>.

An example of a pattern using entity type

PERSON (named|appointed|chose|...) **PERSON** Prep? **POSITION**

will determine a relation "named"[41, sec. 21.2].

A significant disadvantage of the pattern-based recognition is that the patterns need to be written by hand, which takes a lot of work. On the other side, the precision of such method is usually high since humans write the patterns.

Supervised machine learning

The second approach suggests to annotate a corpus of relations and entities and consider it as a training set in the machine learning. More precisely, features are extracted from the sentences and are served to the machine learning model as an input. With the relation types as the output labels, the model is prepared to be trained automatically [41, sec 21.2].

The features can consist of unigrams or bigrams of the words in the sentence, especially between the subject and object. Next, the named entity types, POS tags or features from the syntax tree can be included as well. Sometimes the number of words between the subject and object or stemmed version of the words is also considered [41, sec. 21.2].
The algorithm uses two models for the relations extraction. The first decides whether there exists a relation between two given entities, and the second one recognizes its type. First, the algorithm finds all pairs of entities in the sentence, test it with the first model, and only the positive results are tested on type with the second model [41, sec. 21.2].

The supervised method has a potential to be very accurate on general inputs; however, to achieve better results than pattern-based models, an extensive training dataset is required. Therefore, the *Dis- tant Supervision for Relation Extraction* was invented to solve the problem [45]. It is an elegant way, how to obtain a large training dataset. The DBpedia and Google Knowledge Graph[4] provide enough information about some of the relations between real-world objects. The algorithm can extract full relations with a subject, object and relation type. To gather the training data, the algorithm searches a page about the subject on Wikipedia and finds a sentence with the specified object. Then, it is ready to adopt the text as an input data for the training phase. This approach delivers a large number of sentences, in which the confidence about the relation type is high, and thus the training dataset is rich enough to create a reliable model.

4.1.4 Topic Modeling

The *Topic modeling* is a statistical approach for finding general topics that appear in the document. The keywords are already able to dis- cover some trends in a document; however, unlike the topics, they have to appear in the text.
Before the topic recognition, a fixed set of topics have to be defined. The more advanced algorithms create a hierarchical topic structure rather than a simple list. The usual process of topic modeling is to assign probabilities of all topics to every word in the dictionary. Then, considering a document as a bag-of-words[5], one can compute the overall probability that a document has a specific topic only from its words.
One of the well-known algorithms for the topic modeling is called the *Latent Dirichlet allocation* developed by Blei; Ng; Jordan in 2003 [46]. The model assumes that every document has a fixed

distribution of topics it belongs to. Furthermore, it assumes that the words of a document were withdrawn from a specific distribution defined in the following way.

1. Pick a topic distribution of each document according to a Dirichlet distribution.
2. Pick a word distribution for each topic.
3. For all words in all documents, generate the word as follows:

(a) Pick a random topic according to the distribution of topics for the document.

(b) Pick a random word from the distribution of words for the topic chosen in step a).

The generation scheme tells us the probability theory for documents and topics; however, it does not tell us how to recognize the topics from the fixed documents. To apply the theory, we need to reverse the process and assume that the documents were created by this generation process, and eventually guess the topic. One of the iterative methods, which achieves this goal, is called the *Collapsed Gibbs Sampling* [47]. Put simply, the initialization stage picks a random topic for each word in all documents, and each next stage goes over every word and changes its topic by computing the probabilities, assuming that all the other words are correctly labeled.

4.1.5 Document Category

There are two types of categories of interest. The first is more related to the structure of the text and tells us where the document is coming from. Such type can be, for instance, email, medical record, police record, or affidavit. The second type is related to the meaning of the text. Semantically, it is very similar to the topics introduced in the section 4.1.4, and the only difference is that the category is the most dominant topic.

4.2 INTRODUCTION OF THE TOOLS

4.2.1 NLTK: The Natural Language Toolkit

As a baseline for the commercial NLP tools, the open source python framework *NLTK* is included[48]. Although it was initially developed for educational purposes, it is now broadly used

by researchers and public to process the textual data in python.

NLTK consists of multiple modules, which can be applied in a pipeline during the text processing. The basic modules apply the tokenizer, POS tagging, and syntactic analysis. More advanced modules are dealing with the information extraction and knowledge representation. The framework comes with large preprocessed corpora that can be used as reference corpora during the keyword or NE extraction [49].

The commercial NLP tools do not need any settings since they are already configured, and in fact, the configuration is the key to their success. Nevertheless, the *NLTK* is more a framework than a tool and one has to take the time to set the parameters correctly. It also brings a significant advantage of high configurability. The next two paragraphs explain the approach of extracting the keywords and entities in the *NLTK* for the purpose of the comparison.

Firstly, we removed the stop words and filtered only meaningful words. The list of stop words was used as defined by the *Scikit-learn library*[6]. To remove the nonsense words, we compared them against a dictionary of the *Brown corpus*, which is the first text corpus of Ameri- can English of about 1 million words [51]. Next, we filtered only nouns and finally, computed the *TF-IDF* score against the *Reuters* reference corpus [49, sec. 2.1], which contains more than 10,000 news articles with 1.3 million words. We have chosen the journalistic context because it was the closest domain to the legal environment. This procedure outputs a score for each word in a document, and the final keywords are those, which *TF-IDF* score exceeded a threshold of 0.15.

The entity extraction requires fewer settings since the NE tagger is already trained by the library [49, sec. 7.5]. We split the text into sentences and consequently sentences into tokens[7]. Next step is to extract the part-of-speech tags for each token. The tokens together with the *POS tags* are the input data into the *NE chunker*, which can consequently mark words as named entities and also determine its type.

4.2.2 Google Cloud Natural Language

Google has a long-standing experience with the text processing, especially in the field of information retrieval. The company decided to provide their knowledge of semantic text analysis in a service called *Cloud Natural Language* [52]. Google makes no secret of their approach: the same Deep Learning models that power the Google Search are also employed in the NLP service.

The tool can recognize entities, syntactic structure of sentences, category and sentiment of the text. The sentiment analysis has recently become a point of interest for companies as they want to know public opinion of themselves. However, the sentiment in legal documents is usually neutral and does not play a vital role in the relevance classification. Therefore we decided not to consider sentiment as a feature.

The *Cloud Natural Language* tool also adds a score to every entity and category so that a user sees how confident the service is about each feature. Moreover, the service replies with a Wikipedia article for every entity, if it exists. Another useful feature is the Translation API, which converts text among many different languages and is able to recognize sources of an unknown language. Worth noting that the input does not have to be a pure text but also speech in the audio format or text in a picture.

4.2.3 Watson Natural Language Understanding

In March 2017, the IBM announced retirement of the *AlchemyAPI* service, which was designed to understand the semantics of text and image by advanced techniques [53]. One component of the service was also the *AlchemyLanguage*, providing all NLP features, such as entity, sentiment and topic recognition. As a replacement, a new service emerged: the *Natural Language Understanding* service [54].

Besides typical NLP features, such as keywords, entities, relations, category, sentiment or topics (which are called concepts in the *NLU*), the service provides emotion recognition (joy, anger, fear, etc.) and metadata recognition. The latter includes information about the author, title, prominent page image, and publication date. Unfortunately, this feature is available only for HTML pages, which is usually not the format of legal documents, hence we will not take advantage of the metadata recognition.

One of the most promising features of the *NLU* service is the model customization. The service always recognizes NLP features with respect to some general language model, which consists of words from all aspects and domains of our language. This might be a disadvantage, especially when the documents are always of one type and contain words from a specific domain. For instance, the general model would always consider law, justice or sentence as keywords in the legal documents; however, they might appear in most of the documents and become not relevant anymore in the

legal domain. The *NLU* service enables a user to create its own language model and load it into the cloud. More than one model can be active, and the user can switch between them before each query.

4.2.4 Aylien

Aylien is a software package of information retrieval, machine learning, and natural language processing [55], which provides competitive features to the mentioned NLP services from Google and IBM.

In addition to the classical NLP features, *Aylien* provides summarization, which essentially picks the most important sentences from the text. Next innovative feature looks for related phrases, which are semantically as close to the original expression as possible. Another advantage of *Aylien* is that it includes time-based entities in the named entity recognition, specifically, the dates. The previously mentioned services surprisingly ignore the time and date expressions most of the time.

Next compelling feature of *Aylien* is the hashtag suggestion, which turns out to be the topic modeling as we described it in 4.1.4. Next feature – the article extraction is another name for the metadata recognition, applied by IBM (4.2.3). Genuinely new feature is the aspect-based sentiment analysis, in which the sentiment is recognized for each aspect, not generally for the whole text. For example, an analysis of hotel reviews can discover angry opinions on Wi-Fi signal but an excellent rating of the staff.

4.3 COMPARISON

To compare the tools, we had to restrict the evaluation set only to the common features included in all the tools. Specifically, the *keywords* and *named entities*: persons, organizations, and locations. Without any comparison, we also evaluated the relations, which were supported only by *Watson NLU*, and dates supported only by *Aylien*.

4.3.1 Testing data: Mock legal case

Indeed, we could not use any real-world case for our testing and analysis due to privacy reasons. Therefore, we analyzed a mock legal case called *Davis v. HappyLand Toy Company*[56] created by the *American Mock Trial Association*.

Briefly described, the case puts a father of a child, who was killed by swallowed part of a toy,

against the company that produced the dangerous toy. For our use case, we picked five most interesting and distinct documents:

An email of the inter-company communication about the sub- stances used in the toy.

An affidavit (testimony) of the father.

An affidavit of the babysitter, who looked after the child at the moment of the fatality.

Medical record from the autopsy of the child.

Journal paper about the origin, behavior, look, and impacts on a human body of the substance used in the toy.

4.3.2 Survey

The survey was created so that no prior knowledge of the law is necessary. The questions are simple and clearly stated. Some of the documents are several pages long, and thus the survey can be saved and finished later. The respondents were not provided with any extra information about the legal case so that the results are comparable with the NLP tools, which have no information as well. Five developers of the Legato system completed the survey.

We asked the same set of questions for every document. First two questions asked about the keywords. We have split the one-word phrases and multi-word phrases and required at least ten entries for both. Next, the respondent is supposed to type in people, organizations, and locations that appear in the text. Furthermore, the survey asks for relationships between people, dates, and event related to each date. Next two questions were related to the semantics and were not used to evaluate the tools. We asked the respondents to copy a short passage from the text that could be used as evidence at the court. The second question asks for a summary of the whole document. A comprehensive information about the survey are included in the attachment A.4.

4.3.3 Statistical Comparison

Since only five respondents completed the survey, we decided to unify results for each answer. More precisely, an entry was considered as a keyword (person, organization, etc.) if and only if at least one respondent typed it in the survey. The standard approach how to compute agreement between

two sources is the *Cohens kappa measure* [57]. The problem with *kappa measure* is that the sources assign a fixed number of items to classes, which is not the situation here. In this case, the number of items is variable, and there are no classes. In contrast, here the agreement between the sets is of interest. Therefore, the analysis applies a measure for the distance between sets, the *Jaccard index* [58], defined as follows:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Since the union is always larger than or equal to the intersection, it holds for the result that $0 \quad J(A, B) \quad 1$ for any sets A, B. When $J(A, B) = 1$, the sets are identical and when $J(A, B) = 0$, the sets do not share any item.

The next step is to define when two items (keywords and entities) are equal. We have defined two approaches in comparing the equality: *strict approach*, and *subset tolerance approach*.

Table 4.1: Jaccard index of the NLP tools evaluated against the survey with strict string comparison.

Feature	Tool Document	NLTK	Google	Watson	Aylien
Keywords	Miller affidavit	0.11		0.11	0.02
	Davis affidavit	0.06		0.03	0.03
	Email	0.05		0.24	0.20
	Toxicology paper	0.00		0.02	0.03
	Medical record	0.09		0.04	0.05
	Average	**0.06**		**0.09**	**0.07**
People	Miller affidavit	0.00	0.29	0.54	0.38
	Davis affidavit	0.00	0.13	0.25	0.24
	Email	0.29	0.44	1.00	0.43
	Toxicology paper	0.00	0.00	0.29	0.00
	Medical record	0.00	0.27	0.71	0.57
	Average	**0.06**	**0.23**	**0.56**	**0.32**
Organizations	Miller affidavit	0.07	0.40	0.50	0.29
	Davis affidavit	0.06	0.17	0.33	0.17
	Email	0.00	0.25	0.50	0.00
	Toxicology paper	0.00	0.33	0.33	0.14
	Medical record	0.00	0.00	0.00	0.00
	Average	**0.03**	**0.23**	**0.33**	**0.12**
Locations	Miller affidavit	0.00	0.06	0.00	0.00
	Davis affidavit	0.00	0.00	0.00	0.00
	Email	0.00	0.00	0.00	0.00
	Toxicology paper	0.00	0.00	0.00	0.00
	Medical record	0.14	0.00	0.00	0.00
	Average	**0.03**	**0.01**	**0.00**	**0.00**

Table 4.2: Jaccard index of the NLP tools evaluated against the survey with subset tolerance in string comparison.

