Argumentation and Judgement Factors: LLM-based Discovery and Application in Insurance Disputes

Basit Ali[*], Anubhav Sinha, Nitin Ramrakhiyani,
Sachin Pawar, Girish K. Palshikar[*], Manoj Apte
TCS Research, Tata Consultancy Services Limited, India.
{s.anubhav2, nitin.ramrakhiyani, sachin7.p, manoj.apte}@tcs.com
{alibasit78, girishpalshikar}@gmail.com

Abstract

In this work, we focus on discovery of legal factors for a specific case type under consideration (e.g., vehicle insurance disputes). We refer to these legal factors more explicitly as "Argumentation and Judgement Factors" (AJFs). AJFs encode specific legal knowledge that is important for legal argumentation and judicial decision making. We propose a multi-step approach for discovering a list of AJFs for a given case type using a set of relevant legal documents (e.g., past judgements, relevant acts) and Symbolic Knowledge Distillation (SKD) from a Large Language Model (LLM). We propose a novel geneRatE-CRitic-reviEW (RECREW) prompting strategy for effective SKD. We construct and evaluate the discovered list of AJFs on two different types of cases (*auto-insurance* and *life insurance*) and show their utility in a dispute resolution application.

1 Introduction

Legal argumentation (Feteris, 2018; Walton, 2012) and judicial decision making (Lovegrove, 1989; Friedman et al., 2020; Bystranowski et al., 2022) are very complex tasks, requiring a wide spectrum of knowledge and skills. Both tasks require creation, analysis, comparison and critique of (i) the presented evidence, (ii) presented witness testimonies, (iii) cited prior cases, (iv) knowledge of specific legal terms used and sections/sub-sections of relevant statutes/acts/regulations invoked (v) arguments presented by both sides and so forth. Apart from legal knowledge, general knowledge of the domain and understanding of domain-specific documents (e.g., insurance policies) are also required.

An important basis for legal argumentation and judicial decision making consists of the various "legal factors" relevant to the case at hand. In this paper, we focus on "Argumentation and Judgement Factors" (AJFs) (Table 1), which are essentially *legal factors* well known in legal NLP literature (Westermann et al., 2019; Ashley, 1990; Aleven, 2003; Gray et al., 2024). We propose to refer them as AJFs mainly because (i) unlike the generic term "legal factors", the term AJF includes both the aspects of argumentation and judgement giving the right context, and (ii) the more explicit naming helps a language model understand the desired semantics of the factors, better.

Computationally, AJFs are linguistic terms and phrases that form a basis of both legal argumentation and judicial decision-making. On these lines, we propose NLP techniques to automatically extract an initial seed set of AJFs from a given set of relevant documents, such as past cases, legal statutes, domain-specific documents, etc. Using these extracted seed AJFs, we expand them further using symbolic distillation from an LLM. We believe that the proposed AJFs can form basis of a useful, human-understandable and editable legal knowledge-base (KB) which can be queried, searched and referred in legal applications such as argumentation assistance and legal decision prediction.

It is important to highlight the AJFs in light of existing literature on legal factors. Traditional works such as HYPO and CATO (Ashley, 1991, 1990; Aleven, 2003) propose legal factors as aspects designed to model legal reasoning in the domain of US trade secret law. The works also state that the factors are stereotypical factual patterns that strengthen or weaken a side's position. In another work (Westermann et al., 2019), the authors use a fix set of manually identified factors in tenant-landlord disputes. However, these factors do not have a *tilt* towards any side, different from the CATO's and HYPO's treatment of the factors. In our case, firstly we are automatically discovering such factors for a case type (e.g. vehicle insur-

[*] Work done while working at TCS Research

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 2789–2804
March 24-29, 2026 ©2026 Association for Computational Linguistics

	Basic AJF	**Complex AJF**
Definition	Atomic base concepts in the domain of the case type which are important considerations in legal arguments and judicial decisions	An event or a scenario involving one or more basic AJFs which inherently favours one of the contesting parties
Syntax	Noun phrase	Verb phrase or noun phrase
Semantics	• Key concept or entity for a case type • Generally corresponds to a physical object, a document, a process, a role, a neutral event, etc.	• Corresponds to an event or scenario • Favours one of the contesting parties
Examples	• `insurance policy` • `ignition key` • `stolen vehicle` • `accident`	• ⟨`original ignition key of the stolen vehicle was not submitted`, *favours*, *insurance company*⟩ • ⟨`the stolen vehicle was parked at a safe location`, *favours*, *insured party*⟩ • ⟨`carelessness of the driver`, *favours*, *insurance company*⟩

Table 1: Basic and Complex AJFs - Description and Examples

ance or life insurance) with a generic technique which can work on any other domain and scale better than manual factor identification. Secondly, we distinguish the legal factors into the two types – basic AJFs and complex AJFs (Table 1). Basic AJFs are those key concepts relevant to the given case type which are influential in argumentation and decision making, but do not explicitly indicate any tilt towards a specific side. On the other hand, the complex AJFs which are designed to indicate a tilt towards a specific side. Moreover, this division into basic and complex AJFs provides a more structured and clearer view of legal factors than considered previously in the existing literature. Syntactically, basic AJFs are generally noun phrases (`policy premium`) whereas complex AJFs can be verb or noun phrases both (`delay in filing insurance claim` or `insurance claim filing was delayed`). Semantically, basic AJFs are "concepts or entities" whereas complex AJFs are "events or scenarios". A recent work in literature that comes closest to our proposed ideas is by Gray et al. (2024) where the authors also employ LLMs to automatically discover legal factors, however they neither make any distinction into basic or complex factors nor consider any tilt towards either side in the factors.

In this paper, we focus on two types of insurance disputes – vehicle insurance and life insurance, where the two contesting parties are an *insurance company* and the *insured party*. We discover the AJFs for each case type starting from a set of domain specific documents (judgements, insurance policies, etc.) The construction process involves discovery of basic AJFs using information extraction from these documents followed by Symbolic Knowledge Distillation (SKD) from an LLM. Further, another step of SKD helps discover complex AJFs based on the basic AJFs gathered in the previous step. We evaluate the quality of the discovered AJFs and also validate their utility through a real-life application. The specific contributions of the paper are listed as follows:

- We propose a Symbolic Knowledge Distillation (SKD) based approach to discover Basic and Complex AJFs from LLMs, using a novel geneRatE-CRitic-reviEW (RECREW) prompting strategy (Section 2).
- We carry out the AJF discovery for two different case types using two LLMs and provide detailed *intrinsic* evaluation results for all four LLM-case type combinations (Section 3.5).
- As part of the *extrinsic* evaluation of the discovered AJFs, we show their effectiveness in an application on dispute resolution where the objective is to identify the stronger party in a dispute. The application uses a Retrieval Augmented Generation (RAG) (Lewis et al., 2020) based technique using the list of discovered AJFs for constructing the context (Section 3.7).

