Automated construction contract analysis for risk and responsibility assessment using natural language processing and machine learning

Computers in Industry 166 (2025) 104251

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.sciencedirect.com/journal/computers-in-industry

Automated construction contract analysis for risk and responsibility assessment using natural language processing and machine learning

Irem Dikmen [a,*], Gorkem Eken [b], Huseyin Erol [c], M. Talat Birgonul [d]

[a] University of Reading, Whiteknights Campus, School of the Built Environment, Chancellor's Building, Reading, Berkshire RG6 6AH, UK
[b] Middle East Technical University, Civil Engineering Department, K1 Building, Ankara 06800, Turkey
[c] Postdoc researcher in University of Reading, Whiteknights Campus, School of the Built Environment, Chancellor's Building, Reading, Berkshire RG6 6AH, UK
[d] Emeritus Professor in Middle East Technical University, Civil Engineering Department, K1 Building, Ankara 06800, Turkey

ARTICLE INFO

Keywords:
Automated contract review
Natural Language Processing (NLP)
Machine Learning (ML)
Artificial Intelligence (AI)
Text classification
Construction risk management

ABSTRACT

Construction contracts contain critical risk-related information that requires in-depth examination, yet tight schedules for bidding limit the possibility of comprehensive review of extensive documents manually. This research aims to develop models for automating the review of construction contracts to extract information on risk and responsibility that will provide inputs for risk management plans. Models were trained on 2268 sentences from International Federation of Consulting Engineers templates and tested on an actual construction project contract containing 1217 sentences. A taxonomy classified sentences into Heading, Definition, Obligation, Risk, and Right categories with related parties of Contractor, Employer, and Shared. Twelve models employing diverse Natural Language Processing vectorization techniques and Machine Learning algorithms were implemented and benchmarked based on accuracy and F1 score. Binary classification of sentence types and an ensemble method integrating top models were further applied to improve performance. The best model achieved 89 % accuracy for sentence types and 83 % for related parties, demonstrating the capabilities of automated contract review for identification of risk and responsibilities. Adopting the proposed approach can significantly expedite contract reviews to support risk management activities, bid preparation processes and prevent disputes caused by overlooking risks and responsibilities.

1. Introduction

Contracts play a critical governance role in construction projects by delineating the scope, payments, responsibilities, dispute resolution processes, and other binding terms between the employer and contractor. As Mendis et al. (2015) explain, contract documents communicate the employer's expectations and requirements to the project teams executing the work. Formal agreements between clients and contractors are thus essential for successful project delivery. On the other hand, contracts can become a source of risk if they involve ambiguity about rights and responsibilities, and/or parties to the contract are not aware of the contractual conditions and risk allocation.

According to Akintoye and MacLeod (1997), contractual terms are the second most important driver of risk premiums in the construction sector. This underscores the need to thoroughly review construction contracts from a risk perspective. Contracts intrinsically bear interpretation risks due to the subjective nature of comprehending lengthy texts drafted in natural language (Al Qady and Kandil, 2010). Differences in interpreting the applicable clauses amongst contracting parties can trigger conflicts (Grant et al., 2014). Inadequate definitions or specifications about the scope and procedures might result in disputes and claims (Hayati et al., 2019). Furthermore, employers may impose amended clauses in standard-form contracts to transfer greater liabilities onto contractors (Mendis et al., 2013; Rameezdeen and Rodrigo, 2014). If such one-sided clauses against their interests go unnoticed, contractors face avoidable financial, operational, and legal risks during project execution, leading to further conflicts.

Analyzing construction contracts in a comprehensive way is thus essential. Traditionally, construction professionals manually review contracts to identify risks and risk allocation between the parties which requires substantial expertise and effort. However, as Lee et al. (2019) point out, the limited bidding periods constrain manual reviews of

* Corresponding author.
E-mail addresses: i.dikmen@reading.ac.uk (I. Dikmen), gorkemeken@mail.com (G. Eken), huseyinerol@yahoo.com (H. Erol), birgonul@metu.edu.tr (M.T. Birgonul).

https://doi.org/10.1016/j.compind.2025.104251
Received 27 May 2024; Received in revised form 15 December 2024; Accepted 4 January 2025
Available online 25 January 2025

extensive documents spanning hundreds of pages with interdependent sections. Permitting undetected risks and responsibilities to remain can have severe consequences later. Hence, there is a growing need for automated systems that can accurately analyze contract texts with minimal manual intervention (Chakrabarti et al., 2018). In this regard, Artificial Intelligence (AI) presents a solution to enhance risk management by rapidly identifying the clauses that are related with risk and its allocation. Rule-based and Machine Learning (ML) classifiers using text mining and Natural Language Processing (NLP) approaches can help reliably analyze and categorize text-based information in contract clauses (Shamshiri et al., 2024), which has been a popular research theme. Although there are several previous studies regarding construction contract management as will be discussed Section 2 on literature review, most of the studies are about requirements identification from technical specifications, and automated contract review with limited focus on facilitating risk management processes. In this study, the aim is to demonstrate how information about risk, responsibility and shared ownership can be automatically retrieved so that contract review as a part of the risk management process is facilitated.

This research aims to develop models combining various NLP techniques and ML algorithms for the automated review of construction contract clauses to identify risk, right and responsibility, which are critical for formulating risk management strategies and preparation of risk management plans. The study uses FIDIC (International Federation of Consulting Engineers) standard form of contracts to train models and test them on an actual construction project contract. Sentences within the contract are classified by using a taxonomy of "Headings, Definitions, Obligations, Rights, and Risks" with a parallel labeling scheme assigning the Obligation (Responsibility), Right, and Risk sentences to related parties of the contract as Contractor, Employer, and Shared. Classification accuracy is tried to be improved with techniques like binary partitioning and classifier ensembles.

The remainder of the paper is organized as follows. The literature review synthesizes past applications of NLP and ML in different domains. Next, the research methodology details the overall framework, including dataset development, model implementation with vectorization techniques and algorithms, and performance benchmarking. The key results regarding sentence type and related party classifications are then presented, along with improvements from the binary classification and ensemble method. Finally, the conclusions section discusses major findings, contributions, limitations, and future research directions toward automating construction contract analysis.

2. Literature review

As a subfield of AI, NLP draws from linguistics and computer science to develop algorithms that enable computers to understand, interpret, and manipulate human language by processing textual data (Salama and El-Gohary, 2016). Key application areas of NLP include language translation, text classification, speech recognition, and information extraction. In recent years, the popularity of NLP has surged with the advancement of computational techniques. When coupled with ML algorithms, NLP can extract insights and patterns from unstructured text data (Shamshiri et al., 2024). This review compiles NLP and ML-related studies across different domains grouping them under two clusters and discusses how the current research can fill a gap in the existing literature.

Cluster 1 Requirements Engineering: Several studies have applied NLP techniques to address issues in Requirements Engineering (RE), which refers to the elicitation, analysis, specification, validation, and management of requirements for a system. Studies under this cluster demonstrate NLP's capabilities for ambiguity resolution, automated template checks, and extracting models from text to improve requirement quality. Both ML and rule-based techniques have been implemented with various NLP techniques. Ambiguity in requirements documents can undermine project success, as vague requirements increase the risk of misunderstandings. NLP has been used in RE to automatically detect different forms of ambiguity. Zait and Zarour (2018) developed an NLP approach to detect lexical and semantic ambiguity. Using part-of-speech tagging, requirement normalization, and BabelNet lexical database, they identified words with multiple meanings and sentences prone to multiple interpretations. Yang et al. (2010) focused on detecting coordination ambiguity in requirements text. After preprocessing the sentences with various NLP techniques, they used the LogitBoost ML algorithm with labeled samples to train a classifier for pinpointing ambiguous instances. Huertas and Juárez-Ramírez (2012) introduced the Natural Language Automatic Requirement Evaluator (NLARE) model, which leverages several NLP techniques to evaluate requirements in terms of atomicity, ambiguity, and completeness. Rosadini et al. (2017) utilized General Architecture for Text Engineering (GATE) as an NLP tool to identify ten ambiguity classes concerning the requirement documents of railway signaling systems. In addition to ambiguity detection, Arora et al. (2015) proposed an NLP-based tool called Requirement Template Analyzer (RETA) to assess the conformance of requirements text to predefined templates. The tool flags problematic syntactic structures violating template guidelines. Robeer et al. (2016) extracted conceptual models from textual user stories using heuristic rules. Their NLP-based Visual Narrator tool identified key entities and relationships to create models automatically, facilitating RE.

