T.R.

GEBZE TECHNICAL UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

EXTRACTION OF ARTICLE BASIS OF THE VERDICT FROM LAW CASES

YUSUF FATIH ŞIŞMAN

SUPERVISOR
PROF. YUSUF SINAN AKGÜL

GEBZE
2023

T.R.
GEBZE TECHNICAL UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

EXTRACTION OF ARTICLE BASIS OF THE VERDICT FROM LAW CASES

YUSUF FATIH ŞIŞMAN

SUPERVISOR
PROF. YUSUF SINAN AKGÜL

2023
GEBZE

GEBZE TECHNICAL UNIVERSITY	GRADUATION PROJECT JURY APPROVAL FORM

This study has been accepted as an Undergraduate Graduation Project in the Department of Computer Engineering on 19/01/2023 by the following jury.

JURY

Member
(Supervisor) : Prof. Yusuf Sinan Akgül

Member : Prof. İbrahim Soğukpınar

ABSTRACT

It is a laborious task for lawyers to find the case files similar to their current cases among many case files to strengthen their defenses. This paper propose a sequence to sequence system to solve this problem. The system extract which articles have been violated and which have not from law cases. Furthermore, a dataset consists of law cases is manually annotated to train a model. Experiments show that the proposed system can be used instead of reading a lot of case files to find similar case files. The Rogue-1 F1, Rogue-2 F1 and Rogue-3 F1 score of the proposed system finally reaches to 0.94, 0.88 and 0.82 respectively. And average runtime of the model is 0.015 seconds for a law case.

Keywords: Natural Language Processing, Machine Learning, sequence-to-sequence, Law.

ÖZET

Savunmalarını güçlendirmek için bir çok dava dosyası arasından kendi davalarına benzer dava dosyalarını bulmak avukatlar için zahmetli bir iş. Bu rapor bu sorunu çözmek için bir diziden diziye sistemi öneriyor. Bu sistem dava dosylarından hangi maddelerin ihlal edildiğini ve hangilerini ihlal edilmediğini çıkarıyor. Ayrıca, modeli eğitmek için dava dosyalarından oluşan bir veri seti manuel olarak etiketleniyor. Deneyler benzer dava dosyaları bulmak için bir çok dava dosyası okumaktansa bu önerilen sistemin kullanlabileceğini gösteriyor. Önerilen sistemin Rogue-1 F1, Rogue-2 F1 ve Rogue-3 F1 skorları sırasıyla 0.94, 0.88 ve 0.82'ye ulaşıyor. Ve bir dava dosyası için ortalama çalışma süresi 0.015 saniye sürüyor.

Anahtar Kelimeler: Doğal Dil İşleme, Makine Öğrenmesi, diziden diziye, Hukuk.

ACKNOWLEDGEMENT

I would like to thank Prof. Yusuf Sinan Akgül for being reachable throughout the semester and his assistance during the project and Niyazi Mehmet Cansever for providing the case files I used to create the dataset.

Yusuf Fatih Şişman

LIST OF SYMBOLS AND ABBREVIATIONS

Symbol or Abbreviation	:	**Explanation**
LSTM	:	Long Short Term Memory
s_k	:	Encoder state at encoder step k
h_t	:	Hidden state of the decoder at step t
a_{tk}	:	Attention weight for encoder state k at decoder step t
c_t	:	Attention output at decoder step t
p_t	:	Predicted output at decoder step t

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

There are a lot of European Court on Human Rights law cases and these cases are unstructured or not well structured. This situation makes it difficult for lawyers to find law cases similar to their current case and thus strengthen their defenses. Knowing which articles have been violated or which have not in a case file makes it easier to search for case files similar to their current case.

In this project a sequence to sequence approach is proposed to solve this problem. In the recent years, sequence to sequence neural networks have achieved great success in a number of natural language processing tasks. Sequence to sequence neural networks are trained to convert input sequences to output sequences and used to solve complex language problems like machine translation, text summarization, question answering etc. In this project it is used to extract which articles have been violated and which have not from European Court of Human Rights law cases. Thus, lawyers can find law cases similar to their current case easily.

