UNIVERSITY OF KONSTANZ

MASTER THESIS

Predicting Court Decision: a Multimodal Deep Learning Perspective

Author:
Wenting WANG

Matr.-Nr:
01/772868

Period of completion:
29 April 2019 - 29 August 2019

1st Assessor:
Prof. Dr. Lyudmila GRIGORYEVA

2nd Assessor:
Prof. Dr. Daniel A. KEIM

A thesis submitted in partial fulfillments of the requirements
for the degree of Master of Science (M.Sc.)
Social and Economic Data Science

at the

Department of Economics
University of Konstanz

Konstanz, 29 August 2019

Abstract

Judicial decision prediction is one of the key research areas in legal technology, which plays an important role in legal assistance and oral arguments strategy analysis. In the field of artificial intelligence, deep learning is springing up in recent years. By building hierarchical structures, deep learning can extract features hierarchically, from the lower level to higher level, and automatically recognise patterns of input data, thus improve the classification performance. Multimodel deep learning (MMDL) is a particular deep learning technique based on multiple sources of data. By implementing fusion on data, feature or decision level, it enables the most use of data. In this paper, a feature-level fusion and decision-level fusion experiments based on multimodal deep learning are proposed. It is one of the first attempts to apply the MMDL approach to court decision prediction. The cases information of the Supreme Court of the United States and related oral arguments corpus were used for experiments. The result shows that the proposed MMDL method based on multi-source data and fusion techniques outperform machine learning and simple neural network.

Keywords multimodal deep learning; legal technology; court decision prediction

Acknowledgements

I would first like to thank my thesis advisor Prof. Grigoryeva of the Graduate School of Decision Sciences at University Konstanz. The door to Prof. Grigoryeva's office was always open whenever I ran into a trouble spot or had a question about my research or writing. She consistently allowed this paper to be my work but steered me in the right the direction whenever she thought I needed it.

I would also like to thank Prof. Keim and his Ph.D. student Mennatallah El-Assady who were involved in the thesis topic proposal for this research project. Without their passionate participation and input, the proposal could not have been successfully conducted.

I would also like to acknowledge Dr. N.J. Wang of the Law Department and Y. Wang of the Computer Science Department at University Konstanz as the second readers of this thesis, and I am gratefully indebted to them for their very valuable comments on this thesis.

Finally, I must express my very profound gratitude to my parents and to my boyfriend for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Contents

Abstract iii

Acknowledgements v

1 Introduction 1
1.1 Background . . . 1
1.2 Research Questions . . . 1
1.3 Structure of the Thesis . . . 2

2 Related Work 3
2.1 Law . . . 3
2.1.1 Legal Technology . . . 3
Background . . . 3
Legal information retrieval . . . 3
Legal documents automation . . . 4
Alternative legal service . . . 4
Others . . . 5
2.1.2 Decision Prediction . . . 5
2.1.3 Judiciary Oral Arguments before the SCOTUS . . . 6
Procedure of oral arguments . . . 6
Purposes of the oral arguments . . . 7
2.1.4 Emotion and Law . . . 7
Emotion . . . 7
Law and emotion . . . 8
2.2 Methodology, Models and Frameworks . . . 8
2.2.1 The Development of Deep Learning . . . 8
2.2.2 Basic Elements of Neural Networks . . . 10
Multilayer perception . . . 10
Convolutional neural network . . . 10
Recurrent neural network . . . 12
Word embedding . . . 13
2.2.3 Applications of Deep Learning . . . 13
Speech Processing . . . 13
Computer Vision . . . 13
Natural Language Processing . . . 14
Others . . . 14
2.2.4 Multimodal Deep Learning . . . 14
Overview of multimodal deep learning . . . 14
Structures and models . . . 15
Applications in diverse areas . . . 17
Advantages of multimodal deep learning . . . 18
Methods in multimodal learning . . . 18
2.2.5 Framework . . . 19

3 Data and Models **21**
3.1 Data . . . 21
3.1.1 Data Source . . . 21
U.S. Supreme Court Database (SCDB) . . . 21
SCOTUS Oral Arguments Corpus . . . 22
3.1.2 Data Processing . . . 22
Target . . . 22
Structured data . . . 23
Text data . . . 23
Audio data . . . 24
Mixed data alignment . . . 24
3.2 Models Construction . . . 24
3.2.1 Model Components . . . 25
Multilayer perceptron . . . 25
Models for natural language processing . . . 25
3.2.2 Multimodal Deep Learning . . . 27
Feature-level fusion . . . 27
Decision-level fusion . . . 27
3.2.3 Evaluation . . . 29

4 Results and Discussion **31**
4.1 Data Exploration . . . 31
4.2 Feature-fusion Results . . . 32
Justice level results of feature-fusion experiments . . . 32
Case level results of feature-fusion experiments . . . 32
Inferred case level results . . . 33
4.3 Decision-fusion Results . . . 33
Justice level results of decision-fusion experiments . . . 33
Case level results of decision-fusion experiments . . . 34
Inferred case level results . . . 35

5 Conclusion **37**

A Appendix Title **39**
A.1 Hardwares . . . 39
A.2 Variables of SCDB . . . 39
A.3 Codes . . . 40

List of Figures

2.1 The relation of Artificial Intelligence, Machine Learning and Deep Learning . . . 10
2.2 Function illustration of convolutional layers and pooling layers . . . 11
2.3 Architecture of LeNet-5, a Convolutional Neural Network . . . 12
2.4 Data-level fusion . . . 15
2.5 Figure-level fusion . . . 16
2.6 Decision-level fusion . . . 17

3.1 The feature-level fusion experiment procedure . . . 28
3.2 The decision-level fusion experiment procedure . . . 30

4.1 The trend of number of cases . . . 32

List of Tables

4.1 The results of feature-level fusion experiments on justice level 33
4.2 The results of feature-level fusion experiments on case level 33
4.3 The results of case level prediction inferred from feature-level fusion experiments on justice level 34
4.4 The results of decision-level fusion experiments on justice level 34
4.5 The results of decision-level fusion experiments on case level 35
4.6 The results of case level prediction inferred from decision-level fusion experiments on justice level 35

A.1 The hardwares used in experiments 39

List of Abbreviations

AI	**A**rtificial **I**ntelligence
RNN	**R**ecurrent **N**eural **N**etwork
CNN	**C**onvolutional **N**eural **N**etwork
1D-CNN	**One-D**imensional **C**onvolutional **N**eural **N**etwork
GRU	**G**ated **N**eural **N**etwork
LSTM	**L**ong **S**hort **T**erm **Memory**
DL	**D**eep **L**earning
ML	**M**achine **L**earning
NLP	**N**atural **L**anguage **P**rocessing

List of Symbols

y_j^l	output of neuron j of layer l
z_i^{l+1}	value before activated by neuron i of layer $l+1$
W_{ji}^l	weights between neuron j of layer l and neuron i of layer $l+1$
b_i^l	bias
$f(\cdot)$ or $f_h(\cdot)$	nonlinear activation function
x_i^t	neuron i of input layer at time t
$a_{h'}^{t-1}$	neuron h' of hidden layer at time $t-1$
h'	a neuron of hidden layer
h_t^j	activation of the GRU at time t
$\bar{h}_t^j$	candidate activation of the GRU at time t
z_h^t	neuron h of hidden layer before activated at time t
p_t^j	an update gate of GRU
y_k^t	neuron k of output layer at time t
w_{ih}	weights between the input layer and the hidden layer
$w_{h'h}$	weights between the hidden layers
w_{hk}	weights and between the hidden layer and the output layer
r_t^j	the reset gate of GRU

Chapter 1

Introduction

1.1 Background

With the fast development of technologies such as big data, artificial intelligence, and cloud computing in recent years, judicial institutes, law offices, and technology companies have introduced such techniques into the judicial field and legal services. The transition of the expert system based on rules, to the legal artificial intelligence system based on big data, is constantly making new achievements in applications such as assisting judges for cases judgment, serving lawyers for materials retrievals and writing, and enabling faster, cheaper and more reliable legal advice to the public.

The judicial prediction, as a subsidiary research area developed since decades, has been applying techniques to many legal applications such as court decision prediction, criminal prediction, trademark lawsuit prediction. The prediction facilitates lawyers to optimise appeal and oral arguments, helps justices to clarify the cases, ensures more objective legal judgement, reduces unnecessary cost of some legal services. Moreover, investors can make investment decisions based on the predictive analysis, and the appellant can decision wether to file a lawsuit based on the probability of winning, etc. Aletras et al. (2016) predicts the judicial decisions of the European court of human right with an accuracy of 79%. By predicting the case outcomes Byrd and Howard (2014) can assist to improve strategies of oral arguments based on the certain styles of justices or lawyers. Katz et al. (2017) applied random forest classifier to predict the decisions of the Supreme Court of the United States, which reaches an accuracy of 71 %.

However, judicial prediction research still faces many challenges. Although the volume of data is large, the diversity, relevance, and meaning of data are hard and seldom to be associated, extracted. For example, there are applications (e.g., Prediction of sentencing in criminal cases) based on millions of open judicial adjudicative document, but other data such as cases interpretations, articles explanation or the trial recording, is insufficient utilised.

From the data aspect, there is no standardised system, unstructured data processing required particular techniques, and labeled data is still in short. On the technique side, machine learning and shallow neural network are mainly used for prediction task, yet these methods require manual feature engineering and are inappropriate to process unstructured data, such as images, audio, video, and texts.

1.2 Research Questions

Different from the previous methods, this paper proposes a multimodel deep learning model based on multiple types of input data (cases background information, oral arguments audio records and transcripts). Moreover, two experiment procedures

with the fusion process in the different period were implemented. In summary, this paper is attempting to answer the questions as follows.

(1) How does the subsidiary model perform on the court decision prediction, using cases background information, oral arguments audio records, and oral arguments transcripts, independently? Which subsidiary model structure gives a better result for the corresponding type of data?

(2) How does the concatenated model based on multimodal deep learning performance on the court decision prediction, using two or three types of data?

(3) Does the fusion period influence the performance of models on prediction? Which fusion procedure performs better, feature-level or decision-level fusion?

(4) How do the models perform on justice level and case level prediction, independently? Is the results on case level inferred by the justice level experiments, the same as the results obtained by justice level experiments themselves?

1.3 Structure of the Thesis

In the following chapters, the Related Work reviews relevant research literature in the field of legal technology and deep learning methods. The Data and Methods introduce the collecting and processing of data, and the methods and experimental procedures. The Results and Discussion shows the results of the experiments, on justice and case level, and feature-level and decision-level fusion, independently. The Conclusion states the contribution and limitation of this paper.

All the codes used in this paper can be found in the Github repository:
https://github.com/wwendi/thesis-code/tree/master/code

Chapter 2

Related Work

2.1 Law

2.1.1 Legal Technology

Background

The corporation of law and informatics dates back to 1987, when the first International Conference on Artificial Intelligence and Law (ICAIL) [1] was conducted in Boston, which facilitated the establish of International Association for Artificial Intelligence and Law (IAAIL) [2] in 1991 and the research and application of the cross-disciplinary field of law and technology, such as formal model of law reasoning, computational model for argument and decision making, computational model for evidence inference and reasoning, law reasoning in multi-intelligence system, automatic legal document classification and abstraction, automatic information retrieval from legal datasets, machine learning and data mining on legal application (e.g., evidence obtaining), law robots for trivial and repetitive tasks, etc.

In such environment, legal technology as a new field is fast growing up. Supported by artificial intelligence, technology is expected to bring more innovative and deep changes to law area. Though investment in legal-tech is still far less than finance and medicine technology industries, the number of legal technology public companies is globally fast boosting, from 15 in 2009 to from than a thousand in 2018, mainly focus on online legal service, electronic evidence obtaining, intellectual property/trademark management, artificial intelligence law technology, legal search, lawyer recommendation, notary, etc. Moreover, legal institutions such as law office, law departments of companies and court have also been joining the research, development and investment of the legal-tech area. The trend topics of legal technology are briefly introduced in follows.

Legal information retrieval

The digitalization of law material such as Legal documents, judgement opinions and articles has supported information search in massive datasets. The traditional legal datasets service such as Westlaw (Arewa, 2006) [3] and PKULaw [4] provides search results based on keywords, which is inefficient and not accurate. New legal information retrieval service is based on techniques such as natural language processing

[1] ICAIL: https://icail2019-cyberjustice.com
[2] IAAIL: www.iaail.org
[3] Westlaw: https://legal.thomsonreuters.com › products › westlaw
[4] PKULaw: https://www.pkulaw.cn/

(NLP) and understanding, semantics retrieval and legal question and answer system, for example, the first robot layer ROSS Intelligence [5], an intelligent information retrieval API based on IBM Watson system, is able to provide most relevant and valuable results instead of plenty of plain items. Moreover, semantics and text analysis, video and image processing techniques provides automatic retrieval workflow for trademark and intellectual property and copyright (Schafer, 2016) [6].

The development of legal information retrieval based on speech conversation has two steps. The first step is the intelligence retrieval, where layers will provides questions or keywords for retrieval system, collect relevant cases, evaluate and give professional answers.The second step is automatic retrieval, an end-to-end workflow without human involvement, where retrieval system will automatically understand the fact description, recognize legal questions, search in datasets for most relevant and valuable information and provides final report.

Legal documents automation

Automatically investigation and analysis on legal documents, evidences and contracts has obviously increased working efficiency. For example, for cases of merger and acquisition, a massive digital evidences and legal documents are ought to be collected, cleaned up and edited, which will cost large human force and time. Automatically evidence extracting techniques based on natural language processing, techniques assistant read (TAR), predictive coding, etc. largely increase the efficiency and provide even higher accuracy. Another attention of legal documents automation is on contract analysis, which is significant in risk management, investigation, litigation, etc. however rather time costly. By using machine learning techniques, Deloitte enables Kira Systems (Chan et al., 2018) [7] to complete contract analysis in 15 minutes, which is a 12-hour task for a human layer. Gradually more research and institutes are developing applications on contract analysis such as KMStandards [8], RAVN [9], Seal Software [10], Legal Beagle [11], LawGeex [12].

Automatic text generation and writing assistant of legal documents such as plaint, judgement opinion, legal notes will largely increase efficiency. Layers' works transfer from writing to check, edit and approve. Fenwick & West LLP [13] provides a system enabling documents generation for start-ups' initial public offerings, which reduce writing time from 20-40 hours to 5 hours. As data increases, intelligence machine is able to learn and continuously update; while relevant cases and judgements can be connected and enable an automatic dynamic connecting and updating system.

Alternative legal service

Alternative legal service such as online law office and robots lawyers provides customers general legal consultant service (e.g., testament, marriage consultant, traffic accident consultant). Legal robots DoNotPay () assist customers to prepare appeal

[5] ROSS Intelligence: https://rossintelligence.com
[6] TrademarkNow: https://www.trademarknow.com
[7] Deloitte Alliance Kira Systems: https://kirasystems.com
[8] KMStandards: kmstandards.com
[9] RAVN: https://imanage.com/product/ravn
[10] Seal Software: https://www.seal-software.com
[11] Legal Beagle: https://legalbeagle.com
[12] LawGeex: https://www.lawgeex.com
[13] Fenwick & West LLP: https://www.fenwick.com

materials to traffic fares. The alternative legal service is standardized, commercial, automatic and democratic, which provides legal service with lower cost and is not depends on experience and professionals of certain layers, and further remove justice gap and improve access to general justice. For example, LegalZoom (), an online legal service provider, satisfies large amount of legal demand or demand that is costly in law offices. The United Kingdom passed the Legal Services Act in 2007 [14], aiming to liberalize the legal market, reform the legal industry organization model, and introduce competition to promote the affordability of legal services. In this context, some international law firms have established low-cost legal service centers to provide legal services at lower prices through technology, in addition to hourly billing and fixed fees.

Generally speaking, from rule-based legal expert system, which transform the legal articles and knowledge as rules to formal language compatible to computers, to automatic system leveraging on machine learning, deep learning and massive data, technology gradually influences the law industries, not only from techniques aspect, but also shapes new opinions and extent of acceptance of technology in law area.