Feature	Tool / Document	NLTK	Google	Watson	Aylien
Keywords	Miller affidavit	0.30		0.47	0.57
	Davis affidavit	0.35		0.26	0.33
	Email	0.14		0.57	0.50
	Toxicology paper	0.09		0.37	0.25
	Medical record	0.23		0.24	0.15
	Average	**0.22**		**0.38**	**0.36**
People	Miller affidavit	0.79	0.32	0.85	0.94
	Davis affidavit	0.64	0.29	0.63	0.47
	Email	1.00	0.56	1.00	0.71
	Toxicology paper	0.00	0.06	0.29	0.29
	Medical record	0.57	0.40	0.71	0.83
	Average	**0.60**	**0.32**	**0.69**	**0.65**
Organizations	Miller affidavit	0.36	0.40	0.50	0.71
	Davis affidavit	0.06	0.17	0.33	0.17
	Email	0.50	0.50	1.00	0.50
	Toxicology paper	0.29	0.33	0.33	0.29
	Medical record	0.16	0.00	0.00	0.00
	Average	**0.27**	**0.28**	**0.43**	**0.33**
Locations	Miller affidavit	0.27	0.61	0.67	0.29
	Davis affidavit	0.00	0.10	0.00	0.00
	Email	0.00	0.00	0.00	0.00
	Toxicology paper	0.00	0.00	0.00	0.00
	Medical record	0.14	0.00	0.00	0.00
	Average	**0.08**	**0.14**	**0.13**	**0.06**

In the *strict approach*, all items are converted to lower case, so that names, organizations, locations and other items are compared regardless of the capital letters. Apart from that, no changes are made to the items, and the comparison is based on strict string equality. As a consequence, for instance, the full name and given name are not equal, even if the text represents the same person. E.g. for sets A = "Joey Davis", "Brett" and B = "Joey", "Brett" the Jaccard index is

$$J(A, B) = \frac{|\{\text{"Brett"}\}|}{|\{\text{"Joey Davis", "Joey", "Brett"}\}|} = 1/3$$

Table 4.1 shows *Jaccard indices* using the *strict approach* for each com- bination of tool, feature and document compared with the ground truth, which is the survey output. The "Average" row displays the mean over all documents for each tool and feature.

The *subset tolerance approach* converts the items to lower case in the same manner as the *strict approach*. Then, two items are equal if one is a subset of the other or they are strictly equal. To keep the construction of unions and intersections simple, we include both superset and subset in the union as well as in the intersection. E.g. for sets $A = \{$"Joey Davis", "Brett"$\}$ and $B = \{$"Joey", "Davis"$\}$ the Jaccard index is

$$J(A, B) = \frac{|\{\text{"Joey Davis"}, \text{"Joey"}, \text{"Davis"}\}|}{|\{\text{"Joey Davis"}, \text{"Joey"}, \text{"Brett"}, \text{"Davis"}\}|} = 3/4$$

Results of the *subset tolerance approach* are displayed in the figure 4.2 in the same form as in 4.1. One can notice that the score is always equal or higher compared to the *strict approach*, which is a natural consequence of the approach definitions.

Keywords

All tools except for the *Google Cloud Natural Language* service provide the keywords extraction. The *Jaccard index* is generally very low in the case of the keywords. There is usually a lot of words to choose from, and even the respondents agreed only on few of them. One can notice the best scores in the case of the email due to the short length of the text. The *Watson NLU* achieved the best score; however, it did not perform much better than our baseline *NLTK*. *Aylien* scores high in the *subset tolerance approach* because it extracts very long keywords, which have a high probability of being a superset of some of the real keyword.

People

The scores for the people extraction were surprisingly high. In the *strict approach*, the *NLTK* failed because of its limitation of recognizing only one-word items. People names mostly appeared with both given and family name and the *NLTK* recognized the name separately as two items. On the other hand, when the subsets are tolerated, the *NLTK* achieved a satisfying result. The winner of this

category is the *Watson NLU*, which is superior for every document and approach. *Aylien* service achieved a better score than *Google CNL*, especially in the case of the longer documents.

Organizations

The *NLTK* achieved the lowest score for the same reason as in the case of the people extraction since the names of organizations also consist of more than one word. In the strict variant, all other tools performed significantly better than the baseline, yet the best score again achieved the *Watson NLU* service. In this category, *Google CNL* outperformed the *Aylien* service in the *strict approach*, though we can see a reversed result in the *subset tolerance approach*. One of the possible explanations is the fact that *Google* can precisely determine borders of the entities but the words are not so often shared with the survey output. In comparison, *Aylien* usually fails at determining the borders since the organizations often consist of many words but the items contain the right information in the subsets. Since the Medical record, unfortunately, does not contain many organizations, none services were successful.

Locations

The extraction of the locations was quite chaotic and diverse by the respondents as well as by the tools. The location can be understood as a named entity which is unique in the world, such as Hype park in London, 107 Myers street or Moscow, whereas someone understands the location as any expression specifying the environment, such as liv- ing room, school or a car. The *Google CNL* mostly recognized the latter type and the other tools recognized the former. Moreover, there were not enough location entities in the documents to draw a meaningful conclusion from the results. In the strict variant, we ascribe the win of the *NLTK* service to a coincidence, as its simple model extracted many words starting with capitals, and hence more people than locations appeared in the output set. In the subset tolerance variant, *Google CNL* and *Watson NLU* achieved very similar score; however, the Watson generated approximately ten times fewer items, which brings us to a conclusion that the discriminative power of the *Watson NLU* items might be higher.

Dates

As the *Aylien* service is the only one that is able to extract the time- based entities, no comparison was necessary. Still, it is valuable to see how precise is the extraction, and therefore, we present the *Jaccard index* computed from the *Aylien* output and the survey data.

Table 4.3: Jaccard index of the time-based entities extracted by Aylien.

Document	Jaccard index
Millers affidavit	0.07
Davis affidavit	0.13
Email	0.00
Toxicology paper	0.00
Medical record	0.00
Average	**0.04**

The *Aylien* achieved non-zero results only in the longer documents, and even then, the scores are under our estimation. As an example, a simple date in form "24. 12. 2017" is recognized as a phone rather than a date. Here, it strongly depends on the spaces: after the removal of the space before the month ("24.12. 2017") the day and month are finally recognized as a date; however, the whole string is still reported as a phone. On the other hand, here the analysis takes into consideration only the dates, while *Aylien* recognizes times and other expressions, such as "last week", as well.

4.3.4 Analytical comparison

Aside from the statistical evaluation of the tools accuracy, it is also important to manually analyze the output and check whether it is semantically correct. This section includes a discussion about the tools' characteristics, their benefits, and drawbacks.

NLTK

The previous section already outlined two drawbacks of the *NLTK*: a one-word limitation and simple NE model. The first problem is visible in both affidavits, where the toy name appears under the

title “Princess Beads”, which is an obvious keyword in this context. Interestingly, the *NLTK* extracted both parts: “princess” and “beads” as keywords since the words are unique by itself. However, the key phrase was missed. As the *NLTK* is highly customizable, we believe that the multi-word phrases extraction is, indeed, possible, but it would require an extra amount of work.
On the other hand, the keyword extraction performance was com- parable to the commercial tools. *NLTK* performed considerably worse in the entity extraction. Besides the separation of the given and family name, it also falsely marked “Liquid”, “Myth” or “Identification” as people, probably because of the capital letters. One can notice many false positives in the Medical record, where some words were written in the capitals and *NLTK* recognized all of them as organizations, e.g., “BODY”, “AND” or “OF”.
Although the *NLTK* provides a satisfactory baseline for the NLP
feature extraction, one can see a noticeable line between its hand- written patterns and the supervised machine learning models, which are used by the other tools.

Google Cloud Natural Language

The approach of the Google is little different in the following way. Together with the entity, it outputs also a position in the text, and no duplicates are discarded. Google does not understand the extraction as choosing the most identifying words, but more as highlighting of the words in the text. Therefore, the *Google CNL* usually produces a larger output than others. The true positive rate is almost perfect, for example, in the Millers affidavit, all people were correctly recognized. Even more surprising is that “Hillary Davis” appears only once in the text and the service was able to link the name to many occurrences of her given name. The problem is that words, such as “kids”, “one”, or “friend” are recognized as well, which is rather disturbing in the present use case.

Watson Natural Language Understanding

The *Watson* approach is slightly more moderate. Minimal false pos- itives are introduced as *Watson* outputs a word only in case of high confidence. Similarly as *Google*, *Watson* recognized all people in the Millers affidavit including full name of “Hillary Davis”. Due to the conservativeness, only the family name of “Chase Tuchmont” was recognized since the “chase” is an ordinary word in English as well. However, the overwhelming majority of the recognized entities

makes perfect sense and are truly people, organizations or locations. There- fore, the *Watson NLU* service is not only superior in the statistical evaluation, but also in the analytical overview of the semantics.

Aylien

One can see yet another approach by the *Aylien* service. Keywords are understood as complex pieces of information of arbitrary length. As a result, service incorrectly extracted a key phrase "Davis and that I wanted to speak with someone about Princess", which seems slightly awkward. This was a source of a low score of *Aylien* in the keywords category. On the other hand, *Aylien* excelled in the extraction of people. Despite a few mistakes, mostly recognizing numbers as people, the service was able to keep high true positives and low false positives, including organizations and locations.

5. CLASSIFICATION

Exploring the data in the chapter 3 helped to clarify enough the characteristics of legal documents to design models for the relevance classification. The analysis discovered that the focus should be put on the sentences instead of documents and that the recognition of a document type before relevance prediction might be a beneficial step.

The first section (5.1) presents how the NLP features are encoded into attributes applicable to the models. Next sections describe the system of models that predicts the document relevance. Due to a large number of features, the system is structured into layers, where the top-level models make predictions based on the outputs of the bottom-level models. The section 5.2 introduces the bottom-level models, which are specialized on each feature separately and are able to understand the word embeddings. The top-level layer is formed by three machine learning models that are consecutively making decisions about the documents in order to output the final relevance. The first is the *Law-Case classifier* (5.3), second is the *Sentence classifier* (5.4) and the last is the *Document classifier* (5.5). Each of them needs a different set of inputs and applies a different estimator. The following figure visually explains the system of models. For further details about the individual classifiers, see the related sections.

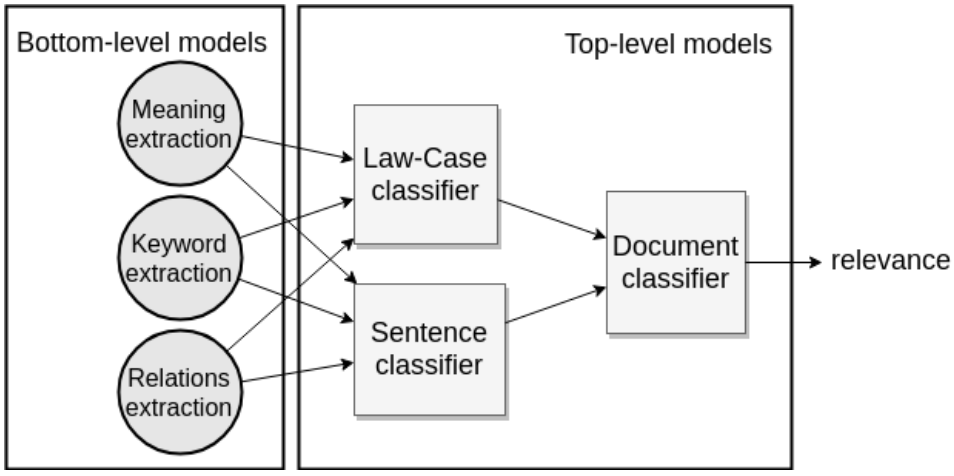


Figure 5.1: Diagram of models

5.1 ATTRIBUTES

To achieve a successful classification, one of the most crucial steps is to convert the data into suitable attributes that will be helpful in determining the classes. Technically, all machine learning estimators are built to process numbers as input variables. Since the working dataset consists of a pure text, the first step needs to convert all key- words, entities, relations, syntactic structure, and other features into the numerical representation. During this process, one has to keep in mind the primary purpose: the numbers must internally capture information about the relevance.

Based on the arguments made in the chapter 3, the working set of data samples consists of sentences rather than documents. As a result, the representation of the attributes and the process of classification itself become simpler.

5.1.1 Sentence Meaning Embedding

When extracting a meaning of a sentence, one will encounter two major problems to be solved: 1) how to represent the meaning of a single word and, 2) how to combine word meanings. The first problem is also known as the *Word Embedding* problem [59]. The simplest solution is to assign unique numbers to all words and train classifiers on the numbers instead of words.

{"knife" → 1, "gun" → 2, "wire" → 3, . . . *}*

This approach does not usually perform well since the classifiers inter- pret a number as a quantitative, not nominative, attribute. As a result, the classifier understands two words with close numbers as close by the meaning as well. Since the words are not generally ordered in any way, there is undoubtedly a considerable misinterpretation. The next solution is often applied in information retrieval systems. Instead of one number, the word is encoded by a vector of length equal to the size of the vocabulary. Then, the embedding contains zeros in all elements except for the index of the word it represents. This approach is called *One-hot encoding* as defined by the Scikit-learn machine learning library [1].

{"knife" → (1, 0, 0, 0), "gun" → (0, 1, 0, 0), "wire" → (0, 0, 1, 0), . . . *}*

Now, a vector uniquely identifies a word and since the vectors are not quantitatively comparable,

the problem with the previous approach is eliminated. This solution can be further improved by counting the words in the training documents, TF-IDF weighting or co-occurrence matrices [59].

The third and final solution goes further in the embedding of semantics. Recent research by Mikolov et al. [60] introduced a revolu- tionary algorithm *Word2vec*, which uses a neural network to create a vector of fixed length that compresses semantics of the word it rep- resents. Each dimension of a vector represents some concept, which can be somehow quantified, such as masculinity, age, or liveliness [61]. The vectors are capturing semantic regularities, and therefore, the classifiers have a potential to perform better even for unknown words. This embedding satisfies our ultimate assumption that close vectors are also close by the meaning of their underlying words.

Both the algorithm and the trained embeddings are publicly avail- able. Though there is a possibility of applying the algorithm to com- pute our embeddings, the obtained dataset is too small to train a model reliably, and thus it is more efficient to adopt the dataset based on available models trained thoroughly on a large number of samples. The dictionary adopted here is trained by the Google company and is available in their official code archive [62]. The pre-trained model contains 3 million unique words of 300-dimensional vectors, which was trained on a 300 billion word news dataset. For our purpose, a trimmed dataset of 300 thousand unique words without the phrases is sufficient.

The second problem relates to the fact that the classifiers usually need fixed input, but the computed vectors exist for each word of a sentence. Researchers Vedantam et al. [63] applied a simple average of *word2vec* vectors to the words in a phrase to get the final embedding. The idea can be extended to whole sentences by computing an average of all words in the sentence.