Cluster 2 Automated contract management: NLP and ML have been leveraged in the contract management domain to enable the automation of laborious manual tasks like contract analysis, and text classification. In general, NLP-based research in the construction contract management domain has generally pursued compliance checking, information extraction, text classification, and assessment of contractual risk due to ambiguity or alterations from standard forms of contract. For contract analysis, Chalkidis et al. (2017) extracted key contract elements like parties, duration, and governing law from 3500 contracts using word embeddings and ML algorithms. Chalkidis and Androutsopoulos (2017) enhanced performance on the same data by employing a deep learning approach. Chakrabarti et al. (2018) developed a framework using paragraph vectors and supervised learning to identify risk-prone clauses in the contract and map them to predefined categories of liability, indemnity, and confidentiality. Regarding text classification, Mok and Mok (2019) categorized court decision sentences to build a domain ontology using NLP and logistic regression. Galser et al. (2018) classified rental contract sentences written in German according to a predefined taxonomy. They used NLP text vectorization techniques and ML algorithms to categorize the sentences into classes such as duties, prohibitions, and definitions. Zhang and El-Gohary (2016) proposed a rule-based NLP approach for automated compliance checking of construction regulatory texts. Moon et al. (2018) used web crawling to extract keywords that summarize international construction documents from various countries. They used an ontology to capture semantic features based on domain knowledge. Al Qady and Kandil (2010) extracted responsibilities from construction contracts with an NLP tool called Concept Relation Identification using Shallow Parsing (CRISP). Salama and El-Gohary (2016) focused on classifying contract clauses as environmental or non-environmental using ML algorithms on an NLP-processed dataset. Jung et al. (2024) developed a Bidirectional Encoder Representations from Transformers (BERT)-based NLP model to automatically link construction schedule activities to Uniformat classes. For the assessment of contractual risk, Lee et al. (2019) detected modified clauses in FIDIC-based contracts, which could be disadvantageous to contractors using syntactic and semantic rules. Lee et al. (2020) used a similar rule-based approach to determine missing contract clauses that actually favor contractors. Zhou et al. (2023) developed an NLP and deep learning-based method to intelligently detect missing clauses in construction project contracts. Shuai (2023) proposed a rational-augmented NLP framework to identify unilateral contractual change risks in construction projects.

The current study differs from the previous studies in Cluster 1 and

Cluster 2 that it aims to automate contract review process by providing information on clauses on risk, responsibility and right and allocated parties Contractor, Employer and Shared as an input for risk management plans in FIDIC types of contracts. Most similar studies within this domain are carried out by Moon et al. (2022) and Pham and Han (2023). Moon et al. (2022) utilized BERT to classify clauses in construction specifications into various risk categories, including payment, temporal, procedure, safety, role and responsibility, definition, and reference. Using 2807 clauses from 56 construction specifications, BERT-based clause classification model returns performances with high accuracy. This study used specifications from two highway projects and 5 national/regional standards from Australia, UK and USA to identify risk in various categories, whereas our study does not cover identification of risk from technical specifications but aims to cluster risk and responsibility information using a standard form of contract which is FIDIC. Pham and Han (2023) used 2586 clauses from 10 FIDIC-based construction contracts to develop a multitask model that simultaneously performs classification tasks for risk identification (6 categories), allocation (4 categories) and response (5 categories). They argued that performance of their multitask model exhibited higher performance than single task models. Our study is different as it automates the contract review process to retrieve clauses using a taxonomy that relates right, responsibility and risk, and allocates to an ownership category of Contractor, Employer or Shared which can further be used to recall/analyse specific clauses, for example about “Shared Risk” and/or “Responsibility of Employer” which was not done in Pham and Han (2023). Unlike the previous studies, our study also uses ensemble methods for improved performance as will be discussed in the following sections. Automated contract review model to support risk management process by providing clauses on risk, responsibility and allocation, coupled with the ensemble approach increasing performance of the model differs from the current state-of-the-art.

3. Research objective and methodology

The research objective is to develop a model for automated contract review that can be used as an input for risk management and bid preparation activities in a contracting firm. The overall research methodology (showing the research steps on the left and methods on the right) comprises of several key steps, as depicted in Fig. 1. With a problem statement developed around the need to expedite construction contract risk analysis amid bidding time constraints, this research targeted a solution based on automated text analysis leveraging NLP and ML capabilities. Since readily available labeled datasets were lacking publicly, foundational methodology phases established training and test data corpora by extracting sentences from three FIDIC standard forms of contracts, namely Red, Yellow, and Silver Books, along with a real project contract through Python libraries. Sentences were categorized into labels spanning sentence types (Heading, Definition, Obligation, Risk, Right) and related parties (Contractor, Employer, Shared) to enable supervised classification model training. Labels underwent selective expert review to validate categorization quality. With thoroughly labeled datasets established, the methodology shifted to developing an array of ML models as well as defining their performance evaluation metrics. Twelve models combining NLP text vectorization techniques with ML algorithms were trained on sentence type and related party classification predictions based on the curated contract sentence datasets. Accuracy and F1 score were designated key metrics to test model performance. A multi-class classification procedure was implemented by decomposing the categorized labels into binary partitions to improve sentence type prediction accuracy over initial results. Finally, the competitive voting ensemble method integrated top-performing models for each contract text classification task to enhance predictions further. This comprehensive methodology enabled iterative performance gains while thoroughly evaluating the effectiveness of the ML models for the automated analysis of construction contracts. The following sections delve into the details of the research steps.

3.1. Text preparation

Text preparation phases involve transforming original contract PDF documents into textual datasets in spreadsheet form, amenable to ML classifiers. In order to automate this process, a tool aligned to the process model architecture in Fig. 2 was designed using Python programming language.

The initial conversion process utilized the Python PDFMiner library (Shinyama, 2019) to extract text files from the PDF versions of the provided contracts. However, the text outputs contained various extraneous artifacts, including heading breaks, segmented sentences, page numbers, and watermarks, requiring systematic removal before splitting the sentences. As such, four hard-coded rules were implemented sequentially by a custom Python script to clean texts from the extra characters that resulted from the conversion process and do not belong to the main text of contracts. The first rule provided the removal of extra paragraph breaks that divide heading texts from their numbering by replacing them with space. The second rule handled numbering within sentences that induced mid-sentence segmentation by searching for clause/sub-clause numbering conventions delimited by colons or semicolons and replacing intruding line break characters with space to rejoin partial divisions. The third rule addressed page numbers inducing text fragmentation by searching for and removing paragraphs with solely digit strings before and after other paragraphs. Finally, the fourth rule stripped watermarks that emerged as individual characters through isolated paragraphs by searching for and removing all single-character paragraphs across texts. Consequently, these four cleaning rules resulted in a polished text file ready for downstream processes.

