ArgAssist: LLM-based Argument Synthesis for Insurance Disputes

Anubhav Sinha, Nitin Ramrakhiyani,
Sachin Pawar, Isha Narang , Manoj Apte
TCS Research, Tata Consultancy Services Limited, India.
{s.anubhav2, nitin.ramrakhiyani, sachin7.p, narang.isha, manoj.apte}@tcs.com

Abstract

Access to timely and affordable dispute resolution is a major challenge for industries such as insurance, finance, and consumer goods, where legal disputes between suppliers and consumers are frequent and often complex. We present **ArgAssist**, an LLM-based system for structured argumentative discourse generation that forms a core component of an alternative dispute resolution (ADR) platform that we are building. ArgAssist assists parties involved in a dispute (such as insurer and insured in an insurance dispute) by synthesizing discourse-structured legal arguments, represented as claims supported by typed premises grounded in case information. ArgAssist first generates *base* arguments using an LLM, which are then strengthened by grounding them in relevant prior cases and statutes to ensure legal soundness and contextual coherence. We introduce a novel evaluation metric for assessing the quality of generated arguments and demonstrate ArgAssist's effectiveness on two real-world datasets in the insurance domain. Our results show that explicitly modeling argumentative discourse structure and grounding, significantly improves alignment with expert-authored legal arguments.

1 Introduction

Legal disputes are pervasive across industries such as insurance, finance, e-commerce, and consumer goods. Most of these disputes end up in litigations, straining the already overburdened judicial systems. As a result, policy makers and regulatory bodies are increasingly advocating the use of alternative dispute resolution (ADR) mechanisms to resolve disputes before formal litigation (NITI-Aayog, 2021; Ashton, 2024). To support this vision, we are developing a dispute resolution platform that leverages advances in Natural Language Processing (NLP) to automate and assist key components of legal reasoning. One such core capability is argumentative discourse generation, i.e., the ability to synthesize plausible, coherent, and legally grounded arguments for a disputing party based on the case material. The ability to construct such arguments is a skill traditionally reserved for trained legal professionals. However, for accessible justice, there is a pressing need to democratize legal reasoning and support individuals who may lack access to professional legal representation. To this end, this paper presents ArgAssist, a core module of our dispute resolution platform, which uses large language models (LLMs) to assist users in formulating structured legal arguments.

Unlike argument extraction, which identifies arguments from existing texts, argument synthesis involves generating new arguments that are legally grounded and tailored to the specifics of a case. This task is particularly challenging due to the need for domain knowledge, contextual relevance, and logical coherence. To tackle this challenge, we propose an approach to argument synthesis that leverages LLMs and curated legal knowledge. Central to our methodology is a *structured representation* of arguments, where arguments are not isolated statements but instances of structured argumentative discourse, where a central *claim* is supported by a set of *typed* premises such as *facts*, *statutes*, *precedents*, or other *evidences/testimonies* (Table 1). This structure not only facilitates systematic generation and evaluation but also aligns with established models of legal argumentation (Palau and Moens, 2009).

Recent advances in LLMs have enabled promising approaches to argument synthesis, moving beyond traditional extraction tasks. Prior works have explored generating persuasive and counter-arguments using retrieval-augmented generation (RAG) frameworks (Verma et al., 2025; Hua et al., 2019), multi-agent systems (Zhang and Ashley, 2025), and prompting strategies inspired by argumentation theory (Miandoab and Sarathy, 2024). These efforts highlight the growing interest in struc-

Proceedings of the 27th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 124–138
August 2–5, 2026.

Argument by an Insured party:
Claim: The insurer cannot raise new grounds for repudiation at the appeal stage that were not mentioned in the original repudiation letter. **Premises**: - **Fact**: The repudiation letter did not mention the issue of the driver's identity or license status as a ground for denial. - **Precedent**: Supreme Court judgment in Glada Power and Telecommunication Ltd. Vs. United India Insurance Co. Ltd. held that insurers are bound by the grounds stated in the repudiation letter.
Argument by an Insurance Company:
Claim: The driver was not holding a valid license at the time of the accident. **Premises**: - **Evidence/Testimony**: Verification report from Mr. RRR, independent investigator, Hyderabad, stated that the license was not issued by any RTA office in Hyderabad. - **Evidence/Testimony**: Detailed report from Additional Licensing Authority, Hyderabad, showed license No. 99999 was issued to a different person (YYY) and not to RRR. - **Statute**: Section 9 Clause 1(1) of the Motor Vehicles Act, 1988 requires a valid driving license for driving a vehicle. - **World knowledge**: A contract of insurance is based on the principle of utmost good faith (uberrima fides) and both parties must act honestly.

Table 1: Examples of legal arguments. An argument consists of a statement making some claim which is supported by one or more premises of types - *fact*, *precedent*, *statute*, *evidence/testimony*, or *world knowledge*.

tured and context-aware argument generation. In contrast to these works, our approach focuses on synthesizing structured legal arguments grounded in real-world dispute data, using a standardized argument format and a novel evaluation metric to assess alignment with actual court-presented arguments. Our key contributions are:

- **ArgAssist**, a multi-stage LLM-powered framework that synthesizes discourse-structured legal arguments grounded in case facts as well as relevant statutes and prior cases.
- **Two curated datasets** from Indian consumer court cases in the insurance domain, enabling realistic evaluation of argument synthesis in industrial dispute scenarios.
- **ArgMatch**, a novel evaluation metric for assessing the quality of synthesized arguments by comparing them with gold-standard arguments.

2 Related Work

Argumentation Mining: Argumentation Mining studies the identification and analysis of argumentative discourse structures, typically consisting of claims, premises, and conclusions (Walton et al., 2008). These structures are foundational in applications such as legal analytics, debate moderation, and scientific discourse analysis. In the legal domain, several works have explored extracting legal arguments and their relations (e.g., support or attack) (Zhang et al., 2023; Elaraby and Litman, 2022; Ali et al., 2022, 2023; Gray et al., 2024). Argumentation Mining has evolved from rule-based systems to modern approaches powered by LLMs (Li et al., 2025), which leverage prompting techniques like zero/few-shot learning, chain-of-thought reasoning, and retrieval-augmented generation (RAG).

Counter-Argument Generation: Generating counter-arguments is a key component in debates, legal reasoning, and ethical discourse. Early work by Hua et al. (2019) used seq2seq models to generate counter-arguments by retrieving relevant information from web sources. More recent work by Verma et al. (2025) proposed a framework that balances analytical (justification-based) and social (reciprocity-based) arguments. Their approach includes RAG-based retrieval from the Change My View (CMV) dataset and fine-tuning LLMs to produce balanced counter-arguments.