5.1.2 Keyword embedding

Next attribute is a vector computed from the keywords that appear in a sentence. The NLP feature extraction determined keywords in the context of the whole documents, which means the first task is to find the working set by intersecting the sentence with the keywords. In the process of intersecting, the *subset tolerance approach* is applied, as defined in the section 4.3.3.

The next step is very similar to the approach mentioned in the previous section – the words are converted into an embedding by the *word2vec* dictionary and then averaged into one 300 dimensional vector. Indeed, the sentence meaning vectors contain the keywords as well, and thus

some information is redundant. However, the choice of right words plays a crucial role in the influence on classification, and the importance of both the attributes will be justified further.

5.1.3 Entity embedding

Watson NLU service is able to recognize extensive number of entity types (462)[3], which include as obscure categories as “WebBrowserEx- tension” or “HockeyConference”. For the purpose of this research, only 6 of them are used:

- Person,
- Organization,
- Location,
- Crime,
- HealthCondition.

Note that also “Company” type is recognized, subsequently classified as “Organization”. In addition, the concepts and dates are included as entities, since there is no difference in the way they might influence the relevance.

The case-specific embedding is based only on the entities that are important for the specific case. During the preprocessing, they are extracted from the case synopsis[4] and stored in a vector of fixed length for each entity type. For instance, a vector for people entities might look as follows:

("joey david", "thomas", "jesse hester", "obama", "", "", "", "", "", "")

The rest of the vector is free for entities from documents, which would be included based on their importance score (*Watson NLU* attaches the score to each entity). More specifically, when an entity appears in a new document with a high score, it is appended to the vector and never deleted. Finally, the embedding itself is a mask array of ones and zeros that indicate which entities are present in the sentence.

One problem with this embedding is that the vector acquires differ- ent meaning for every case. In a

situation when a model learns that a name on the index *i* indicates a relevant document, the model becomes not transferable to another case, where the vector contains different names. Hence, the entities are also encoded in general embeddings independent of case. Fortunately, the *word2vec* dictionary was trained on general texts including the entities as well. Most of the well-known names, such as “David”, “George”, or “Adele”, organizations, such as “Mc Donald”, “IBM”, or “Facebook” and locations, such as “Prague”, “Nigeria”, or “Prater” are present in the dictionary. Not only that it is now possible to train a general model, in which each word has its own fixed representation, but also, the embeddings satisfy the expected semantic rules. For example, vectors representing names will be close together with respect to other entities, and the model has a chance to learn that names are in fact important regardless of the specific word. The figure 5.2 confirms our thought about larger distances between the types of entities. Here, one can immediately see the larger distances between names, locations, and organizations than the distance among the entities of the same type. The general embedding is created by concatenation of a one-hot vector specifying the entity type and the *word2vec* vector for the entity.

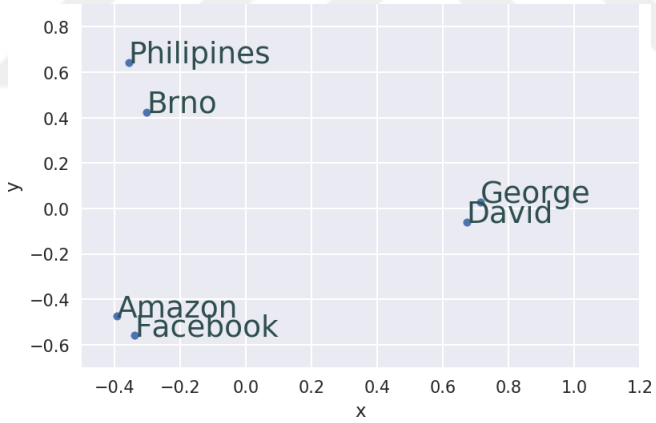


Figure 5.2: Distances among entity vectors in word2vec dictionary after PCA transformation to 2 principal components.

5.1.4 Relation embedding

The relations computed by Watson NLU keep the following structure:

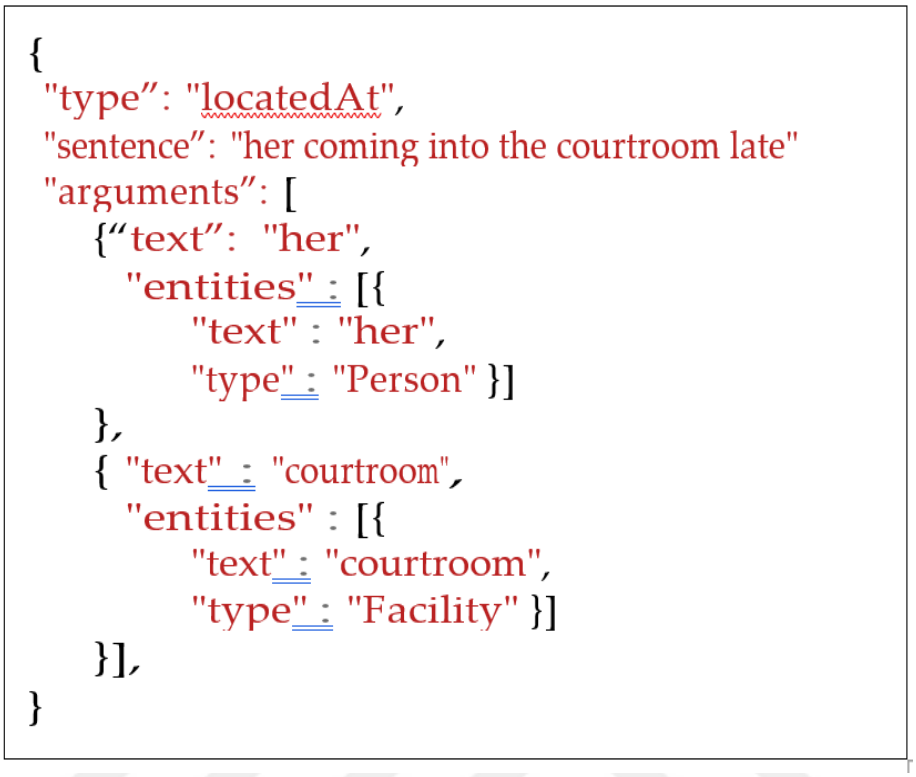


Figure 5.3: Watson NLU relation encoding example in JSON

Note that the fields "location" and "score" were excluded to make the example compact. As the JSON example suggests, the essential parts of the relation are its type ("locatedAt"), types of the entities ("Person", "Facility") and the text of the entities ("her", "courtroom"). The relation embedding, therefore, needs to accommodate relation type, 2x entity type and 2x entities itself. The previous sections presented the *word2vec* dictionary, which can convert common words into a vector. The same approach is used here for embedding of the entity words since there is a high probability of them being in the dictionary. On the other hand, types of relations and entities are rather identifiers than meaningful words, and thus they will hardly occur in the *word2vec* dictionary. In this case, we can take advantage of the lists of all possible types offered by Watson. There is a limited number of relation types (54) and entity types (49), which suggests a straightforward way of converting the type into a vector – the *One-hot encoding* as described earlier.

To summarize, the final embedding of the relation consists of

- relation type in one-hot vector (length 54),
- entity type of the subject in one-hot vector (length 49),

- entity type of the object in one-hot vector (length 49),
- the subject in word2vec vector (length 300),
- the object in word2vec vector (length 300).

5.1.5 Category embedding

The category is an example of an attribute related to the document instead of the sentence. Similarly as the relation and entity types, the Watson NLU defined a fixed set of categories. There are five levels of categories arranged in a hierarchic tree structure, but not all categories contain a full range of levels. An example of a category with four levels might be:

```
law, govt and politics / legal issues / civil rights / privacy
```

Each category level has a fixed number of types, which are stored in a form of ordered lists, where each index belongs to one category. The concatenated numbers determine the indices of categories and the final embeddings. It is worth noting that the fifth level is excluded from the embedding since the last category is mostly unused.

5.1.6 Structural characteristics

All the previous attributes refer to the text and its semantics. Documents are, however, defined also by their structure in the sense of text formatting. For example, some documents possess the structure of a fixed form, such as police records, medical records, or other administrative documents. In this case, the arrangement of text plays a crucial role in determining the type of a document, and therefore, also the relevance.

The following set of structural attributes are explored further in the research to discover their contribution:

- document length in characters,

- document length in words,
- document length in sentences,
- document length in new lines,
- average length of a sentence in characters,
- average length of a sentence in words,
- average length of a word in characters,
- number of empty lines,
- number of special characters.

5.2 BOTTOM-LEVEL MODEL

This section introduces the model that directly processes the embed- dings explained in the previous section and extracts information, which then provides to the top-level models. To achieve the best performance, it is crucial to split the predictions to elementary units and build on top of them (7.1.3). The models are able to learn a limited amount of information, especially in the case of an insufficient number of training instances, and therefore, one should keep the models as simple as possible. In contrast, one could concatenate all attributes in one vector, apply a single large model, and expect that it will be able to extract all discriminative information. This approach is not very likely to work in the environment with thousands of features, as in our case.
Following this idea, we present a series of models specialized on each of the attributes separately. Besides better final performance in predicting the relevance (7.1.3), another advantage is the transparent flow of information and a better understanding of the models. Exam- ining the intermediate results from the models, we can deduce which attributes are the most discriminative in predicting the relevance and which are not. Note that only the attributes containing *word2vec* em- beddings (meaning, keywords, relations) need a bottom-level model, as the others are built of small-sized vectors.

5.2.1 Machine learning estimator selection

The evaluation chapter contains a test on the most suitable machine learning estimators for the bottom-level predictions (7.1.1). Besides the fact, that the neural networks achieved the best

results, it is worth to explain, why they are naturally the most suitable model.

Regarding attributes describing the sentences, the final solution was to apply the *word2vec* dictionary to create an embedding. As mentioned in the section 5.1.1, the dictionary was computed from the texts of news reports that are processed by a neural network. Specifically, in the proposed word2vec algorithm, the vectors are derived from the weights between the hidden and output layer [59, sec. 2.2.1]. Since the embeddings represent an internal state of a neural network, the nat- ural step is to extend the network by virtually connecting additional layers. Even though the idea of extending network is not precise, as the embeddings are understood as inputs instead of weights, the aspect of the follow-up model structure showed to be significant (7.1.1).

Another reason, why neural network best suites our needs, is that it is an incremental estimator [64]. Incremental estimators enable to iteratively learn and improve on a new batch of data without forgetting

the old state. Another incremental estimator solving classification problems is the Naive Bayes (NB) classifier.

The final structure of the neural network has a variable input layer dimension based on the attribute, two hidden layers of 16 and 8 neurons and 2 output neurons representing binary classes. Such a model is prepared to predict law-related vs. case-related sentences as well as relevant vs. irrelevant sentences.

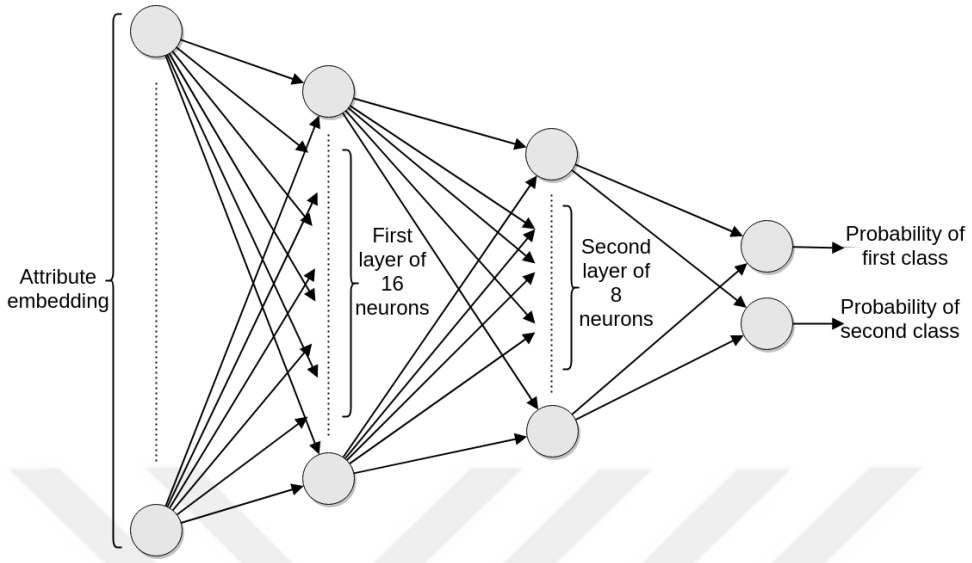


Figure 5.4: Bottom-level model structure

The *sigmoid* activation function showed to be the most successful in all layers. During the model training, an Adamax optimizer with a learning rate of 0.02 was applied in 150 epochs and batch size of 20 to 100 depending on the number of training data. To prevent the overfitting, the model stopped learning in case of increasing loss on a validation set (10% of training data). The applied loss function is called the Cross Entropy (CE), which is defined as follows:

$$H(P, Q) = -\mathrm{E}_{x \sim P} \log Q(x) = -\sum_{x} P(x) \log Q(x)$$

where P is the true probability distribution and Q is the predicted probability distribution over samples x [65]. Note that the roles of true and predicted distributions are given, in contrast to the accuracy measure.

5.1 LAW-CASE CLASSIFIER

The goal of the Law-Case classifier is to assign a document to one of the classes law-related or case-related. As the Data Characteristics chapter defines the classes only loosely, the following paragraphs provide a more formal definition.

Case-related documents contain at least one specific information about the subjects or events

related to the case. The authors of these documents are usually directly involved in the case or at least has come into contact with the subjects. They are mostly confidential, such as emails, medical records, emergency call records, police records or testimonies. Their main characteristic is an occurrence of a known entity, such as a victim, defendant, or location of an event that is part of the case.

Law-related documents do not contain any information specific to the case. They usually state facts that help lawyers argue about the legal aspects of the case and find disagreement and evidence. Examples of such documents are fragments of law code, description of similar cases from the past, policies of a company/school/office, news, books, or articles. Their main characteristic is a specific vocabulary, usually including the legal terminology.