2 Automatic AJF Discovery

We propose a multi-step approach to discover basic and complex AJFs as shown in Figure 1.

2.1 Step 1: Discovering Candidate Basic AJFs

In this step, we discover candidate basic AJFs which are generally atomic noun phrases and correspond to various types of concepts such as physical objects, documents, pieces of evidence, processes,

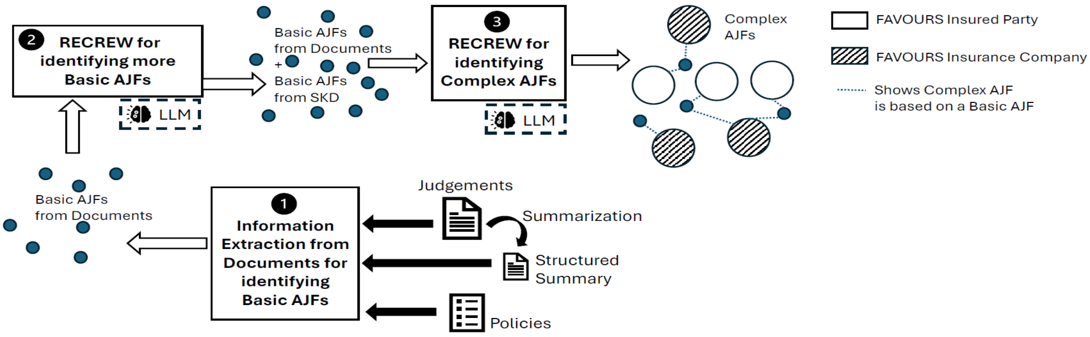
2 RECREW for identifying more Basic AJFs
LLM
Basic AJFs from Documents
+
Basic AJFs from SKD
3 RECREW for identifying Complex AJFs
LLM
Complex AJFs
FAVOURS Insured Party
FAVOURS Insurance Company
Shows Complex AJF is based on a Basic AJF
Basic AJFs from Documents
1 Information Extraction from Documents for identifying Basic AJFs
Judgements
Summarization
Structured Summary
Policies

Figure 1: Steps for AJF discovery for a particular case type

and roles. We consider an input text corpus which comprises of judgements, their LLM generated summaries (more details in Section 3.2) and relevant statutes such as acts, regulations and policies. Firstly, we use a zero-shot entity extraction technique GLiNER (Stepanov and Shtopko, 2024) for extracting entities of the following types from the input text corpus – *object*, *concept*, *document*, *organization*, *role*. Some examples of extracted entities in vehicle insurance are – `motor vehicle` (*object*), `fundamental breach` (*concept*), `driving licence` (*document*), `insurance company` (*organization*), `Regional Transport Officer` (*role*). We observed that the list of GLiNER based extraction had many spurious entities which may not be valid concepts for the case type's domain. To alleviate this, we apply two stages of filtering:

Basic filtering: As part of this filtering, we remove entries which are either too specific to a particular dispute such as names of people or organizations or are too generic and used widely in the legal domain such as the legal roles (e.g. `judge`, `attorney`) or legal entities (`court`, `tribunal`). Specifically, two kinds of filtering are carried out – (i) list/NER based filtering, and (ii) document frequency based filtering. Details about these filtering steps are presented in Appendix A.

Perplexity based filtering: Pawar et al. (2024), in a text classification setting, propose use of conditional perplexity of a text given some context, as a measure of the text's association with the context. We extend the same idea to check plausibility of a candidate entity with respect to a context from the domain of our interest vs. its plausibility with respect to a general context. More specifically, for any candidate phrase p, we compute the following score:

$$score(p) = \frac{PPL_M(p|c_{domain})}{PPL_M(p|c_{general})} \tag{1}$$

where $PPL_M(p|c_{domain})$ is the conditional perplexity of the phrase p using a Small Language Model (SLM) M with respect to c_{domain} which is a small paragraph describing the domain of interest. We explored multiple decoder-only transformer-based SLMs for this task and found Fox-1-1.6B[1] to be qualitatively better. For example, for vehicle insurance domain, we use c_{domain} = *Vehicle Insurance disputes arise between an insured and an insurance company. An insured has bought an insurance policy for his vehicle from the insurance company. An important concept related to such vehicle insurance disputes is....* On the other hand, $PPL_M(p|c_{general})$ corresponds to the conditional perplexity of the phrase p with respect to the general context $d_{general}$ = *Various types of disputes arise between two parties. An important concept related such disputes is.* Here, if $score(p) < 1$ then the phrase p is more likely to associated with the domain of interest. This perplexity based factor filtering is particularly valuable for our work, as it excels in domain-specific classification where labeled data is scarce but high reliability is required.

Finally, to remove duplicates, we cluster the entities retained after the above filtering, using agglomerative clustering with complete linkage based on OpenAI embeddings (text-embedding-3-large). We have chosen the distance thresholds heuristically by visualizing the dendrograms. For each cluster, we retain only its longest length member as the cluster representative which enters the final list of basic AJFs.

RECREW: geneRatE-CRitic-reviEW prompting strategy

LLMs trained on huge text corpora have captured immense lexical and semantic knowledge in the

[1] `https://huggingface.co/tensoropera/Fox-1-1.6B`

text of their training corpora. Also, the scale and domain coverage of the training corpora is larger than any specific domain's corpus allowing LLMs to have an abstract general insight at questions in several domains. We believe that the set of judgements and statutes that we consider, constitutes only a tiny sample of the universe of discourse on any case type. This encourages us to use LLMs to further the discovery of basic AJFs and even discover complex AJFs from scratch. Symbolic Knowledge Distillation (SKD) (Acharya et al., 2024; West et al., 2022) of LLMs is a process to extract symbolic and human comprehensible knowledge from LLMs and we use SKD to enable discovery of new legal factors.

We propose a novel three step method to elicit knowledge about legal factors from an LLM. The method geneRatE-CRitic-reviEW (RECREW), combines few shot in-context learning with a consistency check and reflexion (Shinn et al., 2023) based regeneration for the task at hand. We detail each of the three steps in detail as follows:

- **Generate**: To kick-start generation of the AJFs, we devise a prompt to consider hand-picked valid AJFs as instances for few-shot in-context learning. This allows for the LLM to understand the task and perform generation.
- **Critic**: We can observe that all generations emitted as part of the **Generate** step may not be correct or as desired. This step performs the function of a critic which checks each of the generated AJFs and segregates them as correct or incorrect, along with suitable justification.
- **Review**: We hypothesize that the generations rejected by the critic can be corrected using the justification generated by the Critic and can become part of the set of valid AJFs. This step performs exactly this task of review and correction of the critic reported incorrect generations. It uses a variant of the **Generate** style prompt which combines the definition and dynamically chosen few-shot examples semantically similar to the factor being corrected. Further, the instruction changes from requesting raw generation to correction.