Others

As techniques developing, more artificial intelligence and robotics are applied in legal areas. Online dispute resolution (ODR) (e.g., SquareTrade [15]) offers solutions to clients by processing the factual description and evidence they submitted, without lawyers or courts involving. Evaluation and recommendation of lawyers becomes another significant research area, from which a more transparent and efficient lawyers market is promising (Susskind, 2017; Alarie et al., 2018). A braver hypothesis was put forward by Briggs (2015), that some judgements could be made by AI though computational law, which explores to represent laws by computational logics and codes. Translating the legal cases and articles into codes, so that the judgment is out of personal subjectivity, and is reachable to everyone. Such an open-source system can be set online for public supervision.

2.1.2 Decision Prediction

Prediction techniques based on artificial intelligence has many applications in judicial area such as court decision prediction and criminal prediction. In 2006, a natural language processing model designed by Aletras et al. (2016) can predict the judicial decisions of the European court of human right with an accuracy of 79 %.

Decision prediction was also applied in law industry, for example, by using NLP techniques on judgement documents, Lex Machina (Byrd and Howard, 2014) [16] , a legal technology company, can predict the case outcomes, assist to improve oral arguments strategy based on the certain styles of justices or lawyers. Such prediction techniques has already been used in intellectual property cases. (Posner, 2005).

Katz et al. (2017) applied random forest classifier to predict the decisions of the Supreme Court of the United States. They gathered cases for over two centuries and used related characteristics to predict the result on both case and Justice level, reaching a rather satisfied accuracy compared to prior studies. Though this research

[14]Legal Services Act 2007: https://www.legislation.gov.uk/ukpga/contents
[15]SquareTrade: https://www.squaretrade.com
[16]Legal Analytics by Lex Machina: https://lexmachina.com

analysed information knowable prior to the date of decision, it includes only shallow textual features, for instance, time, location, and issue area of the cases. Other important materials the Justice reply on such as the oral arguments provided before the court was not taken into consideration. Generalisability and consistency was anticipated in the model, which enables the extrapolation of the findings outside the sample. However, explanation power on each case was not achieved, especially considering the dynamics of the cases. Deep learning provides a way of automatically learning representation and extracting information from raw data.

Based on factual descriptions, Luo et al. (2017) proposed an attention-based neural network method to extract relevant cases and predict charges. Similarly, Hu et al. (2018) constructed a neural network model to infer 10 legal attributes (e.g. Violence, Physical Injury, Intentional Crime). Based on these discriminative attributes, the model performed rather well, particularly for few-shot charges and confusing charges. Experimental data from both studies derived from China Judgements Online [17].

The judicial decision prediction can assist petitioners or respondents to construct an optimized appeal or oral argument strategies; while it can also help justices make decisions according to data from plenty of similar cases, which ensures the judgement justice to some extent. Due to the expensive legal service of appeal for litigants, usually an evaluation of cases and winning probability will be implemented before appeal. By training models leveraging on massive data and robust algorithms, computers are able to process all data, instead of a sample of them, to obtain a more precise and credible results. On the other hand, when a predicted winning rate is in a low level such that the continuous appeal is not promising, other solutions will be considered, for example, extra-curial or giving up appeal. The judicial decision prediction, however, may also distort litigation behaviours, which will bring new bias and abuse.

2.1.3 Judiciary Oral Arguments before the SCOTUS

The Supreme Court of the United States (SCOTUS) (Baum, 2018), established in 1789, is the highest court in the federal judiciary of the US. As set by the Judiciary Act of 1869, the Court consists of the chief justice and eight associate justices, each has lifetime tenure. The president, with the advice and consent of the senate, appoints a new justice when there is a vacancy.

Any case comes before the Court is through petitions for writs of certiorari, also termed as "cert". The party that appealed to the Court is the petitioner and the other side is respondent, regardless of which party initiated the lawsuit in the trial court. A cert petition is voted at a conference by the nine justices, where no less than four of nine justices granting a writ of certiorari indicates that, the petition is selected for the oral argument.

Procedure of oral arguments

In an oral argument (Johnson et al., 2006; Johnson, 2004), the lawyers of both petitioner and respondent side will focus on and clarify the facts of a case, submitted beforehand, and argue the reasons they grant or deny the cert petition. The petitioner party will first make a statement for no more than 30 minutes, and choose whether to rebuttal. The party that first discusses should present all the arguments, and make no more statement until the rebuttal. The respondent party will then presents its

[17] China Judgments Online: http://wenshu.court.gov.cn/

argument. With the approval of parties or the Court, amici curiae, as a "friends of the court", may also file briefs. The justices can interrupt and ask questions at any time, even when a lawyer is making a statement. Some justices will even interrupt a lawyer when answering and add some other questions. From October through April, the Court holds two-week oral argument sessions each month. It is usually assumed that the justices are familiar with and have read the submitted briefs of the case.

After the oral argument, the case is submitted for decision, which is by majority vote of the justices. The Court is obligated to issue the final decision by the end of a particular term. The most senior justice in the majority on his or her side will be assigned the initial draft of the Court's opinion.

Purposes of the oral arguments

The main function of oral arguments is to enhance judicial prestige through procedural justice (Greenwald and Schwartz Jr, 2001). Though face-to-face question-and-answer, justices of the Court can clearly hear the claims of two sides of a case. Justices can also present their issues and concerns to the two parties and get immediate response. In other words, the oral argument procedure focus on fast and mutual views communication.

Moreover, due to the requirement of procedural justice, the behaviour of justices will also be viewed by people, media and other professional institutes, who are expecting the justices to raise critical and essential questions during oral arguments. Besides, since the decision procedure of the justices is unpublished and assumed independent, oral arguments offer a chance for the justices not only to communicate with the two parties, but also exchange views among themselves.

The third function of oral argument is to help the Court to clarify the claims of the two parties. As the highest judicial institute, the Supreme Court often meet the cases with major social disputes, whose influences exceed the cases themselves. Therefore, in order to assess the influences of the decisions, the justices usually proposes hypothesis conditions to challenge the two parties. Both sides of the parties are required to apply legal rules, articles and opinions they claim in different ways, to convince the Court that, even in such hypothesis, there is nothing wrong with their claims. In addition, the parties may also be asked by the justices for the unclear arguments in the submitted brief files.

2.1.4 Emotion and Law

The topic of emotion attracted attentions of scholars in various fields including philosophy (Murphy and Hampton, 1990; Nussbaum, 1992), psychology (Clore et al., 1994), and sociology (Thoits, 1989).

Legal scholars, meanwhile, refused that all questions could be answered internally and sought incorporation from other fields (Bandes and Blumenthal, 2012). Legal realists, critical race theorists, and other scholars challenged judicial objectivity (Abrams and Keren, 2009). Feminist jurisprudence argued that law as a pure reasoning process lacks some important qualities, such as empathy and compassion (Henderson, 1987; Minow and Spelman, 1988). According to Bandes (1996), an emotional stance, whether positive or negative, is defined contextually. The field of law and emotion has developed and expanded to several areas – securities (Huang, 2003), risk regulation (Kahan, 2007), foreclosure (White, 2010), family law (Huntington, 2007), and trademark (Bradford, 2008).

Emotion

Emotions are dynamic processes that are integral to decision making (LeDoux, 1998), other than intense, occasional or unpredictable moods. They are formed, interpreted, and communicated in social and cultural contexts, and they assist us in evaluating and reacting to stimuli Bandes and Blumenthal (2012), which could be words, tones, face expressions and body movements. Emotions could be identified and understood and they contain information we intend to convey or reveal implicitly.

Law and emotion

Bandes and Blumenthal (2012) noted that emotions affect law on three levels. (1) Emotions reject the fiction of pure rational reasoning in the identification and implementation of legal norms; it also affects the deliberative process for justices, judges, juries, and other legal actors. (2) Emotions influence behaviours and decision-making both explicitly and implicitly. (3) As a component of social and institutional dynamics, emotions affect collective decision-making.Feigenson and Park (2006) summarised four ways in which emotions influence legal judgement, which are affecting information processing, biasing perception and evaluation, providing cues to blame, and anticipating future emotions that follow a legal judgement.

Empirical studies on decision-making in court is not new to the research field. There has been discussion of emotions impacting negotiation (Ryan, 2005), and the role of empathy in judging (Bandes, 2009; Abrams, 2010). Nevertheless, scant research concerns the influence of emotions on juror and justices regarding oral arguments or reasoning process.

Bandes and Blumenthal (2012) stated that "most current theories adopt a version of a dual-process model, involving some combination of quick, intuitive judgments; and slower, more deliberative judgments. Much of the debate centres on how the two processes interact." The former process is regarded unconscious and automatic (Haidt, 2001). According to this concept, a moral judgement is like an aesthetic judgement –instant feeling of approval or disapproval resulting from stimuli. Two functional magnetic resonance imaging (fMRI) studies (Greene et al., 2001) suggested that emotional engagement influence moral judgments. In addition, such intuitive reasoning tends to involve retributive reasoning, which may be subsequently revised or overridden by slower, more controlled processing (Greene et al., 2001). Therefore, the Justices' questions and judgement during oral arguments follows first process (quick, intuitive), while the reasoning after oral arguments follows the second process (deliberative).

Focusing on the first process, Black et al. (2011) coded the linguistic emotions of Supreme Court Justices' questions at oral argument, and they found that the more unpleasant language Justices use towards the attorney, the less likely that side would prevail. Dietrich et al. (2019) indicated that, besides what the justices say, the way they say those words may be of equal importance. They defined the level of emotional arousal, as measured by vocal pitch of Justices' questions at oral argument.

These methods used in judicial decision prediction, however, are mainly shallow encoded, which cannot extract sufficient representative information correlated with the final votes, or some features may have overlapped information. Methods based on deep learning techniques are used in the following experiment, which enables automatic features representation extraction.

2.2 Methodology, Models and Frameworks

2.2.1 The Development of Deep Learning

For decades, continuous investment in and research on Artificial Intelligence (AI) in both academia and industry (e.g., technology, biology, finance, mobile) has gradually changed the everyday life of human beings. Deep Learning is one of the most important research fields of AI, which enables various applications such as speech recognition, natural language processing, computer vision, and automotive driving

The history of Deep Learning can be traced back to 1943, when McCulloch and Pitts (1943) brought forward the mathematical model based on neural networks of the human brain. In 1958, the first version of single-layer perceptron model was provided by Rosenblatt (1958), which could distinguish simple shapes like triangles and squares. It was at that time people began to image creating intelligent machines with the ability to percept, learn, and memorize. However, the limitations of the first version perceptron are clear. In 1969, Minsky and Papert (2017) stated that the simple perceptron model is unable to solve XOR problem due to the fixed human-designed layers, which brought the research of Deep Learning into a long cooling period. In 1986, Rumelhart et al. (1988) extended the neural network to incorporate multiple hidden layers, sigmoid activation functions, and back-propagation model training process, thus solving the nonlinear classification issues. In 1989, Cybenko (1989); Hornik et al. (1989) put forward the universal approximation theorem stating that a feed-forward network with a single hidden layer containing a finite number of neurons can approximate any continuous function for input within a specific range. In the same year, LeCun et al. (1989, 1990) invented Convolutional Neural Net- works (CNN) and applied it for hand-written digits recognition. From 1991, the development of neural network models slowed down due to the gradient vanishing problem. Various shallow machine learning models were proposed, including the famous Support Vector Machine invented by Cortes and Vapnik (1995). In 2006, Hinton et al. (2005) discussed graphical model in the brains, suggesting a method of auto-encoder (Hinton and Salakhutdinov, 2006) to reduce dimensionality of data to enable fast training of the Deep Belief Nets by pre-training methods. Bengio et al. (2007) showed that the method of pre-training can be extended to unsupervised learning of auto-encoder. Poultney et al. (2007) proposed an efficient learning method of sparse representations with an energy-based model. The above-mentioned studies shaped the future of Deep Learning, leading to its subsequent fast-developing period. In 2011, Glorot et al. (2011) put forward ReLU activation function, restraining gradient vanishing problem. One of the first and most important break-throughs of deep learning applications is on speech recognition. Hinton et al. (2012); Dahl et al. (2011) reduced the error rate of speech recognition to 30% by deep learning, which is the most important improvement in this field within the decade. In 2012, Krizhevsky et al. (2012) reduced the error rate of ImageNet image (Deng et al., 2009) classification from 26% to 15%. Thereafter, Dauphin et al. (2014); Choromanska et al. (2015) independently proved that saddle point problem can be attacked and hence is not a severe issue for optimisation.

Deep learning is part of a broader family of machine learning methods. An essential difference between deep learning and shallow machine learning is that deep learning leverages on distributed representations (Bengio et al., 2006) and does not require manual feature extraction. Since a model itself solves classification or regression problem based on the features, the quality of features affects the whole system

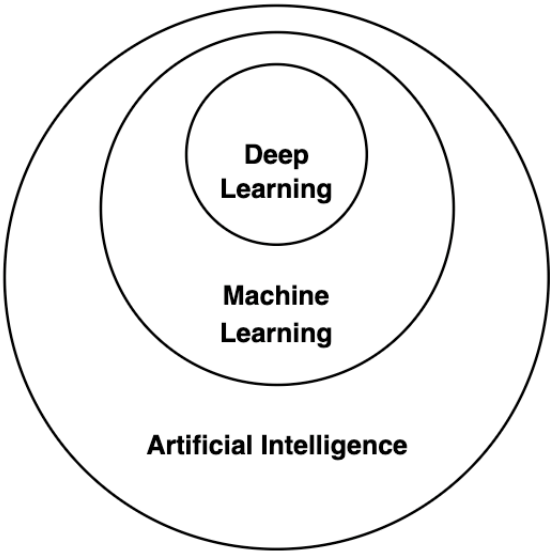

FIGURE 2.1: A Venn diagram showing the relationship of Artificial Intelligence, Machine Learning and Deep Learning

significantly. In addition, feature engineering requires domain knowledge and sufficiently dedicated time, while deep learning as a method of representation learning (Bengio et al., 2013), is able to learn hierarchical concepts by continuously assembling simple concepts to more complex ones and automatically extract features from the data (LeCun et al., 2015; Goodfellow et al., 2016). Deep learning methods have several particular properties. The hidden layers of deep learning are linear combination of input features, while the weights between the hidden layers and in-put layers are the weights of input features in their linear combinations (Bengio et al., 2009). The capacity of deep learning grows exponentially as the depth of model increases.

(Goodfellow et al., 2016) noted that "Deep learning is a particular kind of machine learning that achieves great power and flexibility by learning to represent the world as a nested hierarchy of concepts, with each concept defined in relation to simpler concepts, and more abstract representations computed in terms of less abstract ones."

2.2.2 Basic Elements of Neural Networks

Multilayer perception

Multilayer Perception (MLP)(Rosenblatt, 1958), as a class of feedforward neural network, is the most basic component of deep learning models. An MLP has at least an input layer, a hidden layer and an output layer. Each layer includes neurons that use nonlinear activation functions. MLP utilises a super technique called backpropagation for training. MLPs are sometimes called vanilla neural network, especially when they have a single hidden layer.