All 263 documents were labeled with one of the classes as part of the annotation process. The case-related group is in the majority with 199 document versus 64 law-related documents (32% of all). This ratio implies that a model predicting all documents as case-relevant achieves an accuracy of 0.68. Further in the text, this simple model will be referred to as a baseline model. Note that the evaluation metric inspected here is an accuracy score, in comparison with the relevance prediction where precision and recall are of interest. The reason for that is that the two classes are on the same semantic level – one is not more important than the other. To train the models on sentences, we had to establish a simplified assumption and consider all sentences as case-relevant if the document is case-relevant (similarly for law- related). As a result, the models were trained on all 14,368 sentences. The top-level model of Law-Case classifier is working with the following sets of attributes. The first set is extracted and used as an input for the bottom-level models: sentence meaning, keywords, relations, and entities. Their encoding is explained in the section 5.1. The second set contains the structural characteristics of the document as explained in the section 5.1.6. Next set of attributes is created by counting the entities that appear in the sentence. The assumption is that the law-related documents do not contain as many entities as the case-related, since the explanation of laws uses neutral words, such as "defendant", "subject", "suspect", and similar. Next two attributes are the first and second level of a document category (other levels do not contain enough representatives). The last two attributes are derived from the concepts of documents. First, the concepts are organized into two groups – case concepts and law concepts – such that the case concepts appear in some case-related document but not in any law-related (similarly for law concepts). Then, the attributes express the number of

case and law concepts in the given document.

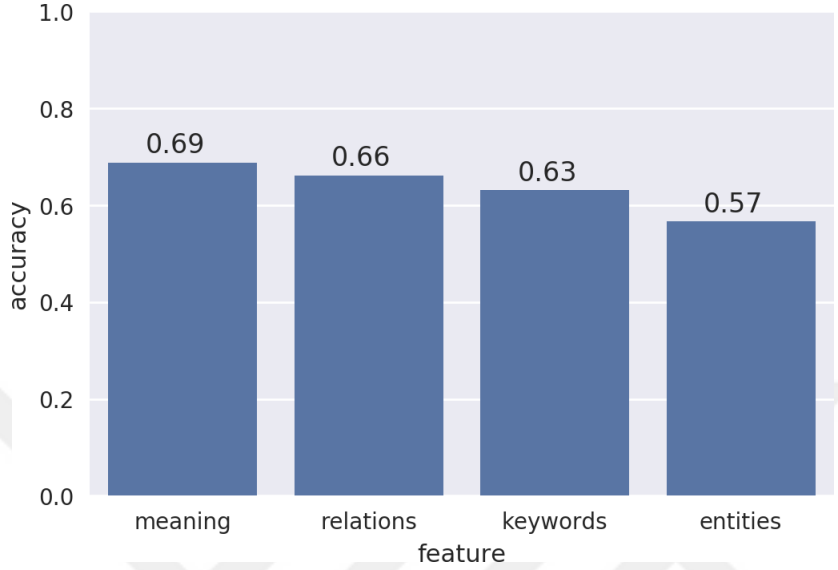


Figure 5.5: Accuracies of bottom-layer models for prediction of case- related and law-related documents. The models itself are trained and evaluated on sentences and the document class is determined from the averaged probabilities of individual sentences.

The figure 5 .4 reveals t he accuracies of bottom-level classifiers in the task of predicting the class of documents. The models were evaluated on each document after training on sentences of all other documents (Leave-One-Out method). One can notice rather poor performance, with *meaning* the most discriminative attribute. On the other side, the *entities* do not contribute very well to the predictions. Even though the results suggest a low performance of the top-level model too, the combination of bottom-level models turns up to be surprisingly powerful.

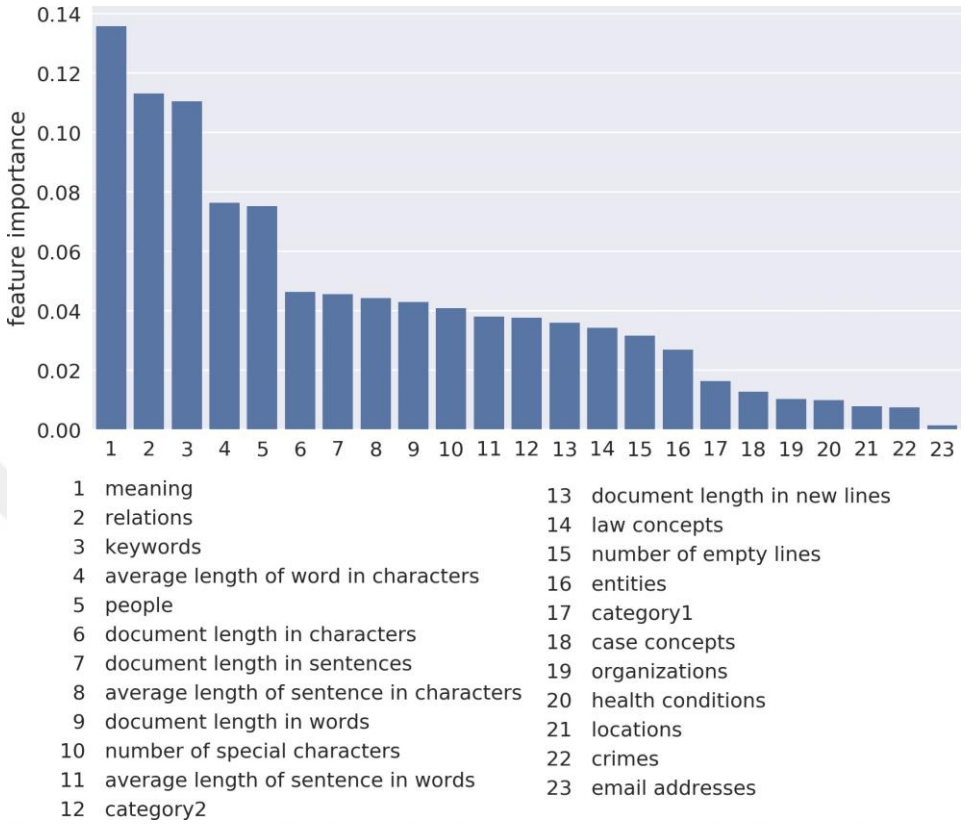


Figure 5.6: Feature importance from Random Forest classification

of case-related and law-related sentences. *Meaning* encodes the averaged word2vec vectors of all words in the sentence. The *people*, *organizations*, *locations*, *crimes*, *health conditions* and *email addresses* represents the number of appearances in the sentence. *Category1* and *category2* represent the two most general levels of the category.

To determine which attributes are beneficial and which are irrel- evant, one can plot the importance of individual features suggested by the *Random Forest* classifier. The classifier computes the *gini index*[5] that not only drives the decision tree creation but also signifies the feature importance. The evaluation in the figure 5.5 takes advantage of this ability and displays the features sorted by its importance.

The meaning, relations, and keywords occupy the first three most important features. The entities model with its importance less than

0.03 is useless in this settings and will not be considered in the final model. Some of the

structural characteristics are successful, such as an average length of words or a document length. The significance of the word length might be caused by more detailed and professional terminology of law-related documents, which often includes longer words. The document length feature might be influenced by the long case-related testimonies, which is a specific artifact of our data set. The model evaluated only the number of people as a relevant feature out of all the entity counts. Interestingly, the first level of the category is not relevant as much as the second level. This might be due to the general nature of the first level, which is probably sharing a single value for a large number of documents, and therefore, it loses the discriminative power. Regarding the concepts, they proved not to be very helpful in this classification task.

As the most discriminative attribute is the meaning, it is worth to look at the contribution of individual dimensions (figure 5.6). The importance is following the Zipf's law, as it seems to be decreasing exponentially with the ordered dimensions. Nevertheless, the dimensionality reduction did not help in the classification and the accuracy slightly decreased.

The top-level model is based on the first 12 features in the figure

5.5. In this case, the model is trained on the set of documents, which has only 263 instances; hence the neural networks are not applicable. As the best classifier showed to be the *Random Forest* (sec. 7.1.2) with 20 base estimators and maximum tree depth of 7 (sec. 7.2.2). The final model achieved a performance of 0.89 on the set of all documents evaluated by the Leave-One-Out method. Now, when the division to the two diverse types of documents is in place, the labeled documents can proceed to the relevance prediction.

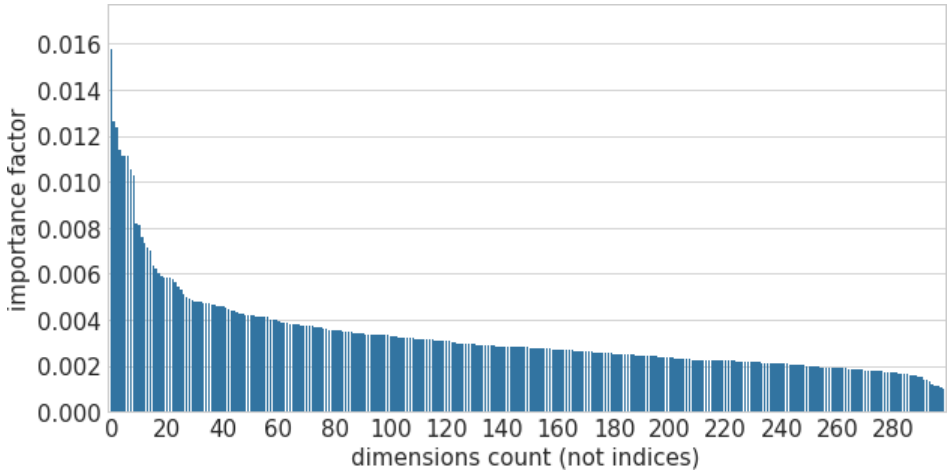


Figure 5.7: Importance of dimensions in the sentence meaning vectors

5.3 SENTENCE RELEVANCE CLASSIFIER

The sentence relevance prediction is a relatively harder task than the previous classification. People consider sentences relevant based on various factors. For the sake of simplicity, the system assumes that the relevance depends on the words which appear in the sentence, especially keywords. In reality, the context of the document, case, and other external knowledge plays a significant role, which is ignored here. Another factor that makes the prediction difficult relates to the process of human annotation: in case there are multiple sentences with relevant information, a lawyer would highlight only one of them. As a result, the model trains on a dataset, where similar sentence appears as relevant and irrelevant at the same time.

Recognizing a sentence relevance can be applied in other fields as well. As an example, a thesis "Automatická sumarizace textu" by Machovec conducts a study on the automatic text summarization based on the relevant sentences [66].

Instead of the accuracy, the observed evaluation score is now pre- cision and recall. The precision expresses a percentage of correctly classified sentences out of all sentences that were classified as relevant. The recall expresses a percentage of correctly classified sentences out of all relevant sentences. As mentioned in the section 1.3, the recall is usually more valued in the E-discovery, since the extra time of the lawyer spent on an irrelevant piece of information outbalance the situation, in which the lawyer is not notified about some relevant in- formation. The mentioned research reveals the human performance in evaluating the relevance: precision ca 20% and recall ca 50%. These values set a baseline in the following experiments.

Two groups of attributes are applied in the sentence predictions: general and case-specific. The general attributes are trained on the full set of documents, and the model is kept the same regardless of the case. The case-specific attributes are trained on documents of a single case, and they capture information specific to the case.

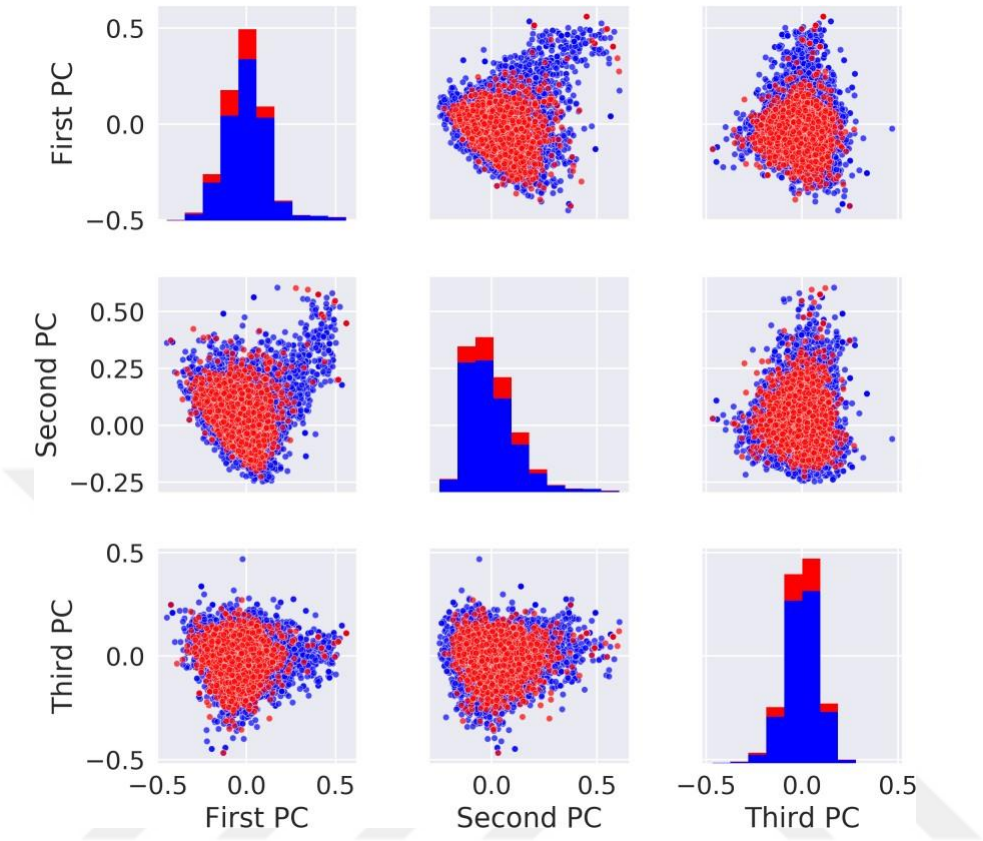
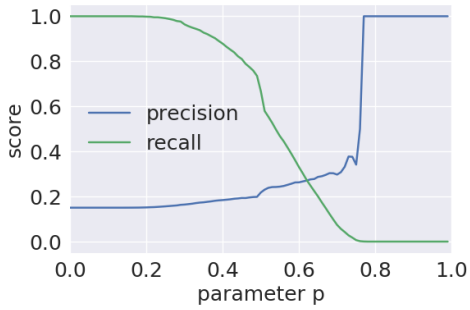


Figure 5.8: The first three components of PCA transformation of the sentence meaning plotted with labels according to the sentence relevance. Red points represent the relevant and blue irrelevant sentences.