It is important to point out that the **Critic** step in the RECREW procedure is aimed at increasing the *precision* of the SKD process as it is tasked with filtering possible false positives from the Generate step. Similarly, we can note that the **Review** step is aimed at increasing the *coverage* of the SKD process as it is tasked with correcting Critic-rejected generations, hence leading to better recall.

2.2 Step 2: RECREW for Basic AJFs

In addition to the basic AJFs extracted from the different document sources, we use LLMs as another source for obtaining additional basic AJFs through the RECREW method. To create the final list of basic AJFs, we first include all basic AJFs obtained as part of Step 1 (extracted from text corpus) to the final list. We then merge the RECREW based basic AJFs to the final list, wherein (i) if a RECREW basic AJF is similar to an extraction basic AJF (cosine similarity match > 0.90) then we ignore that RECREW basic AJF; (ii) in an otherwise case, we add the RECREW basic AJF to the final list. We use a consistency checking type prompt ("Yes" or "No") for the **Critic** step to check if the generated candidate basic AJF is valid or not. Appendix B shows the prompts used for each of the three RECREW steps for eliciting new basic AJFs, particularly for the vehicle insurance domain. It is important to note that the prompts are generic in nature and can be easily ported to another domain with few replacements in the system prompt and instructions. An example of the generate-critic-review cycle is shown in Step 2 of Table 3, where a basic AJF – `offender` is elicited at the generate step, but is discarded by the critic step. However, at the review step, it is corrected to `at-fault driver` which is eventually accepted in the second critic step.

2.3 Step 3: RECREW for Discovering Complex AJFs

It is important to note here that the defining structure of complex AJFs comprises of two parts – (i) an event / scenario related to a basic AJF and (ii) a favouring relation to one of the contesting parties. This complex structure is not directly extract-able from the text documents and hence, we directly resort to the SKD based RECREW approach on the LLM for complex AJF discovery. Appendix C shows the prompts used for each of the three RECREW steps for eliciting complex AJFs. It is important to note that in the **Critic** step, we use a conditional generation style prompt where we expect the LLM to generate the favouring party ("insured party" vs "insurance company" vs "Neutral") given the event/scenario generated during the **Generate** step. If the LLM outputs "Neutral" it signals that there is lesser confidence in the AJF

	VID	LID
# cases where the “insured party” wins	69 (66.3% cases)	38 (53% cases)
# cases where the “insurance company” wins	35 (33.7% cases)	34 (47% cases)
Average number of sentences per dispute description	78.9	121.4
Standard deviation of number of sentences	33.8	40.9

Table 2: Detailed Dataset Statistics

Step 1: Discovering Candidate Basic AJFs **From text corpus**: `survey report`, `surveyor`, `driving licence`, `claim form`, `ignition key`, `police record`, `impugned order`, `judgment`, `petitioner`
Step 2: RECREW for Basic AJFs **Accepted by Critic (after generate)**: `subrogation rights`, `transfer of insurance policy` **Accepted by Critic (after review)**: `at-fault driver`, `vehicle safety features`, `passenger injuries` **Discarded by Critic**: `offender`, `written information`, `individual persons`
Step 3: RECREW for Complex AJFs **Accepted by Critic (after generate)**: ⟨`notification of theft was documented with law enforcement`, *favours*, *insured*⟩ **Accepted by Critic (after review)**: ⟨`survey report indicated damages exceeding the insured's initial claim`, *favours*, *insurance company*⟩ **Discarded by Critic**: ⟨`delay in obtaining a free certified copy of the policy`, *favours*, *insurance company*⟩

Table 3: Illustration of the proposed step-wise AJF discovery through examples

and this needs to be sent for another look to the **Review** step. Otherwise, the favouring party identified by the **Critic** step prompt is used as part of the final accepted AJF. Here the intuition is that as the critic is performing a simpler *discriminative* task of identifying the favouring party for a given event/scenario (as compared to the more complex *generative* task of generating the complete pair of the event/scenario and favouring party), it is more likely to be correct.

3 Experimentation and Evaluation

3.1 Dataset

In this paper, we gather two datasets of disputes, one in the domain of vehicle insurance and another in the domain of life insurance. As discussed earlier, we use initial documents mainly from two sources – *judgements* and *statutes*. For judgements, we collected 1117 insurance disputes from the National Consumer Disputes Redressal Commission (NCDRC), India which is a consumer court in India. The disputes were downloaded from the ConfoNet project website[2] by selecting the sector as “Insurance” for the years 2022, 2023, and 2024. We retained only vehicle and life insurance disputes by removing disputes belonging to other type of insurance (e.g., property or health). We also discarded the disputes where there was no clear winning party so that the focus remains on the disputes where the human judgement was clearly decisive. Finally, we ended up with the dataset of **104** vehicle insurance disputes (VID) and **72** life insurance dispute (LID) judgements. The statistics about both the datasets are detailed in Table 2. For statutes, the second source of documents, we relied on two representative insurance policies for private (IRDAI, 2025d) as well as commercial vehicles (IRDAI, 2025a) in case of vehicle insurance disputes. Similarly, in case of life insurance disputes we considered two representative life insurance policies (IRDAI, 2025b,c).

3.2 LLM-based summarization of dispute judgments

Though all the dispute judgements roughly follow the same outline, they are quite unstructured and long. We identified a set of important commonly occurring *structural elements* of any dispute and got each dispute judgement summarized such that all of them follow the same *standardized structure*, following Pawar et al. (2025).

We use this final structured summary as one of the inputs to the AJF discovery approach (Figure 1) and also as part of the extrinsic evaluation described in Section 3.7. Following are these structural elements considered for both VID and LID datasets:

i. Facts agreed by both parties,

ii. Aspects on which the parties disagree,

iii. Demands of the insurance company,

[2] `https://cms.nic.in/ncdrcusersWeb/search.do?method=loadSearchPub` (accessed on 30-APR-2024)

iv. Demands of the insured party,

v. Arguments of the insurance company,

vi. Arguments of the insured party,

vii. Relevant prior cases referred,

viii. Relevant statutes or policy terms and conditions referred,

ix. Final decision by the National Commission along with justification,

x. Winning party

3.3 Baseline Approach

We consider the work by Gray et al. (2024) as a baseline for our work since it also focuses on extraction of legal factors by leveraging LLMs. However our work differs in 3 key ways:

i) Their approach relies on raw court opinions for extracting legal factors using LLMs, which restricts the scope and coverage of factors to what is present in the legal documents. In contrast, we employ distillation over LLMs to discover factors by probing into an LLM's world knowledge, thereby broadening the coverage and scope of our factors beyond the set of documents itself.

ii) Their approach involves relying on human expertise to obtain the final refined list of factors whereas our method relies on an LLM through the three-step SKD procedure – RECREW, for this task.

iii) Their work involves only an intrinsic evaluation wherein they compare the extracted factors with human labeled gold-standard factors. However, our work evaluates the quality of the discovered factors both intrinsically as well as extrinsically in the context of real-life judicial decision-making.