Although there are several type of natural language processing projects in the field of law, this project is different from them because none of them aim to extract which articles have been violated and which have not in an abstractive way like in this project.

Contributions of this project are summarized as follows:

1. An unique approach is proposed to help lawyers to find cases similar to their cases.

2. The proposed approach is formalized as a sequence to sequence problem.

3. An annotated dataset that consists of Turkish law cases is created to train a sequence to sequence model.

The remaining of this paper organized as follows. Chapter 2 discusses related natural language processing work in the field of law. Chapter 3 describes annotation of dataset and system architecture. Chapter 4 discusses experiments and their results. Finally Chapter 5 concludes project and discusses future work.

2. RELATED WORK

There are a lot of natural language processing researches in the field of the law. In the following, some of the different types of researches is discussed.

In Information Extraction from Legal Documents: A Study in the Context of Common Law Court Judgements [1], sentences in the law cases are classified as fact, reasoning or conclusion to assist lawyers in identifying sections of judgements relevant to their case, and analysis of cases to identify relation between facts and conclusions.

In A Dataset of German Legal Documents for Named Entity Recognition [2], entities like Legal Norms, Court Decision, Legal Literature are extracted from German Legal Documents.

In Information Extraction from Arabic Law Documents [3], a hybrid approach that uses rule based methods and machine learning for named entity recognition and relation extraction tasks on Arabic law documents is proposed.

In Legal Texts Summarization by Exploration of the Thematic Structures and Argumentative Roles [4] and Automatic Text Summarization of Legal Cases: A Hybrid Approach [5], text summarization approach is used on law cases.

In CAIL2019-SCM: A Dataset of Similar Case Matching in Legal Domain [6], a semantic matching system is proposed to find similar case files.

Although some of the above studies have the same goal as this project, this project proposes different approach to realize this goal. And this approach is the extraction of which articles have been violated and which have not.

3. METHODOLOGY

This project aims to extract which articles have been violated and which have not from European Court of Human Rights law cases with using machine learning. Realizing this goal requires two steps: creating a dataset by annotating the law cases and training a model with this dataset.

3.1. Annotation of the Dataset

In the following, different situations and how they are expressed in annotations are described.

Violation of articles have the first priority order in the annotation. One case file may contain violations of more than one article.

- 3 ihlal edildiğine
- 2 , 3 ihlal edildiğine

Absence of violation of articles have the second priority order in the annotation. One case file may contain absence of violations of more than one article.

- 2 ihlal edilmediğine
- 2 ihlal edildiğine ; 3 , 5 ihlal edilmediğine

In some cases court decides whether an action will result in a violation of any article, and these situations have the third and fourth priority order in the annotation.

- 2 ihlal edeceğine ; 3 ihlal etmeyeceğine
- 2 ihlal edildiğine ; 3 ihlal edilmediğine ; 4 , 6 ihlal edeceğine ; 5 ihlal etmeyeceğine

In some cases there are multiple plaintiff or same article is examined in terms of several actions. In annotations of these cases, same article may and may not be violated in same time.

- 2 ihlal edeceğine ; 2 ihlal etmeyeceğine

In some cases the article is indicated together with its paragraph and subparagraph.

- 5 - 3 ihlal edildiğine
- 6 - 1 , 6 - 3 c ihlal edildiğine

In some cases an article is examined substantively and procedurally.

- 3 esas ve usul ihlal edildiğine
- 2 esas ihlal edildiğine ; 2 usul ihlal edilmediğine

There are some additional protocols in European Convention on Human Rights. These protocols are expressed in annotation as follows.

- 1 nolu protokolün 1 ihlal edildiğine
- 7 nolu protokolün 4 ihlal edilmediğine

3.2. System Architecture

Input of the model are law cases which consist of sequence of words and output of the model is violated and not violated articles as sequence of words. Since both input and output are consist of sequences and these sequences have different lengths, a model that can convert a sequence to another sequence with different length is needed. Thus, a sequence to sequence model with encoder-decoder architecture [7] and attention mechanism [8] is used to achieve this task. Attention mechanism allows the model to learn which parts of the input are important and which are not, according to current state of the decoder. Thus it improves performance.