Argument Synthesis: While Argumentation Mining focuses on extraction, argument synthesis involves generating coherent arguments from structured or unstructured input, making it a more complex and challenging task. Tuvey and Sen (2023) introduced a framework for generating legal arguments from case facts by fine-tuning models such as GPT-2 and Flan-T5. Similarly, Saha and Srihari (2023) proposed a neural factual argument generator that adheres to specified stances and argument schemes though their work is not specifically about legal arguments. Argument synthesis is also central to ethical persuasion in legal AI (Rogiers et al., 2024), where the goal is to construct persuasive yet principled arguments. Other works (Gray et al., 2025; Zhang and Ashley, 2025) have explored legal argument generation using LLM prompting, multi-agent frameworks, and hallucination mitigation strategies. Although, these techniques assume availability of cases which are annotated with *legal factors* which is not the case with our datasets of insurance disputes. Miandoab and Sarathy (2024) introduced prompting techniques inspired by argumentation theory to generate strong arguments, while Favero et al. (2025) focused on evaluating argument strength using critical questions generated and judged by LLMs. However, both of these

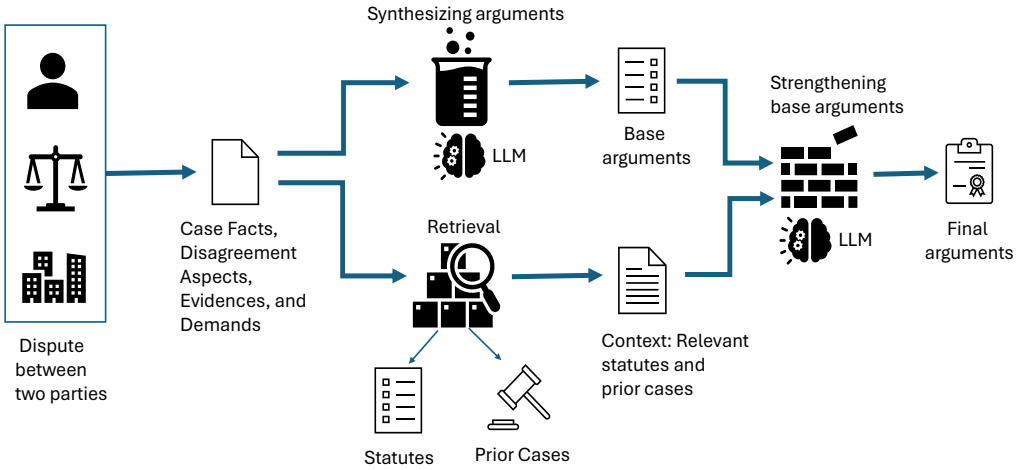

Figure 1: ArgAssist pipeline

works are not specifically focusing on generating legal arguments. Overall, the work by Tuvey and Sen (2023) is the most aligned with our goal of generating legal arguments and hence we use it as a baseline for ArgAssist in our experiments.

3 Proposed Approach

3.1 Structured representation of cases

For synthesizing arguments for a party involved in a dispute, we need case information about the dispute. We assume that this information is available in a structured representation having the following structural elements: (i) *facts agreed by both parties*, (ii) *aspects on which the parties disagree*, (iii) *evidences* (such as documents, physical objects), (iv) *witness testimonies*, (v) *demands* of the insurance company, (vi) *demands* of the insured party. Together, these elements provide the factual and contextual grounding required for synthesizing dispute-specific argumentative discourse.

For argument synthesis, we also need information about *prior cases* that have been adjudicated and resolved by courts. For such cases, we assume the availability of all the above-mentioned elements, along with additional discourse-level information that captures judicial reasoning. Specifically, we assume availability of: (i) *arguments* of the insurance company, (ii) *arguments* of the insured party, and (iii) *the key assessments and observations* made by the judge while arriving at a decision. Here, the arguments are further structured into claim and premises as described in Table 1. The key assessments generally capture (i) evaluation of a specific argument made by a party, (ii) rationale provided by the judge in support of the final decision, or (iii) assessment of the relevance or applicability of any evidence or statute. Section 4.1 provides more details about how we automatically obtain this structured representation of input cases (for which we need to synthesize arguments) and prior cases. Appendix A shows an example of this structured representation for a case.

3.2 Structured Representation of Statutes

Since our work focuses on the insurance domain, the terms and conditions specified in insurance policies play a central role in shaping dispute-specific argumentative discourse. Accordingly, we use the term *statute* to denote an *atomic normative discourse unit* (such as a rule, regulation, condition, or exception) derived from legislative Acts as well as insurance policy documents. While the term traditionally refers to legislative provisions, we deliberately adopt this broader interpretation to capture policy-specific clauses that influence dispute resolution. To support systematic discourse grounding during argument synthesis, we curate a representative set of insurance policy documents and derive a collection of *structured statutes*. Here, each statute is structured into – (i) a *header* capturing the purpose and scope of the statute, and (ii) *content* capturing the actual legal norm (see Appendix Table 6).

3.3 ArgAssist pipeline

Figure 1 illustrates the overall pipeline of the ArgAssist framework. As part of a dispute resolution assistance system, a *party* (insured or insurance company) can present the case facts, disagreement aspects, evidences and its demands to the ArgAssist system and obtain a set of legally sound and strong

Algorithm 1: Retrieving relevant prior cases for an input case

Input: (i) Input case q with elements: Facts F_q, Disagreement aspects D_q, Evidences E_q, Testimonies T_q;
(ii) Set of prior cases $\mathcal{PC} = \{pc_1, pc_2, \ldots, pc_k\}$
Output: L_{PC}: Ranked list of prior cases based on similarity to q

$L_{PC} \leftarrow \{\}$;
foreach *prior case* $pc_i \in \mathcal{PC}$ **do**
  `// Compute similarity for each element`
  $s^i_F \leftarrow \text{MaxAvgSim}(F_{pc_i}, F_q)$;
  $s^i_D \leftarrow \text{MaxAvgSim}(D_{pc_i}, D_q)$;
  $s^i_E \leftarrow \text{MaxAvgSim}(E_{pc_i}, E_q)$;
  $s^i_T \leftarrow \text{MaxAvgSim}(T_{pc_i}, T_q)$;
  `// Weighted aggregation of element scores for each prior case`
  $S_i \leftarrow w_F \cdot s^i_F + w_D \cdot s^i_D + w_E \cdot s^i_E + w_T \cdot s^i_T$;
  $L_{PC} \leftarrow L_{PC} \cup \{\langle pc_i, S_i \rangle\}$
`// Sort prior cases by final score`
$L_{PC} \leftarrow$ Sort L_{PC} in descending order of S_i;
return L_{PC} *i.e. Ranked list of prior cases*

Function `MaxAvgSim`(A, B)**:**
  `//` A `has` m `elements,` B `has` n `elements`
  $s \leftarrow \frac{1}{|A|} \sum_{a \in A} \max_{b \in B} \text{CosSim}(\text{Emb}(a), \text{Emb}(b))$
  return s

arguments which the party can use to strengthen their case. The ArgAssist framework comprises of three important stages of processing which are described in detail as follows.

3.3.1 Synthesizing base arguments

As the first step, we enable synthesis of arguments by prompting (detailed prompt in Table 2) an LLM with two inputs: (i) case facts, disagreement aspects, evidences, testimonies and demands, and (ii) party from whose perspective the arguments need to be generated. Also, the prompt describes the *structured representation* of arguments, where each argument is represented in the form of a *claim* that is supported by a set of *typed* premises (Table 1). Hence, it enables the LLM to generate relevant structured arguments grounded in the case information to avoid hallucination. We denote the output comprising of these base arguments as ***ArgAssist_base***.

3.3.2 Retrieving from different sources

We hypothesize that providing an LLM with the case information as the only input for generating arguments may not be sufficient. It is important to supplement the ask with a context comprising of case relevant legal knowledge. This case relevant legal knowledge can comprise of relevant statues, relevant prior cases and legal factors[1] (Aleven, 2003). This is in line with the process followed by legal practitioners for building their client's case. We consider four elements of a input case's structured summary (facts, disagreement aspects, witness testimonies and evidences) and obtain relevant prior cases based on semantic similarity with the prior cases' corresponding elements. For better semantic similarity, we *generalize* case facts before embeddings computation such that specifics (person and organization names) are ignored. We explain the matching process in Algorithm 1. We computed a weighted average of scores from the corresponding matching score of the four elements as the final score. We sort the final scores from all the prior cases in descending order to obtain a ranked list of most relevant prior cases (L_{PC}) to an input case. On similar lines, we also obtain a ranked list of relevant statutes (L_S) by measuring their semantic similarity with these four elements of the input case's information.