In Appendix G, we provide the detailed procedure used to implement the baseline approach along with the specific prompts. We applied the baseline to both the VID and LID datasets and conducted an intrinsic evaluation similar to the one we do for factors obtained from our proposed approach and report the results in Table 4.

3.4 Experimental setup

For all our experiments, we used the `gpt-4o-mini`[3] as the LLM for symbolic knowledge distillation (SKD) due to our limited budget. For the critic step, we use a temperature setting of 0 and for the generate and review steps, we use a temperature setting of 0.7. For text embeddings, we use OpenAI's `text-embedding-3-large` model[4] that are used for clustering of AJFs. We also try the proposed AJF discovery with an open source small language model – LLama-3-8B-instruct[5] for comparison. For more details, refer to Appendix E.

3.5 Intrinsic Evaluation

In this section, we discuss the intrinsic evaluation of the automatically discovered AJFs following the steps described in Section 2. Table 4 shows the number of basic AJFs and complex AJFs extracted and discovered at each step of the process, using the gpt-4o-mini model. Due to the large number of AJFs, it is not possible to evaluate all of them manually for correctness. Hence, for the intrinsic evaluation, we randomly select K AJFs for evaluation by human experts and report the corresponding sample precision $P|_K$. We also perform fine-grained evaluation of each step in the RECREW process by selecting the random samples for each of the step shown in Table 4. Each individual element in a random sample is verified by two human experts who follow the guidelines below:

- A Basic AJF is marked as *correct* if it is perceived to be an important concept useful in argumentation and/or judicial decision making in an insurance dispute/case.
- A Complex AJF is marked as *correct* if it is perceived to be an important event/scenario useful in argumentation and/or judicial decision making as well as it should be *realistic* and favouring the right party.

The precision numbers in both datasets indicate that Complex AJFs are discovered with better precision than the Basic AJFs. However, the critic for Basic AJF performs better than the critic for Complex AJFs which seems to overly reject valid Complex AJFs too. It is important to highlight that both in terms of the precision and the coverage, the proposed RECREW procedure outperforms the baseline. We detail the results obtained from the Llama-3-8B-Instruct model in Appendix F.

Across the entire annotation exercise (AJFs obtained in both case types from RECREW using gpt-

[3] https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

[4] https://openai.com/index/new-embedding-models-and-api-updates/

[5] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

	VID		LID	
	Count	$P\|_K$	Count	$P\|_K$
Basic AJFs				
Extraction from text corpus	490	NA	578	NA
Generated by RECREW-generate step	39	NA	48	NA
Total	529	NA	626	NA
Selected by Critic in RECREW-generate step	320	$0.79\|_{50}$	306	$0.77\|_{50}$
Selected by Critic in RECREW-review step	158	$0.77\|_{50}$	225	$0.82\|_{50}$
Rejected by Critic in any step	260	$0.87\|_{50}$	415	$0.92\|_{50}$
Complex AJFs				
Generated by RECREW-generate step	18004	NA	18782	NA
Selected by Critic in RECREW-generate step	14421	$0.91\|_{100}$	16929	$0.86\|_{100}$
Selected by Critic in RECREW-review step	1853	$0.89\|_{100}$	573	$0.86\|_{100}$
Rejected by Critic in any step	3583	$0.48\|_{100}$	1853	$0.70\|_{100}$
Baseline approach	176	$0.72\|_{176}$	47	$0.81\|_{47}$

Table 4: Statistics of automatically discovered AJFs in the VID and LID datasets using the gpt-4o-mini model. Also shown in the intrinsic evaluation of the AJFs (in terms of precision $P|_K$ within random sample size K)

4o-mini as well as llama and the baseline), the percentage of AJFs on which both annotators agreed was 76%. We computed the inter-annotator agreement in terms of Cohen's Kappa and found it to be 0.32, signalling a fair agreement. We however stress that the kappa statistic is not an appropriate measure of agreement in this case as it has been shown in literature that it gets affected when there is class imbalance (Feinstein and Cicchetti, 1990; Viera et al., 2005), i.e. when one label is much more prevalent than the other label. In our case, the "correct" label is more prevalent than the "incorrect" label (which can be seen from the high precision values in Table 4). Hence, in our case, instead of the kappa score, the agreement percentage gives a better picture of the inter-annotator agreement. All the annotations were carried out by four authors having a computer science post-graduation background and ages between 25-40.

3.6 Ablation Analysis

It is important to analyze how much do the **Critic** and **Review** stages of the RECREW procedure contribute to in the overall discovery of the AJFs. The two aspects of the intrinsic evaluation – "Selected by Critic in RECREW-generate step" and "Rejected by Critic in any step" enable the ablation for the Critic step. For e.g. in case of the VID dataset, the precision for the generated Basic AJFs which are accepted by the critic is 0.79. Also the precision for the Basic AJFs of the critic for correctly rejecting is 0.87 which implies that precision for the generated Basic AJFs (but rejected by the critic) is just 0.13. This effectively makes the approximate precision for the Generate step to be a weighted mean of these two samples i.e. $\frac{(0.79\times320)\,+\,(0.13\times209)}{(320\,+\,209)} = 0.53$. This indicates that the Critic stage is important and helps push the precision from the only Generate step's 0.53 to 0.79. Similarly, we can see from the "Selected by Critic in RECREW-review step" aspect that those many AJFs weren't missed, thereby pushing the coverage up. In case of the VID dataset's basic AJFs, the no. is 158 which means there was $\frac{158}{(320+158)} = 33\%$ increase in coverage.

3.7 Extrinsic Evaluation

To conduct an extrinsic evaluation and check the real-life effectiveness of the discovered AJFs, we consider a downstream application of dispute resolution where one of the key tasks is to identify the stronger party in a dispute. The task is – given the structured summary of the dispute (Section 3.2) (except the details about the decision by National Commission and the Winning Party), can the stronger party be identified from the details of the dispute. We hypothesize that considering a set of case relevant complex AJFs along with the dispute details, can lead to improved performance of stronger party identification. To check the validity of this hypothesis, we consider two evaluation approaches:

1. **Dispute Details Only (only DD)**: In this approach we construct a prompt (See Appendix D) where we instruct an LLM to identify the stronger party while considering the details of a

given dispute.