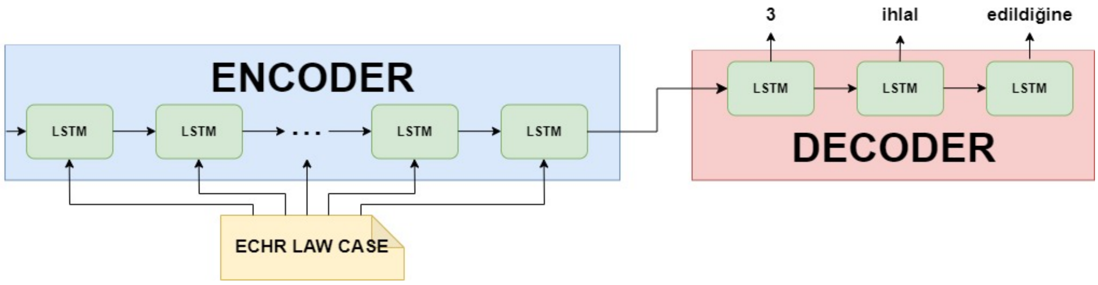

Figure 3.1: Sample of this project on the encoder-decoder architecture

In the following, architecture of model is described in detail.

3.2.1. Encoder

The task of the encoder is converting input sequence to fixed length context vector that contains required information of input sequence to help decoder to make accurate predictions.

In first, sequence of tokens in the input case converted into dense layer with using embedding layer. Then, these embeddings are passed into the LSTM.

In the vanilla encoder-decoder model final hidden state of the LSTM is used as a context vector and the output of the LSTM is ignored. But output of the LSTM is not ignored this project since it is used in the attention mechanism.

3.2.2. Decoder

The task of the decoder is predicting the target sequence by the help of context vector and attention mechanism.

Decoder uses start of sentence token as a first input. This input converted into dense layer with using embedding layer. Then, these embeddings are passed into the LSTM.

After that Luong Attention [8] mechanism is applied with using current hidden state of the LSTM and the output of the encoder. For each encoder state in the outputs(s_k), their relevance to current hidden state of the decoder(h_t), which called attention score is computed.

$$score(h_t, s_k) = h_t^T W s_k$$

Softmax is applied to attention scores to get attention weights(a_{tk}).

$$a_{tk} = \frac{e^{score(h_t, s_k)}}{\sum_{i=0}^{n} e^{score(h_t, s_i)}}, k = 0 \cdots n$$

Attention output(c_t) is computed as weighted sum of encoder states with attention weights.

$$c_t = \sum_{k=0}^{n} a_{tk} s_k$$

Current hidden state and attention output is combined and passed into the linear layer and argmax is used to get predicted output(p_t).

$$p_t = argmax(W_c[h_t, c_t])$$

Then the predicted output is used as an input for next decoder cell and this cycle continue until the predicted output is end of sentence token.