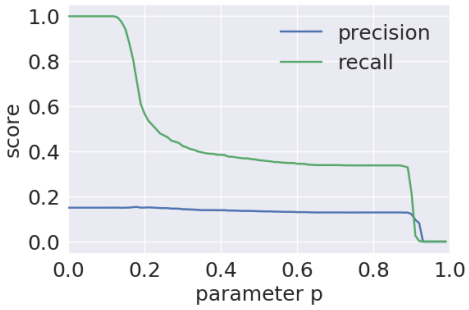
5.3.1 General Attributes

The general attributes include the *word2vec* vectors: meaning, key- words, and relations. The entities vector proved to be inefficient in the Law-Case model, and therefore, the present model does not include it. The bottom-level models (sec. 5.2) are applied to train the sentence relevance. Thanks to the predicted probability, there is a straightforward tool to trade between the precision and recall. Let us define a parameter p as the minimum probability for a sentence to be considered as relevant. With increasing p, the recall increases and precision decreases.

(a) Meaning attribute



(b) Keywords attribute



(c) Relations attribute

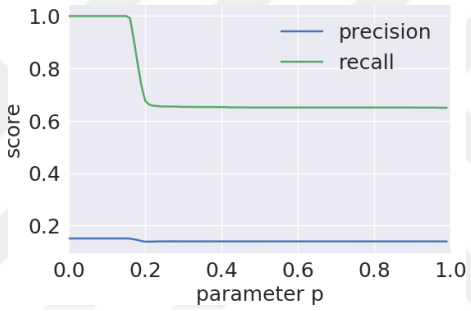


Figure 5.9: Precision and recall of the general attributes for different values of parameter *p* when predicting the relevance of sentences.

The figure 5.8 shows the precisions and recalls for all three general attributes with respect to the parameter *p*. Obviously, the keywords and relations failed to learn the features well, because the precision never increases with higher *p*. In contrast, the figure 5.8a indicates a more promising situation. For $p = 0.5$, the model achieved more than 20% of precision and crossed the 50% of recall, which reached the aforementioned human performance.

Surprisingly, the meaning is again the superior attribute (similarly as in the Law-case classifier). It seems that the neural network does not need to pick keywords or any other feature beforehand from the text, as it is able to extract them itself. The figure 5.7 shows sentences plotted in the space of first three principal components according to the meaning attribute. As only 17.7% of the sentences are relevant, the red points are sparsely distributed in the area. Unfortunately, there is not an evident separation of the sentence according to the relevance, which is an expected outcome

due to the obstacles described earlier.

5.3.2 Case-specific attributes

The already mentioned lack of context is partly solved by the case- specific attributes, which provide a background of the case. The at- tributes are computed as described in the section 5.1.3. Only the case synopsis is applied in the creation of the vectors, in contrast with the application, where the vectors are complemented with keywords from documents. The prediction is evaluated on models that were trained exclusively on the documents from the given case.

Based on the earlier model selection (7.1.2), the Random Forest estimator with 20 estimators and 7 tree depth was chosen to evaluate the relevance. Its performance was not strikingly high with precision 17% and recall 71% for p = 0.45, and precision 20% and recall 35% for p = 0.5. Nevertheless, the performance is expected to be higher in the application, as the keyword and entity vectors will contain extra entries extracted from the documents.

Multiple combinations of general and case-specific model come forward. The figure 5.9 shows two of them: *AND* variant, where a sentence is considered relevant only in case both models agree, and *OR* variant, where one of the models voting for a positive relevance is enough.

(a) *AND* variant of the model (b) *OR* variant of the model

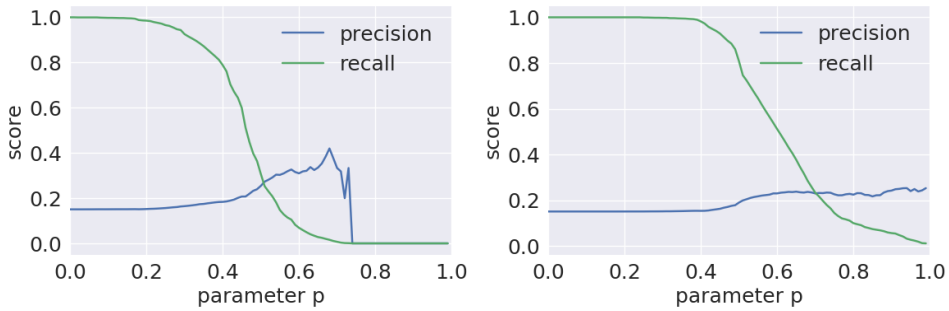


Figure 5.10: Precision and recall of the general and case-specific at- tributes

The *or* models seem to be the more efficient variant. The intuitive idea is that a sentence might be generally relevant without the case- specific keywords (especially in the law-related documents), but also the appearance of some critical keyword might be an alert without the sentence being

actually relevant according to the general characteristics.

The final model is using the *OR* combination of general and case-specific models (neural network for general and Random Forest for case-specific) with $p = 0.55$. This setting yields final precision of 0.22 and recall of 0.65. In comparison with the human performance, the recall is higher, and the precision is comparable.

5.4 DOCUMENT RELEVANCE CLASSIFIER

The previous two classifiers made decisions to guide the last one – the document relevance classifier – to the correct result. The output from the sentence classifier is a direct input for the document classifier, included as the number of relevant sentences. The Law-Case classifier provided another attribute, which is a binary flag of the document's type. Other attributes include the structure as explained in the section 5.1.6, input from the bottom-level attributes, category, and counts of the individual entities. The bottom-level attributes include meaning, keywords, and relations, as in the case of previous classifiers. Here, however, the meaning is computed for each word of the document, and the predictions are averaged. Since the case-specific features were already included in the sentence prediction, this model does not contain any attributes specific to the case.

The figure 5.10 shows the attributes importance and guides us to select the best set of attributes. Indeed, the number of relevant sen- tences is the most important attribute, which is a positive fact and a confirmation that the sentence classifier is beneficial. The structural characteristics cover all the next positions up to the eighth. We antici- pated their importance to be high in advance, though they matter even more than expected. On the other hand, the counts of entities and bottom-level attributes have not proved to be discriminative enough to predict the relevance.

It is worth to pinpoint one remarkable fact. The meaning attribute achieved importance of almost 14% in the sentence classifier, whereas less than 4% in the document classifier. One can notice that the number of words included in the meaning matters – the sentence spans 15.9 words in average, which brings enough specificity to train the models, compared to the 868 words per document.

Due to the similar shape of data to the Law-Case classifier, the document classifier adopts the same estimator – Random Forest (7.1.2)

– with the same parameters (7.2.2). Brief testing of other parameters did not significantly increase the accuracy. The final model is trained by the Leave-One-Out method, which means 262 training

documents and one testing. Based on the feature importance analysis displayed in the figure 5.10, only the first eight attributes were considered in the final model.

The first variant is a classification into one of the three classes (HOT, WARM, COLD). To be able to confirm success, one can compare the result to the baseline model, which always predicts the majority class (HOT) that would achieve an accuracy of 40%. After multiple executions, our model achieved averaged accuracy of 0.738 with standard deviation of 0.02 and confidential interval $(0.726, 0.750)$.

In the second variant, the model takes advantage of the fact that there exists an order among the classes: WARM is semantically be- tween COLD and HOT. Therefore, the model is trained on documents of binary values, which is achieved by converting all WARM documents to HOT. Then, the probability of a document being HOT is

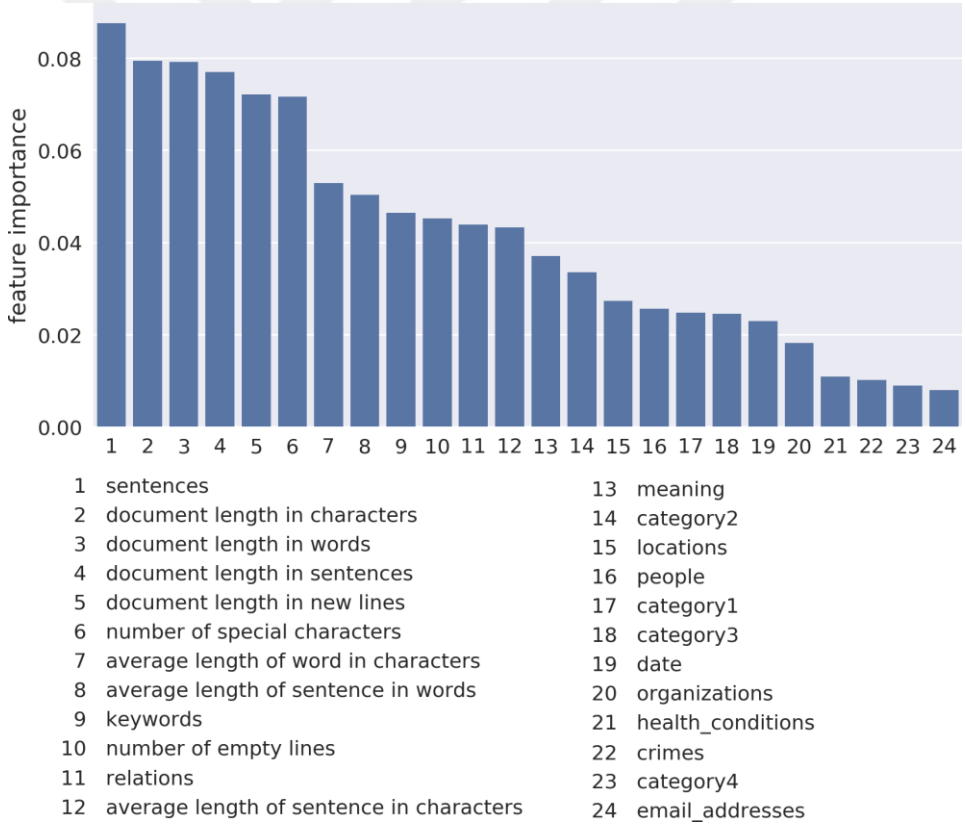


Figure 5.11: Feature importance from Random tree classification of HOT,

WARM and COLD relevance computed, and finally, the classes are decided based on the thresholds h and w. If the probability is higher than h, the class predicted is HOT. If the probability is lower than w than the class is COLD. If neither of the cases is true, the WARM class is the output. The

heat map in the figure 5.11 shows the accuracies achieved when applying various values for *w* and *h*. Even though the accuracies are close to 70% for some combination of *h* and *w*, it never exceeds the performance of the first variant (74%). To summarize, we conclude that the classification of three classes gives better results than the classification of the border two with outputting the middle class in case the probability is near 50%.

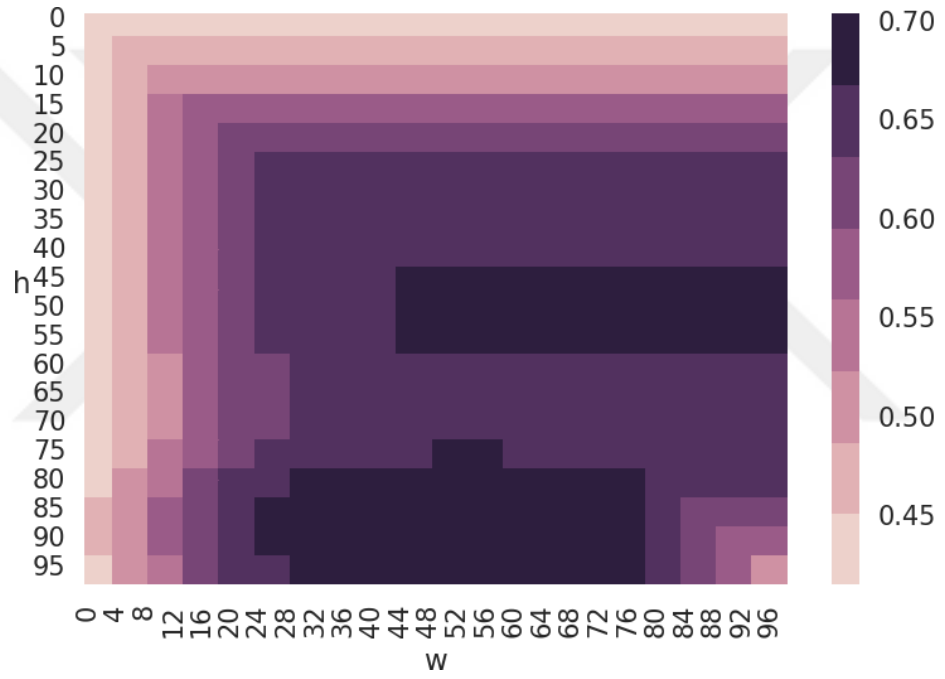


Figure 5.12: Heat map of accuracies for all values of parameters *w* and *h*.

Note that the values are shown in %.

Even though we can present the precision and recall of the ternary classification problem, the result would be misleading, as the error in swapping HOT and WARM would be the same as swapping HOT and COLD. In the evaluation chapter, the WARM and HOT labels will be merged to create a binary problem, which again enables the comparison of precision and recall. Lastly, it is worth to mention that the final model, as applied in the application, includes one more attribute – the commentary. Users can comment on documents in Legato, which creates potentially valuable

information in the relevance prediction. Unfortunately, there are no ground data to either train or evaluate the performance of this attribute.

6. EVALUATION

Besides the final results, this chapter introduces evaluation of parameters for the estimators and also a couple of small experiments focused on particular options during the design of the classification process, which drove the decisions. The classification chapter often states facts without factual support and refers to the experiments here instead.