2. **Dispute Details + AJF Context (DD+AC)**: In this approach we first obtain a list of complex AJFs relevant to each of the disagreement aspects, based on their embeddings' based semantic similarity[6]. We believe that the disagreement aspects are the most important facet of the dispute and hence we obtain complex AJFs relevant only to them. Secondly, we construct a context which presents each of the relevant complex AJFs as a conditional if-then statement. For example, suppose one relevant complex AJF is ⟨`second ignition key was not submitted`, *favours*, *insurance company*⟩, then we arrange it as `If the scenario - "second ignition key was not submitted" holds true in this case then it favours the insurance company`. This guides the LLM to decide whether to consider the AJF for the case and also prevents use of a spuriously predicted AJF into the LLM's reasoning for the task. We ensure to add the context when it contains factors favoring both parties as a context having only one party will bias it. In case when such a biased context is found, we skip context addition. Next, we construct a prompt (See Appendix D) where we instruct an LLM to find the stronger party while considering the details of a given dispute as well as the constructed AJF context. We do not consider a context based on the baseline approach as it has very limited coverage in terms of the factors.

   Note on label leakage risk: Although, we are using the same set of disputes for evaluating stronger party identification as the set which was used for identifying initial set of basic AJFs (Section 2.1), we believe that there is no risk of label leakage. This is because for stronger party identification, only complex AJFs (along with their favouring party) play a part (in the form of AJF context in the prompt) and not basic AJFs. Complex AJFs are not directly extracted from the dispute documents. Rather, complex AJFs are generated through SKD process by the LLM using basic AJFs as seeds. Moveover, only a subset of basic AJFs is getting extracted from the dispute documents and they do not have any "tilt" towards a particular party unlike complex AJFs. For stronger party identification, this "tilt" of complex AJFs is important but it is not derived from the documents in any way.

Dataset →	VID		LID	
	Accuracy	**Macro F1**	**Accuracy**	**Macro F1**
only DD	0.71	0.70	0.60	0.59
only AC	0.44	0.43	0.55	0.54
DD + AC	**0.72**	**0.71**	**0.65**	**0.65**

Table 5: Extrinsic evaluation of the AJFs for the VID and LID datasets while using gpt-4o-mini for stronger party prediction

Baseline setting - Only AJF Context (Only AC): As a baseline for extrinsic evaluation, we also consider a scenario when only the selected AJFs are given as details for deciding the stronger party without any dispute details. The context in this case if formed differently from the earlier context (in DD+AC) where the context was formed by putting each AJF into a conditional if-then statement format. This if-then format would not work here as the dispute details are not present. Hence, we form the context by simply listing the relevant complex AJFs, each in a tuple form e.g. ⟨`original key of the stolen vehicle was not submitted`, FAVOURS, `insurance company`⟩.

We use both the gpt-4o-mini as well as a Llama-3-8B-Instruct LLMs to evaluate the approaches and experiment with all cases in both the datasets. We report the accuracy and macro F1 numbers in Table 5 and Table 14. As can be seen, the addition of the constructed context to the dispute details (*AC+DD*) leads to better performance as against the vanilla only dispute details (*only DD*) approach for both datasets while using gpt-4o-mini. In case of Llama-3-8B-Instruct too, *AC+DD* approach outperforms the *only DD* approach on the LID dataset. However, this is not observed for the VID dataset. We believe that the smaller Llama-3-8B-Instruct model is not able to harness the AJF context as effectively as the gpt-4o-mini is able to. Although, we checked the explanation that the LLM provides for the stronger party decision, it does consider the factor context provided, helping it to infer to the right decision more often. This confirms our hypothesis and underscores the contribution of the discovered AJFs.

[6] We use an empirically judged similarity threshold of 0.55

4 Related Work

Earlier work has shown that identifying legal factors plays an important role in case analysis. HYPO (Ashley, 1991) focused on the comparing and contrasting of cases in terms of "Dimensions" (Legal factors) which either favoured the Plaintiff or Defendant (e.g., `Security Measures` Favours `Plaintiff`). CATO (Ashley, 1999) extended such legal factors related to trade secret law and introduced a factor hierarchy which organizes factors into abstract and specific factors. However, these approaches depends a lot on human experts and may be difficult to scale. Recent work, such as Gray et al. (2024), employed LLMs to extract legal factors directly from court opinions using zero/one shot prompting techniques with human (or LLM) refinement. However, the approach is not generalizable for a specific domain because the factor coverage is bound by the number of cases input to their proposed factor discovery procedure. However, in our case, as we are distilling basic/complex AJFs from an LLM, our coverage of the factors relevant to a domain are not limited to a specific set of input documents. Another related work by Santosh et al. (2025) proposes a structured intermediate planning using encoder-decoder based model (Bertsch et al., 2023) to create event representations in tuple format (subj-verb-obj) for generating coherent summaries. Another work by Joshi et al. (2023) is similar and proposes a method to extract events from a document using dependency parser, to obtain tuples of the form (subj, pred, obj) used in a prior case retrieval application. However, our proposed "eventive" factors i.e. complex AJFs are more flexible and also capture a tilt towards a favouring party making them suitable for argument generation and decision prediction.

5 Conclusion and Future Work

As part of this work, we focus on Argument and Judgement Factors (AJFs) which are the well known "legal factors" from literature but named so (AJF) for better clarity and applicability. We then proposed a step-wise approach for discovering AJFs for a case type (such as vehicle insurance disputes) using Symbolic Knowledge Distillation (SKD) from an LLM. The SKD involves a novel three-step prompting method – RECREW (geneRatE-CRitic-reviEW) based on one-shot reflexion. We evaluated the discovered AJFs intrinsically as well as extrinsically in a real-life dispute resolution application. We demonstrated the effectiveness of the discovered AJFs in enhancing an LLM's ability for better decision prediction. We also show that when compared to a suitable SoTA baseline, our RECREW based discovery provides better coverage in terms of number of different AJFs discovered. In future, we plan to explore the proposed technique's generality to more case-types beyond insurance and also enable other applications such as argument synthesis. We would also like to explore representation of the discovered AJFs in a knowledge graph format where various semantic relations among the AJFs can also be captured such as *causes*, *contradicts*, *entails*, etc.

6 Limitations

- Given our budget constraints and access constraints due to organizational policies, we have tried with only a single large model namely gpt-4o-mini and not with other more capable models such as gpt-4.1 or gpt-5. However, we believe that the proposed techniques are generic and should work with other LLMs as well.
- We have only experimented with cases of two type i.e. vehicle and life insurance. However, as the proposed approaches are generic they should work for other case-types as well.
- In this work, we have simply created the knowledge base of legal factors in the form a *list* wherein we have not captured the relations among the AJFs. We plan to explore this direction as future work.
- Though, we have manually evaluated the discovered AJFs, we believe that qualitative evaluation by legal experts is also very important. In future, we plan to conduct a detailed user study involving legal experts to qualitatively evaluate the discovered AJFs.

7 Ethics Statement

The associated datasets on insurance disputes are available publicly in their raw form without any form of anonymization of personally identifiable information such as person and organization names. We ensure that we do not use any PII information for the aims of legal factor discovery or the application on stronger party identification. Moreover, we filter out all factors that are based on such information using Named Entity Recognition techniques. Apart from this aspect, there are no major ethical considerations associated with this work.