3.3.3 Strengthening base arguments

Using the relevant prior cases and statutes that are now obtained, the base arguments are strengthened as part of this step. Here, the focus is on refining existing argumentative discourse by enriching it with additional, legally grounded premises. For each relevant prior case (and each relevant statute), an LLM is prompted to check if the prior case (statute) is relevant. If it is found to be relevant, the LLM is prompted to add premises of the types "precendent" and "statutes" correspondingly. This strengthens the existing argument with a better set of premises. This process is shown in Algorithm 2 and produces the output ***ArgAssist_final***. The detailed prompt used by this algorithm is shown in Table 3. We would like to posit that though this process looks similar to Retrieval Augmented Generation (RAG) (Lewis et al., 2020), it is actually Retrieval Augmented Refinement (RAR) (Aguilar, 2025) as an existing argument is being refined.

4 Experimentation and Analysis

4.1 Datasets and Annotation

In this paper, we utilize two datasets of insurance-related disputes which have also been used earlier

[1]Legal factors are not currently used as relevant legal knowledge as cases in our dataset do not have factors identified for them.

Algorithm 2: Base argument strengthening

Input: (i) q : input case details (facts, disagreement aspects, evidences, testimonies, demands, etc.);
(ii) $A = \langle C, P = \{\langle p_1, t_1\rangle, \cdots, \langle p_n, t_n\rangle\}\rangle$: base argument (claim C along with set of n premises P where $\langle p_i, t_i\rangle$ represents i^{th} premise and its type);
(iii) L_{PC}, L_S: ranked lists of prior cases and statutes relevant to q;
Output: Strengthened version of the argument A

```
foreach pc ∈ L_PC do
    p_pc ← generate_premise(q, pc, A) ;
      // Prompts LLM to generate a suitable
      premise for A's claim using pc if it
      is relevant (Table 3)
    if p_pc is "No" then
        continue ;        // skip if pc is not
          relevant
    else
        A.P ← A.P ∪ {⟨p_pc, "precedent"⟩} ;
          // add the premise of type
          "precedent"
        break
foreach s ∈ L_S do
    p_s ← generate_premise(q, s, A) ;  // Prompts
      LLM to generate a suitable premise for
      A's claim using s if it is relevant
      (Table 3)
    if p_s is "No" then
        continue ;        // skip if s is not
          relevant
    else
        A.P ← A.P ∪ {⟨p_s, "statute"⟩} ;  // add
          the premise of type "statute"
        break
return A
```

in Ali et al. (2026). We collected a total of 1117 insurance disputes from the National Consumer Disputes Redressal Commission (NCDRC), India, which serves as a consumer court. These disputes were downloaded from the ConfoNet project website[2] by selecting the sector as "Insurance" for the years 2022, 2023, and 2024. To focus on the relevant domains, we filtered out disputes related to other types of insurance (e.g., property or health), retaining only those pertaining to vehicle and life insurance. This resulted in curated datasets of **129** vehicle insurance disputes and **82** life insurance disputes, which we refer to as **VID** and **LID**, respectively[3]. We use 10 disputes each from both VID and LID as our *development set* (for deciding various hyperparameters as well as refinement of prompts) and rest all the disputes are used as our *test set*.

Prior cases: To support argument synthesis with precedents, we compiled a corpus of Indian Supreme Court judgments[4] spanning the years 1952 to 2016, which chronologically precede the disputes in our dataset. We shortlisted cases where one party was an insurance company and the other was an insured party. This filtering yielded a dataset of 211 cases, which we refer to as **PC** and treat as the universe of potential prior cases. Given that the Supreme Court is the highest judicial authority in India, the decisions and observations in these cases serve as binding precedents, making them highly relevant for legal argument synthesis.

Structured representation of cases: Each document in VID, LID, and PC is an unstructured text document detailing the facts of the case, demands and arguments by contesting parties, and the final decision. To convert each dispute in a structured representation (as described in Section 3.1), we perform LLM-based structured summarizationon the similar lines of Pawar et al. (2025). See an example in Appendix A.

Ground truth: As part of the structured summarization, we identify the arguments made by both parties in each dispute in VID and LID. These are the arguments that were *actually presented in court*, and we treat them as the *gold-standard arguments* for evaluation. ArgAssist uses only the facts, disagreement aspects, and demands from the structured representation (excluding the actual arguments) to generate new arguments. These synthesized arguments are then compared against the gold-standard arguments using a newly proposed evaluation metric (Section 4.3). It is important to note that while we do not employ a separate manual annotation process involving legal experts, the gold-standard arguments we use are *implicitly expert-validated*. This is because they originate from real arguments crafted and presented by qualified lawyers in actual legal proceedings. By benchmarking our generated arguments against these authentic, expert-produced arguments, we mitigate the need for any explicit validation by human experts.

Representative Policy documents: Representative insurance policy documents, for obtaining the statutes as discussed in Section 4.1, are as follows:

- *Representative insurance policy for commercial vehicles*
- *Representative insurance policy for personal vehicles*

[2]Earlier website https://confonet.nic.in/, which is now moved to a new website https://e-jagriti.gov.in/judgement

[3]The datasets would be shared on request.

[4]http://www.liiofindia.org/in/cases/cen/INSC/

System prompt: You are a legal expert specializing in [vehicle | life] insurance domain. Your task is to generate proper arguments in the form of claim and premise which should be helpful for parties involved in [vehicle | life] insurance legal case. Also, you are a structured data generator which always responds in valid JSON format.

User prompt: Following is the important information related to a legal case between an insured and an insurance company.
Facts: {facts},
Disagreement aspects: {disagreement_aspects},
Evidences: {evidences},
Testimonies: {testimonies},
Demands: {demands}

###Instruction:
1. Carefully analyse the given information for a legal case.
2. Your task is to construct legal arguments for {party} based on your analysis of the above given information.
3. You must not invent any facts, evidence or any other information which will be different from the above given information.

Response format:
Strictly produce the output in the following JSON format where each element in the list is an argument.

```
[{  "party": {"party"},
    "claim": "<statement making some claim or conclusion>",
    "premises": [
      {"facts": ["<list of one or more facts supporting the case>"]},
      {"evidence/testimony": ["<list of one or more evidences/testimonies supporting the claim>"]},
      {"statute": ["<list of one or more statutes supporting the claim>"]},
      {"precedent": ["<list of one or more precedents supporting the claim>"]},
     {"world knowledge": ["<list of one or more statements based on world knowledge or common sense>"]}]
  },...]
```

Table 2: Prompt used by ArgAssist_base to generate base arguments

System prompt: 1. Following is the important information related to a legal case between an insured and an insurance company.
Facts: {facts},
Disagreement aspects: {disagreement_aspects},
Evidences: {evidences},
Testimonies: {testimonies},
Demands: {demands}
2. Following is the argument made by the {party}: {argument}

User prompt (for prior cases): Following is the argument and court's assessment from a prior case that might be relevant to the current case: {prior_case_info}

Instructions:
1. Your task is to incorporate the prior-case information as a legal precedent into the existing argument made by the {party}, embedding it within the argument's premises in a way that strengthens or strongly supports the claim.
2. Note that prior-case information can only be added in the current argument's premises if and only if it **strongly** supports the claim. **Do not add anything** if the prior case is not strongly relevant.
3. You must not invent any facts, evidence, or other information that is not present in the given case data.

