Claim my Case: Negative Precedent and the Role of a Lawyer in the Judicial Process

Anonymous EMNLP submission

Abstract

The task of legal judgement prediction is to automatically predict the outcome of a case given the facts of the case. In this paper we observe that in reality judges don't determine the case outcome given the facts alone but are constrained in their decision by legal claims, produced by lawyers. Legal claims are the rights the claimant asserts as violated in front of the court. The job of a judge is to decide which of the claimed rights have actually been violated. Not only are the claims valuable on their own as an essential legal product, distinct from the case outcome, they are important with respect to the existing task of legal judgement prediction, because without a knowledge of claims it's impossible to identify negative outcome, ie. the claims that had been rejected. Since, both positive and negative outcome is equally binding under the doctrine of precedent, current legal AI models are learning only half of the relevant information contained in the case outcome. In this paper we therefore introduce a new task, legal claim prediction, and establish a strong baseline for it on the European Court of Human Rights corpus, which we label with claims for this purpose. We observe that claim prediction is harder for neural models, which achieve only 0.76 F1 micro averaged score on this task compared to 0.80 on the positive outcome prediction task. We further show a vanilla models struggle with predicting the negative outcomes at 0.07 F1. We therefore introduce a new Subtractive Network architecture which achieves over 3x improvement on this task at 0.23 F1 by utilising both claims and outcomes. Thus we conclude that claims are conceptually necessary and empirically useful missing piece of legal AI research.

1 Introduction

While case outcome prediction has been investigated extensively in a number of jurisdictions around the world (Zhong et al., 2018; Xu et al., 2020; Chalkidis et al., 2019a), it has also been simplified to be a task of predicting the outcome of the case given the circumstances of the claimant, i.e. the facts of the case. However, this task is completely artificial, as it never actually occurs as part of the legal process.[1] The judge is never tasked with identifying which law has been breached given the facts. Instead, the job of a judge is to establish which law, provided a set of alleged breaches of law put forth by the claimant, i.e. the party that is making a claim, have actually been breached. The judges are therefore constrained in their decision. On the one hand they can't rule that a law which has not been claimed, has been breached. On the other hand, they have to consider all of the claims put forth to them. Conversely, the lawyers are not trying to second guess a judge in crafting the claims, they are selecting all the laws that could have been reasonably breached by the defendant and do go beyond the most straightforward claims to craft creative arguments on behalf of their clients to persuade judges to rule against the wrongdoer.

Crucially, in precedential jurisdictions, via the

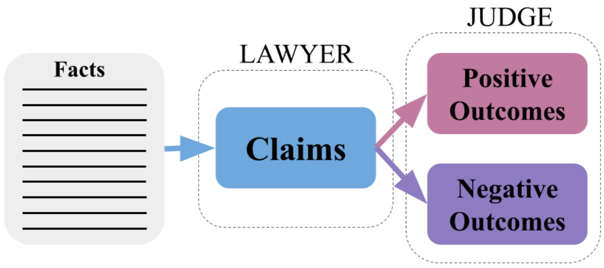

Figure 1: The judicial process can be broken down into two steps. Facts are first turned into claims by lawyers, narrowing down the relevant legal areas. Then, the claims either succeed or fail when assessed by a judge and turn into outcomes.

[1]Thomson Reuters Practical Law provides a comprehensive overview of the legal process: https://uk.practicallaw.thomsonreuters.com/7-502-0631?

doctrine of *stare decisis* (Duxbury, 2008; Lamond, 2016; Black, 2019), this interaction between a judge and a lawyer is responsible for creating new law. A successful claim (positive outcome) extends the scope of the law, while an unsuccessful one (negative outcome) limits it. It is therefore a surprising omission that legal claims remain an unexplored area of legal AI research. Furthermore, under the existing positive outcome classification paradigm, the focus is solely on half of the information legal outcomes contain. Therefore, in this paper we introduce claim prediction as a new text classification task along an annotated corpus of European Court of Human Rights (ECtHR) case law to study both claim and negative outcome prediction task with.