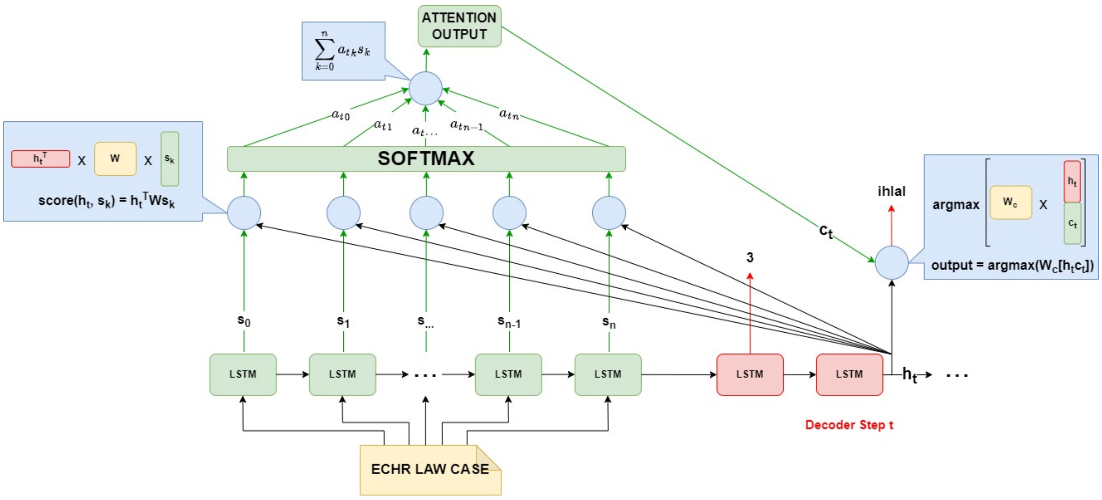

Figure 3.2: Attention mechanism at the decoder step t.

4. EXPERIMENTS

4.1. Dataset

Main dataset consists of Turkish translations of 3000 European Court of Human Rights law cases. 75%:15%:10% split were used for training, validation and testing. All of the case files were preprocessed to remove blank lines and unwanted characters before the training phase to make lower to training time. The dataset was used in different ways during experiments. For some of the models the most organized case files were selected and these 2700 case files were used. For some of the models instead of all of the words in the case files, only words around numbers and ”ihlal” keywords were used for each file.

Table 4.1: Information of every form of the datasets.

Dataset Name	Training Data	Validation Data	Test Data	Number of Tokens
Main	2250	450	300	170289
Best	2032	406	270	152163
Filtered Main	2250	450	300	51184
Filtered Best	2032	406	270	21056

4.2. Experimental Settings

4 different model were trained with different settings. Embedded dimensions of the encoder and the decoder were 256, hidden dimensions of the encoder and the decoder were 512, number of layers in the encoder and the decoder were 2, dropout rates of the encoder and the decoder were 0.5 for all 4 of them. All of the LSTM’s parameters were initialized with the uniform distribution between -0.08 and 0.08. Teacher forcing were used with 0.5 ratio.

For the first model, main dataset and the attention mechanism were used. The model was trained with batch size of 2.

For the second model, best dataset was used and it was trained without attention mechanism with batch size of 4

For the third model, filtered best dataset and the attention mechanism were used. The model was trained with batch size of 4.

For the fourth model, filtered main dataset and the attention mechanism were used. The model was trained with batch size of 4.

4.3. Evaluation Metric

Recall, precision and F1-score of Rouge-1, Rogue-2 and Rogue-3 are used to evaluate models.

$$recall = \frac{number\ of\ overlapping\ n\text{-}grams}{total\ tokens\ in\ reference\ label}$$

$$precision = \frac{number\ of\ overlapping\ n\text{-}grams}{total\ tokens\ in\ predicted\ output}$$

$$F1\text{-}Score = 2 \times \frac{precision \times recall}{precision + recall}$$

4.4. Results

First, the first model was trained with the entire dataset and the attention mechanism.

Table 4.2: First model

Metric	Precision	Recall	F1
Rouge-1	0.41	0.51	0.44
Rouge-2	0.23	0.30	0.25
Rouge-3	0.08	0.11	0.09

After the poor performance of the model on the test set, the most organized cases in the dataset was used in the training of the new model to improve performance on test set. And the batch size was increased from 2 to 4. Attention mechanism was deactivated during training of this model because training with increased batch size and attention mechanism was exceeding the memory limit.

Table 4.3: Second model

Metric	Precision	Recall	F1
Rouge-1	0.74	0.74	0.70
Rouge-2	0.57	0.59	0.54
Rouge-3	0.39	0.41	0.38

The performance increased but it was still not well enough. Instead of using all of the words in the case files, only words arround numbers and ”ihlal” keyword were used training of the new model to improve performance. This filtering process reduced the number of tokens in the dataset considerably. Reducing in the number of tokens simplified the model and made the attention mechanism available without exceeding the memory. The best dataset was used during training since the performance of the second model was better than the first model.

Table 4.4: Third model

Metric	Precision	Recall	F1
Rouge-1	0.95	0.93	0.92
Rouge-2	0.86	0.85	0.84
Rouge-3	0.77	0.77	0.75

The performance of this model was quite good. Since the reduced number of tokens in the dataset speeds up the training process considerably, a new model was trained with the same filtering process and the main dataset.