References

Kamal Acharya, Alvaro Velasquez, and Houbing Herbert Song. 2024. A survey on symbolic knowledge distillation of large language models. *IEEE Transactions on Artificial Intelligence*.

Vincent Aleven. 2003. Using background knowledge in case-based legal reasoning: A computational model and an intelligent learning environment. *Artificial Intelligence*, 150(1):183–237. AI and Law.

Kevin D Ashley. 1990. Modeling legal argument: reasoning with cases and hypotheticals. a bradford book.

Kevin D Ashley. 1991. Reasoning with cases and hypotheticals in hypo. *International journal of man-machine studies*, 34(6):753–796.

Kevin D Ashley. 1999. Designing electronic casebooks that talk back: The cato program. *Jurimetrics*, 40:275.

Amanda Bertsch, Uri Alon, Graham Neubig, and Matthew Gormley. 2023. Unlimiformer: Long-range transformers with unlimited length input. *Advances in Neural Information Processing Systems*, 36:35522–35543.

Piotr Bystranowski, Bartosz Janik, and Maciej Próchnicki. 2022. *Judicial Decision-Making: Integrating Empirical and Theoretical Perspectives*. Springer.

Alvan R Feinstein and Domenic V Cicchetti. 1990. High agreement but low kappa: I. the problems of two paradoxes. *Journal of clinical epidemiology*, 43(6):543–549.

Eveline T. Feteris. 2018. *Fundamentals of Legal Argumentation: A Survey of Theories on the Justification of Judicial Decisions*, 2 edition. Springer.

Barry Friedman, Margaret H. Lemos, Andrew D. Martin, Tom S. Clark, Allison Orr Larsen, and Anna Harvey. 2020. *Judicial Decision-Making: A Coursebook*. West Academic Pres.

Morgan Gray, Jaromir Savelka, Wesley Oliver, and Kevin Ashley. 2024. Using llms to discover legal factors. *arXiv preprint arXiv:2410.07504*.

IRDAI. 2025a. Commercial Vehicle Policy. https://irdai.gov.in/documents/37343/993134/39..+COMMERCIAL+VEHICLE+INSURANCE+POLICY_GEN740.pdf/a2dee364-a9c2-6c20-0bb3-6dd4e6151b8c?version=1.1&t=1668418763850&download=true. [Online; accessed 19-May-2025].

IRDAI. 2025b. Life Insurance Policy Sample FG Group. https://irdai.gov.in/documents/37343/563751/SamplePolicyDocument_FGGroupTermLifePlan_2013-14.pdf/30b9abc7-57eb-51d5-e967-2ccc3784e974?version=1.1&t=1667796726870. [Online; accessed 23-July-2025].

IRDAI. 2025c. Life Insurance Policy Sample LIC. https://licindia.in/documents/20121/118257/Sample-Policy-Document_LIC-s-New-Jeevan-Shanti_UIN-512N338V01-(1).pdf/4bbd92df-648c-9c77-a950-904cd7326d2d?t=1669099607669. [Online; accessed 23-July-2025].

IRDAI. 2025d. Private Vehicle Policy. https://irdai.gov.in/documents/37343/993134/IRDAN103RP0007V01201819_GEN2732.pdf/d2ea7fe2-eadb-2f57-8c14-9d8deef166c4?version=1.2&t=1668328094511&download=true. [Online; accessed 19-May-2025].

Abhinav Joshi, Akshat Sharma, Sai Kiran Tanikella, and Ashutosh Modi. 2023. U-creat: Unsupervised case retrieval using events extraction. *arXiv preprint arXiv:2307.05260*.

Prathamesh Kalamkar, Astha Agarwal, Aman Tiwari, Smita Gupta, Saurabh Karn, and Vivek Raghavan. 2022. Named entity recognition in Indian court judgments. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 184–193, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Austin Lovegrove. 1989. *Judicial Decision Making, Sentencing Policy, and Numerical Guidance*. Springer-Verlag.

Sachin Pawar, Manoj Apte, Girish Keshav Palshikar, Basit Ali, and Nitin Ramrakhiyani. 2025. Drassist: Dispute resolution assistance using large language models. In *Proceedings of the Twentieth International Conference on Artificial Intelligence and Law (ICAIL)*, pages 188–198.

Sachin Pawar, Nitin Ramrakhiyani, Anubhav Sinha, Manoj Apte, and Girish Palshikar. 2024. Why generate when you can discriminate? a novel technique for text classification using language models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1099–1114.

TYSS Santosh, Youssef Farag, and Matthias Grabmair. 2025. Coperlex: Content planning with event-based representations for legal case summarization. *arXiv preprint arXiv:2501.14112*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

Ihor Stepanov and Mykhailo Shtopko. 2024. Gliner multi-task: Generalist lightweight model for various information extraction tasks. *arXiv preprint arXiv:2406.12925*.

Anthony J Viera, Joanne M Garrett, et al. 2005. Understanding interobserver agreement: the kappa statistic. *Fam med*, 37(5):360–363.

Douglas Walton. 2012. *Legal Argumentation and Evidence*. Pennsylvania State University Press.

Peter West, Chandra Bhagavatula, Jack Hessel, Jena Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2022. Symbolic knowledge distillation: from general language models to commonsense models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4602–4625.

Hannes Westermann, Vern R Walker, Kevin D Ashley, and Karim Benyekhlef. 2019. Using factors to predict and analyze landlord-tenant decisions to increase access to justice. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, pages 133–142.

A Details on Basic Filtering of candidate basic AJFs

As part of the basic filtering step, we remove spurious candidate basic AJFs obtained as output of the GLiNER based information extraction. These include candidates which are either too specific to a particular dispute such as names of people or organizations or are to generic and used widely in the legal domain such as legal roles or organizations. We explain the two kinds of filtering which are carried out, as follows:

List/NER based Filtering: As part of the list of extracted candidate AJFs we found (i) generic legal terms such as legal roles (e.g. `judge`, `attorney`) and legal entities (e.g. `court`, `tribunal`, and (ii) specific named entities such as persons names, organization names, locations, dates, money amounts and sections. These phrases are either too specific to a particular judgement (e.g. person names) or too general to be considered as a basic AJF for the case type. Hence, we remove them using matching in fixed lists as well as using the legal NER model proposed by Kalamkar et al. (2022) on the list of extracted entities.

Document Frequency based filtering: We compute the Document Frequency (DF) of each extracted entity in the input corpus. The candidates with a DF of 1 and coming from a judgment are discarded because they generally correspond to a concept specific to that judgement (e.g. `Rs 1 crore`, `consumer protection act`, etc.). However, candidates with a DF of 1 but coming from statute documents (policy or acts) are retained because they can be broadly applicable to the cases (e.g., `statutory liability`, `annual premium`, etc).

B Prompts for Step 2 – RECREW for Basic AJF discovery

The generate, critic and review prompts for discovery of Basic AJFs are shown in Table 6, Table 7 and Table 8 respectively, for the VID dataset. We reiterate that these prompts are generic and were ported easily for the LID dataset with few replacements in the system prompt and instructions.