We define the legal claim prediction as a task of predicting the legal claims given the facts of a case. While similar to legal judgement prediction (Aletras et al., 2016; Chalkidis et al., 2019a), which seeks to predict the positive outcomes of cases from the facts, we shift the focus from the service provided by the judge to the one provided by the lawyer. By doing so, this paper presents a study of how to computationally model the core of any legal advice, i.e. the mapping of the client's circumstances to the relevant legal principles.

Furthermore, the addition of this task allows for a more faithful model of a court process. Judges decide on the outcome of a case given both the case facts and the claims. They can't decide on a violation of any other law than that which is given for their consideration in form of the claim. To model the court, claims are an essential component.

In light of legal AI having already been employed for sentencing criminals (Završnik, 2020), we also argue about what should be the practical application of legal models. We argue that a shift of focus from modelling the role of a judge to the one of the lawyer, is crucial to develop models which are beneficial to the legal process.

The contributions of this paper are therefore as follows:

- Annotated version of the European Court of Human Rights (ECtHR) corpus containing Claims and Negative Outcomes.
- Defend lawyer automation as an ethical application of legal AI research.
- Comparative study of Positive Outcome, Negative Outcome and Claim prediction.
- Introduce Subtractive Network, improving on the negative outcome prediction task by 3x over the existing architectures.

2 The Judge and the Lawyer

Lawyers are sought after for two main reasons. They can provide a legal guarantee on a promise, usually in form of a contract, or they can give advice on how to get a compensation for a breach of a right. In this paper we concentrate on the latter skill. In such instances, the lawyer will offer advice on how does the law apply to the claimants circumstances. If there is a potential for the client to be compensated, the client is willing to litigate and the accused party is not willing to settle, a legal case is born.[2]

The legal process can thus be understood as a process of narrowing down the legal space where the alleged breach of law has occurred. Initially, the space includes all the law there is fig. 1. The lawyer then narrows it down to the subset of all the plausible laws. Finally, the judge asserts which of these, if any, have been violated. We can therefore observe two distinct interactions between the real world and the law. First, when a lawyer aligns them in the form of a claim, second when a judge aligns them in the form of an outcome. While the latter has been studied extensively, we are the first to investigate the former. Furthermore, as we explore below, claims play an important role in both.

2.1 The Judge

Positive outcome prediction models the judge, it captures the information about which legal claims succeed. The potential benefit of such model is for estimating a potential of a claim. Because of the high cost associated with a litigation, estimating the chance of a successful claim is extremely valuable. It is also extremely difficult. We argue that the main source of this difficulty however doesn't lie with modelling the law, but rather in modelling human behaviour. Especially with cases which have climbed some way through the hierarchy of a court appeal system, the outcome can often go either way, since both sides of the case have a strong argument.[3]

[2] This is an oversimplification, but it illustrates the general flow of a legal process. For criminal law for example, the alleged crime has been committed against the society at large and is therefore prosecuted by the State on behalf of the people without the option for settlement.

[3] The very nature of appeal is based on this notion.

It's therefore important not to think of outcome prediction as equivalent to some sort of a medical task, such as cancer detection. It would be misleading to assume this is a situation where the "tumour" of breach of law is either there or not. After all, the decision of the court is binding either way the judge decides.

Herein lies the ethical dubiousness of modelling judges. Much like a jury a judge represents the human element in otherwise relatively sanitised legal process. The judge weighs the competing views of the claimant and defendant and makes a decision. She can be persuaded by the specific circumstances that a case should be decided against expectations. In fact, her ability to do this is a crucial requirement for her job. Some judges, such as Lord Denning for example, became famous for their defiance of legal precedent in favour of ruling of what is 'right'.[4] Unlike in the domain of medicine, where identifying the underlying truth is essential for treatment, and thus a machine diagnostician is in theory a competition for the human one, in the domain of law the validity of the decision is poised solely on the best intentions of the judge, who is the source of an underlying truth. We therefore argue a judge should never be replaced by a machine.