Response format: Strictly produce the output in the following format:
((No)), Nothing can be added in the precedent. | ((Yes)), "precedent" = [<relevant prior-case information supporting the claim>]

User prompt (for statutes): Following is the statute that might be relevant to the current case: {relevant_statute}

Instructions:
1. Your task is to incorporate the statute into the existing argument made by the {party}, embedding it within the argument's premises in a way that strengthens or strongly supports the claim.
2. The statute may be added only if it **strongly** supports the claim. **Do not add anything** when the relevance is weak.
3. You must not invent any facts, evidence, or other information that is not present in the given data.

Response format: Strictly produce the output in the following format:
((No)), Nothing can be added in the statute. | ((Yes)), "statute" = [<relevant statute information supporting the claim>]

Table 3: Prompt used by ArgAssist_final to strengthen a base argument

- *Representative insurance policy for annuity payments*
- *Representative insurance policy for term life insurance*

Structured representation of statutes: To convert the lengthy and semi-structured policy documents into the desired structured form (Section 3.2), we prompted an LLM to break and summarize each policy document into clear *header + content* sections. The model was instructed to retain the factual details exactly as they appear, create a short header that captures the overall meaning of each section,

	Insured	Insurance Company
VID	0.958	0.782
LID	0.972	0.944

Table 4: Evaluation of Retrieval Quality (**RQ**)

and ensure that the sections are mutually exclusive. See an example in Appendix Table 6.

4.2 Baseline approach

As a baseline, we implemented a modified version of the argument generation framework proposed by Tuvey and Sen (2023), as it was the closest existing technique aligned with our task of argument synthesis. Their work introduced a structured approach to generating legal arguments by categorizing case sentences into rhetorical roles, such as facts, arguments, statutes, and precedents, and using those facts as input data and arguments as output data. They fine-tuned GPT-2 and Flan-T5 models on this data, demonstrating the feasibility of automated legal argument generation. Given the similarity in task formulation, we built upon their setup for meaningful comparison.

In our implementation, we constructed the training set using structured summaries from 211 prior cases (PC), each containing facts, points of disagreement, evidence, testimonies, and arguments from both the insurance company and the insured. We split each case into two separate data points, one for each party, while keeping the facts constant and only varying the instruction. This resulted in a total of 422 examples. Unlike Tuvey and Sen (2023), we did not perform additional summarization, as our data was already condensed and structured. We followed their model choice and used the Flan-T5 (Chung et al., 2022) large model (as Flan performed best in their evaluation), introducing special tokens [FACTS] and [ARGUMENT] to guide the model during training and inference. The dataset was split into an 85:15 ratio for training and validation. The model was trained for 7 epochs on an A100 GPU using a simple instruction format: *Generate an argument for the [insurance company/insured] based on the given facts.*

4.3 Proposed evaluation metrics

Metric for evaluating retrieval quality: The strengthening of base arguments is dependent on the quality of retrieved statues and prior cases (Section 3.3.2). As we do not have gold standard relevance judgements for the statutes and prior cases, we conduct an indirect evaluation as follows. We assume that even if one argument in the case gets strengthened due to the retrieved relevant statutes and prior cases, we consider the retrieval to be successful for that case. As the evaluation metric of retrieval quality (**RQ**), we compute the fraction of cases for which retrieval was successful, i.e. at least one argument in those cases was strengthened. The **RQ** metric is computed for each dataset and each contesting party.

Metrics for evaluating synthesized arguments: Firstly, we propose a novel evaluation metric, denoted as **AM** (ArgMatch), for assessing the quality of synthesized *argumentative discourse* by comparing generated arguments against corresponding gold-standard arguments. The computation of this metric for a specific case or dispute is detailed in Algorithm 3. First, a *bipartite matching* is established between the generated and gold-standard arguments using cosine similarity between their *claims*. Then, for each pair of matching arguments, overall argument similarity is computed by additionally considering similarity between their premises. All the gold-standard premises contribute to this similarity if there are matching premises in the generated argument. However, any additional premises in the generated argument do not affect the matching score. To report the performance across the entire dataset, we aggregate the case-level scores using both micro and macro averaging, thereby capturing overall effectiveness as well as per-case consistency.

For AM, the unmatched generated arguments are treated as *false positives*. However, they may not be completely *incorrect* only because similar argument was not presented in the actual court proceedings. Hence, to evaluate quality of such unmatched generated arguments, we used a LLM-as-a-judge strategy to assign a quality score on the scale of 1 to 5 (see the prompt in Appendix Table 7). We report average quality score $\mathbf{QS_{FP}}$ for unmatched generated arguments (FPs) as the second metric for evaluating synthesized arguments.

4.4 Results and Analysis

We use the OpenAI GPT-4.1 LLM for the experiments and text-embedding-3-large embeddings for text similarity. Detailed implementation aspects are covered in Appendix C. Table 4 shows the evaluation results for retrieval quality in terms of the **RQ** metric. It can be observed that in most cases the retrieval quality is good, with very few irrelevant

Dataset	Party	Technique	Micro-averaged			Macro-averaged			$\mathbf{QS_{FP}}$
			AM_P	AM_R	AM_{F1}	AM_P	AM_R	AM_{F1}	
VID	insured	baseline	0.076	0.097	0.086	0.082	0.119	0.092	1.441
		ArgAssist_base	0.271	0.524	0.357	0.275	0.525	0.341	3.155
		ArgAssist_final	**0.288**	**0.558**	**0.380**	**0.293**	**0.568**	**0.365**	**3.622**
	insurance company	baseline	0.152	0.196	0.171	0.178	0.241	0.191	1.445
		ArgAssist_base	0.332	0.627	0.434	0.332	0.629	0.418	3.168
		ArgAssist_final	**0.343**	**0.648**	**0.449**	**0.342**	**0.652**	**0.431**	**3.513**
LID	insured	baseline	0.080	0.113	0.094	0.084	0.128	0.094	1.701
		ArgAssist_base	0.204	0.479	0.286	0.209	0.517	0.286	3.438
		ArgAssist_final	**0.227**	**0.533**	**0.319**	**0.232**	**0.581**	**0.318**	**3.977**
	insurance company	baseline	0.085	0.134	0.104	0.103	0.190	0.124	1.660
		ArgAssist_base	0.241	0.551	0.335	0.243	0.587	0.333	3.472
		ArgAssist_final	**0.260**	**0.595**	**0.362**	**0.262**	**0.638**	**0.360**	**3.771**

Table 5: Evaluation of baseline and proposed argument synthesis techniques; AM indicates the proposed ArgMatch metric and QS_{FP} indicates LLM-as-a-judge scores assigned for unmatched generated arguments (Section 4.3). We have used the lenient version of AM where premise types are not considered (Type_Agnostic = True in Algorithm 3). See Appendix Table 8 for the stricter version of AM.

cases/statutes being retrieved, leading to appropriate strengthening of the arguments.