You are a legal expert in auto-insurance disputes!
Argumentation and Judgement Factor (AJF): It is any linguistic term or phrase which gets used in legal arguments and contributes to reasoning used in judicial decisions.

Following are some example of AJFs specific to vehicle insurance disputes / cases.
(2 few-shot examples)

Instruction: Suggest some more such AJFs related to vehicle insurance disputes / cases.

Ensure the following while generating the response:
1. Generate AJFs separated by a new line.
2. Each AJF should not contain more than four words.
3. Do not generate any extra information such as justification.

Table 6: Prompt used for the **Generate** step of Basic AJF discovery. Overall there are 10 training examples, from which 2 are chosen randomly at a time as few-shot examples in each execution of the prompt.

C Prompts for Step 3 – RECREW for Complex AJF discovery

The generate, critic and review prompts for discovery of Basic AJFs are shown in Table 9, Table 10 and Table 11 respectively.

D Prompts for Extrinsic Evaluation

In Table 12, we detail the prompt used for the only Dispute Details (only DD) technique where only selected elements of the structured summary of the dispute are provided to the LLM for discerning the stronger contesting party. In Table 13, we detail the prompt used for the DD + AC (Dispute Details + AJF Context) technique where both the selected elements of the structured summary and the context

```
You are a legal expert in auto-insurance disputes!
Argumentation and Judgement Factor (AJF): It is any linguistic term or phrase which gets used in legal arguments and contributes to reasoning used in judicial decisions.

Following are the examples followed by the label (AJF or NOT AJF) with the justification:
1. driving license - AJF ( because it is a piece of evidence which can be used in legal arguments and can influence the decision in vehicle insurance disputes/cases )
2. respondent - NOT AJF ( because it is just one of the parties involved in a case )
... (10 more few-shot examples)

For the following example predict the label (AJF or NOT AJF) with a justification, similar to the format shown above.
{basic_AJF} -
```

Table 7: Prompt used for the critic step of Basic AJF discovery

```
You are a legal expert in auto-insurance disputes!
Argumentation and Judgement Factor (AJF): It is any linguistic term or phrase which gets used in legal arguments and contributes to reasoning used in judicial decisions.

Following are some examples of valid AJFs and justification of why they are valid:

{positive_examples_along_with_reasoning}

Following is an incorrect AJF along with the justification about why it is incorrect. Revise the following incorrect AJF to a valid AJF as per the above definition and examples of AJFs.
Incorrect AJF: {basic_incorrect_AJF_with_reasoning}

Ensure the following while generating the response:
1. The revised AJF should not contain more than four words.
2. Do not generate any extra information such as justification.

### Response format:
Revised AJF: ⟨AJF⟩ : ⟨justification⟩
```

Table 8: Prompt used for the review step of Basic AJF discovery

```
You are a legal expert in auto-insurance disputes!
Following tuples in the format ⟨Events or scenarios, favours, insurance company⟩ involving “ignition keys” favours the insurance company in disputes / court cases between an insured and an insurance company.
(2 few-shot examples)

Output: Write more tuples like these involving “{basic_AJF}”.

### Instruction while creating the output:
1. Generated tuples should be separated by a new line.
2. Each event or scenario in the tuples should be meaningful and realistic in practice in vehicle insurance cases.
3. Each event or scenario should be short, clear and atomic and focus on individual events or scenarios. Do not generate complex events or scenarios where multiple events are involved.
```

Table 9: Prompt used for the **Generate** step of Complex AJF discovery. Overall there are 10 example tuples for each favouring party, from which 2 are chosen randomly for a favouring party at a time as few-shot examples in each execution of the prompt.

```
You are a legal expert in auto-insurance disputes!
### Instruction:
Given the auto-insurance Event or Scenario described below, assess which party (the Insured or the Insurance Company) is most likely to be favored in a dispute. If the Event or Scenario does not indicate a clear inclination towards either party, choose “Neutral”. Select one of the following options and provide a clear, concise justification for your decision:
1. Neutral
2. Insured
3. Insurance Company

### Input:
Event or Scenario: {complex_AJF}

### Response Format:
⟨Neutral or Insured or Insurance Company⟩: ⟨justification⟩
```

Table 10: Prompt used for the critic step of Complex AJF discovery

constructed from the retrieved factors is provided for the stronger party prediction task.

E Implementation Details

Due to our budget and hardware constraints, we used only the OpenAI gpt-4o-mini as the larger LM for the experiments and text-embedding-3-large model for text embeddings along. Overall, it cost us about USD 7$ for API calls for all the experiments reported in this paper. Additionally, we used the Llama-3-8B-Instruct model for experiments with a moderate sized model for comparison. We downloaded the Llama model from Huggingface[7]. In the experiments with the open source llama model we use the stella-400M[8] embeddings for semantic similarity. We implemented the code in Python 3.12.4 and used `openai` package (version 1.76.2) for the gpt-4o-mini API calls and the huggingface transformers package (version 4.43.2) for llama experiments.

[7] huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

[8] https://huggingface.co/NovaSearch/stella_en_400M_v5

You are a legal expert in auto-insurance disputes!
Complex Argumentation and Judgement Factor (AJF): It is any Event or Scenario which favours either "Insured" or "Insurance company" and which may get used in legal arguments and reasoning in judicial decisions. It is expressed as a tuple ⟨ `Event or Scenario`, *favours*, *Insured or Insurance company*⟩

Following are some examples of valid Complex AJFs and why they are valid:
{positive_examples_along_with_reasoning}

Following is an incorrect Complex AJF along with the justification about why it is incorrect. Revise the following incorrect Complex AJF to a valid Complex AJF as per the above definition and examples of Complex AJFs.
Input: {incorrect_complex_AJF_with_reasoning}

Instruction while creating the output:
1. Event or Scenario in the revised Complex AJF should be meaningful and realistic in practice in vehicle insurance cases.
2. Event or Scenario in the revised Complex AJF should be short, clear and atomic and focus on individual events or scenarios. Do not generate an event or scenario where multiple events are involved.

Table 11: Prompt used for the review step of Complex AJF discovery

F Results using an open source small language model

We also try the proposed AJF discovery with an open source small language model – LLama-3-8B-instruct for comparison with the gpt-4o-mini based discovery. We detail the results of the intrinsic evaluation of the AJFs discovered using the LLama-3-8B-instruct model in Table 15. We observe that in terms of both sample precision ($P|_K$) and coverage, the larger gpt-4o-mini model is better than the LLama-3-8B-instruct model. This difference can be attributed to the inherent complexity of legal language and to the fact that smaller models are typically trained on smaller text corpora than larger LLMs.