Unfortunately, humans are also prone to a bias. There is a considerable evidence that judges do decide cases based on factors such as gender, enthnicity or simply getting bored of the logic used in a previous judgement (Peer and Gamliel, 2013). Indeed, outcome modelling could be used to help with uncovering and perhaps mitigating such biases. However, teaching the model from the existing data isn't perhaps the best way of uncovering the bias within it, see Chang et al. (2019); Mehrabi et al. (2019).

To better model the judge, whether it is to make an informed decision about which claims to raise in front of which judge, or to keep an eye on the judges to ensure they don't abuse their power, it's necessary to consider the claims. Without a claim as an input, we can't properly assess its chances to succeed at a court. Without knowing which claims are judges evaluating, we can't easily tell if they are abusing their power or not. Let's therefore shift legal modelling efforts from outcomes to claims and from a judge to a lawyer.

[4]Margaret Thatcher called Denning: "Probably the greatest English judge of modern times".

2.2 The Lawyer

Claims capture a crucial component of legal advice given to a client by a lawyer. Modelling the relationship between facts and claims could therefore assist legal practitioners and their clients by improving the speed and reducing the associated costs of legal services. For example, before seeking a lawyer for a consultation, a client could use a claim prediction system to learn about the law that might be relevant to their situation. Liberating the access to such information would be beneficial to the rule of law.

But the utility of claims is not limited to a potential claimant. As discussed above, they are important for modelling the court as well. This is especially important for jurisdictions that rely on legal precedent. By the doctrine of precedent, the past decisions of a court is binding on future decisions. This ensures consistency between cases and a certain level of predictability of law. The choice of what is claimed therefore influences what the law becomes regardless of the outcome. If a judge decides a law has been violated a new precedent is born, but equally, when a judge decides a law was not violated, this sets a precedent too. Without knowing the claims the case outcome only encodes half of the precedent as the legal AI model learns only about the successful claims. To study precedential legal systems, claims are therefore an essential component.

Furthermore, we would argue the job of a lawyer is much less prone to a bias against their client. The explicit task in front of a lawyer is to align the law with the circumstances for the benefit of the client. In that sense the lawyer is compelled to work for their client in a way that a judge is not.

In conclusion, while understanding of human nature is necessary for the job of a lawyer, perhaps being a human is not. Therefore, much like the robotic surgeon, the robotic lawyer is at least conceivable.

3 Legal Corpus

We chose to work with the ECtHR corpus, which contains caselaw pertaining to the European Convention of Human rights. To this end we use the same scrape of the HUDOC database as Chalkidis et al. (2019a) to construct our own claims corpus. Since their dataset only contains the case outcomes, we need to extract claims from the case text ourselves. To do this we split each case by its head-

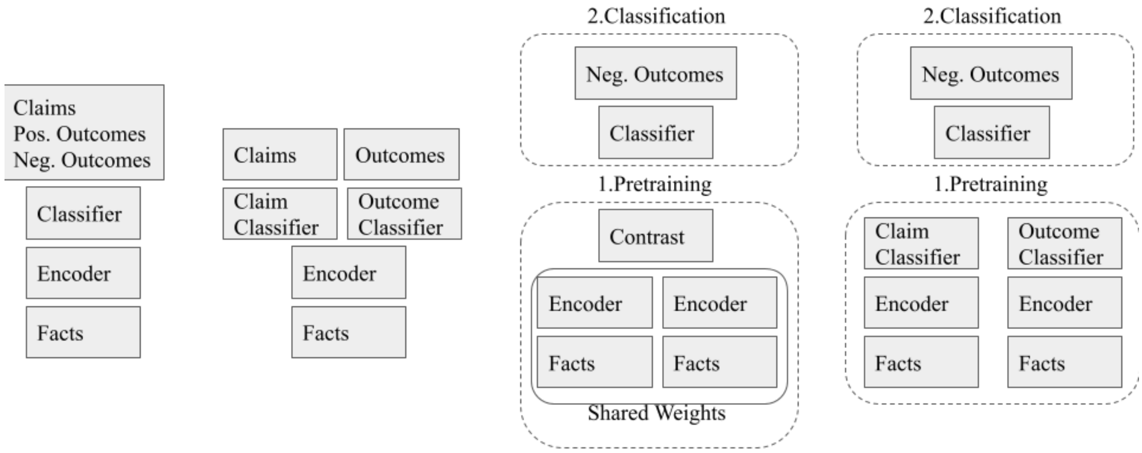

Figure 2: The four architectures considered in this paper. From left to right: vanilla Longformer classifier, MTL classifier, siamese network fine tuned for classification, our own Subtractive Network.