6.1 DECISION POINT EVALUATION

6.1.1 Bottom-level Estimator Selection

The bottom-level estimator selection was based on the classification of case-related and law-related sentences from the sentence meaning. As mentioned earlier, it is expected that the neural network outperforms the other estimators due to the natural essence of the word2vec attribute. Nevertheless, it needs to be shown by an experiment, which results can be viewed in the figure 7.1.

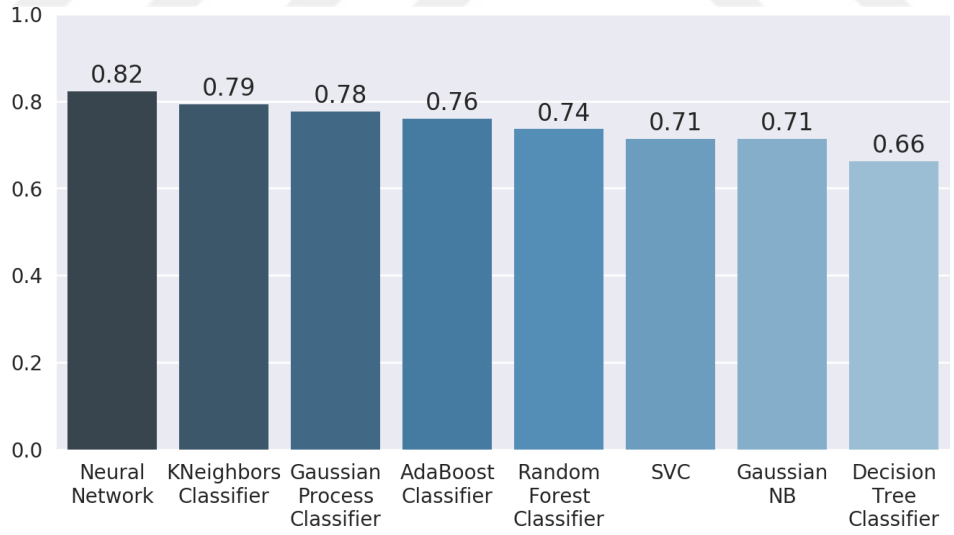


Figure 6.1: Bottom-level estimators' accuracies for case-related or law- related classification of the sentence meaning

In contrast with the top-level estimator selection (sec. 7.1.2), one can notice that the ensemble

estimators and decision tree did not perform very well. That is a consequence of the multidimensional space of continuous variables, which is hard to understand by such simple rules as inequalities. The KNN algorithm, on the other hand, scored high, which might be due to the aforementioned property of the word2vec vectors – the distances among them correspond to their semantic distance.

6.1.2 Law-Case estimator selection

After all values of attributes from bottom-level models were pre- computed, the next step was to pick the best top-level estimator. The situation is the following:

a low number of samples

a low number of features

binary class

quantitative and ordinal features

This combination of features suggests that decision trees or their ensembles could be the best suited. The neural networks need a large number of samples, and the SVM works best with a large number of features. The figure 7.2 shows the comparison.

One can notice that the two included ensembles (AdaBoost and Random Forest) are the most successful. The Neural Network (MLP) and Support Vector Machines (SVC) performed worse than the expectation. All estimators ran with default parameters as they are adjusted by Scikit learn library.

Undoubtedly, the AdaBoost Classifier outperformed a ll others, especially the Random Forest. However, during the parameter tuning of both of these two classifiers, the Random Forest achieved better performance when tuned than the AdaBoost. Due to this finding, the final model is based on the Random Forest estimator.

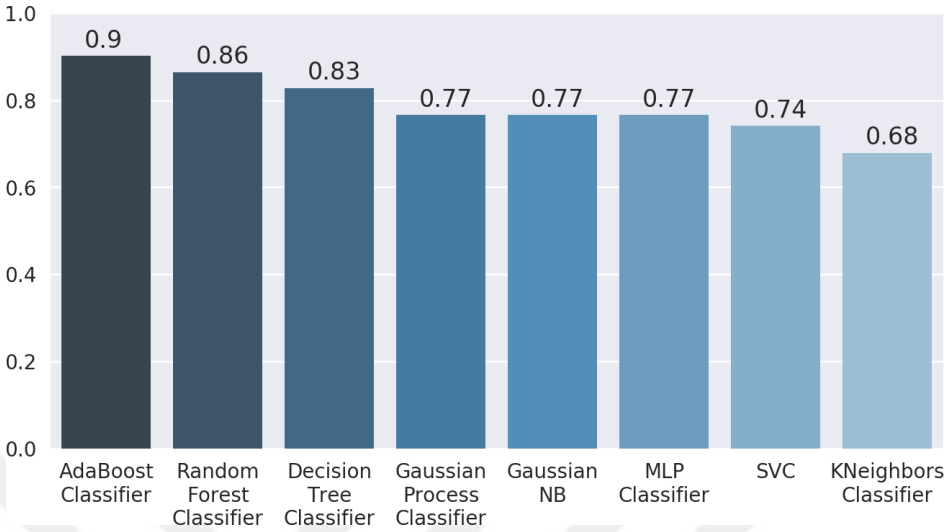


Figure 6.2: Estimator accuracies for the case-related or law-related classification of the sentence meaning.

6.1.3 Layered Architecture

This experiment shows a comparison of two model architectures clas- sifying case-related and law-related sentences. One architecture con- sists of only one model, which analyses all attributes to predict class, whereas the second architecture is designed in two layers, where the top-level model predicts class based on the output of the underlying models.

The bottom-level models, as well as the model in one-layer architecture, are implemented by a neural network estimator with parameters as tunned in the section 7.2.1. The top-level model is implemented by Random Forest classifier with 200 base estimators and maximal tree depth 70. From the dataset of 263 documents, 40 case-related and 40 law-related were randomly selected to be the training set, and the rest served as the testing set. With sentences as data points, one has to be careful not to include sentences of a single document in both training and testing data. Before the training process, the data sets were balanced in order to include the same number of sentences from both classes. Therefore, a baseline model would achieve 0.50 accuracy.

The experiment was concluded in the following way. For the *All- in-one* architecture, the attributes

were being consecutively appended to the vector that served as an input to the model. The *Separate* architecture, on the other hand, always trained bottom-level models on the individual attributes added so far, which output is used in the top-level model for predictions. Thanks to this design, one can easily compare the two architectures with increasing number of attributes.

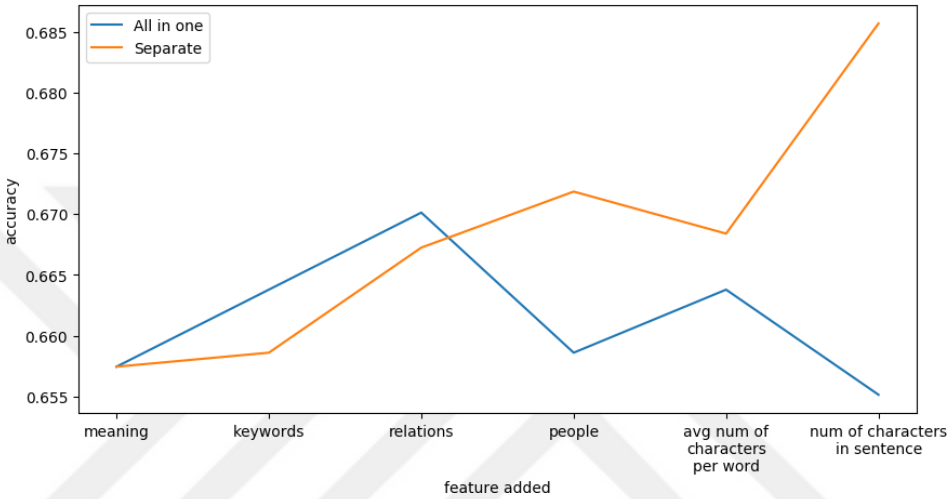


Figure 6.3: Performance of all-in-one model (one layer) vs separate features (two layers). Attributes are consecutively appended to the input, so that one can see the scaling of both methods.

The figure 6.3 shows accuracies of the model architectures. Notice that *All-in-one* architecture is slightly more accurate up to the first three attributes; however, as the input vector prolongs, the ability to recognize patterns decreases. The *Separate* architecture has a dis- advantage that the bottom-level models cannot cooperate together and extract the cross-information. The advantage is that a few data is enough to train bottom-level models, and therefore the method scales better for increasing number of attributes.

6.1.4 Relations with Types

As described in the section 5.1.4, the relation consists of embedded words and types. This experiment helped us confirm the theory that the types are bringing a significant information value in the classification. The accuracy of a single model predicting case-related and law-related

classes of sentences achieved the accuracy of 0.82 with both types and words (dimension 758) and only 0.79 with only words (dimension 600). The cross-validation with stratified folds was applied to evaluate the result. Based on this small test, all further experiments and the final model includes relations with types.

6.1.5 Word2vec Contribution

To show the contribution of *Word2vec* embedding, this experiment compares it with the conventional one-hot embedding. The accuracy score is computed for the same set of sentences used to classify the case-related and law-related labels by the neural network explained in the section 5.2.1.

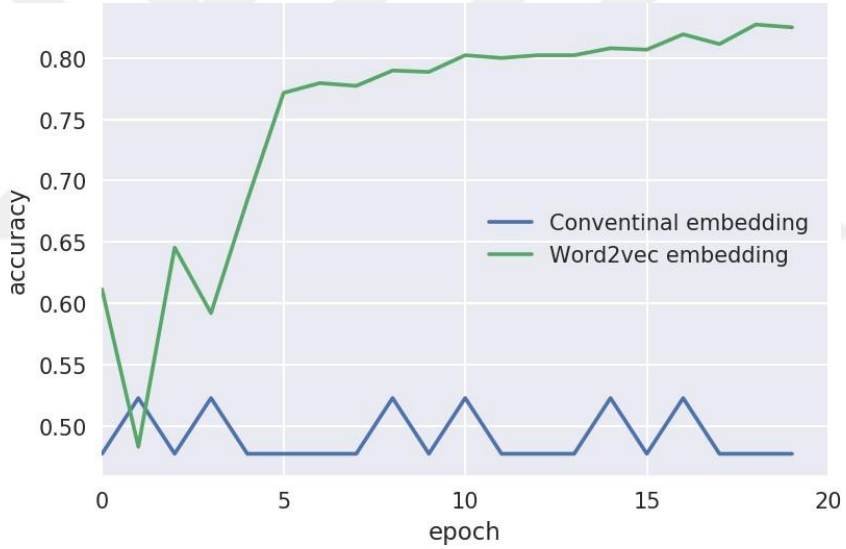


Figure 6.4: Performance of a conventional embedding in one-hot vectors vs. the word2vec embedding evaluated on a validation set during the epochs of neural network learning.

6.1.6 Contribution of Separate Case and Law-Related Models

This experiment evaluates the contribution of the idea to separate the documents to case-related and law-related. The results were obtained by measuring the accuracy of the document classifier on attributes as explained in the section 5.5. The two variants differ in the inclusion or exclusion of a binary attribute specifying the case-related or law- related affiliation. Ten runs of the same settings (Random Forest with 20 estimators and 7 tree depth) were executed, which provided us with confidence intervals to conclude the contribution.

The mean of the accuracy of the variant without the label is 0.736 and with the label 0.743. The average accuracy of the variant with the label is only by 0.7% higher, which is a negligible increase since this amount corresponds to one or two documents out of all 263. To conclude that the label flag is not a real contribution, one can check if the confidential intervals intersect: the 95% interval for the variant with the label (0.735, 0.751) indeed intersects the 95% interval of the other option (0.729, 0.742). Based on this experiment, the Law-Case classifier is not included in the application.

6.2 PARAMETER TUNING

6.2.1 Bottom-level Model

Following the analysis from the estimator selection experiment (sec. 7.1.1), this section presents the parameter tuning of the estimator. Based on the previous experiments with the neural network, the chosen parameters and values to tune are the following:

- # of neurons in the first layer: 128, 64, 32, 16
- # of neurons in the second layer: 16, 8, 4, 2
- learning rate: 0.002, 0.005, 0.01, 0.02
- batch size: 20, 60, 100, 300, 500, 1000
- # of epochs: 20, 100, 150

The activation function was not tunned since only the *sigmoid* per- formed well, and therefore, was the only option.

For the purpose of the tuning, the sentence meaning was applied as the training data with labels according to the case-related and law-related classes. The *Scikit-learn* library provided a tool called *Grid- SearchCV*, which tries all possible combinations of the parameters and computes its accuracy in a parallel threads.

The top ten results are shown in the table 7.5. Notice that the accuracy does not necessarily increase with the increasing number of neurons. The first variant with the learning rate of 0.02, 150 epochs, 16 neurons in the first layer and 8 in the second layer is applied in the research, as

well as in the application. It is worth noting that the batch size will be changed for different settings according to the size of the training dataset. Thanks to the early stopping criteria, which terminates the learning when the accuracy does not increase in 4 consecutive epochs, the number of epochs can be understood as an upper bound rather than a precise number.

Table 6.1: The neural network parameter tuning on sentence meaning predicting the Law-Case classes.