MLPs are universal function approximations shown by Chybenko's theorem (Cybenko, 1989), and they enable mathematical models to perform classification or regression analysis in various fields. In particular, classification is a case of general regression where the target variable is categorical. MLPs are a popular machine learning solution back in the 1980s, providing solutions in diverse fields such as speech

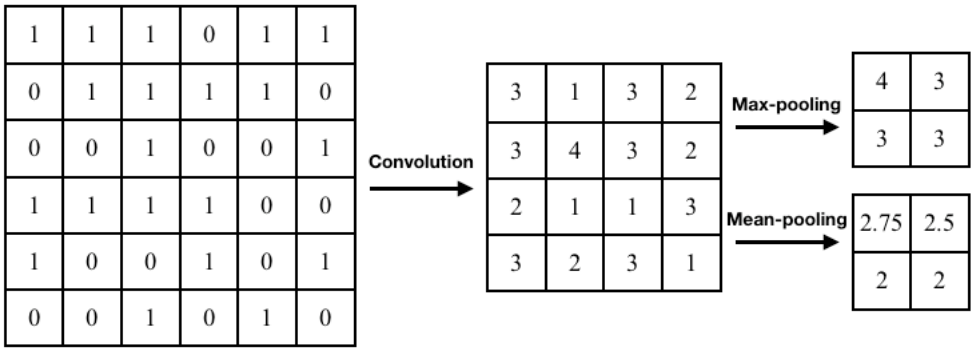

FIGURE 2.2: Function illustration of convolutional layers and pooling layers

recognition, image recognition, machine translation, etc. However, MLPs were then challenged by support vector machines (Cortes and Vapnik, 1995), a related method (Collobert and Bengio, 2004). Recently, interest in neural networks grows back due to the successes of deep learning.

Convolutional neural network

Convolutional Neural Network (CNN) (LeCun et al., 1998) is a class of neural network model and the regularised versions of multilayer perceptrons, most commonly used on spatial data, such as image, video, and natural languages. A single dimensional CNN is also called time delay neural network, which is used to process one-dimensional data. The architecture of CNNs was inspired by biological processes (Fukushima, 2007; Hubel and Wiesel, 1968; Fukushima, 1980; Matsugu et al., 2003) in that "the connectivity pattern between neurons resembles the organisation of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field." The neurons in networks model response simultaneously just as the receptive fields of different neurons partially overlap and work at the same time, which cover the entire visual field. CNNs consist of convolutional layers and pooling layers, where the convolutional layers ensure that an image is spatial continuous so that the local representations can be extracted from it; while by reducing dimensionality of hidden layers, the pooling layers (e.g, max-pooling, mean-pooling) decrease computational complexity in the following steps and maintain rotational invariance of the images. The functions of convolutional and pooling layers are illustrated in Figure 2.2, with the configuration of 3×3 convolutional kernel and a 2×2 pooling layer.

The earliest version of CNN, named LeNet-5, is designed by LeCun et al. (1998) in 1998. The LeNet-5 model architecture is shown in Figure 2.3. At the beginning it was used for image recognition. A MNIST image with size 32×32 is inputted, processed by a convolutional layer with a kernel sized 5×5, obtaining a panel with a size of 28×28, thereafter processed by a pooling layer, obtaining a panel sized 14×14, then through one more convolutional and pooling layer, ending with a panel with the size of 5×5. Finally, the panel goes through the fully connected layers with neurons 120, 84 and 10, respectively, and a Softmax function to get the possibility

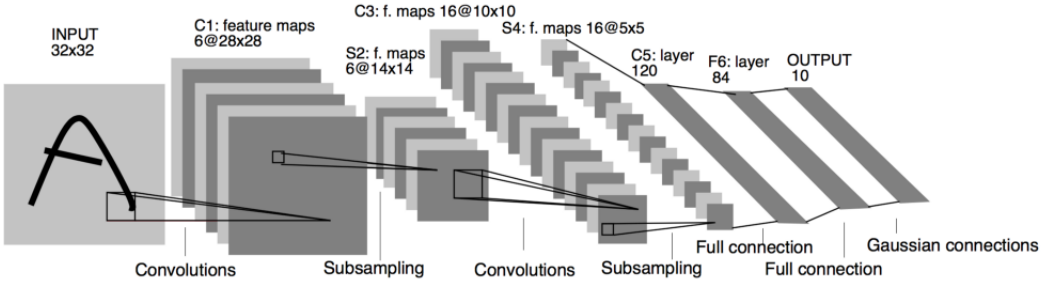

FIGURE 2.3: Architecture of LeNet-5, a Convolutional Neural Network. Each plane is a feature map, i.e., a set of units whose weights are constrained to be identical.

of digit 0 to 9, where the digit with maximum possibility is the predicted result. As iteratively processed by convolutional and pooling layers, sizes (heights and widths) of the panels gradually decrease while depths of the panels increase.

CNN provides hierarchical representations of visual data, where weights in each layer learn a certain component of an image. Therefore, the deeper a layer is in a neural network, the more concrete components the layer can learn. In this way, CNN is able to recognise an object from its separate parts to the entire entity. For example, from a visualization result Zeiler and Fergus (2014) of a CNN model, the second layer responds to corners and other edge/colour conjunctions, the third layer captures similar textures (e.g., mesh patterns, text), the fourth layer shows significant variation (e.g., dog faces, bird legs), the fifth layer shows entire objects with significant pose variation (e.g., dogs, keyboards). CNNs have diverse advantages: training process is efficient on hardwares such as FPGA (DiCecco et al., 2016); the weights in the same convolutional layer share the weights of the convolutional kernel (Goodfellow et al., 2016); the total numbers of parameters largely decrease due to CNNs' local connection, weights sharing and pooling, which also ensures invariance of rotation, skewing, scaling and translating of the images.

Recurrent neural network

Conventional recurrent neural network has issues with gradient vanishing or boosting (Bengio et al., 1994) during the training. Therefore, it cannot use information in the long past sufficiently. For example, when a sigmoid activation function is with a derivative less than 1, the repeated multiplication of such derivatives will lead to gradient vanishing. Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Hierarchical Recurrent Neural Network (El Hihi and Bengio, 1996; Chung et al., 2016) are designed to solve such gradient vanishing/boosting issues. RNNs can only process single dimension data such as time series. However, the multi-dimensional RNN (Graves and Schmidhuber, 2009) as a revised version of RNN, can process multi-dimensional data such as images. Usually the understanding of natural language depends on the meaning of context, Bi-directional RNN is constructed by (Schuster and Paliwal, 1997) to process text sequences in both forward and backward directions. Echo-state RNNs (Jaeger, 2001) offers an efficient training methods to reduce repeated computation on gradient significantly. Simple RNNs are not able to infer. To cope with this, Neural Turing Machines (Graves and Schmidhuber, 2009) and Memory networks (Weston et al., 2014) have been designed with an extra memory block to enable the inference function.

(1) Recurrent neural network (RNN) (Rumelhart et al., 1988). RNN is usually used for processing time series data, and thus has various applications on speech recognition, natural language processing, and weather and stock forecasting. By using the output of the previous layers as the input of the next layers, RNNs can take advantages of past information and memorise some important components from the past. Furthermore, the number of parameters decrease greatly as RNNs share weights across time spans, which reduces training time. The training procedure is, however, more complex than simple multi-layer perceptrons. Therefore, (Sutskever, 2013; Pascanu et al., 2013) put forward some revised training methods for RNNs.

(2) Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997). LSTM is a revised version of RNN architecture, equipped with feedback connections. LSTM replaces the neurons of RNN by LSTM units, which is commonly composed of a cell, an input gate, an output gate and a forget gate. The cell is responsible for keeping track of connections between input and elements of models, and it remembers values overtime. The three gates control the flow of information into and out of the cell; the input gate controls to which extent a value gets into the cell; the forget gate controls to which extent a value remains in the cell; and the output controls to which extent a value in the cell gets into an activation function of the LSTM unit, which is usually a logistic function. LSTM has the advantages of remembering values over long time intervals, giving up insignificant information, partially solving gradient vanishing or booting issues. Therefore, LSTM often performs better than simple RNN on long sequential data.

(3) Gated Recurrent Unit (GRU) (Chung et al., 2014) is a revised and light version of LSTM, which lacks an output gate. It has an update gate and a reset gate: the update gate controls to which extent a value in the past remains and that a new value gets into the cell; the reset gate controls to which extent a value remains in the current cell, similar to the forget gate in LSTM unit. Since there is no output gate in GRU, the output value is complete. GRU architecture has less connection through the entire network, hence it has less parameters and can be trained faster than LSTM. GRUs have been shown to perform better on certain smaller datasets.

Word embedding

Word embedding is representation learning of language modelling in natural language processing. It transforms words in high dimensionality to a lower-dimensional space, with each word as a vector in the real number space. The methods of word embedding include neural network (Mikolov et al., 2013), co-occurrence matrix dimensionality reduction (Lebret and Collobert, 2013; Levy and Goldberg, 2014b; Li et al., 2015), probability model (Globerson et al., 2007) and word representations in context (Levy and Goldberg, 2014a). By using word embedding to represent words or phrase, the performance of text processor, such as grammar parser (Socher et al., 2013a) and sentiment analysis (Socher et al., 2013b), has improved.

GloVe (Global Vectors for word representation) is an open-source unsupervised learning algorithm for obtaining vector representations for words, developed by Stanford in 2014 (Pennington et al., 2014). Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

2.2.3 Applications of Deep Learning

Speech Processing

Deep learning had its first breakthrough application in the field of speech processing (Hinton et al., 2012), on both standard small datasets (Mohamed et al., 2011), and large datasets (Dahl et al., 2011). Two main tasks are involved in this area: speech recognition and speech synthesis (also known as Text-to-Speech). In 2016, Microsoft speech recognition system (Xiong et al., 2016) on conservational speech got an accuracy of 5.6 %, achieving human parity for the first time. In 2018, Google introduced a state-of-art speech recognition end-to-end model (Chiu et al., 2018). Technology companies such as Google, Apple, iFlytek have successfully implemented deep learning on speech synthesis. Google DeepMind put forward a parallel computing model named WaveNet (Oord et al., 2017), Baidu presented a real-time speech synthesis product Deep Voice 3 (Ping et al., 2017).

Computer Vision

Deep learning is broadly applied on computer vision tasks, including traffic-sign detection and classification (Zhu et al., 2016), face recognition (Parkhi et al., 2015), face detection (Yang et al., 2017), image classification (He et al., 2016), image fusion (LIN and HAN, 2017), object detection (He et al., 2017a), semantic segmentation (Long et al., 2015), realtime multi-person 2D pose estimation (Cao et al., 2017), pedestrian detection (Tian et al., 2015), scene recognition (Zhou et al., 2014), visual tracking (Nam et al., 2016), end-to-end video classification (Fernando and Gould, 2016), and video action recognition (Fernando and Gould, 2016). There are also diverse works in arts, including image colourisation (Cheng et al., 2015), turning doodles to fine artworks (Champandard, 2016), image style transfer (Gatys et al., 2016) and pixel recursive super resolution (Dahl et al., 2017). Oxford University and Google DeepMind put forward LipNet (Assael et al., 2016) to implement lip reading, with an accuracy of 93 %, which is higher than the human lip-reading accuracy of 52 %.

Natural Language Processing

NEC Labs America (Collobert et al., 2011) gave the earliest attempt of implementing deep learning on the Natural Language Processing (NLP) field. Nowadays, diverse tasks are based on this technique, such as word embedding models Word2Vec (Mikolov et al., 2013), Part-of-Speech (POS) tagging (Toutanova et al., 2003), transition-based parsing (Weiss et al., 2015), named entity recognition (Lample et al., 2016), role labelling (He et al., 2017b), character-aware neural language models (Kim et al., 2016), sentiment analysis (Severyn and Moschitti, 2015), document classification (Joulin et al., 2016), machine translation (Gehring et al., 2017), read and comprehend (Hermann et al., 2015), question-answering services (Weston et al., 2015; Kumar et al., 2016; Sorokin et al., 2015), and human-computer conversation system (Zhou et al., 2016).

Others

In the field of bioinformatics, deep learning is used for quantitative structure–activity relationships estimation (Ma et al., 2015), predicting eye fixations (Liu et al., 2015), predicting splicing patterns based on genomic features and cellular context (Leung

et al., 2014). The economic and finance field has continuously applied deep learning in analyse important data, examples include financial markets prediction (Dixon et al., 2017), portfolio analysis (Heaton et al., 2017), and insurance churn prediction (Zhang et al., 2017).

2.2.4 Multimodal Deep Learning

Overview of multimodal deep learning

Multi-source information fusion is put forward in the 1970s for military practices (Srivastava and Salakhutdinov, 2012). Cognitive processes behind human or animal observations of objects are multi-source information fusions. Human perceives an object or a concept from various aspects, through senses like sight, hearing, taste, smell, and touch; and once a large amount of information, either complementary or redundant, is collected, the brain will combine and process the data of senses according to certain rules and generate a consistent perception of that object or concept. Apart from general human senses, physical devices or sources can also have multiple sensors, which are also termed modalities (Ramachandram and Taylor, 2017), Examples of a modality include audio, video, text, radar, infrared, and accelerometer. Each sensor out-put is a modality linked with a single dataset. In a broader definition, a language can be one modality and datasets collected from various conditions are different modalities. The process from multi-source sensors to perceptions interest researchers in the field of Multimodal Machine Learning (MMML) (Baltrušaitis et al., 2018) from the 1970s. The overall aim is to leverage machines to obtain the ability of processing and understanding multi-source data. From 2010, deep learning (Srivastava and Salakhutdinov, 2012; Ramachandram and Taylor, 2017; Atrey et al., 2010) accelerates multimodal learning in the following research areas.

(1) Multimodal representation. This field of work uses complementarity, remove redundancy of multimodality, and learn more precise representations, including joint representations and coordinate representations. For example, Srivastava and Salakhutdinov (2012) used multimodal deep boltzmann machine to learn the joint probability distribution of texts and images. Kiros et al. (2014) found feature properties learned from models which enable semantic calculations.

(2) Translation or mapping. Multimodality is applied in translate information from one modality to another. For example, language translation (LeCun et al., 2015), lip reading (Chung et al., 2017), speech-to-text translation (Chorowski et al., 2015), image/video captioning (Yu et al., 2016), and speech synthesis (Tokuda et al., 2000).

(3) Alignment. Multiple modalities are collected from the same entities and aligned according to sets of rules. Examples of such practices include video-audio-subtitle alignment and image semantic segmentation (vision and text label alignment) (Long et al., 2015).

(4) Multimodal fusion. Multiple modalities are combined for classification or regression tasks. According to fusion hierarchy, fusion can be divided into pixel level, feature level (early feature level and late feature level), and decision level. Conventional machine learning methods are suitable for multimodal fusion, for example, visual-audio recognition, multimodal sentiment analysis (using speech, facial expression, etc.) (Poria et al., 2016) and mobile identity authentication (Neverova et al., 2016).

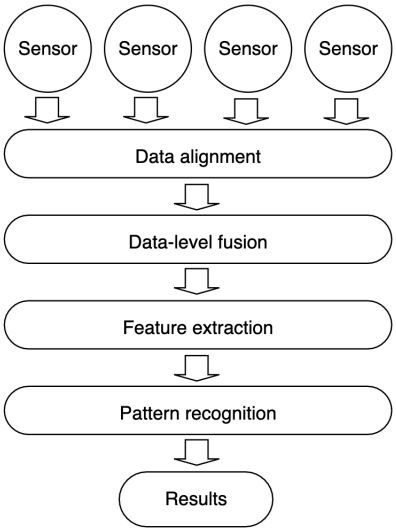

FIGURE 2.4: Data-level fusion procedure

(5) Co-learning. Modalities with richer data are used to assistant the learning processes of other modalities. Examples include transfer learning (Torrey and Shavlik, 2010), and zero-shot learning, one-shot learning.

Structures and models

According to the abstract hierarchy, information fusion in multimodal deep learning is classified to three types: data-level fusion, feature-level fusion, and decision-level fusion (Ramachandram and Taylor, 2017). The structures of the three types of fusion process are shown in follows.

(1) Data-level fusion

Data-level fusion is a low-level fusion (see Figure 2.4), in which information obtained from sensors are directly combined and processed. Based on the fusion results, features can then be extracted for decision predicting or supporting. Compared to the fusions at other levels, data-level fusion has minimal information loss and provides more subtle details. Theoretically, data-level fusion results reach the highest precision. However, it also has many limitations. Large amount of data is required from sensors, adding to the cost of processing, computing and time. In addition, it is inappropriate for online or real-time applications. As the fusion process occurs in low-level, the uncertainty, incompleteness and instability of sensors require further corrections while combining. Usually data from different types of sensors are hard to be composed. Since the communication traffic of raw data is large, noises and interferences are inevitable. In practice, the data-level fusion is applied mainly in multi-source image composing, analysing and understanding as well as same-type radar wave composing.