ings. Under the heading of 'THE LAW' we extract the numbers of Article's which the claimant alleges as violated. These form our claim labels. We keep Chalkidis et al. (2019a) annotations for outcome. For negative outcomes, we simply take the difference between claims and outcomes. Since outcomes are subset of claims, this simple method suffices.

To ensure that our extraction process captures the information we are interested in, we manually check the extracted Article numbers. We also remove any cases where the extracted claims are only a subset of extracted outcomes. We choose to focus only on the first 18 Articles of the European Convention of Human Rights, which corresponds to Section 1, containing the core rights of the convention. Additionally we include Article 1 of the first protocol to the convention, relating to protection of property, which is another frequently claimed Article. We have decided to focus on this subset because it contains Articles that applicants to the court are most likely to claim. However, not all of the first 18 Articles are represented in the dataset. Only the following 14 are: 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 18. Together with protection of property this makes a total of 15 Articles we classify into.

From their original corpus of containing $11,000$ cases which are split into $9,000$ training cases and $1,000$ validation and test cases each, we end up with $6,948$ train cases, 773 validation cases and 858 test cases, totalling $8,579$ cases.

4 Claims and Outcomes

To study the relationship between the claims, positive outcomes and negative outcomes we propose the following experiments. The models described below are also contained in fig. 2. The technical details are in section 5.

josef
This is only a template, I'm working on a nice diagram for all the models.

Claims and Outcomes. We train three architectures to compare the claim and positive outcome prediction. First we formalise the task of claim prediction as a standard machine learning classification task and benchmark it using a state-of-the-art Transformer based model. We then define a model of a court where outcome depends both on the facts and claims and investigate if knowing claims can improve outcome classification. Finally, we train a model with a multi task learning (MTL) objective to predict both facts and claims together to investigate if training on both has an effect on the performance of either.

Negative Outcomes. We test four different architectures on negative outcome prediction. First the same Transformer classification model as above to see how hard it is to learn negative outcome prediction compared to its positive counterpart. Second, the same model but trained using a MTL objective on both positive and negative outcome prediction. The intuition is that knowing the positive outcome categories could help learning the negative ones. Third, we propose a siamese network pretrained directly on distinguishing negative outcome cases from the positive outcome cases using a contrastive loss function. Finally we propose a novel Subtraction Network. The Subtraction Network combines the MTL objective, training to predict both claims and positive outcomes, but uses the difference between the claims and outcomes as the prediction of the negative outcomes.

5 Experimental Setup

Notation. Denoting the set of cases in ECtHR corpus as $\mathcal{C}$, we denote each of its elements a c. We further consider four sets of random variables. First, we consider O, a random variable ranging over a binary outcome space $\mathcal{O} = \{0, 1\}^K$, where K is the number of Articles. Instances of $\mathcal{O}$ are o and denote which Articles have been violated. Second, we consider $\mathcal{L}$, a random variable ranging over the same binary claim space $\mathcal{O} = \{0, 1\}^K$. Instances of $\mathcal{L}$ are l and denote which Article has been claimed as violated. Third, we consider $\mathcal{N}$, with instance of n which are the difference between l and o and denote which Article has been unsuccessfully claimed as violated, i.e. the negative outcomes. We also define a binary random variable Y, ranging over the binary space $= 0, 1$. If Y is 1 it denotes two cases, f_i and f_y share a positive outcome o_k and if it is 0, one case has positive outcome o_k, while the other has negative outcome n_k. Finally, we consider $\mathcal{F}$, a random variable that ranges over the space of facts. We denote its instances as f while we denote facts of a case as f_c. We denote the space of all facts as $\mathcal{F} = \Sigma^*$, where Σ is a set of sub-word units and Σ^* is its Kleene closure.