Then, sentence extraction was operationalized in two stages. In the first stage, the spaCy Python library (Honnibal et al., 2020) was employed for sentence splitting. An NLP pipeline was created to segment the text documents into sentence units based on dots in the text. After sentence splitting with NLP, the complicated sentences were rearranged through syntactic rules, as proposed by Kim et al. (2020). Accordingly, complex multipart FIDIC sentences were algorithmically broken down into digestible self-contained clauses based on bullets, connectors, and punctuations, increasing the number of sentences by 300–400 per document. These operations resulted in four Pandas Data Frames (The Pandas Development Team, 2020) created in the Python environment for each contract text: 1791 sentences for the FIDIC Red Book, 1726 sentences for the FIDIC Silver Book, 1829 sentences for the FIDIC Yellow Book, 1305 sentences for the actual construction project contract.

After converting the PDF contracts into Excel files, three FIDIC Books were combined in one file containing 5346 sentences to be used as the training dataset, while 1305 sentences in the actual construction project contract comprised the test dataset. However, there were repeating sentences in these datasets (especially in the training dataset due to the same provisions and definitions in different FIDIC Books). Before eliminating the duplicates with a matching algorithm, a rule set was defined to account for minor inconsistencies that inhibited naive exact matching of otherwise identical sentences. In this direction, some words, punctuations, special characters, bullet indicators, numbers, and connectors were removed from sentences, and texts were converted into lowercase to normalize case sensitivity mismatches. Implementing such transformations in systematic order generated a comparative sentence column to yield unique entries, enabling accurate duplication removal through semantic equivalence.

The final datasets thus encompassed Excel files with a unique sentence per row. This exhaustive procedure resulted in 3485 sentences ready for categorization: 2268 from FIDIC Red, Silver, and Yellow Books in the training dataset and 1217 from the construction project contract in the test dataset.

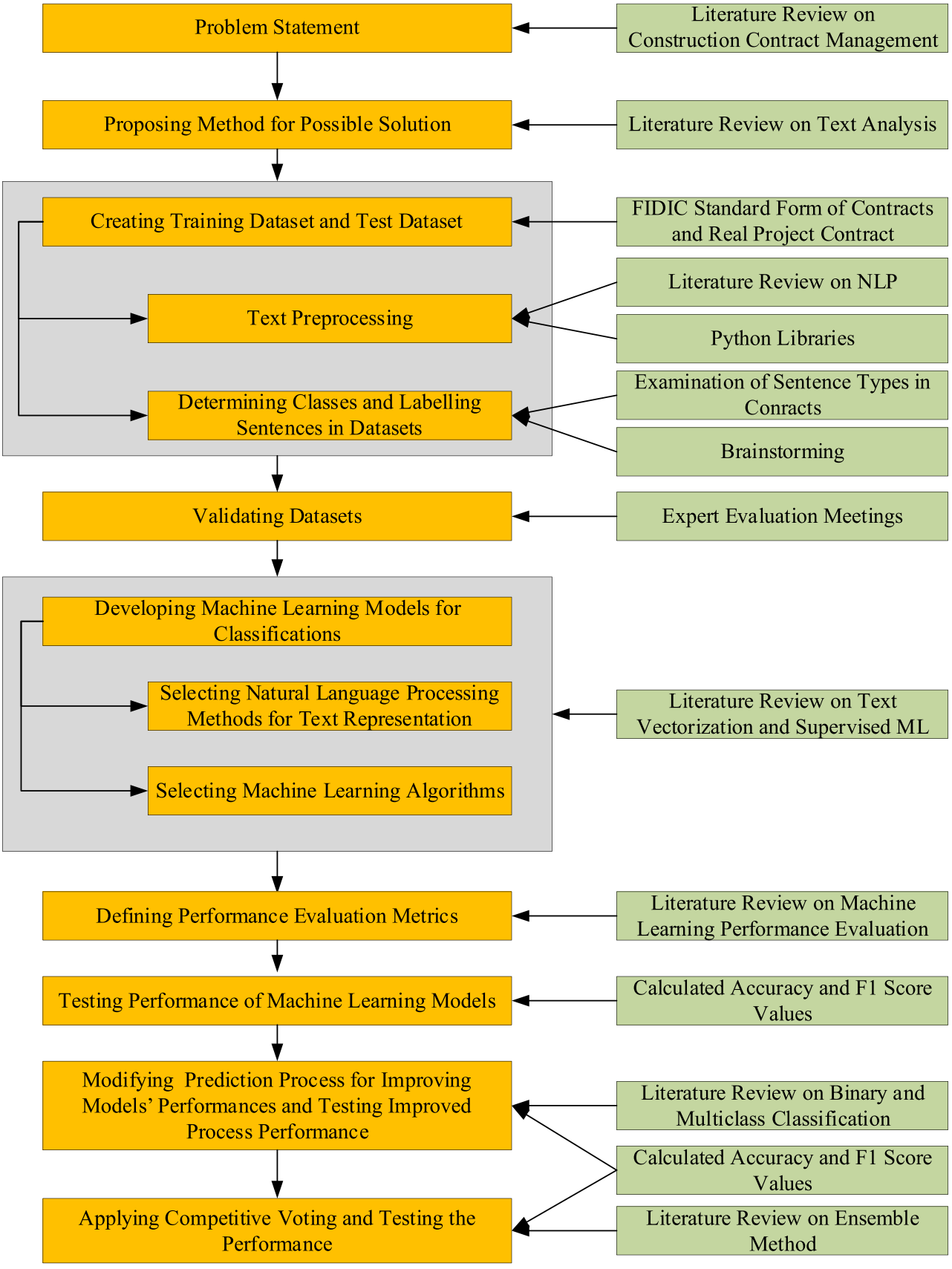

Fig. 1. Research steps.

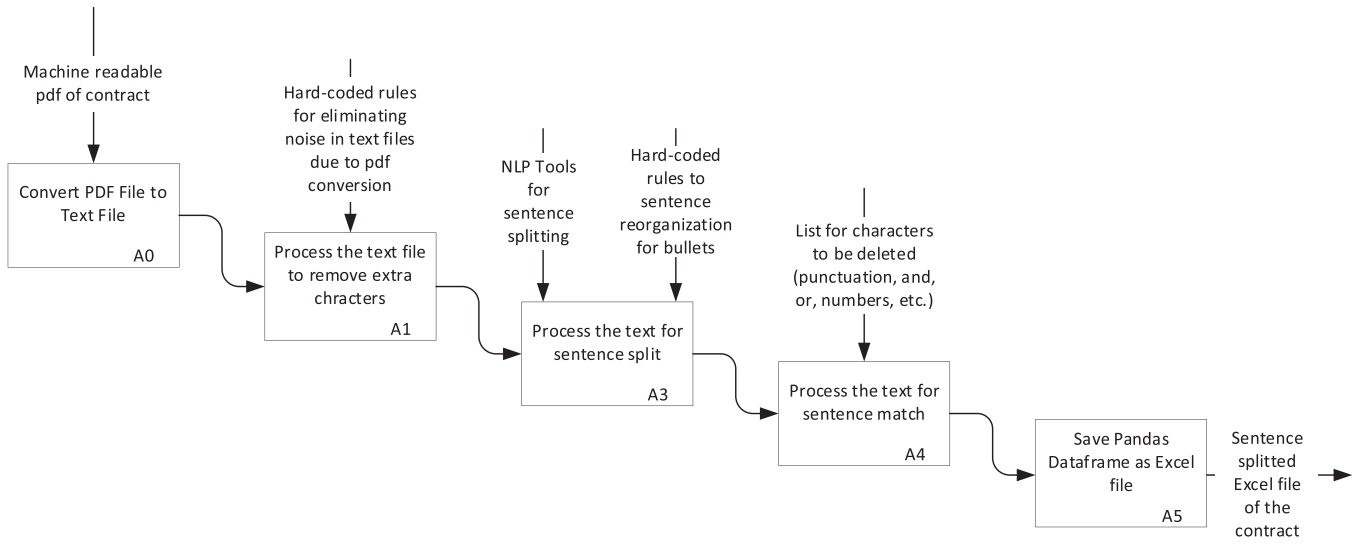

Fig. 2. The process model for converting contract documents to Excel file.