#	Learning rate	Batch size	Epochs	First layer	Second layer	Accuracy
1	0.02	20	150	16	8	0.835
2	0.02	20	100	16	8	0.832
3	0.02	20	100	32	4	0.832
4	0.02	20	100	64	16	0.831
5	0.02	20	150	32	8	0.831
6	0.02	60	150	32	8	0.83
7	0.02	20	100	32	8	0.83
8	0.02	60	150	64	2	0.83
9	0.02	100	150	32	4	0.829
10	0.02	20	100	32	16	0.829

6.2.2 Top-level Model

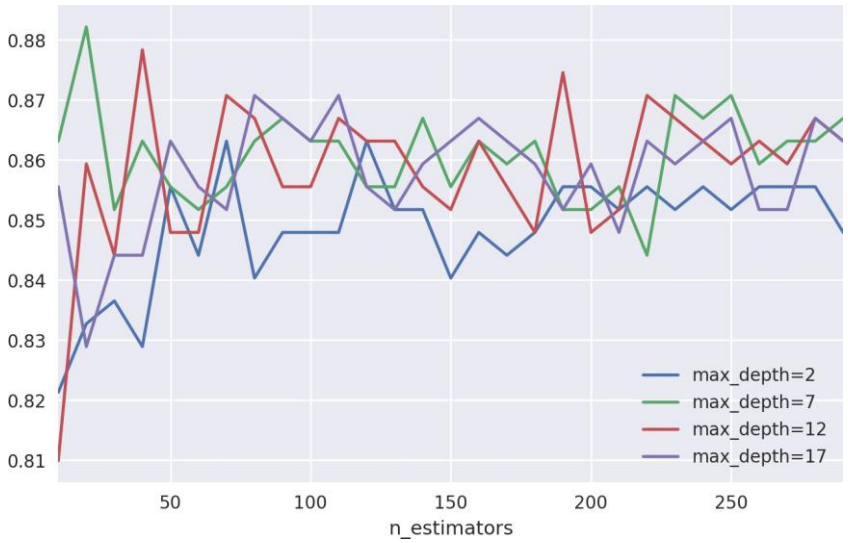


Figure 6.5: Parameter tuning of the Random Forest classifier for the case-related or law-related classification of documents.

The outcome of the model selection in the section 7.1.2 suggested Random Forests as the top-level estimator. This section presents tuning of its parameters, specifically, the number of estimators and the maximum depth of decision trees.

The figure 6.6 shows accuracy scores for different depths of trees. The maximum level of 7 together with 20 estimators (individual decision trees) performed the best with 0.882, and hence these parameters were plugged in the final model. Note that the cross-validation with three folds was applied, which means the model was trained on only 2/3 of data. In case the model is trained on all but one document (Leave-One-Out), the accuracy is 0.890.

6.3 RESULTS

This section summarizes the final results of the classification, evaluates the individual models and show the performance on selected sentences and documents.

6.3.1 Evaluation of Law-Case classifier

Unfortunately, the Law-Case classifier did not show to be useful (sec. 6.1.6) in the relevance prediction. As the application is focused primarily on the relevance, the contribution is too low to be worth the implementation, and therefore, the final application is left without the Law-Case model.

However, the experiments were not pointless, since they discover a surprisingly poor relation between the intuition about the types of documents and their relevance. Also, the information about the class might be useful for lawyers in other tasks than a relevance prediction. Possibly, the model would be more efficient with several times larger corpus of documents, but this research lacks resources to conclude such claim.

6.3.2 Evaluation of Sentence Classifier

As mentioned in the section 5.4, the classification of sentences is a challenging task, primarily due to a subjective view on their relevance.

The proposed Sentence classifier in the final setting achieved a precision of 0.22 and recall of 0.65. This means that there is a 65% chance that a relevant sentence will be recognized by the

tool. The positive fact is that if multiple sentences in a document are relevant, then there is a high probability that they will be recognized, and thus the whole document is classified as HOT. The negative fact is that only one out of four or five sentences is genuinely relevant. This imprecision might be tolerated by users as the primary focus is on the recall.

To obtain a better insight into the algorithm, we present examples showing specific sentences with their predicted and true relevances. All sentences are from the case "Davis v. HappyLand Toy Company" already presented in the section 4.3.1.

The first document is a testimony of a babysitter, who was looking after Joey when the incident happened. An example of a correctly classified sentence follows (note that the sentences are already preprocessed and sometimes cut to individual clauses):

"much worse though was that joey seemed to have some breathing problems"

The sentence contains two words of interest: the name of the victim and a medical term that relates to his health (breathing problems). As the medical term probably appeared in other documents as well, its appearance could be the reason, why the sentence is predicted as relevant.

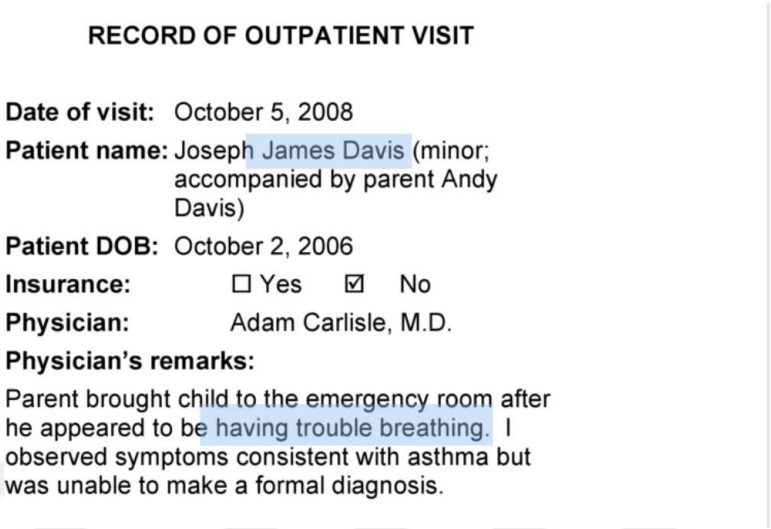
Next sentence was recognized by the classification as relevant, though it is not:

"i do a lot of babysitting for the palmer family and they recommended me to the davis family"

Here, possibly the names ("palmer" and "davis") caused the higher relevance, although it is not necessarily relevant for the case.

No relevant sentence was classified as irrelevant in this document, which means the recall is 1. However, the precision is only 4% with 138 sentences classified as relevant out of 186. In this case, the high relevance is beneficial, since the document is subsequently correctly classified as HOT.

The second document is a record of a doctor visit, where the an- notator picked the name of the patient as relevant and the breathing troubles, as can be seen from the view 6.7.



RECORD OF OUTPATIENT VISIT

Date of visit: October 5, 2008

Patient name: Joseph James Davis (minor; accompanied by parent Andy Davis)

Patient DOB: October 2, 2006

Insurance: ☐ Yes ☑ No

Physician: Adam Carlisle, M.D.

Physician's remarks:

Parent brought child to the emergency room after he appeared to be having trouble breathing. I observed symptoms consistent with asthma but was unable to make a formal diagnosis.

Figure 6.6: Example of a sentence relevance in WARM document

Our algorithm missed the patience name, but highlighted both sentences of the bottom paragraph. Therefore, both the precision and recall are 0.5. The algorithm has troubles with the structured documents, because the OCR analysis sometimes adds strange signs to the sentences. For example, the above check boxes are present in the sentence as "iii yes izl no".

6.3.3 Evaluation of Document classifier

The document classifier achieved an accuracy of 0.74 with a baseline of 0.40, which is a significant contribution (5.5); however, to compare the results with the previous studies, we need to compute the precision and recall. Hence, the following evaluation considers all WARM documents as HOT for the sake of the high-recall strategy. Again, the results are varying according to a threshold *p*, which is a limit that a sentence needs to exceed to be considered as relevant. Results are displayed in the graph 6.8.

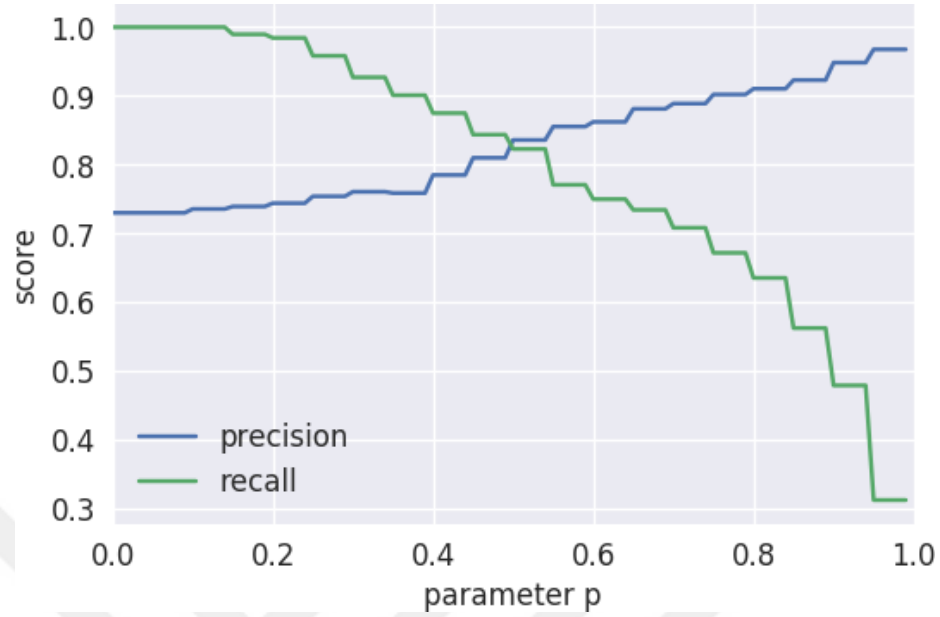


Figure 6.7: Precision and recall of the binary relevance prediction of documents

with respect to a parameter *p* determining a threshold probability for a document to be considered as relevant.

The results are considerably higher than in the case of sentences according to the expectation. On the other hand, the ratio of relevant documents is more than 73%, which makes the classification easier task than in the sentence environment, where only 17% sentences are relevant. For the same reason, the results are hard to compare with the previous research on the document relevance (1.3), where the precision was only 20% and recall 50%, because the percentage of relevant documents was as low as 9%.

Next paragraph reviews the success rate of the ternary-class version of the model in the mentioned case “Davis v. HappyLand Toy Company". Even though only 55% of all HOT cases were recognized as HOT, 82% from the rest was classified as WARM. From 20 COLD documents, 11 were correctly recognized as COLD, but more importantly, only 2 of the misclassified documents were marked as HOT. The conclusion is that the prediction is not 100% precise; however, it was able to learn basic patterns and behaves reasonably.

7 CONCLUSION

This thesis searched answers on the questions about the application of artificial intelligence in the E-discovery process, especially, how are the legal documents different from others, what makes a legal document relevant, what features of a legal document matters the most, if the structure is important in the relevance prediction, and what tools and estimators work best for relevance prediction in the legal domain. Some of the questions about the nature of the legal data are elaborated in the chapter Data Characteristics (3), the essential attributes for the relevance prediction are studied in the section 5.1, and the most suited estimators are examined in the Evaluation chapter (7).

During the data analysis, two ideas on improvement of the relevance prediction were proposed: determining a document type prior to the relevance, and shifting the focus on sentences rather than documents. The first idea failed to prove to be useful, which influenced the further research and implementation. On the other hand, focus- ing on sentences showed to be a beneficial step towards an efficient relevance prediction. As a result, the performance of the presented models reached similar efficiency as other tools in the same domain as well as human annotators.

Besides the research, a simultaneous outcome of the thesis was also the implementation of a system that applies the mentioned techniques and is integrated with an E-discovery system Legato. The system presented here is already included in the instance of the Legato that contains innovative features. If the user testing confirms usability and usefulness of the extension, it will reach the final release.

Many aspects of the system, such as training of the models, parameter tuning, or overall system configuration, can be further improved in the future. Due to the fact that the models were trained on a relatively small dataset, they will tend to overfit and most likely fail on new types of documents. There are still multiple possibilities how to improve the system. The future work might consider more extensive range of entities since the *Watsion NLU* provides 462 types, from which only six were applied. Another improvement could train a recurrent neural network to learn patterns in the sequence of words. Also, the *Watson NLU* provides a possibility to create a custom language model, which could recognize entities, keywords and relations even more precise when focused exclusively on the legal text. The present research could not utilize this feature due to a small amount of training data.

The relevance prediction, as well as other advanced features ap- plying artificial intelligence, will become an essential part of each innovative E-discovery system. There are still opportunities for further enhancements, which might rapidly speed up and ease the work for lawyers. Hopefully, the presented research helps lawyers and soft- ware developers in future decisions to achieve these enhancements.

REFERENCES

[1] CONRAD, Jack G. E-Discovery revisited: the need for artificial intel- ligence beyond information retrieval. *Artificial Intelligence and Law*. 2010, vol. 18, no. 4, pp. 321–345. ISSN 1572-8382. Available from DOI: 10.1007/s10506-010-9096-6.

[2] ROITBLAT, Herbert L; KERSHAW, Anne; OOT, Patrick. Document categorization in legal electronic discovery: computer classification vs. manual review. *Journal of the Association for Information Science and Technology*. 2010, vol. 61, no. 1, pp. 70–80.

[3] PRAKKEN, Henry. On the problem of making autonomous vehicles conform to traffic law. *Artificial Intelligence and Law*. 2017, vol. 25, no. 3, pp. 341–363. ISSN 1572-8382. Available from DOI: 10.1007/ s10506-017-9210-0.

[4] FRANK, Lily; NYHOLM, Sven. Robot sex and consent: Is consent to sex between a robot and a human conceivable, possible, and desirable? *Artificial Intelligence and Law*. 2017, vol. 25, no. 3, pp. 305–323. ISSN 1572-8382. Available from DOI: 10.1007/s10506-017-9212-y.

[5] BROŻEK, Bartosz; JAKUBIEC, Marek. On the legal responsibility of autonomous machines. *Artificial Intelligence and Law*. 2017, vol. 25, no. 3, pp. 293–304. ISSN 1572-8382. Available from DOI: 10.1007/ s10506-017-9207-8.

[6] TURTLE, Howard. Text retrieval in the legal world. *Artificial Intelli- gence and Law*. 1995, vol. 3, no. 1, pp. 5–54. ISSN 1572-8382. Available from DOI: 10.1007/BF00877694.

[7] SMITH, J. C.; GELBART, Daphne; MACCRIMMON, Keith; ATHER- TON, Bruce; MCCLEAN, John; SHINEHOFT, Michelle; QUINTANA, Lincoln. Artificial intelligence and legal discourse: The Flexlaw legal text management system. *Artificial Intelligence and Law*. 1995, vol. 3, no. 1, pp. 55–95. ISSN 1572-8382. Available from DOI: 10.1007/ BF00877695.

[8] GREENLEAF, Graham; MOWBRAY, Andrew; DIJK, Peter van. Rep- resenting and using legal knowledge in integrated decision sup- port systems: DataLex WorkStations. *Artificial Intelligence and Law*. 1995, vol. 3, no. 1, pp. 97–142. ISSN 1572-8382. Available from DOI: 10.1007/BF00877696.