Classification We formulate the outcome and claim prediction as a classification task. Given the facts of a case f we use a deep learning models described below to compute probability of an outcome or claims as follows:

$$p_\theta(o \mid f) = \prod_{k=1}^{K} p_\theta(o_k \mid f) \tag{1}$$

$$p_\theta(l \mid f) = \prod_{k=1}^{K} p_\theta(l_k \mid f) \tag{2}$$

$$p_\theta(n \mid f) = \prod_{k=1}^{K} p_\theta(n_k \mid f) \tag{3}$$

To find the probability of violation of the K Articles we compute:

$$\begin{aligned} \mathbf{h} &= Longformer(f) \qquad (4) \\ p_\theta(\bullet \mid f) &= \sigma(W^{(1)}\,\mathrm{ReLU}(W^{(2)}\,\mathbf{h})) \end{aligned}$$

where $\mathbf{h} \in R^{d_1}$ is a high dimensional representation, $W^{(1)} \in R^{K \times d_2}$ and $W^{(2)} \in R^{d_2 \times d_1}$ are learnable parameters in linear projections, and σ is the sigmoid function.

Multi Task Learning For the MTL objective we use the same Longformer model as above, but we train two separate linear classification layers on top of it simultaneously:

$$\begin{aligned} \mathbf{h} &= Longformer(f) \qquad (5) \\ p_\theta(o \mid f) &= \sigma(W^{(1)}\,\mathrm{ReLU}(W^{(2)}\,\mathbf{h})) \\ p_\theta(l \mid f) &= \sigma(W^{(3)}\,\mathrm{ReLU}(W^{(4)}\,\mathbf{h})) \end{aligned}$$

The Longformer weights are shared between the models, but the parameters $W^{(1)}$ $W^{(2)}$ and $W^{(3)}$, $W^{(4)}$ are distinct and learned separately. Similarly, for negative MTL we train:

$$\begin{aligned} \mathbf{h} &= Longformer(f) \qquad (6) \\ p_\theta(o \mid f) &= \sigma(W^{(1)}\,\mathrm{ReLU}(W^{(2)}\,\mathbf{h})) \\ p_\theta(n \mid f) &= \sigma(W^{(3)}\,\mathrm{ReLU}(W^{(4)}\,\mathbf{h})) \end{aligned}$$

Siamese Networks For our siamese network experiments we use the same Longformer encoder again, but we first pre-train it on distinguishing positive and negative outcomes by optimizing the following contrastive loss function:

$$\begin{aligned} \mathbf{h}_1 &= Longformer(f_i) \qquad (7) \\ \mathbf{h}_2 &= Longformer(f_y) \\ L(\mathbf{h}_1, \mathbf{h}_2, y) &= y \mid\mid \mathbf{h}_1 - \mathbf{h}_2 \mid\mid + \\ &\quad (1-y)\max(0, m - \mid\mid \mathbf{h}_1 - \mathbf{h}_2 \mid\mid) \end{aligned}$$

where m is the margin, which enforces a minimal distance placed between positive and negative outcome cases. We then add a classification layer on top of the Longformer and train the model same as above.

Subtraction Network Finally, we train a MTL network with three objectives:

$$\begin{aligned} \mathbf{h} &= Longformer(f) \qquad (8) \\ p_\theta(o \mid f) &= W^{(1)}\,\mathrm{ReLU}(W^{(2)}\,\mathbf{h}) \\ p_\theta(l \mid f) &= W^{(3)}\,\mathrm{ReLU}(W^{(4)}\,\mathbf{h}) \\ p_\theta(n \mid f) &= \sigma(l - \mid\mid o \mid\mid) \end{aligned}$$

The loss is computed with cross entropy over all three objectives by adding individual losses for positive outcome, negative outcome and claims predictions and dividing them by 3.