Table 4.5: Fourth model

Metric	Precision	Recall	F1
Rouge-1	0.98	0.94	0.94
Rouge-2	0.91	0.88	0.88
Rouge-3	0.84	0.82	0.82

And the best performance achieved with this final settings. As can be seen in the table below, the filtering process also significantly reduced the runtimes of the models.

Table 4.6: Average runtimes of the models for one file

Model	Run Time
First	0.149 s
Second	0.126 s
Third	0.015 s
Fourth	0.031 s

5. CONCLUSIONS

In this paper, a system that extracts which articles have been violated and which have not from Turkish translation of European Court of Human Rights law cases with using a sequence to sequence neural network was introduced. In addition, the 3000 law cases were manually annotated in order to train the sequence the sequence model and evaluate the model performance. As a result of poor performances during the experiments, a filtering process was applied to the files before they were fed into the model, thus good performance is achieved.

For future work, more files could be collected and annotated to improve performance, or by working with a lawyer, the annotations could be simplified to better fit the purpose of the project and improve performance.

BIBLIOGRAPHY

[1] M. Mistica, G. Z. Zhang, H. Chia, *et al.*, "Information extraction from legal documents: A study in the context of common law court judgements," pp. 98–103, Dec. 2020. [Online]. Available: https://aclanthology.org/2020.alta-1.12.

[2] E. Leitner, G. Rehm, and J. Moreno-Schneider, "A dataset of german legal documents for named entity recognition," Mar. 2020. DOI: 10.48550/ARXIV.2003.13016. [Online]. Available: https://arxiv.org/abs/2003.13016.

[3] S. A. Shamma, A. Ayasa, W. Sleem, and A. Yahya, "Information extraction from arabic law documents," pp. 1–6, 2020. DOI: 10.1109/AICT50176.2020.9368577.

[4] A. Farzindar and G. Lapalme, "Legal text summarization by exploration of the thematic structure and argumentative roles," pp. 27–34, Jul. 2004. [Online]. Available: https://aclanthology.org/W04-1006.

[5] V. Pandya, "Automatic text summarization of legal cases: A hybrid approach," Aug. 2019. DOI: 10.5121/csit.2019.91004. [Online]. Available: https://doi.org/10.5121%2Fcsit.2019.91004.

[6] C. Xiao, H. Zhong, Z. Guo, *et al.*, "Cail2019-scm: A dataset of similar case matching in legal domain," 2019. DOI: 10.48550/ARXIV.1911.08962. [Online]. Available: https://arxiv.org/abs/1911.08962.

[7] K. Cho, B. van Merrienboer, C. Gulcehre, *et al.*, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," 2014. DOI: 10.48550/ARXIV.1406.1078. [Online]. Available: https://arxiv.org/abs/1406.1078.

[8] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," 2015. DOI: 10.48550/ARXIV.1508.04025. [Online]. Available: https://arxiv.org/abs/1508.04025.

CV

Yusuf Fatih Şişman

4th Year, Computer Engineering Student
Maltepe / Istanbul
yusuf.fatih.sisman@gmail.com (+90)542 714 6151
github.com/YusufFatihSisman

Education

Computer Engineering at Gebze Technical University (2017 - ...)

Experience

Profe Bilgi Sistemleri (Internship August - September 2020)

I completed an internship about mobile application development. During internship I used Dart and Flutter.

Medyasoft (Internship July - August 2021)

The internship was about backend development. During internship I used C# and SQL.

Skills

C, C++, Java, Python, Unity, Common Lisp, Prolog, Flutter

Projects

Boulder Dash Game

I made a clone of Commodore 64 game, Boulder Dash with using C and SDL library.

Manage (Team Project)

We developed an application that facilitates to managing projects with 4 people. I worked at front-end development part of this project. I used Dart and Flutter during work.

Obstacle Detection For Visually Impaired People

I developed a mobile application that detects obstacles and warns the users via auditory informations.

Ray Tracer

I developed a multi thread ray tracer that renders scenes from descriptor files with C++.