(2) Feature-level fusion

Feature-level fusion is the fusion at the middle-level (see Figure 2.5), including early feature-level fusion and late feature-level fusion. Features (e.g., edges, direction, speed of a car) of data from each sensor are extracted separately and further composed in the fusion process. Generally speaking, features extracted from data are sufficient representations or sufficient statistics. By processing fusion at the feature level, dimensionality of the data is reduced, communication traffic become

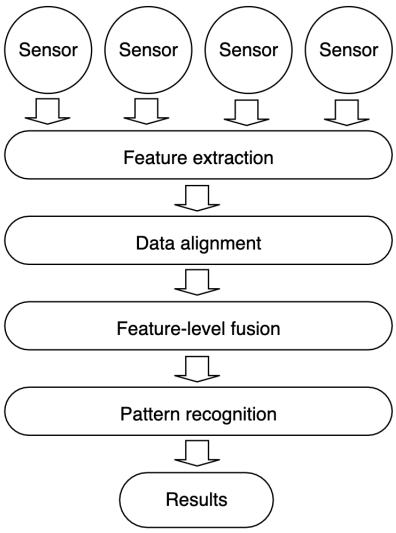

FIGURE 2.5: Feature-level fusion procedure

smaller and real-time applications are feasible. On the other side, the fusion performance decrease due to the fact that some of the information is lost during the raw data processing period. There are mainly two types of feature-level fusion: target state information fusion and target multi-feature fusion. The target state information is usually adopted in multi-sensor target tracing, where data from sensors are processed and adjusted, then used for analysis and states estimation, through methods such as Kalman filter (Welch et al., 1995), joint probability distribution association(Musicki and Evans, 2002), multi-target tracing (Pellegrini et al., 2009; Benfold and Reid, 2011), multiple hypothesis tracking (Blackman, 2004), interacting multiple model (Blom and Bar-Shalom, 1988) and sequential estimation. The target multi-feature fusion is a particular type of pattern recognition task. The methods of target multi-feature fusion includes principal components analysis (Price et al., 2006), singular value decomposition (De Lathauwer et al., 2000), and K nearest neighbour (Cover et al., 1967; Liu et al., 2013).

(3) Decision-level fusion

Decision-level fusion is a high-level fusion (see Figure 2.6). Decisions are made independently according to the sensors which received corresponding data. Therefore, local decisions are gathered and processed for final decision. Decision-level fusion directly produces the final result, aiming at the task targets. Compared with other hierarchical fusions, this level loses the most amount of data due to frictional alignment, lowering the result precision. However, given its smaller communication traffic, the decision-level fusion method is more robust, less sensor-dependent, and more cost-friendly for processing in fusion procedure. The methods of decision-level fusion include Bayes inference (Morris, 1983), expert system (Hayes-Roth et al., 1983; Waterman, 1986), D-S evidence theory (Zhuge et al., 2006), and fuzzy set algorithm (Zimmermann, 2011; Ragin, 2000).

Decision-level and feature-level fusion does not require identical sensors. As fusions on different levels have their own advantages and disadvantages, highly efficient local sensors and fusion optimisation rules have been developed to improve precision and save processing time.

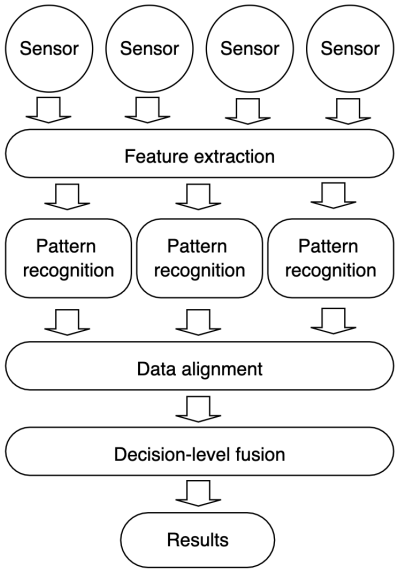

FIGURE 2.6: Decision-level fusion procedure

Applications in diverse areas

In the beginning, applications of multimodal fusion and learning are put forward in military areas (Ramachandram and Taylor, 2017)such as military objects (e.g., naval vessels, aeroplanes and missiles) detection, localising, tracing and recognition. In particular, it includes satellite ocean surveillance system and defence system. After years of research, multimodal learning develops fast in civil applications.

(1) Image fusion. Two or more images are collected to generate a new fusion image by using certain algorithms. By combining diverse spatial, temporal and spectrum resolution, redundancy and contradiction of modalities from different sensors are removed or decreased, improving information precision and reliability. For example, in medical diagnose, by using image fusion techniques on ultrasonic imaging, MRI and X-Ray (Lee et al., 2017)doctors are able to give diagnose results faster and more accuratly, which is hard to accomplish using single modality.

(2) Industrial intelligent robots. By combining and inferring from data of videos, images, texts, etc., robots are able to automatically implement tasks such as moving materials, manufacturing, diagnosing and installation. Under a complex circumstance, robots are required to complement tasks of grab, moving or touch, using on sensors of sights, taste, proximation and momenta, and thus enable fast, reliable and precise actions of robots. For example, spatial ROTEX from Germany (Hirzinger et al., 1993) and engineering test satellite ETS-VII from Japan (Yoon et al., 2004). By using data fusion of 2D colour images from CCD camera calibration and 3D distance images from high-speech 3D imaging radar, robots' cognition and understand capacities to environment are improved.

(3) Remote sensing. An image of high dimensional resolution and high spectrum resolution is obtained by getting fusion of high dimensional resolution colour image and low spectrum resolution image, while introducing multiple air-band and time interval can further improve accuracy of remote sensing. For example, SAR images and visual image sequence fusion (Zhu et al., 2017) enables mineral prospecting and resource detection.

(4) Criminal investigation. Multimodal learning applications in criminal investigation (Bates, 2003) mainly focus on hidden weapon detection, drug detection using devices including infrared, and microwave sensors. Human or device identification is leveraged on biological characteristics such as face, fingerprint, voice, movements, and iris.

(5) Abnormality detection. For industrial monitoring, multiple sensors response to different failure features and detect abnormality by fusion of these features. Intelligent traffic system (ITS) utilises multiple sensors data fusion to applications such as autonomous road recognition, speed control, and localisation.

Advantages of multimodal deep learning

Compared to single modality learning, multimodal learning system has the following advantages (Srivastava and Salakhutdinov, 2012; Ramachandram and Taylor, 2017; Baltrušaitis et al., 2018). (1) Robust system: multiple sensors allow some space for error. For example, when a sensor is defect or interfered, or when objects are out of scope, other sensors can still receive information; (2) Spatial/temporal span extension: coordinated sensors receive more extensive and specific information than single sensor; (3) Decreased reliability improvement and ambiguity: diverse sensors verify information reliability amongst themselves; (4) Increased spatial resolution increase: multimodal has higher resolution than single modality; (5) Lower cost and flexible combination: the choice of sensors is more flexible and efficient since various types of sensors with diverse information can be used.

Methods in multimodal learning

Multi-source information fusion, as a comprehensive data processing procedure, is essentially complex. Conventional estimation theories and recognition algorithms build a theorem and framework foundation of multimodal fusion. Moreover, in recent years, many new methods are proposed and developed based on statistical inference, artificial intelligence, and information theory. The methods boosting development of the multimodal learning areas are briefly introduced as follows.

(1) Signal processing and estimation theories. It includes Multiresolution-based image fusion (Núñez et al., 1999; Petrovic and Xydeas, 2004), linear estimation such as weighted average, ordinary least square, and nonlinear estimation such as Kalman filter estimation, extensive Kalman filter (EKF), gaussian sum approximation (Alspach and Sorenson, 1972). Recently, more complex and valuable methods are developed such as the unscented article filter (Van Der Merwe et al., 2001), Markov Monte Carlo (MCMC) (Neal, 1993; Doucet et al., 2000), and particle filter based on MCMC sampling (Vo et al., 2005; Djuric and Chun, 2002).

Expectation maximization (EM) algorithm (Molnar and Modestino, 1998; Logothetis and Krishnamurthy, 1999), enable parameter estimation and information fusion based on incomplete observed data. Moreover, by building certain indices, optimization methods can be used for best parameter estimations, for example, empirical risk minimum (Richardson and Marsh, 1988; Clark and Yuille, 2013).

(2) Statistical inference. It includes Bayes inference, Dempster-Shafer reasoning (Murphy, 1998; Bloch, 1996), random set theories (Goutsias et al., 2012; Mori, 1997), support vector machine (SVM).

(3) Information theory. By optimizing information measurement, information theory methods enable data fusion, for example, entropy methods (Manyika and

Durrant-Whyte, 1995), minimum description length (MDL) (Barron et al., 1998; Joshi and Sanderson, 1999).

(4) Decision theory. It is often used for high-level fusion such as decision fusion, Nelson and Fitzgerald (1996) applies decision fusion of visible light, infrared and millimeter wave radar on alarm detection and analysis.

(5) Artificial intelligence. In includes fuzzy logic, neural network, genetic algorithm, inference based on rules, expert system, templating (Hall and Llinas, 1997) and figure of merits (FOM).

(6) Geometric methods. By analyzing environment and geometric characteristics of sensor model, multi-source data are processed to fusion procedure. (Abidi and Gonzalez, 1992; Kim and Vachtsevanos, 1998).

2.2.5 Framework

The main deep learning develop framework includes: (1) Theano (Bastien et al., 2012), which is based on Python, developed from 2007, and benefit efficient data exploration. It is however not distribution computing supported; (2) Tensorflow (Abadi et al., 2016) is an open source system particular for large-scale machine learning from 2015, which is flexible, generalized, visualization supported and appropriate for product development; (3) Keras (Chollet et al., 2015) is a high-level neural network API, written in Python, capable of running on top of Theano, Tensorflow or CNTK, supports CPU and GPU, and provides several modules, functions and pre-trained neural network weights. Keras allows easy and fast prototype and enable flexible model architecture building, which includes two types of manners: sequential model enables adding layer one by one, while functional API enables more complex structure such as multiple output branches, sharing layer weights, loading pre-trained networks weights or embedding weights; (4) Caffe (Jia et al., 2014) is developed from 2013 and particular for image processing, it is however not suitable for other deep learning tasks such as speech and text processing; Torch (Bojarski et al., 2016) is developed based on Lua, which is unfamiliar for most developers. On the other hand, it is flexible, easy to build customized hierarchy and provides pre-trained models.

Chapter 3

Data and Models

The feature-level fusion experiment procedure is shown in Figure 3.1, while the decision-level fusion experiment is shown in Figure 3.2. The data is from the SCOTUS Oral Arguments Corpus (Johnson and Goldman, 2009) and the U.S. Supreme Court Database (Harold J. Spaeth, 2018). In the experiment setup, the structured, text and audio data are collected, processed using corresponding methods, and aligned along the same index order. Then three subsidiary models are designed and built according to the characteristics of the data types. The subsidiary models are then combined to a multimodal, and it is fed with the prepared mixed data, fitted and evaluated on case and justice, independently. There are four groups of experiments, according to the data granularity (justice level or case level) and fusion period (feature-level fusion or decision-level fusion).

3.1 Data

3.1.1 Data Source

U.S. Supreme Court Database (SCDB)

The U.S. Supreme Court Database (Harold J. Spaeth, 2018), founded by the U.S. National Science Foundation, was made public in the 1980s. For years, Professor Spaeth and his colleagues were dedicated to collect and code data, which amassed a rich-content dataset with 247 pieces of information for each case. It contains roughly six categories: (1) identification variables (e.g., citations and docket numbers), (2) background variables (e.g., origin and source of the case), (3) chronological variables (e.g., the date of decision, term of Court), (4) substantive variables (e.g., legal provisions, issues, direction of decision), (5) outcome variables (e.g., disposition of the case, winning party), and (6) voting and opinion variables (e.g., how the individual justices voted, their opinions and inter-agreements).

Since the database is consistently validated by the high quality and rich variety of the data, it has been used by hundreds of systematic analyses for Supreme Court. It has been used in social science (Epstein et al., 2007) and legal academics (Segal and Spaeth, 2002; Bailey and Maltzman, 2008; Katz et al., 2017), for both quantitative (Martin and Quinn, 2002; Lee et al., 2015) and qualitative studies (Segal and Spaeth, 1996).

There are two releases of database: the Legacy and the Modern. The Legacy database includes cases decided by the Court between the 1791 and 1945 terms, while the Modern database contains cases from 1946 term onwards. With the intention of predicting court decision, all structured, text and audio data were considered to put into the models. However, the oral arguments data (text and audio) prior 1977 was not available. To take advantage of all three types of data, the Modern SCDB

[1] of structured data was adopted. It provides 13,453 entries of case level data and 120,504 entries of justice level data.

SCOTUS Oral Arguments Corpus

The Supreme Court of the United States (SCOTUS) [2] Oral Arguments Corpus (Johnson and Goldman, 2009) includes 38 years (1977 to 2014) of audio records linked to text transcripts of the oral arguments at the Supreme Court of the United States. The corpus is provided by Professor Goldman, founder and director of the Oyez Project [3]. For each case, they carefully coded the background information (e.g., date, location, docket number, case name), speakers (Justices, attorney and amicus curiae) of segments of oral arguments, and other necessary comments (e.g., attorney is on behalf on petitioner or respondent, timestamps of audio utterances).

As a complete and authoritative source for oral arguments recorded in the Court since the installation of a recording system, the SCOTUS corpus has been used by millions of users and studies worldwide, including "speaker identification" (Yuan and Liberman, 2008), "variation in vowel-to-vowel phonology" (Yu et al., 2015) and many other languages and speech research (Vipperla et al., 2008; Yuan and Liberman, 2011, 2010). It has also supported vast amount of and wide range of studies of the Supreme Court. Black et al. (2011) used Justices' questions and comments (in text) made during oral arguments to predict their final votes. They founnd that the more unpleasant words the Justices use towards an attorney, the less likely the side s/he represents would prevail, either in terms of the overall case outcome or justice individual votes. Dietrich et al. (2019) measured the vocal pitch of each Justice' voice during the oral arguments to construct emotional arousal and predicted many of their votes on these cases.

3.1.2 Data Processing

Various methods and different procedures were used to process different types of data, allowing sufficient information to be extracted.

Target

All decision outcomes related variable are from the SCDB. The decision the Court made is contained in the variable disposition, which has 11 possible values — affirmed, vacated, reversed, and remanded, etc. For the sake of simplicity, the target indicator "partyWinning" was introduced in replace of the complicated dispositions. At the case level, the target indicator partyWinning was coded 1 if the petitioner party received a favourable disposition in that petitioner was regarded as winning party. It includes situations when the Supreme Court (1) reversed, (2) reversed and remanded, (3) vacated and remanded, (4) affirmed and reversed in part, (5) affirmed and reverse in part and remanded, or (6) vacated. The indicator partyWinning was coded 0 if the petitioner did not receive favourable disposition in that respondent was regarded as winning party. In includes situation when the Supreme Court (1) affirmed or (2) dismissed the case or denied the petition. Similarly, at justice level,

[1] Modern Database, 2018 Release 02, Cases Organized by Docket were used. Both case-level and justice-level files are available at http://supremecourtdatabase.org

[2] SCOTUS Oral Arguments Corpus: https://ca.talkbank.org/access/SCOTUS/OralArguments.html

[3] Oyez (www.oyez.org): a free law project from Cornell's Legal Information Institute (LII), Justia, and Chicago-Kent College of Law, is a multimedia archive devoted to making the Supreme Court of the United States accessible to everyone.

the target indicator was coded 1 if the justice votes to reserve a lower court decision, and the target indicator was coded 0 if the justice votes to affirm. Some case outcome or justice vote is unclear when the Court certified to or from a lower court. Only a rare occurrence of 21 out of 13,453 cases (0.15 %) had unclear outcomes, which were simply removed from the data.