Table 5 presents a detailed comparison of ArgAssist and the baseline system using the two evaluation metrics, AM (ArgMatch) and QS_{FP}. We report both the variations of the AM metric – (i) using $Type\_Agnostic = True$, i.e., ignoring premise types and (ii) using $Type\_Agnostic = False$, i.e., considering premise types, in Tables 5 and 8, respectively. As the baseline technique does not generate premise types, we consider the type agnostic version of AM for comparison. The results show that the final, strengthened arguments generated by ArgAssist consistently outperform both the baseline and the base arguments across all evaluation settings. The recall component of AM ranges from approximately 0.53 to 0.65, indicating that ArgAssist is able to generate 53–65% of the arguments that were actually presented in court. Considering the inherent complexity of legal argument synthesis, these recall values are promising and demonstrate the practical viability of ArgAssist in real-world scenarios. Appendix B provides an illustrative example, demonstrating how ArgAssist_final is able to synthesize a legally sound argument by augmenting the base argument with relevant premises. A number of additional arguments are also generated that don't have corresponding gold-standard matches and are thus counted as false positives, which lowers the overall AM_P (precision) scores. However, qualitative assessment using QS_{FP} reveals that these unmatched arguments are of decent quality, with average scores in the range of 3.5–4. See an example in Appendix D.

5 Conclusions and Future Work

We presented ArgAssist, an LLM-powered module for the synthesis of *structured argumentative discourse*, designed as a core component of an alternative dispute resolution (ADR) platform. ArgAssist enables parties involved in disputes to generate structured and legally sound arguments grounded in case facts, statutes, and relevant precedents. Our approach leverages multi-stage prompting, retrieval-augmented refinement, and a novel discourse-level evaluation metric (ArgMatch) to systematically assess alignment between synthesized arguments and expert-authored arguments found in real-world court proceedings. Evaluation on two real-world insurance dispute datasets demonstrates the effectiveness of ArgAssist in generating arguments that closely align with those presented in actual court proceedings. From an industrial perspective, ArgAssist has the potential to enhance the efficiency, consistency, and accessibility of dispute resolution processes in domains such as insurance, finance, e-commerce, and other consumer services and to support contesting parties, reducing dependency on legal professionals.

In future, we plan to explore disputes from other domains such as consumer goods, e-commerce, and taxation. We also plan to consider *legal factors* as another premise type in our argument structure, so as to better contextualize our work with existing legal factors based argument generation literature.

Algorithm 3: The proposed evaluation metric ArgMatch (AM) to compare generated arguments for a specific case/dispute with the corresponding gold-standard arguments

Input: (i) $\mathcal{G} = [A_{\text{gold}}^{(1)}, \ldots, A_{\text{gold}}^{(k)}]$: list of k gold-standard arguments;
(ii) $\mathcal{A} = [A_{\text{gen}}^{(1)}, \ldots, A_{\text{gen}}^{(k')}]$: list of k' generated arguments;
(iii) $Th \in [0, 1]$: minimum similarity threshold for claims (default: 0.6);
(iv) $\lambda \in [0, 1]$: relative importance weight for claims vs. premises (default: 0.5);
(v) $Type\_Agnostic \in \{\text{True}, \text{False}\}$: whether to ignore premise types for computing ArgMatch (default: True);
Output: (AM_P, AM_R, AM_{F1}): precision, recall and F1-score for ArgMatch metric

Compute similarity matrix S such that $S_{ij} = \text{CosSim}(A_{\text{gold}}^{(i)}\text{'s claim}, A_{\text{gen}}^{(j)}\text{'s claim})$;
Compute bipartite matching matrix X such that $X_{ij} = 1$ iff i^{th} gold-standard argument is matched with j^{th} generated argument, maximizing $\sum_i \sum_j S_{ij} \cdot X_{ij}$;
$TP \leftarrow 0$; $FP \leftarrow 0$; $FN \leftarrow 0$; // true positives, false positives, false negatives
foreach (i, j) *such that* $X_{ij} = 1$ **do**
  $TP \leftarrow TP + \text{ArgumentSimilarity}(A_{\text{gold}}^{(i)}, A_{\text{gen}}^{(j)}, Th, \lambda, Type\_Agnostic)$;
$FP \leftarrow k' - TP$; // unmatched generated arguments
$FN \leftarrow k - TP$; // unmatched gold-standard arguments
return $AM_P \leftarrow \frac{TP}{TP+FP}$, $AM_R \leftarrow \frac{TP}{TP+FN}$, $AM_{F1} \leftarrow \frac{2 \cdot AM_P \cdot AM_R}{AM_P + AM_R}$;

Function ArgumentSimilarity$(A_{\text{gold}}, A_{\text{gen}}, Th, \lambda, Type\_Agnostic)$;
Input: $A_{\text{gold}} = \langle C, \{\langle p_1, t_1\rangle, \cdots, \langle p_m, t_m\rangle\}\rangle$: gold-standard argument with m premises;
$A_{\text{gen}} = \langle C', \{\langle p'_1, t'_1\rangle, \cdots, \langle p'_n, t'_n\rangle\}\rangle$: generated argument with n premises;
Here, the tuple $\langle p_i, t_i\rangle$ represents i^{th} premise and its type;
Output: $S(A_{\text{gold}}, A_{\text{gen}})$: matching score between two arguments

claim_match_score $\leftarrow \text{CosSim}(C, C')$;
if *claim_match_score* $< Th$ **then**
  return 0; // Argument similarity is 0 if claims similarity is below minimum similarity threshold
else
  if *Type_Agnostic* **then**
    premises_match_score $\leftarrow \frac{1}{m} \sum_{i=1}^{m} \left(\max_{j=1}^{n} \text{CosSim}(p_i, p'_j)\right)$; // premise types are ignored
  else
    premises_match_score $\leftarrow \frac{1}{m} \sum_{i=1}^{m} \left(\max_{j=1}^{n} \left(\mathbb{I}(t_i = t'_j) \cdot \text{CosSim}(p_i, p'_j)\right)\right)$; // $\mathbb{I}(t_i = t'_j) = 1$ when the premise types are same and 0 otherwise.
  argument_match_score $\leftarrow \lambda \cdot$ claim_match_score $+ (1 - \lambda) \cdot$ premises_match_score;
  if *argument_match_score* $< Th$ **then**
    return 0; // Argument similarity is 0 if overall similarity is below minimum similarity threshold
  return argument_match_score; // returns a real number in the range $[Th, 1]$

6 Limitations

1. **Dataset limitations**: Currently, the datasets (VID and LID) used in our experiments are limited to a single domain, i.e., insurance. Also, the number of disputes in each dataset is limited because of constraints on scraping the disputes from NCDRC website. In future, we plan to extend our experiments to other domains as well as larger number of disputes.

2. **Limited universe of prior cases**: The PC dataset acts a universe of prior cases as relevant precedents in ArgAssist's arguments are retrieved from this set. PC consists of only 211 judgements from India's Supreme Court. Though these judgements provide a good starting point, they may not be covering all the aspects of insurance disputes. Ideally, we should gather a larger set of judgements which can act as a better universe of prior cases. In future, we plan to expand this set by gathering court judgements from some lower courts like High Courts, District Courts, and Consumer Courts.

3. **Number of LLMs used**: Currently, we use only one LLM, i.e., GPT-4.1 in our experiments due to budget constraints. Although, we believe our proposed technique is not specific to any particular LLM, it would be interesting to expand the scope of experiments to include other LLMs of comparable capability such as Claude, Gemini, etc and open-source LLMs such as Llama, Qwen, Mistral, etc.