6 Implementation Details

We use the Longformer language model as the starting point for training all our classifiers (Beltagy et al., 2020). The Longformer is built on the same Transformer (Vaswani et al., 2017) architecture as BERT (Devlin et al., 2019), but it can process up to 4,096 tokens. Since the facts of legal documents go often well beyond 512 tokens, having the ability to encode the full document without need for pooling is crucial. This way, the global attention in our implementation can attend to all the word-pieces in each document.

Since this architecture achieves state-of-the-art performance in tasks similar to ours, e.g. on the IMDB sentiment classification (Maas et al., 2011) and Hyperpartisan news detection (Kiesel et al., 2019). We believe it to be a good choice for the tasks tackled in this paper. Naturally, the encoder is interchangeable and different architecture candidates are available. For example the BigBird (Zaheer et al., 2020) or Perciever (Jaegle et al., 2021) could be interesting candidates for future investigation.

To be able to train the models using our limited resources we set the models' hidden size to 50 and batch size to 16. We truncate individual cases to 4,096 tokens, the Longformer maximum size. Our Longformer models are implemented using the Pytorch (Paszke et al., 2019) and Hugginface (Wolf et al., 2020) Python libraries. We train all our models on 4 Nvidia P100 16GiB GPU's for a maximum of 12 hours using Longformer-base model.

With the exception of our vanilla siamese model pre-training step, our our models are trained using cross-entropy.

Model	Precision	Recall	F1
Claim	0.87	0.67	0.76
Outcome	0.82	0.77	0.80
MTL Claim	0.86	0.65	0.74
MTL Outcome	0.83	0.76	0.80
Outclaim	0.84	0.76	0.80

Table 1: Longformer performance on predicting the claims and positive outcomes given the facts alone, multi task learning to predict both using facts and modelling the court outcome using both claims and facts as an input.

7 Results

The bulk of our results can be found in table 1 and table 2. We discuss them in detail below. All results are reported with F1 micro, we measure significance using the two tailed paired permutation tests with $p < 0.05$.

7.1 Claims vs. Outcomes

We find that classifying claims is harder than classifying outcomes. Our best model achieves only 0.76 F1 micro average score on claim prediction, compared to the 0.80 for the outcome prediction, which is 0.04 in terms of absolute difference, see table 1.

We have originally anticipated the claims to be easier to predict. Our rationale was that since in our formalisation the task of a Lawyer is to identify all the law that could interact with the circumstances at hand, while the Judge needs to figure out if a given breach actually occurred, arguably deeper level of legal understanding, the task of the Lawyer would be easier. In reality, we found that Lawyers invent creative ways of connecting facts with law which our models simply cannot replicate. This can be observed when we compare the predicted Claims to the actual Claims made by lawyers. Our model consistently under-claims, when compared to its human counterpart, see Recall in table 1.

Another reason is that the Judges have an incentive to keep more consistency in deciding their outcomes. Distinguishing a case from previous law threatens reliability in precedential legal system. In retrospect, it's therefore not that surprising that we can observe the drop in performance from positive outcomes prediction to claims prediction.

7.2 Claims and Outcomes

Our experiments with MTL objective have lead to no significant increases in F1 for claims and positive outcomes. In fact, the performance for claims

Model	Precision	Recall	F1
Classifier	0.50	0.04	0.07
MTL Classifier	0.38	0.07	0.12
Siamese	0.31	0.01	0.02
SubNet	0.27	0.20	0.23

Table 2: Longformer performance on predicting the negative outcomes using facts alone, with MTL objective, using a contrastively pretrained siamese network and with our Subtraction Network.

decreases to 0.74. However, the MTL objective seems to benefit outcome precision, improving it by 1%. This improvement is counterbalanced by the reduction in recall. Similarly, using claims as an input to the model (Outclaim) improves the precision on outcome prediction further by another 1%. However the F1 remains constant throughout at 0.80. This is interesting because it looks like it's not significantly easier to learn the case outcomes, knowing the case claims.