3.2. Dataset labeling

Aligned with the overall goal of creating ML models to automate construction contract analysis, unique sentences extracted previously were systematically labeled for supervised classification training. As shown in Fig. 3, two taxonomic labels were manually compiled using the sentence types of Heading, Definition, Obligation, Risk, and Right, and the related parties of Contractor, Employer, and Shared.

First, sentence types were compiled across all 2268 sentences in the training dataset and 1217 sentences in the test dataset. The Heading and Definition groups comprised structural components and terms, whereas the Obligation, Right, and Risk groups indicated conditions affecting different contractual parties. The categorical distribution of the sentences within these groups is presented in Table 1. Label proportions were fairly similar between the training and test datasets, indicating consistency.

Related party assignments were then attached for Obligation, Right, and Risk sentences only, as Headings and Definitions do not imply any responsibility to any party. As shown in Table 2, the training set comprised 269 Shared items affecting both Contractor and Employer, together with 1044 Contractor-specific and 549 Employer-specific statements, compared to 118 Shared, 617 Contractor-related, and 186 Employer-related elements in the test dataset. The comparison of the distribution ratios between the two sets was again consistent without a significant skew.

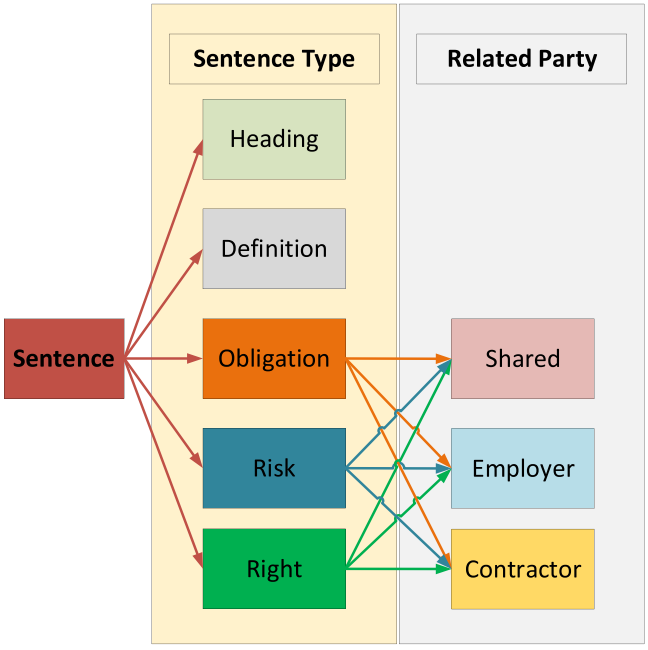

Fig. 3. Labels used to create supervised machine learning models.

Table 1
Categorical distribution of the datasets in terms of sentence type.

Sentence type category	Number of sentences Training dataset	Test dataset
Heading	228	205
Definition	178	91
Obligation	1033	565
Risk	488	242
Right	341	114

This manual labeling effort transformed extracted contract sentences into comprehensively annotated datasets coded for sentence types and related parties, forming categorized datasets for model training and evaluation. Example sentences taken from FIDIC Red Book for each label are given below:

Heading: “4.7 Setting Out”

Definition: “1.1.11 "Contract Agreement" means the agreement entered into by both Parties in accordance with Sub-Clause 1.6 [Contract Agreement].”

Obligation-Employer: “The Employer shall promptly make available to the Contractor all such data which comes into the Employer’s possession after the Base Date.”

Obligation-Contractor: “The Contractor shall be responsible for the adequacy, stability and safety of all the Contractor’s operations and activities, of all methods of construction and of all the Temporary Works.”

Table 2
Categorical distribution of the datasets in terms of related party.

Related party category	Number of sentences Training dataset	Test dataset
Shared	269	118
Contractor	1044	617
Employer	549	186

Obligation-Shared: "Each Party shall advise the other and the Engineer, and the Engineer shall advise the Parties, in advance of any known or probable future events or circumstances which may (b) adversely affect the performance of the Works when completed."

Right-Employer: "The Employer's Personnel shall, during all the normal working hours stated in the Contract Data and at all other reasonable times (a) have full access to all parts of the Site and to all places from which natural Materials are being obtained"

Right-Contractor: "The Contractor shall be entitled to use, for the purposes of the Works, the utilities on the Site for which details and prices are given in the Specification."

Right-Shared: "If either Party is dissatisfied with a determination of the Engineer and (d) thereafter, either Party may proceed under Sub-Clause 21.4 [Obtaining DAAB's Decision]."

Risk-Employer: "If the Contractor suffers delay and/or incurs Cost as a result of a failure by the Employer to give any such right or possession within such time, the Contractor shall be entitled subject to Sub-Clause 20.2 [Claims For Payment and/or EOT] to EOT and/or payment of such Cost Plus Profit."

Risk-Contractor: "The Contractor shall then promptly rectify the error or defect at the Contractor's risk and cost."

Risk-Shared: "Subject to the following provisions of this Sub-Clause, the Contract Price shall be adjusted to take account of any increase or decrease in Cost resulting from a change in or (d) the requirements for any permit, permission, licence and/or approval to be obtained by the Contractor under sub-paragraph (b) of Sub-Clause 1.13 [Compliance with Laws], made and/or officially published after the Base Date, which affect the Contractor in the performance of obligations under the Contract."

3.3. Validation of dataset labels

Although the proportion of the categories showed a satisfactory alignment between the training and test datasets, a comprehensive validation procedure was instituted to affirm the quality of manual labeling through an expert evaluation before final model usage. A total of 280 sentences, involving 10 % of the sentences from each label category, were randomly sampled from the Obligation, Risk, and Right categories across training and test datasets for external review. Headings and Definitions were excluded from the validation study as they are readily identifiable.

Six domain experts currently working in contract management roles in construction companies evaluated the labels. As shown in Table 3, they possess advanced graduate degrees and up to 20 years of tenure in their positions. A partitioned validation methodology separated the participants into control and label groups to prevent confirmation bias.

First, the unlabeled sample of 280 sentences (185 from the training dataset and 95 from the test dataset) was categorized by the label group with the consensus of all three participants. Then, the control group assessed the categories assigned by both the label group and the researchers in a follow-up meeting to ratify them or propose modifications. As a result, the control group logged just eight different labels (six in the training subset and two in the test subset) compared to the researchers, as presented in Table 4. The deviations of 3 % in the training sample and 2 % in the test sample were minor margins of error, indicating substantive data integrity. This multi-phase external validation process enabled robust evaluation of dataset label quality. The limited inconsistencies within reasonable margins provided confidence in the suitability of the categorizations to proceed with implementing supervised ML models.

Table 3
Participant profile of the validation study.

Participant	Education	Experience	Position	Group
Participant 1	M.Sc.	16–20	Chief contract manager	Control
Participant 2	M.Sc.	10–15	Chief contracting officer	Control
Participant 3	Ph.D.	10–15	Senior contract specialist	Control
Participant 4	B.Sc.	5–10	Senior contract specialist	Label
Participant 5	B.Sc.	0–5	Contract specialist	Label
Participant 6	M.Sc.	0–5	Assistant contract specialist	Label

3.4. Preprocessing and vectorization of text data with NLP

Developing effective ML models for contractual text data requires thorough preprocessing to clean noise in sentences, encompassing extra characters like punctuations, cases, and stop words, followed by feature engineering using mathematical vectorization approaches to numerically encode the textual data. As depicted in Fig. 4, sentences in the datasets underwent a sequence of transformations to make them digestible for algorithms. As detailed below, these steps allowed ML models to use meaningful inputs, called X values, for analysis.