Structured data

After remove duplicated entries, entries with missing data or unclear information, the structured data contains 92,554 justice votes cast in 10,336 cases, termed from 1946 to 2018.

To take advantages of cases background data, 23 categorical variables were taken from the Modern SCDB and converted into 1-0 dummy variables, which enabled the following numeric computation. For example, variable "issueArea" contained 14 values (e.g., criminal procedure, civil rights, economic activity, etc.), which were converted to 13 variables with 1-0 values indicating the specific issue areas. In total, 1488 variables were obtained after the one-hot encoding, resulting a sparse matrix.

Since a sparse matrix is costly in computation, moreover, a revised version of Singular Value Decomposition (SVD), named Truncated SVD (Halko et al., 2009), was used to reduce matrix dimensionality. The converted matrix fixed the sparsity issue, extracted and compressed the information without losing the sufficient components. As Halko et al. (2009) pointed out, in particular, the method "works on term count/tf-idf matrices, and is known as latent semantic analysis (LSA)". In this study, 600 components were chosen as a desired dimensionality of output data, and the new matrix with 600 components represented 96.90 % of information the original sparse matrix containing 1488 variables.

Text data

Using the speakers' annotation of oral arguments transcripts from SCOTUS, cases were parsed into discrete segments uttered by (1) the Justices themselves, (2) the attorney and amicus curiae on behalf of petitioner, and (3) the attorney and amicus curiae on behalf of respondent. Further, the utterances [4] of the Justices were split into two parts — words towards petitioner side; and words towards respondent side.

For each utterance of a justice in a certain case, the whole transcript texts were tokenised to word level, with punctuations (e.g., !"() * +, −./ :;<=>?@[]'|) and non- alphabetic characters (e.g., numbers) removed. All uppercases were changed to lowercases, and stop words (e.g., the, and, or, that, etc.) were filtered out since these words occurred too frequent yet seldom conveyed concrete meaning. The entries with a certain Justice saying nothing in a case were removed.

After careful cleaning and processing, the final aggregate text data spanning the 1977 - 2014 terms, which contains 10,742 justice votes cast in 3,006 cases, 10,742 justice utterances towards petitioner side; and 10,742 justice utterances towards respondent side.

In an attempt to extract and compute the information from text, vector representations for text were essential and were obtained by a word embedding method — Global Vectors for Word Representation (GloVe) (Pennington et al., 2014) [5]. GloVe

[4] Here, an utterance is defined as the words a Justice spoke to one side of petitioner or respondent, in a given case.

[5] The GloVe-1.2 and glove.6B.100d pre-trained vectors was used, the materials are available at https://nlp.stanford.edu/projects/glove/

is a count-based unsupervised learning algorithm for obtaining vector representations of words, which can capture the relation of words in a converted vector space. Since various high-quality pre-trained word vectors obtained from respective corpora (e.g., massive web datasets, Wikipedia, Twitter) are available, a robust version of vectors trained from 42 billion tokens (1.9 million words) was applied in this study. By setting the Maximum Number of Words of corpus as 20,000, the Maximum Length of any utterance as 400, the Embedding Dimensionality as 100, each utterance was turned into a sequence of integers where each integer was linked to a word. An utterance was padded to Maximum Length with 0 if it has less than 400 words. By carrying out the word embedding using the pre-trained word representations, information from oral arguments transcripts was captured by extracting patterns from embedding matrix.

Audio data

To extract the emotions hidden in the voice, audio recordings of oral arguments from SCOTUS were collected, which amounted to 5,158 utterance for each party, cast in 1,000 cases spanning from 1982 to 2014. Using the timestamps provided by Oyez Project, audio recordings were also parsed into discrete segments uttered by the Justices and attorneys. In total, the Justices spoke for 502 hours. In addition, the Justices' audio recordings were further parsed to two parts depends on the side (petitioner or respondent) they spoke to.

As proposed by Dietrich et al. (2019), a measure of voiced speech as *Vocal Pitch* was obtained from fundamental frequency using Praat, "a free computer software package for speech analysis in phonetics [6]". For each Justice, his or her vocal pitch was standardised (by standard deviation) for avoiding the systematic differences between Justices (for example between male and female Justices) or any measurement error associated with frequency.

Mixed data alignment

Along the "docket" numbers (cases identification) and the "justiceID", data of structured information, text, and audio were collected and aligned to the same order, and entries were removed if they were in lack of any one of three data types. Finally, 5,158 entries including cases information and Justices behaviours from 997 cases were adopted. For the 5158 cases in justice level, Justices vote to petitioner side counted 2,812, while to respondent side counted 2346, showing a balanced target outcomes.

The aligned mixed datasets are directly used in feature-level fusion experiment after data pre-processing, for the court decision prediction; while they are also used at the second step of decision-level fusion experiment, to fine tune the concatenated final model.

3.2 Models Construction

According to the fusion period and data granularity, there are four groups of experiments are implemented independently: feature-level fusion on justice level, feature-level fusion on case level, decision-level fusion on justice level, and decision-level fusion on case level.

[6] Praat is available at http://www.fon.hum.uva.nl/praat/

Since the target of results is a binary variable, the court decision prediction problem can be regarded as a binary classification task, where the inputs are pre-processed structure data, text data and audio data, and the output is 1 (petitioner party wins) or 0 (respondent party wins).

In the experiment, multiple models are constructed according to the corresponding data processed by them. The models are further composed for fusion procedure. The model components and fusion procedure used in the four experiment are introduced as follows.

3.2.1 Model Components

Multilayer perceptron

The feedforward process of a multilayer perceptron (MLP) is as follow.

$$z_i^{l+1} = \sum_j W_{ji}^l y_j^l + b_i^l$$

$$y_i^{l+1} = f(z_i^{l+1})$$

y_j^l is the output of neuron j of layer l, z_i^{l+1} is value before activation by neuron i of layer $l+1$, W_{ji}^l is the weight between neuron j of layer l and neuron i of layer $l+1$. b_i^l is bias. $f(\cdot)$ is the nonlinear activation function, including radial basis function, ReLU, PReLU, Tanh, Sigmoid, etc.

When using Mean Square Error loss function is as follows.

$$J = \frac{1}{2}\sum_i (y_i^L - y_i)^2$$

. y_i^L is the output of neuron i of the last layer, y_i is the true value of the target. The goal of neural network training is to minimise the value of loss function, where Stochastic Gradient Descent (SGD) is commonly used as an optimisation method.

For each experiment with structured data as the input, a MLP with three layers, each fully connected with a previous layer (except the input layer), was used for feature extracting. The numbers of neurons of the three layers are 64, 32 and 16. Each activation function between layers is Rectified Linear Unit (ReLu).

For each experiment with audio data (vocal pitch) as the input, a MLP with one hidden layer of 2 neuron is used to extract the patterns.

Models for natural language processing

As suggested by (Chollet, 2017), a strategy to combine the speed and lightness of convolutional neural networks, with the order-sensitivity of RNNs, is to use a one-dimensional convolutional neural networks (1D-CNN) as a preprocessing step before an recurrent neural network (RNN). This strategy works especially well when the sequence is so long that it cannot realistically be processed with RNNs, for example, a document with thousands of words. By taking advantages of special structure of 1D-CNN, long sequences will be turned into much shorter representations with high-level features (downsampled), which is a form of information compression procedure. The shorter sequences of extracted features will then become the input to the following RNNs part of the model. The structure reduces expensive cost of RNNs

training on very long sequences by adding a 1D-CNN layer, which maintains the performance of models as well.

The one-dimensional convolution is an operation between a vector of weights $m \in \mathbb{R}^m$ and a vector of inputs viewed as a sequence $s \in \mathbb{R}^s$. The vector m is a filter of the convolution (Kalchbrenner et al., 2014). More precise, s is the input sentence and $s_i \in \mathbb{R}$ is a single feature value associated with the i-th word in the sentence. The idea of the one-dimensional convolutional neural network (1D-CNN)

is to take the dot product of the vector m with each m-gram (a bag of m words) in the sentence s to obtain another sequence c.

$$c_j = m^T s_{j-m+1:j}$$

The narrow type of 1D-CNN requires that $s \geq m$ and yields a sequence $c \in \mathbb{R}^{s-m+1}$ with j ranging from m to s. The wide type of 1D-CNN do not require the range of s or m and yields a sequence $c \in \mathbb{R}^{s-m+1}$ where the index j ranges from 1 to $s+m-1$. Out-of-range input values s_i where $i < 1$ or $i > s$ are taken to be zero.

The feed-forward functions of a recurrent neural network (RNN) are as follow.

$$z_h^t = \sum_{i=1}^{l} w_{ih} x_i^t + \sum_{h'=1}^{H} w_{h'h} a_{h'}^{t-1}$$

$$a_h^t = f_h(z_h^t)$$

$$y_k^t = \sum_{h=1}^{H} w_{hk} a_h^t$$

x_i^t is neuron i of input layer at time t, $a_{h'}^{t-1}$ is neuron h' of hidden layer at time $t-1$, z_h^t is neuron h of hidden layer before activated at time t, y_k^t is neuron k of output layer at time t, w_{ih}, $w_{h'h}$ and w_{hk} are weights between the input layer and the hidden layers, between the hidden layers, and between the hidden layer and the output layer, respectively. $f_h(\cdot)$ are nonlinear activation functions.

Gated recurrent unit (GRU) is a revised version of RNN, which is designed to adaptively capture dependencies of different time scales. It has gating units that modulate the flow of information inside the unit. The activation h_t^j of the GRU at time t is a linear interpolation between the previous activation h_{t-1}^j and the candidate activation $\bar{h}_t^j$,

$$h_t^j = (1 - p_t^j) h_{t-1}^j + p_t^j \bar{h}_t^j$$

, where an update gate p_t^j decides how much the unit updates its activation, or content. The update gate is computed by

$$p_t^j = \sigma(W_p x_t + U_p h_{t-1})^j$$

.

The candidate activation $\bar{h}_t^j$ is computed similarly to that of the traditional recurrent unit,

$$\bar{h}_t^j = tahn(W x_t + U(r_t \bigodot h_{t-1}))^j$$

, where r_t is a set of reset gates and $\bigodot$ is an element-wise multiplication. When off (r_t^j close to 0), the reset gate effectively makes the unit act as if it is reading the first symbol of an input sequence, allowing it to forget the previously computed state.

The reset gate r_t^j is computed similarly to the update gate,

$$r_t^j = \sigma(W_r x_t + U_r h_{t-1})^j$$

.

In this paper, a 1D-CNN layer with 32 neurons (kernel sized 3, padding as the same, activation function as ReLu), followed by a 5×5 max pooling layer and a GRU with 16 neurons (dropout rate as 0.1, recurrent dropout rate as 0.5).

3.2.2 Multimodal Deep Learning

Feature-level fusion

The procedure of feature-level fusion experiment is shown in Figure 3.1. In turn, the structured data, text data and audio data were used in four processes: feature extraction, data alignment, feature-level fusion and pattern recognition.

Due to the distinction of structure and characteristics of three types of data, they were prepared by corresponding techniques independently. In the feature extraction process, the structured data was processed by one-hot encoding and SVD dimensionality reduction; the text data was tokenised and transformed by GloVe embedding into an embedding matrix; the audio data was extracted and transformed as vocal pitch.

The processed and prepared data are then collected, aligned and sorted according to the "docket" number and "justiceID". On the justice level, in total, 5,158 entries were aligned from 91,838 entries of structured data, 10,742 entries of text data and 5,158 entries of audio data. On the case level, in total, 1,000 entries were aligned from 10,336 entries of structured data, 3,006 entries of text data and 1,000 entries of audio data.

Each type of data obtained from the alignment step was fed into corresponding model components, and then the final concatenated model, for feature-level fusion processing. The prepared structured data was put into a MLP model, the prepared embedding matrix from text data was put into a 1-D CNN followed by a GRU model, the prepared vocal pitch was put into a distinct MLP model. For either text or audio data obtained from oral arguments, they are separated to petitioner and respondent group. Each group of data was put into a subsidiary model which is later concatenated with another. Thereafter, three model components were concatenated and followed by two more layers, the first with 4 neurons and ReLu activation and the second with 1 neuron with sigmoid activation, both are fully connected with the previous layer.

The overall data was split to training, validation and test sets (0.64: 0.16: 0.2). After building and compiling the whole architecture, feed the model with training datasets and targets, iteratively evaluate and tune models with validation datasets. Each experiment was implemented 10 times, and the average accuracy as the evaluation metrics were recorded. Besides, subsidiary models with corresponding parts of data were also trained, independently, and the accuracies are also recorded. Results are shown in the next chapter.

Decision-level fusion

The procedure of decision-level fusion is shown in Figure 3.2. The process is similar with the feature-level fusion experiment mentioned above, but is distinctive in steps including different fusion period, data alignment period and fine tuning process.

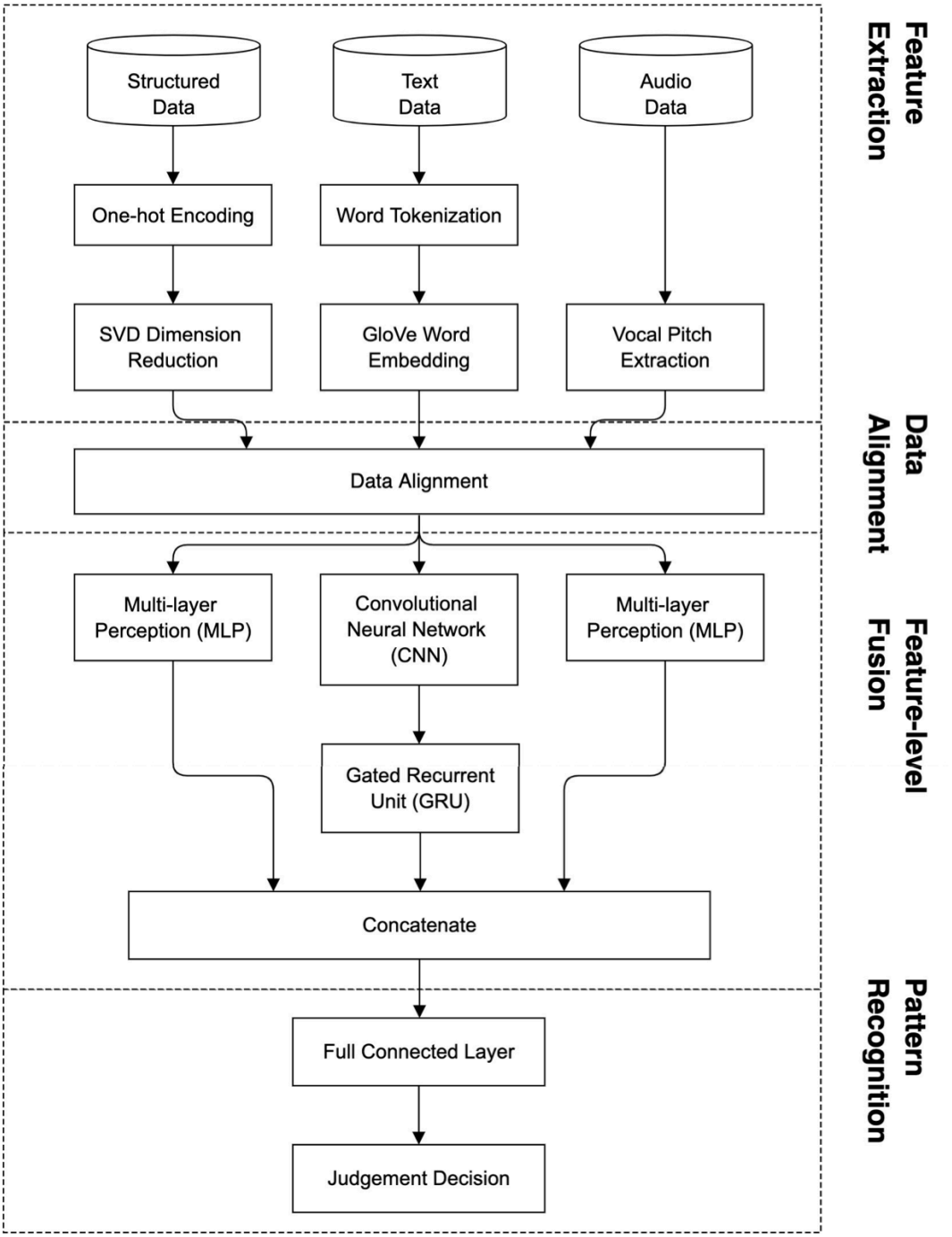

FIGURE 3.1: The **feature-level fusion** experiment procedure of judgement decision prediction

Each type of data after pre-processing is put into corresponding subsidiary model, independently. The three subsidiary models are fed with targets and complete data independently, trained and tuned in parallel and separately, then concatenated and followed by two more layers.