Why is that? Knowing the claims should help if the model has trouble narrowing down the space of Articles where the potential breach of law could have occurred. But our models don't seem to struggle with this. We can see this by calculating how well can one predict outcomes by using claims directly. This is effectively simulating the human lawyer as an outcome prediction model. We find that at 0.79 F1 this human baseline still 'underperforms' our model on outcome prediction. How is that possible? The answer is simple, human lawyers are not trying to merely predict the outcomes of cases. They try to identify all the possible claims to benefit their client and argue them as well as they can. That's indeed what the clients pay them for.

needs expansion and clarification

While the F1 improvements are minimal, the precision increases are less negligible. Our results imply that the MTL and Outclaim setup could be beneficial for tasks where precision is particularly important. For example in a neural ranking model, where the typical evaluation is done on mean averaged precision (MAP) or precision@k (P@k).

add experiments with ordering

7.3 Negative Outcomes

Need to expand

While the relationship between claims and positive outcomes seems to have limited practical use for enhancing the classification of either, the importance of training on both becomes apparent for the task of negative outcome prediction.

The same classifier that achieves 0.80 F1 on positive outcome prediction performs incredibly poorly on predicting negative outcomes at 0.07 F1 micro. This is more than an order of magnitude worse than for positive outcomes! We believe that this is due to the fact that it's very hard to learn which Articles don't apply if you don't know which ones do apply.

The MTL objective improves the performance on this task by over 70%, to 0.12 F1. We believe that this improvement is thanks to the model receiving information about which Articles relate to the facts of the case as well as which don't. While indirectly, the embeddings generated by the shared Longformer encoder provide significant advantage for classifying negative outcomes.

However, the issue with the MTL objective is that it only provides this information indirectly in terms of the outcome. This is why we have used the siamese network setup, where we can train directly on distinguishing the negative and positive outcomes. Unfortunately, we find the siamese network fails to generalise for negative outcome prediction and achieves the lowest performance at 0.02 F1.

On the other hand, we achieve the best performance with our Subtractive Network (SubNet) architecture at 0.23 F1, over a three times improvement on the vanilla classifier. The advantage here is that unlike with the siamese network the SubNet knows about both claims and positive outcomes and that it is the difference between the two that reveals negative outcomes.

8 Related Work

Juris-informatics can trace its origins all the way to the early 1960's (Kort, 1957; Nagel, 1963). The pioneers have used rule based systems to successfully capture aspects of legal reasoning in thousands of hand crafted rules (Ashley, 1988). Yet these systems have been too brittle to be employed in practice due to the ever changing rules of law. Especially in common law countries, majority of law is contained in case law, where cases are transcripts of the judicial decisions, making the law change constantly with every new decision. With the advances of natural language processing (NLP) in the past two decades, the interest in developing NLP applications for the legal domain have been rejuvenated by the research aiming towards more robust models of law.

Aspects of legal reasoning which have been explored include question answering (Monroy et al., 2009), legal entity recognition (Cardellino et al., 2017), text summarisation (Hachey and Grover, 2006), judgement prediction (Xu et al., 2020) and majority opinion prediction (Valvoda et al., 2018).

Our work is in particular similar to the recent research on Chinese law judgement prediction by (Zhong et al., 2018) and (Xu et al., 2020), who break down the court judgement into the applicable law, charges and terms of penalty. Operating in the civil law system (Germany, China, France etc.)

they argue that predicting applicable law is one of the fundamental subtasks, which will guide the prediction for other subtasks. In the context of ECHR law, we argue that the legal claim prediction is one such fundamental subtask as lawyers develop claims as part of their legal advice.

However, legal claims are not part of the research above as claims are not available in the Chinese law datasets. Yet, that doesn't make claim prediction a unique task for the ECHR law. It is as much a part of the ECHR system as any other adversarial legal system. ECHR cases do conveniently provide this information as part of the judgement, but in most legal systems, the information will be submitted directly to the court and to the other party.

The ECHR dataset has been collected by (Chalkidis et al., 2019b), who have predicted binary outcomes of the ECHR law and the corresponding articles using neural architectures. Our work builds on their research by focusing on the prediction of the claims which precede the judgement outcome.

9 Conclusion

We have conducted the first study to investigate the relationship between legal claims and outcomes. We argue that replacing a human judge is inconceivable even if a legal model could match a human judge on any existing metric. On the other hand, we contend that replacing a lawyer is at least conceivable, though by no means we recommend the use of the current legal AI models, including ours, towards this end.