References

Sergio Torres Aguilar. 2025. "don't teach minerva": Guiding llms through complex syntax for faithful latin translation with rag. *arXiv preprint arXiv:2511.01454*.

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.

Basit Ali, Sachin Pawar, Girish Palshikar, and Rituraj Singh. 2022. Constructing a dataset of support and attack relations in legal arguments in court judgements using linguistic rules. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 491–500, Marseille, France. European Language Resources Association.

Basit Ali, Sachin Pawar, Girish Palshikar, Anindita Sinha Banerjee, and Dhirendra Singh. 2023. Legal argument extraction from court judgements using integer linear programming. In *Proceedings of the 10th Workshop on Argument Mining*, pages 52–63, Singapore. Association for Computational Linguistics.

Basit Ali, Anubhav Sinha, Nitin Ramrakhiyani, Sachin Pawar, Girish Keshav Palshikar, and Manoj Apte. 2026. Argumentation and judgement factors: LLM-based discovery and application in insurance disputes. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2789–2804, Rabat, Morocco. Association for Computational Linguistics.

David Ashton. 2024. Alternative dispute resolution. In *European Parliamentary Research Service (EPRS)*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, and 16 others. 2022. Scaling instruction-finetuned language models. *Preprint*, arXiv:2210.11416.

Mohamed Elaraby and Diane Litman. 2022. Arglegalsumm: Improving abstractive summarization of legal documents with argument mining. *arXiv preprint arXiv:2209.01650*.

Lucile Favero, Daniel Frases, Juan Antonio Pérez-Ortiz, and Tanja Käser. 2025. ELLIS Alicante at CQs-gen 2025: Winning the critical thinking questions shared task: LLM-based question generation and selection. In *Proceedings of the 12th Argument mining Workshop*, pages 322–331, Vienna, Austria. Association for Computational Linguistics.

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

Morgan Gray, Li Zhang, and Kevin D Ashley. 2025. Generating case-based legal arguments with llms. In *Proceedings of the 2025 Symposium on Computer Science and Law*, pages 160–168.

Xinyu Hua, Zhe Hu, and Lu Wang. 2019. Argument generation with retrieval, planning, and realization. *arXiv preprint arXiv:1906.03717*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Hao Li, Viktor Schlegel, Yizheng Sun, Riza Batista-Navarro, and Goran Nenadic. 2025. Large language models in argument mining: A survey. *arXiv preprint arXiv:2506.16383*.

Kaveh Eskandari Miandoab and Vasanth Sarathy. 2024. "let's argue both sides": Argument generation can force small models to utilize previously inaccessible reasoning capabilities. In *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*, pages 269–283.

NITI-Aayog. 2021. Designing the Future of Dispute Resolution: The ODR Policy Plan for India. In *The NITI Aayog Expert Committee Report on ODR*.

Raquel Mochales Palau and Marie-Francine Moens. 2009. Argumentation mining: the detection, classification and structure of arguments in text. In *Proceedings of the 12th international conference on artificial intelligence and law*, pages 98–107.

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.

Alexander Rogiers, Sander Noels, Maarten Buyl, and Tijl De Bie. 2024. Persuasion with large language models: a survey. *Preprint*, arXiv:2411.06837.

Sougata Saha and Rohini K Srihari. 2023. ArgU: A controllable factual argument generator. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8373–8388.

Oscar Tuvey and Procheta Sen. 2023. Automated argument generation from legal facts. *Preprint*, arXiv:2310.05680.

Preetika Verma, Kokil Jaidka, and Svetlana Churina. 2025. "reasoning" with rhetoric: On the style-evidence tradeoff in llm-generated counter-arguments. In *Proceedings of the International AAAI*

Conference on Web and Social Media, volume 19, pages 1966–1989.

Douglas Walton, Christopher Reed, and Fabrizio Macagno. 2008. *Argumentation schemes*. Cambridge University Press.

Gechuan Zhang, Paul Nulty, and David Lillis. 2023. Argument mining with graph representation learning. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, pages 371–380.

Li Zhang and Kevin D Ashley. 2025. Mitigating manipulation and enhancing persuasion: A reflective multi-agent approach for legal argument generation. *arXiv preprint arXiv:2506.02992*.

A Example of structured representation of a input case dispute


```
"facts": [
  "Chiman Patel (respondent no. 2) was the owner of
  truck No. CIL 5248.",
  "The truck was insured with National Insurance
  Company Ltd. for the period 18.1.1993 to 17.1.1994
  .",
  "Sunder Singh, aged about 35 years, was employed
  as a driver of the truck on a salary of Rs. 1,500
  /- per month.",
  "On 9.11.1993, the truck overturned while carrying
   a heavy load of firewood, resulting in the death
  of Sunder Singh.",
  "Respondent nos. 3 to 6 (parents, widow, and son
  of Sunder Singh) filed a claim petition under
  Section 166 of the Motor Vehicles Act, 1988,
  seeking compensation for Sunder Singh's death.",
  "The insurance company contested the claim
  petition."
],
"disagreement aspects": [
  "Whether the accident was caused due to the
  negligence of Sunder Singh (driver) or due to
  mechanical failure (breaking of the arm bolt).",
  "Whether the truck was properly maintained and
  roadworthy or not.",
  "Whether the insurance company's liability is
  limited to the amount prescribed under the Workmen
  's Compensation Act, 1923, or extends to the
  entire compensation awarded by the Motor Accident
  Claims Tribunal.",
  "Whether the insurance policy taken was for 'Act
  Liability' only or for a higher/unlimited
  liability."
],
"evidences": [
  "Insurance policy document showing it was a policy
   for 'Act Liability' only.",
  "Endorsements and terms in the insurance policy
  regarding the extent of coverage and premium paid.
  "
],
"testimonies": [],
"demands": [
  {
    "party": "insurance company",
    "demand": "The liability of the insurance
    company should be restricted to the amount
    prescribed under the Workmen's Compensation Act,
     1923, as the insurance policy taken was for '
    Act Liability' only and not for unlimited or
    higher liability."
  },
  {
    "party": "insured",
    "demand": "The insurance company should be
    directed to pay the entire amount of
    compensation awarded by the Motor Accident
    Claims Tribunal for the death of Sunder Singh,
    as the truck was comprehensively insured."
  }
],
"arguments": [
  {
    "party": "insurance company",
    "claim": "The liability of the insurance company
     is limited to the amount prescribed under the
    Workmen's Compensation Act, 1923, and does not
    extend to the entire compensation awarded.",
    "premises": [
      {
        "fact": [
          "The owner paid only the premium required
          to cover liability under the Workmen's
          Compensation Act, not for higher or
          unlimited liability."
        ]
      },
      {
        "evidence/testimony": [
          "The insurance policy document and
          endorsements specify it is a policy for '
          Act Liability' only."
        ]
      },
      {
        "statute": [
          "Section 147(1)(b) of the Motor Vehicles
          Act, 1988, and its proviso, which state
          that a policy is not required to cover
          liability beyond that arising under the
          Workmen's Compensation Act for employees
          such as drivers."
        ]
      },
      {
        "precedent": [
          "New India Assurance Co. Ltd. vs. C.M.
          Jaya and others (2002) 2 SCC 278, which
          held that in the absence of a clause for
          higher liability and payment of extra
          premium, the insurer's liability is
          limited to statutory liability."
        ]
      }
    ]
  },
  {
    "party": "insured",
    "claim": "The insurance company is liable to pay
     the entire amount of compensation awarded by
    the Tribunal as the truck was comprehensively
    insured.",
    "premises": [
      {
        "fact": [
          "The owner had insured the vehicle and a
          valid insurance policy was in force at the
           time of the accident."
        ]
      },
      {
        "statute": [
          "Sections 147 and 149 of the Motor
          Vehicles Act, 1988, which require the
          insurer to satisfy judgments and awards
          against persons insured in respect of
          third party risks."
        ]
      }
    ]
  }
],
"key_assessments": [
  "The Supreme Court assessed that the insurance
  policy in question was specifically for 'Act
  Liability' only, as indicated by the policy
  document and endorsements, and therefore the
  insurer's liability is limited to that arising
  under the Workmen's Compensation Act, 1923.",
  "The Court reasoned that under Section 147(1)(b)
  of the Motor Vehicles Act, 1988, a policy is not
  required to cover liability beyond that arising
  under the Workmen's Compensation Act for employees
   such as drivers, unless a higher premium is paid
  and the policy expressly provides for such
  coverage.",
```

```
  "The judge observed that the statutory liability
  under the Workmen's Compensation Act is the
  minimum required by law, and any extension of
  liability beyond this must be based on the terms
  of the insurance contract and the premium paid.",
  "The Court cited the precedent set in New India
  Assurance Co. Ltd. vs. C.M. Jaya and others (2002)
   2 SCC 278, affirming that in the absence of a
  clause for higher liability and payment of extra
  premium, the insurer's liability cannot be
  expanded beyond the statutory minimum.",
  "The Supreme Court found the High Court's view
  that the insurer was liable for the entire award
  to be incorrect, as it failed to consider the
  specific terms of the insurance policy and the
  statutory framework.",
  "The judge clarified that if the owner desires the
   insurer to bear unlimited or higher liability, a
  specific policy with additional premium and clear
  terms to that effect is required.",
  "The Court affirmed the quantum of compensation
  and interest awarded by the High Court but
  modified the judgment to restrict the insurer's
  liability to the amount arising under the Workmen'
  s Compensation Act, making the vehicle owner
  liable for the remaining amount."
]
```