Preprocessing employs programmed textual editing techniques to strip away unnecessary elements of sentences so that ML models can focus on the most pertinent data content. Specific techniques applied here included:

- Expanding contractions: Shortened verbal conjugations of words like "can't" need to be expanded to "cannot" to simplify grammatical complexity. However, the datasets of this study did not contain such informal abbreviations as they were derived from legal contract documents.
- Lowercasing: All sentences were converted to lowercase letter formatting using the Natural Language Tool Kit (NLTK) in Python (Bird et al., 2009). This step created consistent casing, as some ML algorithms interpret and process uppercase and lowercase terms distinctly.
- Removing punctuations: The regular expression library in Python stripped out punctuation symbols by replacing 32 common types, like periods, parentheses, brackets, etc., with spaces. This step helped focus the sentences on the words themselves rather than non-alphanumeric characters.
- Removing digits: Despite numbers carrying essential legal and contractual meanings like monetary values or dates, variation in their use across different sentences can confuse ML algorithms regarding the selection of related labels. Thus, digits were globally replaced by a single generic numeric value using the regular expression library in Python.
- Removing stop words: Some words like "this," "they," and "where" frequently appear in sentences without contributing substantive informational value. Python's NLTK filtered out such words based on its pre-defined lists of generic stop words.
- Lemmatization: Using Python's NLTK, words were converted into their simplest root dictionary form. For example, "paying," "paid," and "pays" were all simplified to the root word "pay" to aggregate different inflections of terms.
- Removing extra spaces: All the prior preprocessing steps often introduce extra spacing between words. The regular expression library in Python was used to replace them with single standard spaces.

Upon completion of the preprocessing steps, the data frame containing modified X values in a new column was exported to an Excel file for the vectorization phase, which allows computers to process text data. Six methods described below were used to mathematically encode the preprocessed text into numeric vector representations readily digestible by ML algorithms.

- Bag of Words (BoW): The BoW method represents entire sentences through fixed-length vectors containing the counts of unique words appearing in them. The values of a vector reflect the frequency of these words within the related sentence. BoW logic was implemented through the scikit-learn Python library (Pedregosa et al., 2011) in this study. Using the CountVectorizer function, each sentence was

Table 4
Results of the validation study.

Label	Number of sentences (Whole dataset)		Number of sentences (Training subset)			Number of sentences (Test subset)		
	Training	Test	Researchers	Label group	Control group	Researchers	Label group	Control group
Obligation-Shared	142	81	14	12	13	9	11	10
Obligation-Contractor	624	401	62	59	61	40	39	39
Obligation-Employer	267	83	26	27	26	9	8	9
Risk-Shared	72	23	8	9	8	3	2	3
Risk-Contractor	285	177	28	33	29	17	19	17
Risk-Employer	131	42	13	11	13	5	5	5
Right-Shared	55	14	6	6	7	2	3	2
Right-Contractor	135	39	13	16	14	4	3	4
Right-Employer	151	61	15	12	14	6	5	6
Total	1862	921	185	185	185	95	95	95

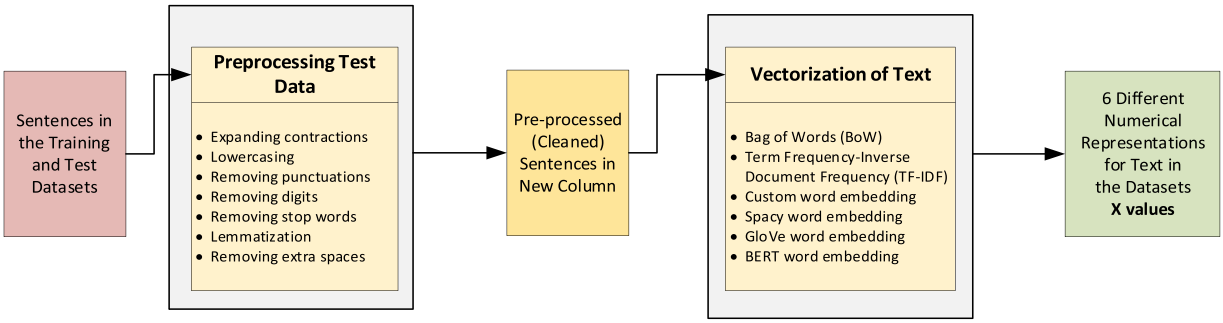

Fig. 4. Steps to convert sentences in the datasets to numerical representation.

converted to a vector with a dimension matching the overall size of 1680 vocabularies across the training dataset.

- Term Frequency-Inverse Document Frequency (TF-IDF): The TF-IDF method also builds vocabulary vectors from sentences but further applies a weighting model to balance frequently and rarely used words. The term frequency portion calculates how often words occur in each text, while document frequency downweighs terms used broadly across many sentences and texts. Multiplication of these terms gives the final TF-IDF score for each word to embed in the vector of the related sentence. Similar to BoW, the scikit-learn Python library was utilized to implement TF-IDF logic. With the TfidfVectorizer function, each sentence was represented by a vector containing 1680 values.

Although BoW and TF-IDF are effective methods for converting text into machine-readable vectors, they lack the ability to capture the context of words. In order to address this limitation, word embeddings have been introduced in 1986 (Landthaler et al., 2016). Word embeddings collectively encompass techniques in NLP that map words or phrases to vectors, with the primary aim of capturing and characterizing the semantic relationships between them based on their distributional properties within large language corpora. Several research groups have developed pre-trained word embedding models, including Word2vec by Google (Mikolov et al., 2013), GloVe by Stanford University (Pennington et al., 2014), fastText by Facebook AI Research (Bojanowski et al., 2016), and BERT by Google (Devlin et al., 2019). In this study, a custom word embedding was employed together with three pre-trained word embeddings, as described below:

- Custom word embedding: Unlike pre-trained embeddings, such as GloVe or BERT, which are trained on large, general-purpose corpora, custom embeddings are developed by training directly on the specific dataset being analyzed—in this case, sentences extracted from FIDIC construction contracts and an actual project contract. This approach allows the model to capture domain-specific language, jargon, and nuanced meanings that may not be well-represented in more general embeddings. By training custom embeddings specifically on the FIDIC contract dataset, we aimed that the resulting vector representations of words and sentences are aligned with the unique linguistic patterns in construction contracts. This specialization may enable the model to better understand contract-specific terminology and thus improves its ability to classify sentences by type and related party. The custom word embeddings in this study were generated using the Word2Vec model, implemented through the Keras library in Python. The training process involved the following steps:

1. Preprocessing the Text Data: The sentences from the FIDIC contracts were first preprocessed, which involved expanding contractions, lowercasing, removing punctuation and digits, removing stop words, and lemmatization. This was essential to clean the text and prepare it for effective embedding generation.
2. Training on the Contract Dataset: The cleaned corpus of sentences from the FIDIC contracts was used as the training data for the Word2Vec model. Word2Vec uses a shallow neural network to learn vector representations for words based on their context in the text. In this case, the model learned word vectors by analyzing the co-occurrence of words in sentences from the contracts, capturing the relationships between key terms and phrases.
3. Dimensionality of Word Vectors: The dimensionality of the word vectors was set to 200, meaning that each word was represented by a 200-dimensional vector. This dimension size was selected based on a balance between computational efficiency and the model's ability to capture complex relationships between words in the contract language.
4. Training Parameters: The Word2Vec model was trained using the Continuous Bag of Words (CBOW) approach, which predicts a target word based on its surrounding context. The model also used a window size of 5, meaning it considered five words to the left and right of the target word to generate the word's vector representation. A skip-gram approach could also be considered for future work, depending on the desired focus on rare words and phrases.

5. Output: After training, each word in the FIDIC dataset was represented by a unique 200-dimensional vector. These vectors were then used as input features for subsequent classification tasks, enabling the machine learning models to make predictions about sentence types and related parties based on the contract's specific language.