[9] RISSLAND, Edwina L.; SKALAK, David B.; FRIEDMAN, M. Timur. BankXX: Supporting legal arguments through heuristic retrieval. *Artificial Intelligence and Law*. 1996, vol. 4, no. 1, pp. 1–71. ISSN 1572- 8382. Available from DOI: 10.1007/BF00123994.

[10] FREEMAN, Kathleen; FARLEY, Arthur M. A model of argumentation and its application to legal reasoning. *Artificial Intelligence and Law*. 1996, vol. 4, no. 3, pp. 163–197. ISSN 1572-8382. Available from DOI: 10.1007/BF00118492.

[11] BENCH-CAPON, Trevor. Argument in Artificial Intelligence and Law. *Artificial Intelligence and Law*. 1997, vol. 5, no. 4, pp. 249–261. ISSN 1572-8382. Available from DOI: 10.1023/A:1008242417011.

[12] ASHLEY, Kevin D; WALKER, Vern R. From Information Retrieval (IR) to Argument Retrieval (AR) for Legal Cases: Report on a Baseline Study. In: *JURIX*. 2013, pp. 29–38.

[13] VISSER, Pepijn R. S.; BENCH-CAPON, Trevor J. M. A Comparison of Four Ontologies for the Design of Legal Knowledge Systems. *Artificial Intelligence and Law*. 1998, vol. 6, no. 1, pp. 27–57. ISSN 1572-8382. Available from DOI: 10.1023/A:1008251913710.

[14] VAN ENGERS, Tom; BOER, Alexander; BREUKER, Joost; VALENTE, André; WINKELS, Radboud. Ontologies in the Legal Domain. In: *Digital Government: E-Government Research, Case Studies, and Imple- mentation*. Ed. by CHEN, Hsinchun; BRANDT, Lawrence; GREGG, Valerie; TRAUNMÜLLER, Roland; DAWES, Sharon; HOVY, Eduard; MACINTOSH, Ann; LARSON, Catherine A. Boston, MA: Springer US, 2008, pp. 233–261. ISBN 978-0-387-71611-4. Available from DOI: 10.1007/978-0-387-71611-4_13.

[15] SHAPIRA, Ron A. Fuzzy measurement in the Mishnah and the Tal- mud. *Artificial Intelligence and Law*. 1999, vol. 7, no. 2, pp. 273–288. ISSN 1572-8382. Available from DOI: 10.1023/A:1008321725690.

[16] PHILIPPS, Lothar; SARTOR, Giovanni. Introduction: from legal the- ories to neural networks and fuzzy reasoning. *Artificial Intelligence and Law*. 1999, vol. 7, no. 2, pp. 115–128. ISSN 1572-8382. Available from DOI: 10.1023/A:1008371600675.

[17] STRANIERI, Andrew; ZELEZNIKOW, John; GAWLER, Mark; LEWIS, Bryn. A hybrid rule neural approach for the automation of legal reasoning in the discretionary domain of

family law in Australia. *Artificial Intelligence and Law*. 1999, vol. 7, no. 2, pp. 153–183. ISSN 1572-8382. Available from DOI:10.1023/A:1008325826599.

[18] BOURCIER, Daniéle; CLERGUE, Gérard. From a rule-based conception to dynamic patterns. Analyzing the self-organization of legal systems. *Artificial Intelligence and Law*. 1999, vol. 7, no. 2, pp. 211–225. ISSN 1572-8382. Available from DOI: 10.1023/A:1008388719330.

[19] BORGULYA, István. Two examples of decision support in the law. *Artificial Intelligence and Law*. 1999, vol. 7, no. 2, pp. 303–321. ISSN 1572-8382. Available from DOI:10.1023/A:1008384601583.

[20] OSKAMP, Anja; LAURITSEN, Marc. AI in law practice? So far, not much. *Artificial Intelligence and Law*. 2002, vol. 10, no. 4, pp. 227–236.

[21] DAHBUR, Kamal; MUSCARELLO, Thomas. Classification System for Serial Criminal Patterns. *Artificial Intelligence and Law*. 2003, vol. 11, no. 4, pp. 251–269. ISSN 1572-8382. Available from DOI: 10.1023/B: ARTI.0000045994.96685.21.

[22] CHORLEY, Alison; BENCH-CAPON, Trevor. AGATHA: Using heuristic search to automate the construction of case law theories. *Artificial Intelligence and Law*. 2005, vol. 13, no. 1, pp. 9–51. ISSN 1572-8382. Available from DOI: 10.1007/s10506-006-9004-2.

[23] HACHEY, Ben; GROVER, Claire. Extractive summarisation of legal texts. *Artificial Intelligence and Law*. 2006, vol. 14, no. 4, pp. 305–345. ISSN 1572-8382. Available from DOI: 10.1007/s10506-007-9039-z.

[24] ABOOD, Aaron; FELTENBERGER, Dave. Automated patent landscaping. *Artificial Intelligence and Law*. 2018. ISSN 1572-8382. Available from DOI: 10.1007/s10506-018-9222-4.

[25] HEMBERG, Erik; ROSEN, Jacob; WARNER, Geoff; WIJESINGHE, Sanith; O'REILLY, Una-May. Detecting tax evasion: a co-evolutionary approach. *Artificial Intelligence and Law*. 2016, vol. 24, no. 2, pp. 149– 182. ISSN 1572-8382. Available from DOI: 10.1007/s10506-016- 9181-6.

[26] CAREY, Matthew. Holdings about holdings: modeling contradictions in judicial precedent.

Artificial Intelligence and Law. 2013, vol. 21, no. 3, pp. 341–365. ISSN 1572-8382. Available from DOI: 10.1007/ s10506-013-9141-3.

[27] MARNEFFE, Marie-Catherine; RAFFERTY, Anna N; MANNING, Christo- pher D. Finding contradictions in text. *Proceedings of ACL-08: HLT*. 2008, pp. 1039–1047.

[28] FENTON, Norman; NEIL, Martin; LAGNADO, David A. A General Structure for Legal Arguments About Evidence Using Bayesian Networks. *Cognitive Science*. 2012, vol. 37, no. 1, pp. 61–102. Available from DOI: 10.1111/cogs.12004.

[29] VLEK, Charlotte S.; PRAKKEN, Henry; RENOOIJ, Silja; VERHEIJ, Bart. A method for explaining Bayesian networks for legal evidence with scenarios. *Artificial Intelligence and Law*. 2016, vol. 24, no. 3, pp. 285–324. ISSN 1572-8382. Available from DOI: 10.1007/s10506- 016-9183-4.

[30] BARNETT, Thomas; GODJEVAC, Svetlana; RENDERS, Jean-Michel; PRIVAULT, Caroline; SCHNEIDER, John; WICKSTROM, Robert. Machine learning classification for document review. In: *DESI III: The ICAIL Workshop on Globaal E-Discovery/E-Disclosure*. 2009.

[31] NOORTWIJK, Kees van; VISSER, Johanna; DE MULDER, Richard V. Ranking and classifying legal documents using conceptual informa- tion. *The Journal of Information, Law & Technology*. 2006.

[32] THOMPSON, Paul. Automatic categorization of case law. In: *Proceed- ings of the 8th international conference on Artificial intelligence and law*. 2001, pp. 70–77.

[33] ZHAO, Feng C; OARD, Douglas W; BARON, Jason R. Improving search effectiveness in the legal e-discovery process using relevance feedback. In: *ICAIL 2009 DESI III Global E-Discovery/E-Disclosure Workshop*. 2009.

[34] OPIJNEN, Marc van; SANTOS, Cristiana. On the concept of relevance in legal information retrieval. *Artificial Intelligence and Law*. 2017, vol. 25, no. 1, pp. 65–87. ISSN 1572-8382. Available from DOI: 10. 1007/s10506-017-9195-8.

[35] HOGAN, Christopher; BAUER, Robert S.; BRASSIL, Dan. Automation of legal sensemaking in e-discovery. *Artificial Intelligence and Law*. 2010, vol. 18, no. 4, pp. 431–457.

ISSN 1572-8382. Available from DOI: 10.1007/s10506-010-9100-1.

[36] ASHLEY, Kevin D.; BRIDEWELL, Will. Emerging AI & Law approaches to automating analysis and retrieval of electronically stored information in discovery proceedings. *Artificial Intelligence and Law*. 2010, vol. 18, no. 4, pp. 311–320. ISSN 1572-8382. Available from DOI: 10.1007/s10506-010-9098-4.

[37] BRASSIL, Dan; HOGAN, Christopher; ATTFIELD, Simon. The centrality of user modeling to high recall with high precision search. In: *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*. 2009, pp. 91–96.

[38] *How many words are there in the English language?* [online] [visited on 2018-02-27]. Available from: https://en.oxforddictionaries. com/explore/how-many-words-are-there-in-the-english- language.

[39] SCOTT, Mike; TRIBBLE, Christopher. *Textual patterns: Key words and corpus analysis in language education*. John Benjamins Publishing, 2006.

[40] MANNING, Christopher D; RAGHAVAN, Prabhakar; SCHÜTZE, Hinrich, et al. *Introduction to information retrieval*. Cambridge university press Cambridge, 2008. No. 1.

[41] JURAFSKY, Dan; MARTIN, James H. *Speech and language processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson London, 2014.

[42] CHITICARIU, Laura; LI, Yunyao; REISS, Frederick R. Rule-based in- formation extraction is dead! long live rule-based information ex- traction systems! In: *Proceedings of the 2013 conference on empirical methods in natural language processing*. 2013, pp. 827–832.

[43] LASSILA, Ora; SWICK, Ralph R. Resource description framework (RDF) model and syntax specification. 1999.

[44] HEARST, Marti A. Automatic acquisition of hyponyms from large text corpora. In: *Proceedings of the 14th conference on Computational linguistics-Volume 2*. 1992, pp. 539–545.

[45] MINTZ, Mike; BILLS, Steven; SNOW, Rion; JURAFSKY, Dan. Distant supervision for

relation extraction without labeled data. In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. 2009, pp. 1003–1011.

[46] BLEI, David M; NG, Andrew Y; JORDAN, Michael I. Latent dirichlet allocation. *Journal of machine Learning research*. 2003, vol. 3, no. Jan, pp. 993–1022.

[47] XIAO, Han; STIBOR, Thomas. Efficient collapsed gibbs sampling for latent dirichlet allocation. In: *Proceedings of 2nd Asian Conference on Machine Learning*. 2010, pp. 63–78.

[48] BIRD, Steven; LOPER, Edward. NLTK: the natural language toolkit. In: *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*. 2004, p. 31.

[49] BIRD, Steven; KLEIN, Ewan; LOPER, Edward. *Natural language processing with Python: analyzing text with the natural language toolkit*. "O'Reilly Media, Inc.", 2009.

[50] PEDREGOSA, Fabian et al. Scikit-learn: Machine learning in Python. *Journal of machine learning research*. 2011, vol. 12, no. Oct, pp. 2825– 2830.

[51] *Brown corpus: Corpus of American English* [online] [visited on 2020-02-12]. Available from: https://www.sketchengine.co.uk/brown-corpus/.

[52] *Google Cloud Natural Language* [online] [visited on 2020-01-19]. Avail- able from: https://cloud.google.com/natural-language/.

[53] DEVARAJAN, Deepika. *Retirement of AlchemyAPI service* [online]. 2017 [visited on 2018-02-08]. Available from: https://www.ibm.com/ blogs/bluemix/2017/03/bye-bye-alchemyapi/.

[54] *Watson Natural Language Understanding* [online] [visited on 2020-01-19]. Available from: https://www.ibm.com/watson/services/natural- language-understanding/.

[55] *Aylien* [online]. 2018 [visited on 2020-01-19]. Available from: https: //aylien.com/.

[56] AMERICAN MOCK TRIAL ASSOCIATION AND NATIONAL IN- STITUTE FOR TRIAL ADVOCACY (U.S.) *Davis V. HappyLand Toy Company*. NITA, 2015. Case file /American Mock Trial Association. ISBN 9781601565204. Available also from:

https://books.google. cz/books?id=DrfljgEACAAJ.

[57] SMEETON, Nigel C. *Early history of the kappa statistic*. JSTOR, 1985.

[58] LEVANDOWSKY, Michael; WINTER, David. Distance between sets. *Nature*. 1971, vol. 234, no. 5323, pp. 34.

[59] SARWAN, Neeraj Singh. *An Intuitive Understanding of Word Embed- dings: From Count Vectors to Word2Vec* [online] [visited on 2018-03-19]. Available from: https://www.analyticsvidhya.com/blog/2017/ 06/word-embeddings-count-word2veec/.

[60] MIKOLOV, Tomas; CHEN, Kai; CORRADO, Greg; DEAN, Jeffrey. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*. 2013.

[61] MIKOLOV, Tomas; YIH, Wen-tau; ZWEIG, Geoffrey. Linguistic reg- ularities in continuous space word representations. In: *Proceedings of the 2013 Conference of the North American Chapter of the Associa- tion for Computational Linguistics: Human Language Technologies*. 2013, pp. 746–751.

[62] GOOGLE. *Google code archive: word2vec* [online] [visited on 2020-02-19]. Available from: https://code.google.com/archive/p/word2vec/.

[63] VEDANTAM, Ramakrishna; LIN, Xiao; BATRA, Tanmay; LAWRENCE ZITNICK, C; PARIKH, Devi. Learning common sense through vi- sual abstraction. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2542–2550.

[64] *Strategies to scale computationally: bigger data* [online] [visited on 2020-02-19]. Available from: http://scikit-learn.org/stable/modules/ scaling_strategies.html.

[65] GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[66] MACHOVEC, Petr. *Automatická sumarizace textu*. 2015 [cit. 2020-02- 17]. Available also from: https://is.muni.cz/th/uorx4/. Master Thesis. Masaryk University, Faculty of informatics, Brno. Supervised by Zuzana NEVĚŘILOVÁ.