B Examples of gold-standard and ArgAssist generated arguments (from LID Dataset)

Gold-standard argument of **insurance company**:

```
"claim": "The insurance claim was rightfully
repudiated due to suppression of material facts
regarding the insured's pre-existing illnesses.",
"premises": [
  {
  "fact": [
    "The contract of insurance is based on utmost
    good faith, requiring full disclosure of
    material facts by the insured."
    ]
  },
  {
  "evidence/testimony": [
    "Investigation report dated 21.01.2013 indicated
     the insured suffered from ARTS syndrome, mild
    mental retardation, CP-Spastic Paraparesis, and
    frequent respiratory tract infections prior to
    the policy."
    ]
  },
  {
  "statute": [
    "Clause 8(3) of the Insurance Regulatory and
    Development Authority allows for statutory
    investigation to verify claim authenticity."
    ]
  },
  {
  "precedent": [
    "P.C. Chacko and Anr. Vs. Chairman, Life
    Insurance Corporation of India & Ors. (AIR 2008
    SC 424) - supports repudiation for non-
    disclosure.",
    "Satwant Kaur Sandhu Vs. New India Assurance Co.
     Ltd. (2009) 8 SCC 316 - emphasizes duty of
    disclosure.",
    "Reliance Life Insurance Company Ltd. Vs.
    Rekhaben Nareshbhai Rathod (2019) 6 SCC 175 -
    upholds repudiation for suppression of material
    facts."
    ]
  }
]
```

Argument generated by ArgAssist_base for **insurance company**:

```
"claim": "The insurance claim was rightfully
repudiated due to alleged suppression of material
facts regarding the insured's pre-existing illnesses
 at the time of filling the proposal form.",
"premises":
[
  {
  "fact": [
    "The insured was diagnosed with ART (Anti-
    Retroviral Therapy), as per the prescription
    dated 11.12.2012 from Shree Balaji Hospital.",
    "The prescription slip dated 11.12.2012
    indicates the insured was suffering from a
    serious illness at the time of death."
    ]
  },
  {
  "evidence/testimony": [
    "Prescription dated 11.12.2012 from Shree Balaji
     Hospital stating the patient was suffering from
     ART.",
    "Undated certificate from Shree Balaji Hospital
    signed by Dr. Banwari Lal Sharma."
  ]
  },
  {
  "world knowledge": [
    "ART is typically prescribed for individuals
    diagnosed with HIV/AIDS, which is a material
    fact in assessing life insurance risk.",
    "Non-disclosure of serious illnesses such as HIV
    /AIDS at the proposal stage is material to the
    insurer's decision to accept the risk."
    ]
  }
]
```

Strengthened argument by ArgAssist_final for **insurance company**:

```
"claim": "The insurance claim was rightfully
repudiated due to alleged suppression of material
facts regarding the insured's pre-existing illnesses
 at the time of filling the proposal form.",
"premises": [
  {
  "fact": [
    "The insured was diagnosed with ART (Anti-
    Retroviral Therapy), as per the prescription
    dated 11.12.2012 from Shree Balaji Hospital.",
    "The prescription slip dated 11.12.2012
    indicates the insured was suffering from a
    serious illness at the time of death."
    ]
  },
  {
  "evidence/testimony": [
  "Prescription dated 11.12.2012 from Shree Balaji
  Hospital stating the patient was suffering from
  ART.",
  "Undated certificate from Shree Balaji Hospital
  signed by Dr. Banwari Lal Sharma."
    ]
  },
  {
  "world knowledge": [
    "ART is typically prescribed for individuals
    diagnosed with HIV/AIDS, which is a material
    fact in assessing life insurance risk.",
    "Non-disclosure of serious illnesses such as HIV
    /AIDS at the proposal stage is material to the
    insurer's decision to accept the risk."
    ]
  },
  {
  "precedent": [
    "In Satwant Kaur Sandhu vs. New India Assurance
    Company Ltd. (2009), the Supreme Court held that
     insurance contracts are contracts of utmost
    good faith (uberrimae fidei), requiring the
    insured to make true and full disclosure of all
    material facts within their knowledge in the
    proposal form. The Court further reasoned that
    any information specifically sought in the
    proposal form is presumed to be material, and
    inaccurate answers entitle the insurer to
```

```
    repudiate liability. The Court found that non-
    disclosure of material facts known to the
    insured, particularly serious illnesses,
    justified repudiation of the contract by the
    insurer. This precedent supports the insurance
    company's claim that suppression of material
    facts regarding pre-existing illnesses, if
    proven, would entitle the insurer to repudiate
    the claim."
    ]
  },
  {
  "statute": [
    "Forfeiture in Certain Events: If any policy
    condition is contravened or if any untrue or
    incorrect statement is found in the proposal, or
     material information is withheld, the policy
    will be void and all claims subject to Section 4
    5 of the Insurance Act, 1938. This statute
    supports the insurance company's claim by
    establishing that non-disclosure or suppression
    of material facts, 2014 such as serious pre-
    existing illnesses, 2014 at the proposal stage
    entitles the insurer to repudiate the policy and
     deny claims."
    ]
  }
]
```


C Implementation details

In the ArgAssist pipeline, we used the OpenAI GPT-4.1[5] LLM and execute all prompts using it. We set the temperature of 0.7 for all prompt executions as the task of argument synthesis is a creative text generation task. However, for structured summarization of cases, we used the temperature setting of 0 as the task requires to accurately extract from and summarize the given documents and hence requires higher determinism. For the retrieval stage in ArgAssist pipeline as well as for ArgMatch (AM) metric, we compute semantic similarity between text pairs using the OpenAI text-embedding-3-large[6] text embeddings. Further, we use the following python packages and corresponding versions in the experimentation: openai (v2.2.0), numpy (v2.3.3) and pandas (v2.3.3).