- spaCy word embedding: spaCy (Honnibal et al., 2020) is an open-source Python library that provides pre-trained models in various languages. In this study, the "en_core_web_lg" English language model (Explosion, 2022) was employed, which contains 514, 000 unique vectors, each with a dimension of 300, compiled from diverse web texts.
- GloVe word embedding: GloVe provides pre-trained word vectors developed with an unsupervised learning algorithm for broad corpora (Pennington et al., 2014). Despite the existence of different GloVe word embeddings, the Wikipedia model containing 300-dimensional vectors was utilized to match the contractual language of this study better.
- BERT word embedding: BERT is a pre-trained model developed by Google (Devlin et al., 2019). It includes vector encodings based on its massive Wikipedia and book corpora, containing contextual usage of 3300 million words, and has been used by the Google search engine since 2020. The BERT model as the basic model of BERT series, employed in this research has converted each word in the datasets into 768-dimensional vectors.

3.5. Machine learning algorithms

ML enables automated pattern discovery from data to make predictive decisions, with core learning approaches spanning supervised, unsupervised, and reinforcement techniques. Supervised learning entails creating predictive models from labeled training data. Algorithms learn decision rules that link inputs to output categories. New unlabeled data can then be classified based on learned patterns. Unsupervised learning identifies intrinsic structures within unlabeled data. Algorithms cluster or segment data based only on input patterns. Reinforcement learning, on the other hand, optimizes actions in a reward-driven environment. Agents learn behaviors by maximizing the cumulative future reward through trial-and-error interaction, with feedback guiding progressive improvement. This research employed supervised learning using the labeled contract sentences for algorithm training. Specifically, five main algorithms were implemented, as detailed below:

- Logistic Regression is a statistical learning algorithm used for classification tasks. It uses the logistic function to predict the probability of categorical outcomes rather than continuous numeric outputs. Logistic regression is well-suited for text classification as it can handle discrete textual inputs and map them to categorical target classes.
- Support Vector Machine (SVM) is a supervised learning algorithm that can be applied to both classification and regression tasks. It constructs a hyperplane or a set of hyperplanes in high-dimensional space to separate different classes or predict numerical values. The strength of SVM lies in its ability to handle non-linear separable data by using kernel functions to map inputs into higher dimensional feature spaces.
- Decision Tree is a widely used supervised learning algorithm that can perform both classification and regression tasks. It builds a model of decisions or rules using a hierarchical structure of nodes, with each leaf node corresponding to an outcome or class label. Decision trees naturally handle discrete and categorical variables, making them suitable for text classification.
- Recurrent Neural Networks (RNNs) are a type of deep neural network that works well with sequential data such as text and time series. RNNs have cyclic connections that enable them to retain the memory of previous inputs when processing new ones. This makes RNNs useful for tasks such as speech recognition, language translation, and NLP applications.
- BERT is a deep bidirectional transformer model suitable for NLP tasks. It leverages vast volumes of text data to learn contextual word representations. Fine-tuning BERT with pre-trained models, as performed in this study, can substantially improve prediction performances in relatively small datasets.

Thus, five ML algorithms, including three statistical and two deep learning methods, were utilized to predict contractual sentence types and related parties.

3.6. Trained models

This study used 12 models for contract text classification by combining six text vectorization techniques with five ML algorithms, as outlined in Table 5. Specifically, BoW, TF-IDF, and spaCy word embedding were each paired with Logistic Regression, SVM, and Decision Tree algorithms, creating 9 models (Models 1–9). RNN was matched with Keras custom word embedding and GloVe word embedding as Models 10 and 11. Finally, the BERT algorithm was matched with pre-trained BERT word embedding (Model 12). This broad set of combinations allowed the comparison of different alternatives for contract text classification.

These 12 models were implemented using the training and test datasets. The training dataset, derived from FIDIC contracts was split into 90 % training and 10 % test sets. Sentences vectorized with the selected NLP technique (X Values) and their predefined labels (Y Values) served as inputs to train the corresponding ML algorithm of a model. Once the model was trained using the training split, its internal performance was assessed on the test split by generating the confusion matrix and classification report. The model was then evaluated on the test dataset derived from a real project contract. The test dataset sentences vectorized by the same NLP technique were fed into the trained classification model to predict labels. Model predictions were compared to the actual dataset labels to test the external performance of the model. This process allowed systematic development and internal validation of the classification models before testing them on the real contract. Consequently, the most reliable models were identified in terms of the performance evaluation criteria detailed in the next section.

3.7. Performance evaluation method

ML classification models are evaluated based on four parameters: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). These parameters are derived from a confusion matrix, which compares the predicted and actual classes of a dataset. For binary classification with two possible classes (positive and negative), the confusion matrix is a 2×2 table. TP counts positive examples that are correctly classified as positive. In FP, negative examples are incorrectly predicted as positive. TN is accurately predicting negatives as negative.

Table 5
Trained models.

Model	Text vectorization technique	Machine learning algorithm
Model 1	BoW	Logistic Regression
Model 2	BoW	SVM
Model 3	BoW	Decision Tree
Model 4	TF-IDF	Logistic Regression
Model 5	TF-IDF	SVM
Model 6	TF-IDF	Decision Tree
Model 7	spaCy word embedding	Logistic Regression
Model 8	spaCy word embedding	SVM
Model 9	spaCy word embedding	Decision Tree
Model 10	Keras custom word embedding	RNN
Model 11	GloVe word embedding	RNN
Model 12	BERT word embedding	BERT

FN is incorrectly identifying positives as negative. For multi-class problems, as in this research, the confusion matrix expands to a n x n grid of cells, with n being the number of classes. The TP, FP, TN, and FN can be obtained for each class by summing appropriate cells. For example, in a 3-class problem with classes A, B, and C, the metrics for class A are calculated as follows:

- TP is the cell where actual A examples are correctly predicted as A.
- FP is found by summing cells where actual B and C examples are incorrectly predicted as A.
- TN sums all cells where actual B and C examples are not incorrectly predicted as A.
- FN sums cells where actual A examples are incorrectly predicted as B or C.

After deriving the TP, FP, TN, and FN values from the confusion matrix, key classification metrics can be calculated. This study utilized four metrics to evaluate model performance. Accuracy measures the overall performance of the model based on the ratio of TP and TN predictions to all predictions. Precision refers to the ratio of true positives to total positive predictions. It shows how good a model is in positive identification. Recall determines the ratio of true positives to actual positives. It shows the capture rate of a model on positives. Finally, F1 score balances precision and recall as the harmonic mean of the two. These four metrics are calculated according to Eqs. (1)-(4):

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{4}$$

In summary, accuracy and F1 score, derived from precision and recall, constituted the main measures to evaluate the performance of the ML models. Although accuracy is a widely used indicator of overall performance, it can be misleading if classes are imbalanced. For instance, a case with 100 positives and 900 negatives could yield a 90 % accuracy rate with a consistent "negative" prediction. Compared to accuracy alone, F1 score provides a more nuanced understanding of the performance in unevenly distributed classes. Therefore, both metrics were taken into account for performance evaluation.

4. Results

4.1. Initial classification results

The results of sentence type classification (Heading, Definition, Obligation, Risk, and Right) for all 12 models on the internal test split and external test dataset are presented in Table 6. The worst-performing model was Model 9 using the Decision Tree algorithm. It had the lowest accuracy and F1 score across both the test split and test dataset. In contrast, Model 12 leveraging BERT word embeddings and the BERT deep learning algorithm attained the highest accuracy of 0.82 and F1 score of 0.79, which were established as benchmark values to improve upon in subsequent steps. Model 5 with TF-IDF vectorization and SVM algorithm also obtained reliable results with 0.80 accuracy and 0.78 F1 score on the test dataset. Although the internal testing of Model 2 and Model 4 achieved results higher than 0.80 for both accuracy and F1 score, these values dropped to 0.73 for F1 score and 0.76 and 0.75 for accuracy, respectively. For other models, the difference between test split and test dataset values was less than 0.06, indicating consistent behavior across different test conditions.