Thereafter, in the decision-level fusion procedure, the weights of the three subsidiary models are frozen. Only the two layers followed by them are fed with aligned data and targets for training and fine tuning.

The results details are in the next chapter.

3.2.3 Evaluation

Overall accuracy, as an evaluation metrics, indicates the general performance of a model. The formula of overall accuracy is

$$p = \sum_{i=1}^{N} m_i / N$$

, where the N is the number of experiment times, m_i is the accuracy of model in experiment i. In this paper, each model under a certain condition is implemented in a 10-times experiment, and the three overall accuracies of training set, validation set and test set are exported, separately.

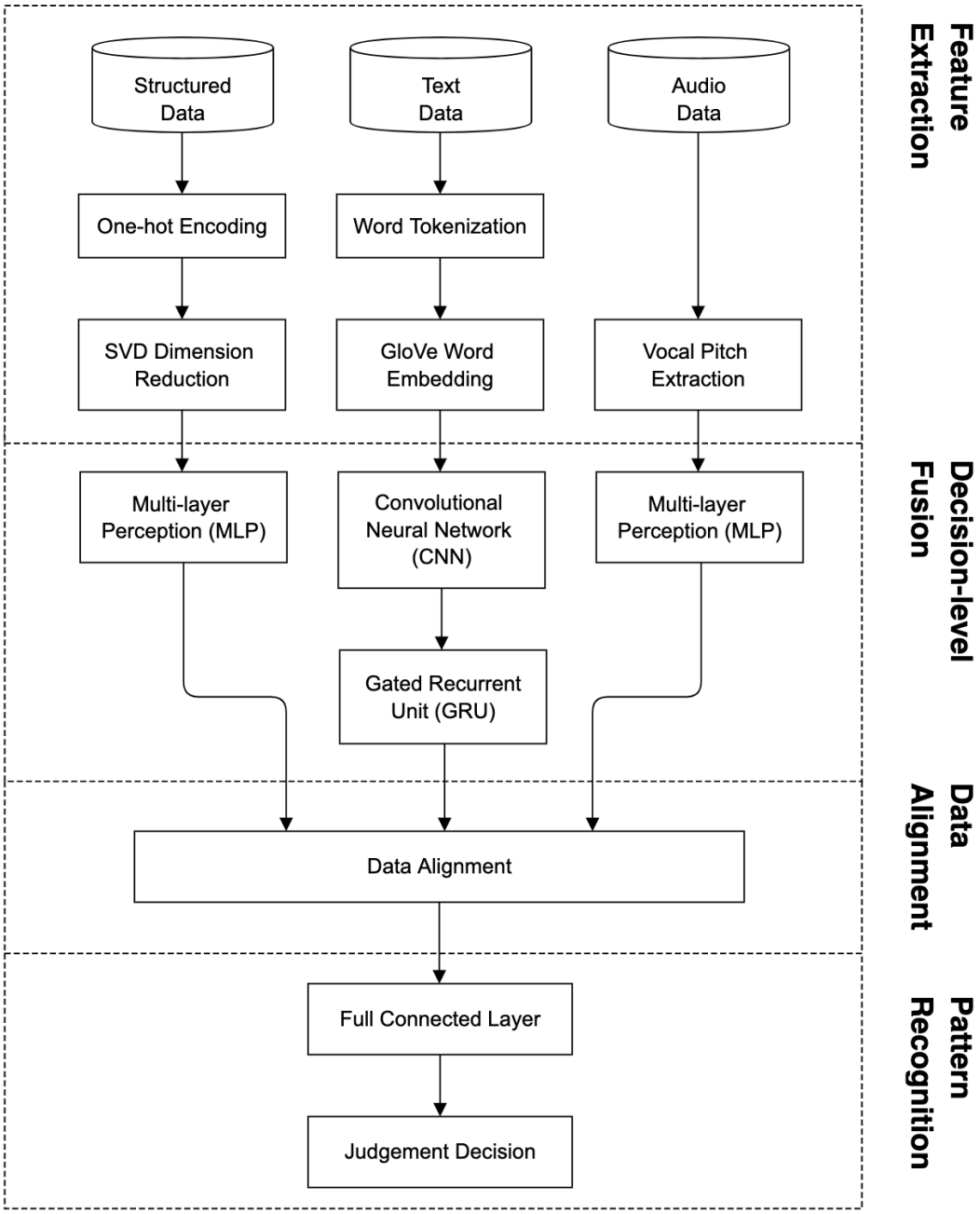

FIGURE 3.2: The **decision-level fusion** experiment procedure of judgement decision prediction

Chapter 4

Results and Discussion

In order to predict justices' decisions of cases on the Supreme Court of the United States, multiple type of data, including structured background information, oral arguments transcripts and audio, were used to train several models, leveraging on features fusion and decision fusion architectures.

Before model training experiments, categorical variables of structured data were one-hot encoded and the dimensionality of the encoded matrix was reduced to 600 using singular value decomposition. Oral arguments transcripts were tokenised to word level, removed from punctuations and non-alphabetic characters; uppercases were transferred to to lowercases; stop words were filtered out; and all words were embedded through pre-trained GloVe embedding vectors word representation. Oral arguments audio were parsed to vocal pitch and standardised.

According to the procedure of data fusions in workflow, two experiments were implemented: feature-level fusion and decision-level fusion. Each experiment was carried out on justice level and case level, separately. Moreover, an extra decision prediction on case level was exported from prediction results on justice level, obtained by model trained on the justice level.

As suggested by Chollet et al. (2015), the following setting were adopted to facilitate the model training. Experiments hardwares: AMD Ryzen Threadripper 1920X (CPU), Gainward GeForce RTX 2070 (GPU), Corsair DIMM 32 GB DDR4-3200 Quad-Kit (RAM), Samsung 970 EVO Plus 500 GB (SSD). Environments: Ubuntu 18.04.2 LTS, Anaconda 5.2, Python 3.6 (IDE: PyCharm), Tensorflow-GPU 1.12, CUDA Toolkit 9.0, cuDNN 7.1.2, Keras 2.2.4 with Tensorflow backend. More details of environment requirements and codes can be found in the appendix.

4.1 Data Exploration

Cases background data and oral arguments on the Supreme Court of the United States were collected through SCOTUS Oral Arguments Corpus (Johnson and Goldman, 2009) and the U.S. Supreme Court Database (Harold J. Spaeth, 2018).

The justice centered background data includes 92,554 justices decisions. Among them, 58,483 votes of justices supported petitioners (petitioning party received a favourable disposition) while 33,882 votes supported respondents (no favourable disposition for petitioning party apparent). 36 votes with no unclear disposition is removed. The accuracy of the baseline model on justice level is then the rate of major votes, which is 0.633.

The case centered background data includes 10,336 cases with their judicial decisions, among which in 6,531 cases justices supported petitioners while 3,784 cases justices supported respondents. 4 votes with no unclear disposition is removed. The accuracy of the baseline model on case level is then the rate of major votes, which is 0.633.

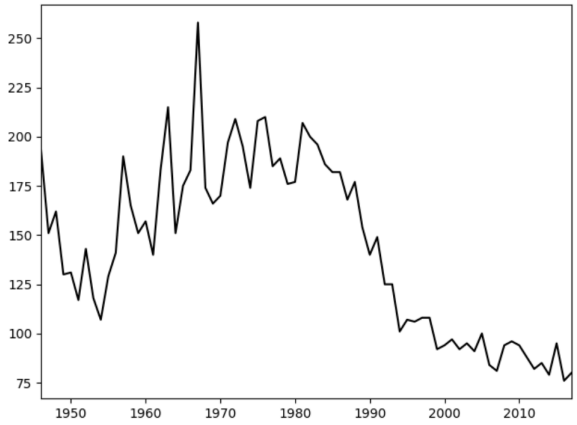

FIGURE 4.1: The yearly trend of number of cases

The yearly number of cases is shown in Figure 4.1, which generally increases from 1946 to 1970, and decreases until 2018.

4.2 Feature-fusion Results

Justice level results of feature-fusion experiments

According to the indices of justices (Justice ID) and cases (Docket number), entries of three types of data (structured data, text and audio) were aligned and sorted by indices. There remained 5,158 entries for each type of data. Each type of data was processed with corresponding methods and prepared for the further model training.

The structured data was inputed into a multilayer perceptron (MLP) model with three layers (64, 32 and 16 neurons, fully connected). The text data was inputed into a model particular for natural language processing, which using a one-dimensional convolutional neural network (CNN) as a preprocessing step before a gated recurrent unit (GRU), shortening the sequences and extracting essential patterns. The audio data parsed as vocal pitch was input into a MLP model with two layers (4 and 2 neurons, fully connected).

The Table 4.1 shows that, when structured data is input into the model for training, the test accuracy of the model is larger than 0.633, which means that the model performs better than the baseline model in prediction task. The results present that the structured data containing cases background provides useful information for court decision prediction.

The test accuracy of model using either text or audio data is less than 0.633, which means that, on justice level, the test transcripts or audio of oral arguments, singly and independently, or together, was not helpful for decision prediction.

When text and/or audio data was joint with structured data used in model training, however, results in a larger test accuracy than using only the structured data. The test accuracy of model using structured data and text data as input is the largest as 0.702. The result implies the effectiveness of feature-level fusion on decision prediction.

Case level results of feature-fusion experiments

The Table 4.2 points that, test accuracy of each experiment is larger than baseline accuracy of 0.633, which means that, on case level, each type of datasets (structured,

TABLE 4.1: The results of feature-level fusion experiments on justice level

Data	Model	Train Acc.	Val. Acc.	Test Acc.
Info + Text + Audio	Baseline	0.633	0.633	0.633
Info	MLP	0.822	0.712	0.690
Text	CNN + GRU	0.569	0.545	0.528
Audio	MLP	0.554	0.542	0.518
Info + Text	MLP + CNN + GRU	0.878	0.712	0.702
Info + Audio	MLP + MLP	0.818	0.715	0.696
Text + Audio	CNN + GRU + MLP	0.634	0.523	0.53
Info + Text + Audio	MLP + CNN + GRU	0.905	0.722	0.698

Abbreviation explanation. Val.: Validation, Acc.: Accuracy. MLP: Multilayer Perceptron; CNN: Convolutional Neural Network; GRU: Gated Recurrent Unit. Info: structured data of case background; Text: oral arguments transcripts; Audio: oral argument voice.

TABLE 4.2: The results of feature-level fusion experiments on case level

Data	Model	Train Acc.	Val. Acc.	Test Acc.
Info + Text + Audio	Baseline	0.633	0.633	0.633
Info	MLP	0.970	0.662	0.640
Text	CNN + GRU	0.652	0.694	0.685
Audio	MLP	0.611	0.662	0.640
Info + Text	MLP + CNN + GRU	0.650	0.694	0.685
Info + Audio	MLP + MLP	0.983	0.619	0.640
Text + Audio	CNN + GRU + MLP	0.694	0.694	0.690
Info + Text + Audio	MLP + CNN + GRU	0.995	0.606	0.630

Abbreviation explanation see Table 4.1.

text or audio data) provides useful information for decision prediction. The model using structured data and audio data has the largest test accuracy as 0.690, which is a slightly smaller than 0.696, the test accuracy of model using justice level structured and audio data.

Inferred case level results

The Table 4.3 presents the case level results inferred from feature-fusion experiments on justice level. The model with structured data has the highest test accuracy as 0.836. The other accuracy is all around 0.594, which may due to the untrained and biased threshold of sigmoid function in the last layer.

4.3 Decision-fusion Results

Justice level results of decision-fusion experiments

On justice level, the structured data has 92,554 justices decisions, the text data has 10,742 entries for each party of petitioner and respondent, and the audio data has 5,158 entries for each party.

TABLE 4.3: The results of case level prediction inferred from feature-level fusion experiments on justice level

Data	Model	Inferred Predicted Accuracy
Info + Text + Audio	Baseline	0.633
Info	MLP	0.836
Text	CNN + GRU	0.593
Audio	MLP	0.594
Info + Text	MLP + CNN + GRU	0.594
Info + Audio	MLP + MLP	0.594
Text + Audio	CNN + GRU + MLP	0.594
Info + Text + Audio	MLP + CNN + GRU	0.594

Abbreviation explanation see Table 4.1.

TABLE 4.4: The results of decision-level fusion experiments on justice level

Data	Model	Train Acc.	Val. Acc.	Test Acc.
Info + Text + Audio	Baseline	0.633	0.633	0.633
Info	MLP	0.994	0.993	0.993
Text	CNN + GRU	0.561	0.541	0.528
Audio	MLP	0.554	0.542	0.518
Info + Text	MLP + CNN + GRU	0.554	0.542	0.518
Info + Audio	MLP + MLP	0.554	0.542	0.518
Text + Audio	CNN + GRU + MLP	0.554	0.542	0.518
Info + Text + Audio	MLP + CNN + GRU	0.546	0.559	0.527

Abbreviation explanation see Table 4.1.

For the model based on a single type of data, all entries of that data were used for model training and tuning. For model based on two or three types of data, the overlapped part of multiple types of data will be collected and set as input to a corresponding multimodal learning model.

The test accuracy of model with structured data is the largest as 0.993. When joint with more type of data, the test accuracy sharply decreases to 0.518-0.528, which is less than the baseline. It may due to that the fusion of multiple type of data removes essential information and brings extra noise to the multimodal model. Another possible reason is that the test accuracy of either the text model or the audio model is too small, which brings bias to the joint decisions during the fusion process.

The Table 4.4 shows that

Case level results of decision-fusion experiments

On the case level, structured data (info) has 10,336 entries; oral arguments transcripts (text) has 3,006 utterances of justices speaking to each side of petitioner and respondent, thus text data has 6,012 utterances in total; oral arguments voice (audio) has 1,000 utterances of justices speaking to each side of petitioner and respondent, thus text data has 2,000 utterances in total.

The Table 4.5 shows that the text model and audio model performs better than the baseline, while the other models performs slightly worse than the baseline. It

TABLE 4.5: The results of decision-level fusion experiments on case level

Data	Model	Train Acc.	Val. Acc.	Test Acc.
Info + Text + Audio	Baseline	0.633	0.633	0.633
Info	MLP	0.934	0.568	0.578
Text	CNN + GRU	0.650	0.694	0.685
Audio	MLP	0.650	0.694	0.685
Info + Text	MLP + CNN + GRU	0.633	0.644	0.619
Info + Audio	MLP + MLP	0.633	0.644	0.619
Text + Audio	CNN + GRU + MLP	0.633	0.644	0.619
Info + Text + Audio	MLP + CNN + GRU	0.633	0.644	0.619

Abbreviation explanation see Table 4.1.