Regarding the weights used in computing the final score in Algorithm 1, we have decided them such that a higher weight (0.35) is assigned to matching facts and disagreement aspects than weight (0.15) assigned to evidences and testimonies, as per our assumption that a higher match of the facts and disagreements indicates higher case relevance than the match between evidences and testimonies. Also in case of the ArgMatch evaluation, we have the default values for all hyperparameters shown in Algorithm 3 which are as follows: $Th = 0.6$ (minimum similarity threshold for claims), $\lambda = 0.5$ (relative importance weight for claims vs. premises), $Type\_Agnostic = True$ (ignore premise types for computing ArgMatch).

[5] https://platform.openai.com/docs/models/gpt-4.1

[6] https://platform.openai.com/docs/models/text-embedding-3-large

D Example of unmatched generated argument

Following is an example of an unmatched generated argument by ArgAssist_final for **insurance company**, which was not present in the gold-standard arguments, but was assigned a high-quality score of 5 as per QS_{FP}.


```
"claim": "There is no admissible evidence to support
 the allegation that the surveyor demanded a bribe
from the petitioner, and such allegation cannot be a
 ground for enhancing the compensation or
disregarding the surveyor 2019s report.",
"premises": [
  {
  "fact": [
    "The insured has alleged that the surveyor
    demanded a bribe but has not provided any
    supporting evidence or testimony."
    ]
  },
  {
  "world knowledge": [
    "Allegations of misconduct must be substantiated
     by clear evidence before being considered by a
    court or tribunal.",
    "Unsubstantiated allegations do not invalidate
    official documents or reports prepared in the
    due course of proceedings."
    ]
  },
  {
  "precedent": [
    "In New India Assurance Company Limited vs.
    Pradeep Kumar (2009), the Supreme Court held
    that while a surveyor's report is not conclusive
     or binding, it serves as a basis for settlement
     and must be departed from only if justified.
    Unsubstantiated allegations against the surveyor
    , such as demands for a bribe, do not by
    themselves justify disregarding the surveyor's
    report unless supported by credible evidence.
    The Court emphasized that surveyors are
    independent professionals whose assessments are
    generally relied upon in insurance matters, and
    only credible, substantiated evidence can
    undermine their reports."
    ]
  },
  {
  "statute": [
    "The Company's Right to Repair, Replace, or Pay
    Cash: The Company may at its own option repair,
    reinstate, or replace the vehicle or part
    thereof and/or its accessories, or may pay in
    cash the amount of the loss or damage. The
    liability of the Company shall not exceed: (b)
    For partial losses, i.e., losses other than
    total loss/constructive total loss of the
    vehicle: actual and reasonable costs of repair
    and/or replacement of parts lost/damaged subject
     to depreciation as per limits specified. This
    statute supports the reliance on the surveyor's
    report for assessing the loss for partial damage
    , as the insurer's liability is limited to the
    actual and reasonable repair costs, and not
    necessarily the full IDV or the workshop
    estimate."
    ]
  }
]
```

Table 6: Structured representation of statutes utilized by ArgAssist_final to strengthen the argument

From vehicle insurance policy:
Header: Personal Accident Cover for Paid Drivers, Cleaners, and Conductors **Content**: In consideration of the payment of an additional premium by the insured as mentioned in the schedule and realization thereof by the Company, it is hereby understood and agreed that the Company undertakes to pay compensation on the scale provided below for bodily injury as hereinafter defined, sustained by the paid driver, cleaner, or conductor in the employment of the insured in direct connection with the insured vehicle, whilst mounting into, dismounting from, or travelling in the insured vehicle, and caused by violent accidental external and visible means which, independently of any other cause, shall within six calendar months of the occurrence of such injury result in: - Death: 100% of the Capital Sum Insured (CSI) - Loss of two limbs or sight of two eyes, or one limb and sight of one eye: 100% of the CSI - Loss of one limb or sight of one eye: 50% of the CSI - Permanent total disablement from injuries other than those named above: 100% of the CSI
From life insurance policy:
Header: Grace Periods, Policy Termination, and Renewal **Content**: A grace period of 15 days is allowed for monthly premium payment mode and 30 days for quarterly and half-yearly modes. There is no grace period for annual mode. The Policy may be terminated at any anniversary by either party with at least three months' written notice. Termination does not affect claims originating before the termination date. The Policy is automatically renewed if the Company issues a receipt for the premium due on the next Policy Anniversary, provided payment is made on time or within the grace period.

System prompt: You are an expert legal reviewer specializing in evaluating any given argument based on the information of the corresponding legal case.
User prompt: 1. Followings are the important information related to a legal case between an insured party and an insurance company. Facts: {facts}, Disagreement aspects: {disagreement_aspects}, Evidences: {evidences}, Testimonies: {testimonies}, Demands: {demands} 2. Following is a legal argument for {party} in JSON format to be evaluated: Argument: {argument}. ### Instructions: 1. Evaluate the above argument made by the {party} based on the following criteria. Answer only in Yes|No: - Logical Coherence: Is the argument sound and free of logical fallacies? - Legal Foundation: Does the argument cite and apply relevant statutes or precedents mentioned in the case information? - Factual Accuracy: Is the argument grounded in the facts provided in the case information? - Persuasiveness: Is the language strong, convincing, and appropriate for a legal setting? 2. Give only those comments which can be helpful to improve the existing argument of {party}. 3. Based on the above criteria, assign a strength score between 1 and 5 where 1 is "very weak" and 5 is "very strong and needs no improvement." ### Response format: Strictly produce the output in the following format: Criteria: Logical Coherence: Yes|No Legal Foundation: Yes|No Factual Accuracy: Yes|No Persuasiveness: Yes|No Comments: Comment1\nComment2\n . . . Argument strength score: 1|2|3|4|5

Table 7: LLM-as-a-judge prompt used for evaluating an argument

Table 8: Evaluation of baseline and proposed argument synthesis techniques; AM indicates the proposed ArgMatch metric. We have used the stricter version of AM where premise types are considered (Type_Agnostic = False in Algorithm 3).

Dataset	**Party**	**Technique**	**Micro-averaged**			**Macro-averaged**		
			AM_P	AM_R	AM_{F1}	AM_P	AM_R	AM_{F1}
VID	insured	baseline						
		ArgAssist_base	0.229	0.444	0.302	0.234	0.441	0.289
		ArgAssist_final	0.251	0.486	0.331	0.256	0.500	0.320
	insurance company	baseline						
		ArgAssist_base	0.286	0.539	0.373	0.287	0.550	0.363
		ArgAssist_final	0.301	0.568	0.393	0.301	0.588	0.383
LID	insured	baseline						
		ArgAssist_base	0.166	0.389	0.233	0.171	0.423	0.233
		ArgAssist_final	0.192	0.450	0.269	0.197	0.488	0.269
	insurance company	baseline						
		ArgAssist_base	0.207	0.474	0.289	0.211	0.517	0.291
		ArgAssist_final	0.235	0.537	0.327	0.238	0.584	0.329