Our experiments reveal that there is a substantial difference in performance for claim and outcome prediction. Claims, it turns out, are harder to predict. In light of our discussion on the merits of each, the claim prediction should therefore be the focus on future research not only because its potential utility, but because it poses more of a challenge compared to its well established sibling.

We have investigated if MTL could lead towards improvement for either task and have found that while the overall effect on performance is negligible, there is a small benefit of using this method on improving precision of the models for outcome prediction. This could be useful for example for document retrieval tasks, where higher precision is desirable.

Finally we have formulated the task of negative precedent prediction. Inspired by the legal process we propose a novel architecture to tackle this problem and improve over other classification methods (which are otherwise very successful on claim and positive outcome prediction tasks) by over three times.

None the less there are two problems arising from our work worth further investigation. They can be conveniently described by the performance gaps described above. First, further work should seek to close the gap between claim prediction and outcome prediction tasks. Second, and we believe more important issue, is that more research is necessary to close the performance gap between the positive and negative outcome prediction.

References

Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preoţiuc-Pietro, and Vasileios Lampos. 2016. Predicting judicial decisions of the european court of human rights: a natural language processing perspective. *PeerJ Computer Science*, 2:e93.

Kevin D Ashley. 1988. Modelling legal argument: reasoning with cases and hypotheticals.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Henry Black. 2019. *Black's Law Dictionary*, 11th edition. Thomson Reuters.

Cristian Cardellino, Milagro Teruel, Laura Alonso Alemany, and Serena Villata. 2017. Legal NERC with ontologies, Wikipedia and curriculum learning. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 254–259, Valencia, Spain. Association for Computational Linguistics.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019a. Neural legal judgment prediction in English. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4317–4323, Florence, Italy. Association for Computational Linguistics.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019b. Neural legal judgment prediction in english. *arXiv preprint arXiv:1906.02059*.

Kai-Wei Chang, Vinod Prabhakaran, and Vicente Ordonez. 2019. Bias and fairness in natural language processing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): Tutorial Abstracts*, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Neil Duxbury. 2008. *Distinguishing, overruling and the problem of self-reference*, page 111–149. Cambridge University Press.

Ben Hachey and Claire Grover. 2006. Extractive summarisation of legal texts. *Artificial Intelligence and Law*, 14(4):305–345.

Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. 2021. Perceiver: General perception with iterative attention.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. 2019. SemEval-2019 task 4: Hyperpartisan news detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 829–839, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Fred Kort. 1957. Predicting supreme court decisions mathematically: A quantitative analysis of the "right to counsel" cases. *American Political Science Review*, 51(1):1–12.

Grant Lamond. 2016. Precedent and Analogy in Legal Reasoning. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, spring 2016 edition. Metaphysics Research Lab, Stanford University.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2019. A survey on bias and fairness in machine learning.

Alfredo Monroy, Hiram Calvo, and Alexander Gelbukh. 2009. Nlp for shallow question answering of legal documents using graphs. volume 5449, pages 498–508.

Stuart S Nagel. 1963. Applying correlation analysis to case prediction. *Tex. L. Rev.*, 42:1006.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8026–8037.

Eyal Peer and Eyal Gamliel. 2013. Heuristics and biases in judicial decisions. *Ct. Rev.*, 49:114.

Josef Valvoda, Oliver Ray, and Ken Satoh. 2018. Using agreement statements to identify majority opinion in UKHL case law. In *to be published in proceedings of JURIX 2018 conference*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Nuo Xu, Pinghui Wang, Long Chen, Li Pan, Xiaoyan Wang, and Junzhou Zhao. 2020. Distinguish confusing law articles for legal judgment prediction. *arXiv preprint arXiv:2004.02557*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33.

Aleš Završnik. 2020. Criminal justice, artificial intelligence systems, and human rights. In *ERA Forum*, volume 20, pages 567–583. Springer.

Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. 2018. Legal judgment prediction via topological learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3540–3549, Brussels, Belgium. Association for Computational Linguistics.