TABLE 4.6: The results of case level prediction inferred from decision-level fusion experiments on justice level

Data	Model	Inferred Predicted Accuracy
Info + Text + Audio	Baseline	0.633
Info	MLP	0.672
Text	CNN + GRU	0.626
Audio	MLP	0.626
Info + Text	MLP + CNN + GRU	0.626
Info + Audio	MLP + MLP	0.626
Text + Audio	CNN + GRU + MLP	0.626
Info + Text + Audio	MLP + CNN + GRU	0.626

Abbreviation explanation see Table 4.1.

may due to that the variation of speaking of oral arguments from different justices is removed after data combination.

Inferred case level results

The Table 4.6 shows the inferred results on case level of decision-fusion experiments on justice level. Except the test accuracy result of model using structured data is 0.72, other model is with test accuracy of 0.626. It may due the the untrainable threshold of sigmoid function.

Chapter 5

Conclusion

This paper explores and discusses the possibility of the court (the Supreme Court of the United States) decision prediction using multimodal deep learning. The experiment is implemented on both justice and case level, while with two types of procedures of multimodal deep learning methods, independently: feature-level fusion and decision level fusion. The result shows that, by using model fusion techniques on multiple types of data (structured data, text data, and audio data), the multimodal deep learning model performs better than the model based on a single type of data, and also better than some related previous research. The fusion of data and model provides more comprehensive data representation, that a single modality of data cannot present, which enables the use of multiple modalities of data from different sensors yet from the same entity, and can reduce the redundancy of data containing the same information.

There are limitations to this study. (1) files submitted by the petitioner and judgment opinions to each case are not involved. (2) the test accuracy is the only evaluation metrics to the model performance, while a system with more metrics considering different aspects should be considered. (3) the attention mechanism for natural language processing is not taken into consideration in this paper yet, which can provide an association between text and target.

Appendix A

Appendix Title

A.1 Hardwares

The hardwares used in all experiments are in A.1.

A.2 Variables of SCDB

The detailed introduction and cookbook containing variables definition and values can be found in the official website of the Supreme Court Database: Online Code Book. The variables include:

Identification Variables (SCDB Case ID, SCDB Docket ID, SCDB Issues ID, SCDB Vote ID, U.S. Reporter Citation, Supreme Court Citation, Lawyers Edition Citation, LEXIS Citation, Docket Number);

Background Variables(Case Name, Petitioner, Petitioner State, Respondent, Respondent State, Manner in which the Court takes Jurisdiction, Administrative Action Preceeding Litigation, Administrative Action Preceeding Litigation State, Three-Judge District Court, Origin of Case, Origin of Case State, Source of Case, Source of Case State, Lower Court Disagreement, Reason for Granting Cert, Lower Court Disposition, Lower Court Disposition Direction);

Chronological Variables (Date of Decision, Term of Court, Natural Court, Chief Justice, Date of Oral Argument, Date of Reargument);

Substantive Variables, Issue, Issue Area, Decision Direction, Decision Direction Dissent, Authority for Decision 1, Authority for Decision 2, Legal Provisions Considered by the Court, Legal Provision Supplement, Legal Provision Minor Supplement);

Outcome Variables (Decision Type, Declaration of Unconstitutionality, Disposition of Case, Unusual Disposition, Winning Party, Formal Alteration of Precedent) ;

TABLE A.1: The hardwares used in experiments

Item	Details
CPU	AMD Ryzen Threadripper 1920X WOF, Prozessor
CPU	Fan Noctua NH-U14S TR4-SP3, CPU-Kühler
GPU	Gainward GeForce RTX 2070, Grafikkarte
Motherboard	GIGABYTE X399 AORUS PRO, Mainboard
RAM	Corsair DIMM 32 GB DDR4-3200 Quad-Kit, Arbeitsspeicher
SSD	Samsung 970 EVO Plus 500 GB, Solid State Drive
Power	Corsair RM1000i 1000W, PC-Netzteil

Voting & Opinion Variables (Vote Not Clearly Specified, Majority Opinion Writer, Majority Opinion Assigner, Split Vote, Majority Votes, Minority Votes, Justice ID, Justice Name, The Vote in the Case, Opinion, Direction of the Individual Justice's Votes, Majority and Minority Voting by Justice, First Agreement, Second Agreement).

A.3 Codes

All the codes (in Python) used in this paper can be found in the Github repository:
https://github.com/wwendi/thesis-code/tree/master/code
An example of a part of codes is as follows:

```
# a part of code, from th_16_model_justice.py
from keras.models import Sequential
from keras.layers.core import Dense
from keras.layers import LSTM, GRU, Input, Bidirectional,
                                    TimeDistributed
from keras.layers.embeddings import Embedding
from keras.layers.convolutional import Conv1D
from keras.layers.convolutional import MaxPooling1D
from keras.initializers import Constant
from keras.models import Model
from keras import initializers
from keras.engine.topology import Layer, InputSpec
from keras import backend as K


def create_cnn(num_words, max_length, embedding_dim, embedding_matrix,
                                    regress=False):
    # load pre-trained embedding
    embedding_layer = Embedding(num_words,
         embedding_dim,
         embeddings_initializer=Constant(embedding_matrix),
         input_length=max_length,
         trainable=False)

    sequence_input = Input(shape=(max_length,), dtype='int32')
    embedded_sequences = embedding_layer(sequence_input)

    x = Conv1D(filters=32, kernel_size=3, padding='same', activation='
                                        relu')(embedded_sequences)
    x = MaxPooling1D(pool_size=5)(x)
    x = GRU(16, dropout=0.1, recurrent_dropout=0.5)(x)

    # check to see if the regression node should be added
    if regress:
        x = Dense(1, activation="sigmoid")(x)

    model = Model(sequence_input, x)
    # return the CNN
    return model
```

Bibliography

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283.

Abidi, M. A. and Gonzalez, R. C. (1992). *Data fusion in robotics and machine intelligence*. Academic Press Professional, Inc.

Abrams, K. (2010). Empathy and experience in the sotomayor hearings. *Ohio NUL Rev.*, 36:263.

Abrams, K. and Keren, H. (2009). Who's afraid of law and the emotions. *Minn. L. Rev.*, 94:1997.

Alarie, B., Niblett, A., and Yoon, A. H. (2018). How artificial intelligence will affect the practice of law. *University of Toronto Law Journal*, 68(supplement 1):106–124.

Aletras, N., Tsarapatsanis, D., Preoţiuc-Pietro, D., and Lampos, V. (2016). Predicting judicial decisions of the european court of human rights: A natural language processing perspective. *PeerJ Computer Science*, 2:e93.

Alspach, D. and Sorenson, H. (1972). Nonlinear bayesian estimation using gaussian sum approximations. *IEEE transactions on automatic control*, 17(4):439–448.

Arewa, O. B. (2006). Open access in a closed universe: Lexis, westlaw, law schools, and the legal information market. *Lewis & Clark L. Rev.*, 10:797.

Assael, Y. M., Shillingford, B., Whiteson, S., and De Freitas, N. (2016). Lipnet: End-to-end sentence-level lipreading. *arXiv preprint arXiv:1611.01599*.

Atrey, P. K., Hossain, M. A., El Saddik, A., and Kankanhalli, M. S. (2010). Multimodal fusion for multimedia analysis: a survey. *Multimedia systems*, 16(6):345–379.

Bailey, M. A. and Maltzman, F. (2008). Does legal doctrine matter? unpacking law and policy preferences on the us supreme court. *American Political Science Review*, 102(3):369–384.

Baltrušaitis, T., Ahuja, C., and Morency, L.-P. (2018). Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443.

Bandes, S. (1996). Empathy, narrative, and victim impact statements, 63 u. chi. l. rev. 361 (1996).

Bandes, S. A. (2009). Empathetic judging and the rule of law. *Cardozo Law Review De Novo*, page 133.

Bandes, S. A. and Blumenthal, J. A. (2012). Emotion and the law. *Annual Review of Law and Social Science*, 8:161–181.

Barron, A., Rissanen, J., and Yu, B. (1998). The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760.

Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I., Bergeron, A., Bouchard, N., Warde-Farley, D., and Bengio, Y. (2012). Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*.

Bates, M. (2003). The assessment of work integrated learning: Symptoms of personal change. *Journal of Criminal Justice Education*, 14(2):303–326.

Baum, L. (2018). *The supreme court*. CQ press.

Benfold, B. and Reid, I. (2011). Stable multi-target tracking in real-time surveillance video. In *CVPR 2011*, pages 3457–3464. IEEE.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

Bengio, Y., Delalleau, O., and Roux, N. L. (2006). The curse of highly variable functions for local kernel machines. In *Advances in neural information processing systems*, pages 107–114.

Bengio, Y. et al. (2009). Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127.

Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. (2007). Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160.

Bengio, Y., Simard, P., Frasconi, P., et al. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Black, R. C., Treul, S. A., Johnson, T. R., and Goldman, J. (2011). Emotions, oral arguments, and supreme court decision making. *The Journal of Politics*, 73(2):572–581.

Blackman, S. S. (2004). Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, 19(1):5–18.

Bloch, I. (1996). Information combination operators for data fusion: a comparative review with classification. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 26(1):52–67.

Blom, H. A. and Bar-Shalom, Y. (1988). The interacting multiple model algorithm for systems with markovian switching coefficients. *IEEE transactions on Automatic Control*, 33(8):780–783.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. (2016). End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

Bradford, L. R. (2008). Emotion, dilution, and the trademark consumer. *Berkeley Tech. LJ*, 23:1227.

Briggs, L. J. (2015). Civil courts structure review: Interim report.

Byrd, O. and Howard, B. (2014). *Lex Machina: 2013 Patent Litigation Year in Review*. Lex Machina.

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7291–7299.

Champandard, A. J. (2016). Semantic style transfer and turning two-bit doodles into fine artworks. *arXiv preprint arXiv:1603.01768*.

Chan, D. Y., Chiu, V., and Vasarhelyi, M. A. (2018). New perspective: Data analytics as a precursor to audit automation. In *Continuous Auditing: Theory and Application*, pages 315–322. Emerald Publishing Limited.

Cheng, Z., Yang, Q., and Sheng, B. (2015). Deep colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 415–423.

Chiu, C.-C., Sainath, T. N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., Kannan, A., Weiss, R. J., Rao, K., Gonina, E., et al. (2018). State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4774–4778. IEEE.

Chollet, F. (2017). Deep learning with python.

Chollet, F. et al. (2015). Keras. https://keras.io.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. (2015). The loss surfaces of multilayer networks. In *Artificial Intelligence and Statistics*, pages 192–204.

Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., and Bengio, Y. (2015). Attention-based models for speech recognition. In *Advances in neural information processing systems*, pages 577–585.

Chung, J., Ahn, S., and Bengio, Y. (2016). Hierarchical multiscale recurrent neural networks. *arXiv preprint arXiv:1609.01704*.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Chung, J. S., Senior, A., Vinyals, O., and Zisserman, A. (2017). Lip reading sentences in the wild. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3444–3453. IEEE.

Clark, J. J. and Yuille, A. L. (2013). *Data fusion for sensory information processing systems*, volume 105. Springer Science & Business Media.

Clore, G. L., Schwarz, N., and Conway, M. (1994). Affective causes and consequences of social information processing. *Handbook of social cognition*, 1:323–417.

Collobert, R. and Bengio, S. (2004). Links between perceptrons, mlps and svms. In *Proceedings of the twenty-first international conference on Machine learning*, page 23. ACM.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Cover, T. M., Hart, P., et al. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.

Dahl, G. E., Yu, D., Deng, L., and Acero, A. (2011). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, 20(1):30–42.

Dahl, R., Norouzi, M., and Shlens, J. (2017). Pixel recursive super resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5439–5448.

Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. (2014). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems*, pages 2933–2941.

De Lathauwer, L., De Moor, B., and Vandewalle, J. (2000). A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

DiCecco, R., Lacey, G., Vasiljevic, J., Chow, P., Taylor, G., and Areibi, S. (2016). Caffeinated fpgas: Fpga framework for convolutional neural networks. In *2016 International Conference on Field-Programmable Technology (FPT)*, pages 265–268. IEEE.

Dietrich, B. J., Enos, R. D., and Sen, M. (2019). Emotional arousal predicts voting on the us supreme court. *Political Analysis*, 27(2):237–243.

Dixon, M., Klabjan, D., and Bang, J. H. (2017). Classification-based financial markets prediction using deep neural networks. *Algorithmic Finance*, 6(3-4):67–77.

Djuric, P. M. and Chun, J.-H. (2002). An mcmc sampling approach to estimation of nonstationary hidden markov models. *IEEE Transactions on Signal Processing*, 50(5):1113–1123.

Doucet, A., Logothetis, A., and Krishnamurthy, V. (2000). Stochastic sampling algorithms for state estimation of jump markov linear systems. *IEEE Transactions on Automatic Control*, 45(2):188–202.

El Hihi, S. and Bengio, Y. (1996). Hierarchical recurrent neural networks for long-term dependencies. In *Advances in neural information processing systems*, pages 493–499.

Epstein, L., Martin, A. D., Quinn, K. M., and Segal, J. A. (2007). Ideological drift among supreme court justices: Who, when, and how important. *Nw. UL Rev.*, 101:1483.

Feigenson, N. and Park, J. (2006). Emotions and attributions of legal responsibility and blame: A research review. *Law and Human Behavior*, 30(2):143.

Fernando, B. and Gould, S. (2016). Learning end-to-end video classification with rank-pooling. In *International Conference on Machine Learning*, pages 1187–1196.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202.

Fukushima, K. (2007). Neocognitron. *Scholarpedia*, 2(1):1717.

Gatys, L. A., Ecker, A. S., and Bethge, M. (2016). Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423.

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org.

Globerson, A., Chechik, G., Pereira, F., and Tishby, N. (2007). Euclidean embedding of co-occurrence data. *Journal of Machine Learning Research*, 8(Oct):2265–2295.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Goutsias, J., Mahler, R. P., and Nguyen, H. T. (2012). *Random sets: theory and applications*, volume 97. Springer Science & Business Media.

Graves, A. and Schmidhuber, J. (2009). Offline handwriting recognition with multidimensional recurrent neural networks. In *Advances in neural information processing systems*, pages 545–552.

Greene, J. D., Sommerville, R. B., Nystrom, L. E., Darley, J. M., and Cohen, J. D. (2001). An fmri investigation of emotional engagement in moral judgment. *Science*, 293(5537):2105–2108.

Greenwald, D. and Schwartz Jr, F. A. (2001). The censorial judiciary. *UC Davis L. Rev.*, 35:1133.

Haidt, J. (2001). The emotional dog and its rational tail: a social intuitionist approach to moral judgment. *Psychological review*, 108(4):814.

Halko, N., Martinsson, P.-G., and Tropp, J. A. (2009). Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions.

Hall, D. L. and Llinas, J. (1997). An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23.

Harold J. Spaeth, Lee Epstein, e. a. (2018). Supreme court database. `http://Supremecourtdatabase.org`. Version 2018 Release 2.

Hayes-Roth, F., Waterman, D. A., and Lenat, D. B. (1983). Building expert system.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017a). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

He, L., Lee, K., Lewis, M., and Zettlemoyer, L. (2017b). Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483.

Heaton, J., Polson, N., and Witte, J. H. (2017). Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*, 33(1):3–12.

Henderson, L. N. (1987). Legality and empathy. *Michigan Law Review*, 85(7):1574–1653.

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Kingsbury, B., et al. (2012). Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29.

Hinton, G. E. et al. (2005). What kind of graphical model is the brain? In *IJCAI*, volume 5, pages 1765–1775.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507.

Hirzinger, G., Brunner, B., Dietrich, J., and Heindl, J. (1993). Sensor-based space robotics-rotex and its telerobotic features. *IEEE Transactions on robotics and automation*, 9(5):649–663.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

Hu, Z., Li, X., Tu, C., Liu, Z., and Sun, M. (2018). Few-shot charge prediction with discriminative legal attributes. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 487–498.

Huang, P. H. (2003). Trust, guilt, and securities regulation. *University of Pennsylvania Law Review*, 151(3):1059–1095.

Hubel, D. H. and Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243.

Huntington, C. (2007). Repairing family law. *Duke LJ*, 57:1245.

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM.