Table 7 shows the results of related party classification (Contractor, Employer, and Shared). Again, Model 9 performed the worst with the lowest accuracy and F1 score on both test sets. The internal test split results of Model 1, Model 2, Model 4, Model 5, Model 10, and Model 11 achieved accuracy and F1 scores of more than 0.80. These values, however, decreased to 0.70 s for accuracy and 0.60 s for F1 score on the external test dataset results. Further investigations revealed that these models were not successful in capturing different representations of the parties in the datasets. In this respect, Model 12 leveraging BERT's context-based predictions was the best model to effectively handle differences between the training and test datasets. The 0.80 accuracy and 0.73 F1 score were targeted for improvement in later steps for related party classification.

4.2. Results with binary classification

With the purpose of enhancing the sentence type classification performance, the multi-class label classification problem was converted into multiple binary classifications using the "one vs rest" method. This involved separating the original five sentence types into four sequential binary label groups, as depicted in Fig. 5. Accordingly, sentences are categorized as either Heading or Clause (covering Definition, Obligation, Risk, and Right) in Label Group 1. Clause sentences were then separated into Definition and Other (Obligation, Risk, and Right) in Label Group 2. Label Group 3 distinguished Obligation from Other (Risk, Right). Finally, Label Group 4 classified Risk and Right. 12 models were trained sequentially across all four label groups. Individual performances of the groups were analyzed before combining predictions into an overall result column in the Pandas Data frame for comparison to the

Table 6
Initial results of sentence type classification.

Model	Internal test split results				External test dataset results			
	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
Model 1	0.79	0.76	0.78	0.77	0.76	0.72	0.76	0.73
Model 2	0.81	0.82	0.82	0.82	0.76	0.73	0.75	0.73
Model 3	0.69	0.75	0.65	0.69	0.70	0.70	0.66	0.63
Model 4	0.83	0.83	0.84	0.83	0.75	0.74	0.75	0.73
Model 5	0.81	0.83	0.79	0.81	0.80	0.79	0.77	0.78
Model 6	0.69	0.75	0.63	0.65	0.70	0.70	0.66	0.65
Model 7	0.71	0.73	0.70	0.71	0.71	0.67	0.67	0.67
Model 8	0.72	0.74	0.66	0.69	0.75	0.70	0.68	0.69
Model 9	0.55	0.38	0.41	0.38	0.62	0.40	0.44	0.41
Model 10	0.66	0.71	0.57	0.58	0.69	0.69	0.60	0.62
Model 11	0.68	0.66	0.60	0.59	0.69	0.62	0.56	0.56
Model 12	0.81	0.81	0.78	0.80	0.82	0.79	0.80	0.79

Table 7
Initial results of related party classification.

Model	Internal test split results				External test dataset results			
	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
Model 1	0.81	0.82	0.80	0.81	0.71	0.66	0.68	0.67
Model 2	0.84	0.85	0.84	0.84	0.69	0.64	0.66	0.64
Model 3	0.69	0.75	0.56	0.55	0.72	0.66	0.68	0.67
Model 4	0.84	0.84	0.85	0.84	0.72	0.65	0.68	0.66
Model 5	0.82	0.85	0.80	0.82	0.77	0.72	0.67	0.69
Model 6	0.67	0.74	0.55	0.54	0.70	0.61	0.45	0.45
Model 7	0.76	0.75	0.73	0.74	0.66	0.58	0.61	0.59
Model 8	0.75	0.74	0.71	0.72	0.70	0.63	0.60	0.61
Model 9	0.61	0.59	0.50	0.52	0.61	0.55	0.47	0.49
Model 10	0.84	0.84	0.81	0.82	0.72	0.64	0.67	0.65
Model 11	0.81	0.82	0.78	0.79	0.73	0.72	0.68	0.67
Model 12	0.85	0.88	0.84	0.86	0.80	0.79	0.69	0.73

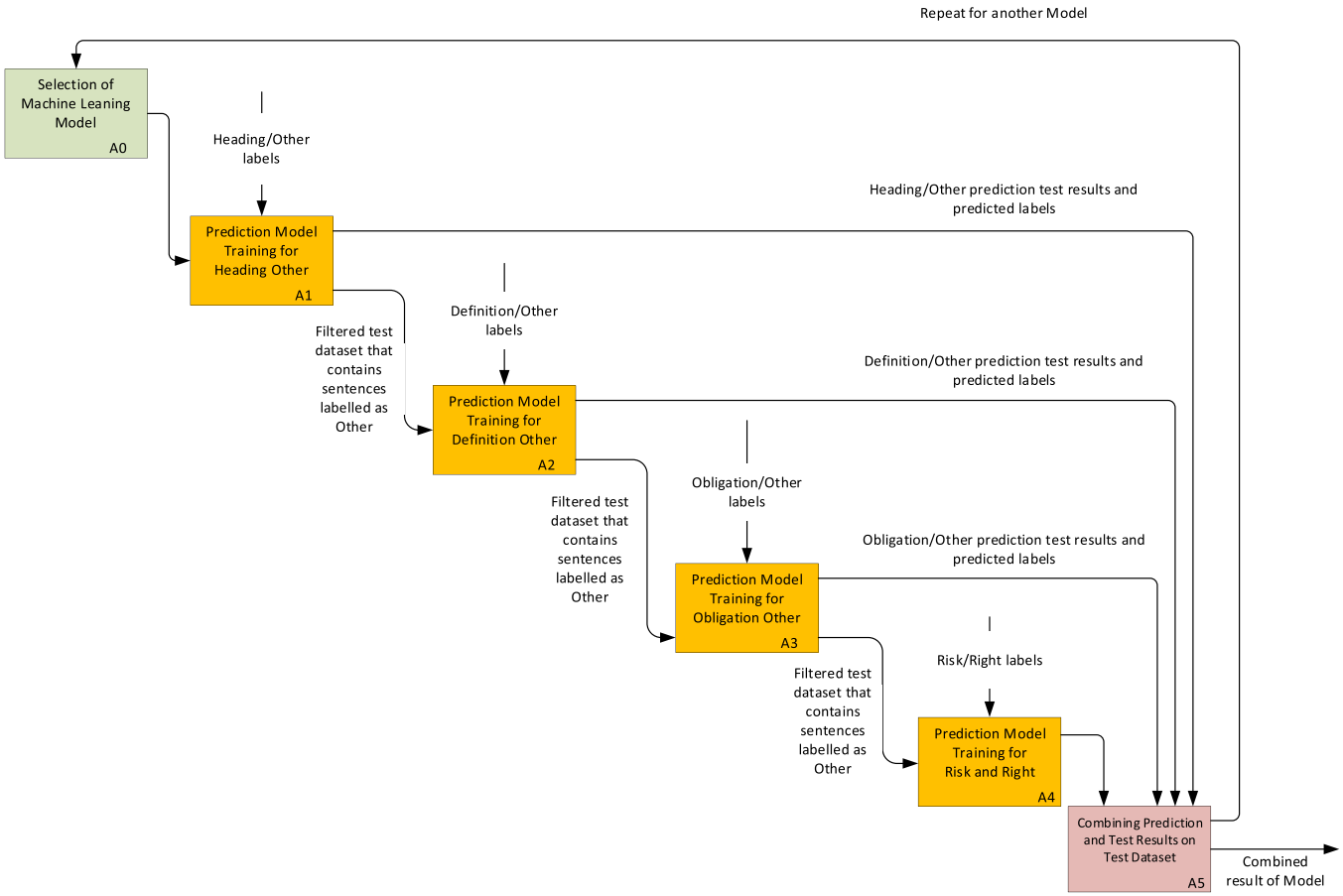

Fig. 5. Conversion process of multi-class classification to binary classification for sentence type labels.

initial classification results.