G Baseline approach

We replicated the technique discussed in (Gray et al., 2024) following the guidelines they provided. However, three modifications were necessary in their work for a meaningful comparison:
(i) Instead of extracting only simple legal factors as in their technique, we modified their prompt instructions to enable the model to extract complex legal factors which includes both the event/scenario and the corresponding favoured party.

Consider the following dispute between an insurance company and an insured party.
Dispute description:

Facts agreed by both parties: {facts}

Aspects on which the parties disagree: {disagreement_aspects}

Arguments of the insurance company: {insurance_company_arguments}

Arguments of the insured party: {insured_arguments}

Relevant prior cases: {relevant_prior_cases}

Relevant statutes or policy terms and conditions: {relevant_statutes}

Instruction: Identify the Overall Stronger Party (insurance company or insured party) using the agreed facts, disagreement aspects, arguments, prior cases and statutes or policy terms and conditions. Include a short justification considering the fairness and logic of the dispute resolution.

Follow the following response format and do not generate any additional output.

Response Format:
Overall Stronger Party: ⟨*insurance company* or *insured party*⟩: ⟨Justification⟩

Table 12: Prompt for the Only DD technique

Consider the following dispute between an insurance company and an insured party.
Dispute description:

Facts agreed by both parties: {facts}

Aspects on which the parties disagree: {disagreement_aspects}

Arguments of the insurance company: {insurance_company_arguments}

Arguments of the insured party: {insured_arguments}

Relevant prior cases: {relevant_prior_cases}

Relevant statutes or policy terms and conditions: {relevant_statutes}

It is important to stress upon relevant legal factors which can be gleaned from the disagreement aspects. They are listed as follows: {context}

Instruction: Identify the Overall Stronger Party (insurance company or insured party) using the agreed facts, disagreement aspects, arguments, prior cases, statutes or policy terms and conditions, and relevant legal factors. Include a short justification considering the fairness and logic of the dispute resolution.

Follow the following response format and do not generate any additional output.

Response Format:
Overall Stronger Party: ⟨*insurance company* or *insured party*⟩: ⟨Justification⟩

Table 13: Prompt for the DD + AC technique. Here the {context} comprises of the retrieved factors listed in the if-then format specified in Section 3.7

Dataset →	VID		LID	
	Accuracy	**Macro F1**	**Accuracy**	**Macro F1**
only DD	**0.71**	**0.70**	0.60	0.58
only AC	0.39	0.38	0.55	0.51
DD + AC	0.66	0.65	**0.62**	**0.61**

Table 14: Extrinsic Evaluation of the AJFs for the VID and LID datasets while using Llama-3-8B-Instruct for stronger party prediction

(ii) We employed the gpt-4o-mini model for factor extraction and refinement as against their use of the gpt-4o model[9]. This is comparable with our proposed technique where we too use the gpt-4o-mini model.
(iii) Instead of relying on human-based factor refinement which yielded the best results in their work, we employed an LLM for this refinement step while adhering to the same guidelines.

The prompts used to extract and refine complex legal factors for the VID datasets are shown in the tables: Table 16 and Table 17. We used the same prompt for factor extraction in both datasets, except for domain-specific legal examples. Cross-domain legal examples were included for each dataset, following guidance to vary example domains. For refinement, a separate system prompt per dataset provided legal background before refining the extracted factors to ensure domain understanding of legal factors.

[9] We operate under strict budget constraints and hence this choice

	VID		LID	
	Count	$P\|_K$	Count	$P\|_K$
Basic AJFs				
Extraction from text corpus	483	NA	179	NA
Generated by RECREW-generate step	20	NA	43	NA
Total	503	NA	222	NA
Selected by Critic in RECREW-generate step	455	$0.71\|_{50}$	216	$0.68\|_{50}$
Selected by Critic in RECREW-review step	40	$0.76\|_{40}$	24	$0.90\|_{24}$
Rejected by Critic in any step	56	$0.92\|_{50}$	36	$0.88\|_{36}$
Complex AJFs				
Generated by RECREW-generate step	8716	NA	4,114	NA
Selected by Critic in RECREW-generate step	8676	$0.72\|_{100}$	4075	$0.74\|_{100}$
Selected by Critic in RECREW-review step	32	$0.65\|_{32}$	32	$0.67\|_{32}$
Rejected by Critic in any step	34	$0.75\|_{34}$	38	$0.62\|_{38}$

Table 15: Statistics of automatically discovered AJFs in the VID and LID datasets using the Llama-3-8B-Instruct model. Also shown in the intrinsic evaluation of the AJFs (in terms of precision within random samples of various types)

You are an expert in analysing legal cases between insurance company and insured parties, identifying critical and complex legal factors that determine judgments in favor of either party!

Legal factor definition and example:
A legal factor is any relevant circumstance, scenario, or piece of information that a court, tribunal, or other adjudicating authority considers when forming its reasoning and arriving at a decision in a legal dispute. Legal factors have been defined as stereotypical fact patterns that tend to strengthen or weaken a side's argument in favor of a legal claim.
Examples:
1. ⟨`insured submitted an unverified medical report`, *favours*, *insurance company*⟩ : The submission of an unverified medical report raises concerns about the accuracy and reliability of the information provided. Insurance companies typically rely on verified medical records to assess risk and determine coverage. If the report is unverified, the insurance company may have grounds to dispute the claim or deny coverage based on the lack of credible evidence regarding the insured's health status.
2. ⟨`insured's cause of death is covered under the policy`, *favours*, *insured*⟩ : The scenario clearly states that the cause of death is covered under the policy. This indicates that the insured's beneficiaries are likely to receive the death benefit, as the insurance company is obligated to pay out for covered causes of death. Therefore, the insured party is favored in this dispute.

Task:
Identify the key complex legal factors based on your understanding from legal factor definition and examples and return them strictly in the format ⟨`Complex legal factor`, *favouring*, *either of the party*⟩ from the below insurance judgement cases focusing on court judgements:
Case 1: {case1}
Case 2: {case2}
Case 3: {case3}
Case 4: {case4}
Case 5: {case5}
Case 6: {case6}
Case 7: {case7}
Case 8: {case8}
Case 9: {case9}
Case 10: {case10}

Table 16: Prompt for the extraction of legal factors from the VID dataset as per the baseline approach

You are a legal expert in auto-insurance disputes!

Complex Legal Factors: {legal_factors}

Task and Instructions:
Review each of the complex Legal Factors presented above, and refine the overall list of Legal Factors based on your understanding and given instructions:
1. Combine obviously identical or highly similar induced factors identified across prompt iterations,
2. identify subtle similarities and differences, and group or distinguish factors accordingly,
3. refine redundant language,
4. capture the full breadth of a factor,
5. avoid combining what a human may identify as separate factors under a single definition,
6. consider counting as a factor something for which the above output contained only a single instance.
7. Return the factors strictly in the format ⟨`complex legal factor`, *favouring*, *either of the party*⟩

Table 17: Prompt for the refinement of legal factors from the VID dataset as per the baseline approach