Johnson, T. R. (2004). *Oral arguments and decision making on the United States Supreme Court*. SUNY Press.

Johnson, T. R. and Goldman, J. (2009). *A good quarrel: America's top legal reporters share stories from inside the Supreme Court*. University of Michigan Press.

Johnson, T. R., Wahlbeck, P. J., and Spriggs, J. F. (2006). The influence of oral arguments on the us supreme court. *American Political Science Review*, 100(1):99–113.

Joshi, R. and Sanderson, A. C. (1999). Minimal representation multisensor fusion using differential evolution. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 29(1):63–76.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Kahan, D. M. (2007). Two conceptions of emotion in risk regulation. *U. Pa. L. Rev.*, 156:741.

Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Katz, D. M., Bommarito II, M. J., and Blackman, J. (2017). A general approach for predicting the behavior of the supreme court of the united states. *PloS one*, 12(4):e0174698.

Kim, I. and Vachtsevanos, G. (1998). Overlapping object recognition: A paradigm for multiple sensor fusion. *IEEE Robotics & Automation Magazine*, 5(3):37–44.

Kim, Y., Jernite, Y., Sontag, D., and Rush, A. M. (2016). Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Kiros, R., Salakhutdinov, R., and Zemel, R. S. (2014). Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., Zhong, V., Paulus, R., and Socher, R. (2016). Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning*, pages 1378–1387.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Lebret, R. and Collobert, R. (2013). Word emdeddings through hellinger pca. *arXiv preprint arXiv:1312.5542*.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., and Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

LeDoux, J. (1998). *The emotional brain: The mysterious underpinnings of emotional life*. Simon and Schuster.

Lee, E. D., Broedersz, C. P., and Bialek, W. (2015). Statistical mechanics of the us supreme court. *Journal of Statistical Physics*, 160(2):275–301.

Lee, J.-G., Jun, S., Cho, Y.-W., Lee, H., Kim, G. B., Seo, J. B., and Kim, N. (2017). Deep learning in medical imaging: general overview. *Korean journal of radiology*, 18(4):570–584.

Leung, M. K., Xiong, H. Y., Lee, L. J., and Frey, B. J. (2014). Deep learning of the tissue-regulated splicing code. *Bioinformatics*, 30(12):i121–i129.

Levy, O. and Goldberg, Y. (2014a). Linguistic regularities in sparse and explicit word representations. In *Proceedings of the eighteenth conference on computational natural language learning*, pages 171–180.

Levy, O. and Goldberg, Y. (2014b). Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185.

Li, Y., Xu, L., Tian, F., Jiang, L., Zhong, X., and Chen, E. (2015). Word embedding revisited: A new representation learning and explicit matrix factorization perspective. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

LIN, S.-Z. and HAN, Z. (2017). Images fusion based on deep stack convolutional neural network. *Chinese Journal of Computers*, 40(11):2506–2518.

Liu, N., Han, J., Zhang, D., Wen, S., and Liu, T. (2015). Predicting eye fixations using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 362–370.

Liu, Z.-G., Pan, Q., and Dezert, J. (2013). A new belief-based k-nearest neighbor classification method. *Pattern Recognition*, 46(3):834–844.

Logothetis, A. and Krishnamurthy, V. (1999). Expectation maximization algorithms for map estimation of jump markov linear systems. *IEEE Transactions on Signal Processing*, 47(8):2139–2156.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

Luo, B., Feng, Y., Xu, J., Zhang, X., and Zhao, D. (2017). Learning to predict charges for criminal cases with legal basis. *arXiv preprint arXiv:1707.09168*.

Ma, J., Sheridan, R. P., Liaw, A., Dahl, G. E., and Svetnik, V. (2015). Deep neural nets as a method for quantitative structure–activity relationships. *Journal of chemical information and modeling*, 55(2):263–274.

Manyika, J. and Durrant-Whyte, H. (1995). *Data Fusion and Sensor Management: a decentralized information-theoretic approach*. Prentice Hall PTR.

Martin, A. D. and Quinn, K. M. (2002). Dynamic ideal point estimation via markov chain monte carlo for the us supreme court, 1953–1999. *Political Analysis*, 10(2):134–153.

Matsugu, M., Mori, K., Mitari, Y., and Kaneda, Y. (2003). Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Networks*, 16(5-6):555–559.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Minow, M. L. and Spelman, E. V. (1988). Passion for justice. *Cardozo L. Rev.*, 10:37.

Minsky, M. and Papert, S. A. (2017). *Perceptrons: An introduction to computational geometry*. MIT press.

Mohamed, A.-r., Dahl, G. E., and Hinton, G. (2011). Acoustic modeling using deep belief networks. *IEEE transactions on audio, speech, and language processing*, 20(1):14–22.

Molnar, K. J. and Modestino, J. W. (1998). Application of the em algorithm for the multitarget/multisensor tracking problem. *IEEE Transactions on Signal Processing*, 46(1):115–129.

Mori, S. (1997). Random sets in data fusion problems. In *Signal and Data Processing of Small Targets 1997*, volume 3163, pages 278–289. International Society for Optics and Photonics.

Morris, C. N. (1983). Parametric empirical bayes inference: theory and applications. *Journal of the American statistical Association*, 78(381):47–55.

Murphy, J. G. and Hampton, J. (1990). *Forgiveness and mercy*. Cambridge University Press.

Murphy, R. R. (1998). Dempster-shafer theory for sensor fusion in autonomous mobile robots. *IEEE Transactions on Robotics and Automation*, 14(2):197–206.

Musicki, D. and Evans, R. (2002). Linear joint integrated probabilistic data association-ljipda. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, volume 3, pages 2415–2420. IEEE.

Nam, H., Baek, M., and Han, B. (2016). Modeling and propagating cnns in a tree structure for visual tracking. *arXiv preprint arXiv:1608.07242*.

Neal, R. M. (1993). *Probabilistic inference using Markov chain Monte Carlo methods*. Department of Computer Science, University of Toronto Toronto, Ontario, Canada.

Nelson, C. L. and Fitzgerald, D. S. (1996). Sensor fusion for intelligent alarm analysis. In *1996 30th Annual International Carnahan Conference on Security Technology*, pages 143–150. IEEE.

Neverova, N., Wolf, C., Lacey, G., Fridman, L., Chandra, D., Barbello, B., and Taylor, G. (2016). Learning human identity from motion patterns. *IEEE Access*, 4:1810–1820.

Núñez, J., Otazu, X., Fors, O., Prades, A., Palà, V., and Arbiol, R. (1999). Multiresolution-based image fusion with additive wavelet decomposition. *IEEE Transactions on Geoscience Remote Sensing*, 37(3):1204–1211.

Nussbaum, M. C. (1992). *Love's knowledge: Essays on philosophy and literature*. OUP USA.

Oord, A. v. d., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., Driessche, G. v. d., Lockhart, E., Cobo, L. C., Stimberg, F., et al. (2017). Parallel wavenet: Fast high-fidelity speech synthesis. *arXiv preprint arXiv:1711.10433*.

Parkhi, O. M., Vedaldi, A., Zisserman, A., et al. (2015). Deep face recognition. In *bmvc*, volume 1, page 6.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318.

Pellegrini, S., Ess, A., Schindler, K., and Van Gool, L. (2009). You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*, pages 261–268. IEEE.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Petrovic, V. S. and Xydeas, C. S. (2004). Gradient-based multiresolution image fusion. *IEEE Transactions on Image Processing*, 13(2):228–237.

Ping, W., Peng, K., Gibiansky, A., Arik, S. O., Kannan, A., Narang, S., Raiman, J., and Miller, J. (2017). Deep voice 3: Scaling text-to-speech with convolutional sequence learning. *arXiv preprint arXiv:1710.07654*.

Poria, S., Cambria, E., Howard, N., Huang, G.-B., and Hussain, A. (2016). Fusing audio, visual and textual clues for sentiment analysis from multimodal content. *Neurocomputing*, 174:50–59.

Posner, R. A. (2005). Intellectual property: The law and economics approach. *Journal of Economic Perspectives*, 19(2):57–73.

Poultney, C., Chopra, S., Cun, Y. L., et al. (2007). Efficient learning of sparse representations with an energy-based model. In *Advances in neural information processing systems*, pages 1137–1144.

Price, A. L., Patterson, N. J., Plenge, R. M., Weinblatt, M. E., Shadick, N. A., and Reich, D. (2006). Principal components analysis corrects for stratification in genome-wide association studies. *Nature genetics*, 38(8):904.

Ragin, C. C. (2000). *Fuzzy-set social science*. University of Chicago Press.

Ramachandram, D. and Taylor, G. W. (2017). Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine*, 34(6):96–108.

Richardson, J. M. and Marsh, K. A. (1988). Fusion of multisensor data. *The International Journal of Robotics Research*, 7(6):78–96.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

Rumelhart, D. E., Hinton, G. E., Williams, R. J., et al. (1988). Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1.

Ryan, E. (2005). The discourse beneath: Emotional epistemology in legal deliberation and negotiation. *Harv. Negot. L. Rev.*, 10:231.

Schafer, B. (2016). The future of ip law in an age of artificial intelligence.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Segal, J. A. and Spaeth, H. J. (1996). The influence of stare decisis on the votes of united states supreme court justices. *American Journal of Political Science*, pages 971–1003.

Segal, J. A. and Spaeth, H. J. (2002). *The Supreme Court and the attitudinal model revisited*. Cambridge University Press.

Severyn, A. and Moschitti, A. (2015). Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 959–962. ACM.

Socher, R., Bauer, J., Manning, C. D., et al. (2013a). Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 455–465.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013b). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Sorokin, I., Seleznev, A., Pavlov, M., Fedorov, A., and Ignateva, A. (2015). Deep attention recurrent q-network. *arXiv preprint arXiv:1512.01693*.

Srivastava, N. and Salakhutdinov, R. R. (2012). Multimodal learning with deep boltzmann machines. In *Advances in neural information processing systems*, pages 2222–2230.

Susskind, R. E. (2017). *Tomorrow's lawyers: An introduction to your future*. Oxford University Press.

Sutskever, I. (2013). *Training recurrent neural networks*. University of Toronto Toronto, Ontario, Canada.

Thoits, P. A. (1989). The sociology of emotions. *Annual review of sociology*, 15(1):317–342.

Tian, Y., Luo, P., Wang, X., and Tang, X. (2015). Deep learning strong parts for pedestrian detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1904–1912.

Tokuda, K., Yoshimura, T., Masuko, T., Kobayashi, T., and Kitamura, T. (2000). Speech parameter generation algorithms for hmm-based speech synthesis. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, volume 3, pages 1315–1318. IEEE.

Torrey, L. and Shavlik, J. (2010). Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI Global.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for computational Linguistics.

Van Der Merwe, R., Doucet, A., De Freitas, N., and Wan, E. A. (2001). The unscented particle filter. In *Advances in neural information processing systems*, pages 584–590.

Vipperla, R., Renals, S., and Frankel, J. (2008). Longitudinal study of asr performance on ageing voices.

Vo, B.-N., Singh, S., and Doucet, A. (2005). Sequential monte carlo methods for multitarget filtering with random finite sets. *IEEE Transactions on Aerospace and electronic systems*, 41(4):1224–1245.

Waterman, D. (1986). A guide to expert systems.

Weiss, D., Alberti, C., Collins, M., and Petrov, S. (2015). Structured training for neural network transition-based parsing. *arXiv preprint arXiv:1506.06158*.

Welch, G., Bishop, G., et al. (1995). An introduction to the kalman filter.

Weston, J., Bordes, A., Chopra, S., Rush, A. M., van Merriënboer, B., Joulin, A., and Mikolov, T. (2015). Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Weston, J., Chopra, S., and Bordes, A. (2014). Memory networks. *arXiv preprint arXiv:1410.3916*.

White, B. T. (2010). Underwater and not walking away: shame, fear, and the social management of the housing crisis. *Wake Forest L. Rev.*, 45:971.

Xiong, W., Droppo, J., Huang, X., Seide, F., Seltzer, M., Stolcke, A., Yu, D., and Zweig, G. (2016). Achieving human parity in conversational speech recognition. *arXiv preprint arXiv:1610.05256*.

Yang, S., Luo, P., Loy, C. C., and Tang, X. (2017). Faceness-net: Face detection through deep facial part responses. *IEEE transactions on pattern analysis and machine intelligence*, 40(8):1845–1859.

Yoon, W.-K., Goshozono, T., Kawabe, H., Kinami, M., Tsumaki, Y., Uchiyama, M., Oda, M., and Doi, T. (2004). Model-based space robot teleoperation of ets-vii manipulator. *IEEE Transactions on Robotics and Automation*, 20(3):602–612.

Yu, A., Abrego-Collier, C., Phillips, J., Pillion, B., and Chen, D. L. (2015). Investigating variation in english vowel-to-vowel coarticulation in a longitudinal phonetic corpus. In *Proceedings of the 18th International Congress of Phonetic Sciences*.

Yu, H., Wang, J., Huang, Z., Yang, Y., and Xu, W. (2016). Video paragraph captioning using hierarchical recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4584–4593.

Yuan, J. and Liberman, M. (2008). Speaker identification on the scotus corpus. *Journal of the Acoustical Society of America*, 123(5):3878.

Yuan, J. and Liberman, M. (2010). Robust speaking rate estimation using broad phonetic class recognition. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4222–4225. IEEE.

Yuan, J. and Liberman, M. (2011). Automatic detection of "g-dropping" in american english using forced alignment. In *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 490–493. IEEE.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.

Zhang, R., Li, W., Tan, W., and Mo, T. (2017). Deep and shallow model for insurance churn prediction service. In *2017 IEEE International Conference on Services Computing (SCC)*, pages 346–353. IEEE.

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. (2014). Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495.

Zhou, X., Dong, D., Wu, H., Zhao, S., Yu, D., Tian, H., Liu, X., and Yan, R. (2016). Multi-view response selection for human-computer conversation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 372–381.

Zhu, X. X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., and Fraundorfer, F. (2017). Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):8–36.

Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., and Hu, S. (2016). Traffic-sign detection and classification in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2110–2118.

Zhuge, J.-W., Wang, D.-W., Chen, Y., Ye, Z.-Y., and Zou, W. (2006). A network anomaly detector based on the ds evidence theory. *Journal of software*, 17(3):463–471.

Zimmermann, H.-J. (2011). *Fuzzy set theory—and its applications*. Springer Science & Business Media.

Declaration

1. I hereby declare that this thesis entitled:

   Predicting Court Decision: a Multimodal Deep Learning Perspective

   is a result of my own work and that no other than the indicated aids have been used for its completion. Material borrowed directly or indirectly from the works of others is indicated in each individual case by acknowledgement of the source and also the secondary literature used.

   This work has not previously been submitted to any other examining authority and has not yet been published.

2. After completion of the examining process, this work will be given to the library of the University of Konstanz, where it will be accessible to the public for viewing and borrowing. As author of this work, I agree / do not agree[*)] to this procedure.

Konstanz, 29.08.2019 (Date) Wenting Wang (Signature)

Erklärung

1. Ich versichere hiermit, dass ich die vorliegende Arbeit mit dem Thema:

   Predicting Court Decision: a Multimodal Deep Learning Perspective

   selbständig verfasst und keine anderen Hilfsmittel als die angegebenen benutzt habe. Die Stellen, die anderen Werken dem Wortlaut oder dem Sinne nach entnommen sind, habe ich in jedem einzelnen Falle durch Angaben der Quelle, auch der benutzten Sekundärliteratur, als Entlehnung kenntlich gemacht.

   Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

2. Diese Arbeit wird nach Abschluss des Prüfungsverfahrens der Universitätsbibliothek Konstanz übergeben und ist durch Einsicht und Ausleihe somit der Öffentlichkeit zugänglich. Als Urheber der anliegenden Arbeit stimme ich diesem Verfahren zu / nicht zu[*)].

Konstanz, den 29.08.2019 (Unterschrift) Wenting Wang

[*)] Please delete as applicable / Nichtzutreffendes bitte streichen.