According to the results of Label Group 1, headings represented the simplest case of differentiating from clause-based sentences. While the BERT model (Model 12) was the best model with 100 % heading classification accuracy, eight other models (Model 1, Model 2, Model 4, Model 5, Model 7, Model 8, Model 10, and Model 11) attained more than 99 % accuracy on the test dataset. This demonstrates ML models can reliably distinguish headings from clauses in contractual texts. The worst-performing model was the Decision Tree-based Model 9 with 0.94 accuracy and 0.90 F1 score.

For Label Group 2, where definitions were labeled against a broad "Other" class, performance understandably fell but remained strong. All 12 models obtained more than 90 % accuracy, indicating robust identification capabilities for definitions. Particularly, the BERT model (Model 12) and Keras-RNN model (Model 10) both achieved 0.98 accuracy and 0.95 F1 score on the test dataset.

The classification of obligations in Label Group 3 against the "Other" group encompassing risks and rights represented a bigger challenge. Although Model 12 had the highest performance with 0.84 accuracy and F1 score, 8 models (Model 2, Model 3, Model 4, Model 6, Model 7, Model 8, Model 9, and Model 10) scored less than 80 % for both, highlighting the difficulty in identifying the obligation sentences.

The final binary step separated the sentences related to risks and rights, where Model 12 achieved 0.79 accuracy and 0.77 F1 score. Four more models (Model 1, Model 4, Model 5, and Model 10) reached comparable performance levels. Similar to the previous results, Model 9 lagged behind with 0.66 accuracy and 0.52 F1 score. The general drop in classification performance shows that there is still progress to be made

in distinguishing between risks and rights.

Following the individual performance evaluations, the binary classification outputs were combined by checking the label groups sequentially to choose from the Heading, Definition, Obligation, Risk, and Right types for each sentence. As shown in Table 8, Model 1, Model 5, Model 10, Model 11, and Model 12 obtained more than 80 % accuracy on the test dataset through the binary classification approach. F1 scores of these models were also very close to or greater than the benchmark value established in the initial classification results. Although the performance of all 12 models was improved, the best and worst-performing models remained the same. While the BERT model (Model 12) came out on top with 0.87 accuracy and 0.83 F1 score, Model 9, which combined the spaCy word embedding and the Decision Tree algorithm, got the lowest scores for both. There were also remarkable improvements in the RNN variants, Model 10 and Model 11, with significant increases in accuracy values and f1 scores. Thus, binary classification proved an effective strategy to boost sentence type prediction in construction contracts.

4.3. Results with ensemble method

Ensemble methods leverage the combined capabilities of multiple ML models to improve classification performance over individual models. Voting is a simple yet powerful ensemble technique where the predictions from selected models are pooled together (Zhou, 2012). Despite the presence of different voting classifiers, this study employed a competitive voting scheme tailored for multi-class labeling (Zhang et al., 2020). As demonstrated in Fig. 6, competitive voting relies on the top three highest-performing models. If all models agree, the unanimous class is selected. If two models select the same class, that majority class is chosen. Finally, if all models disagree, the prediction of the model with the highest individual accuracy prevails.

The competitive voting process began by identifying the top three models for each classification task based on their individual test dataset results (Table 7 and Table 8). For sentence type classification, the selected models were SVM-based Model 5 leveraging TF-IDF vectorization, RNN Model 11 with GloVe embedding, and BERT-based Model 12. The same models were also chosen for related party classification as they outperformed others. The predictions of these models on the test dataset were fed into the competitive voting algorithm in Fig. 6 to produce the ensemble classifications. This structured procedure exploited the relative strengths of the top models on contractual text classification to optimize the decisions. The ensemble results were then evaluated to determine performance enhancements.

Table 9 compiles the accuracy and F1 score improvements resulting from competitive voting on sentence types. In comparison to the best individual model (Model 12), the ensemble method improved accuracy to 0.89 from 0.87 and F1 score to 0.86 from 0.83.

Applying competitive voting to related party classification produced similar improvements as documented in Table 10. The benchmark values of 0.80 for accuracy and 0.73 for F1 score increased to 0.83 and 0.76, respectively, when the ensemble method was used. These enhancements demonstrate how competitive voting can improve model performance by synergizing different predictions.

Table 8
Test dataset results with binary classification of sentence type.

Model	Accuracy	Precision	Recall	F1 score
Model 1	0.82	0.77	0.79	0.78
Model 2	0.79	0.75	0.77	0.75
Model 3	0.74	0.72	0.68	0.69
Model 4	0.80	0.78	0.77	0.77
Model 5	0.84	0.83	0.81	0.81
Model 6	0.72	0.70	0.66	0.67
Model 7	0.76	0.70	0.71	0.70
Model 8	0.79	0.75	0.72	0.73
Model 9	0.68	0.61	0.52	0.52
Model 10	0.82	0.80	0.79	0.79
Model 11	0.83	0.80	0.78	0.78
Model 12	0.87	0.83	0.85	0.83

4.4. Evaluation of results

By examining class-specific performance, we identified several common causes for misclassification. One of the primary sources of misclassification stems from the inherent overlap in the language used in different categories. Another source of misclassification was observed in sentences that are inherently ambiguous or complex, involving multiple clauses or legal dependencies. Certain categories that share conceptual overlap but differ in subtle legal meaning, such as "Shared Responsibilities" versus "Contractor Responsibilities," also contributed to a notable portion of misclassifications. Sentences that require understanding previous or subsequent clauses (e.g., conditions that depend on other clauses) were more likely to be misclassified.

As shown in the previous sections, the classification performance of the ML models was gradually improved through successive steps. Fig. 7 visualizes the improvements in sentence type classification based on four key metrics. The accuracy improved from 82 % to 87 % after binary conversion, culminating at 89 % with the competitive voting ensemble method. F1 score likewise increased from 79 % to 83 % and 86 % throughout these steps. The performance improvements demonstrate that converting sentence types into binary groups allowed more focused learning for the trained models. Furthermore, the ensemble method optimized aggregate predictions across the top models.

For related party classification, although binary classification was not possible, the progression in Fig. 8 shows the ensemble method outperformed the initial best results achieved with Model 12. Accuracy increased to 83 % from 80 %, whereas F1 score climbed to 76 % from 73 %. The findings confirm that competitive voting is an effective ensemble method for advancing the classification of contractual parties beyond individual models by counterbalancing their limitations.

5. Discussion of key findings

The initial classification results showed that deep learning methods involving contextual word embeddings like Model 12, leveraging BERT embeddings and the BERT deep learning architecture, delivered remarkably higher performance compared to statistical learning methods across both sentence type and related party classifications., underscoring the importance of semantic relationships within contract sentences. This can be attributed to BERT's ability to capture the contextual relationships within sentences through its bidirectional transformer architecture, which is particularly effective in handling complex contract language that involves nuanced meanings and multiple dependencies.

Model 11 (RNN with GloVe word embeddings) performed better than Model 10 (RNN with custom word embeddings). This is likely due to the GloVe embeddings being trained on a massive corpus of general language, enabling the model to capture a wider range of semantic relationships. The custom embeddings, while tailored to the FIDIC dataset, may have missed some broader linguistic patterns captured by GloVe, especially for sentences with more general legal language. However, both Models 11 and 10 benefited from RNN's ability to capture sequential dependencies, making them more suitable for handling contract language, which often involves multiple clauses and conditional statements. Decomposing multi-class tasks into narrower binary classifications improved the accuracy further. Integrating predictions from the top models through the competitive voting ensemble, By aggregating the predictions, the ensemble method compensated for individual model weaknesses, exploited their complementary strengths and enhanced overall predictions better than individual algorithms. Although it is not an immediate replacement for human-based review, attaining 0.89 accuracy and 0.86 F1 score for such a broad classification problem with a

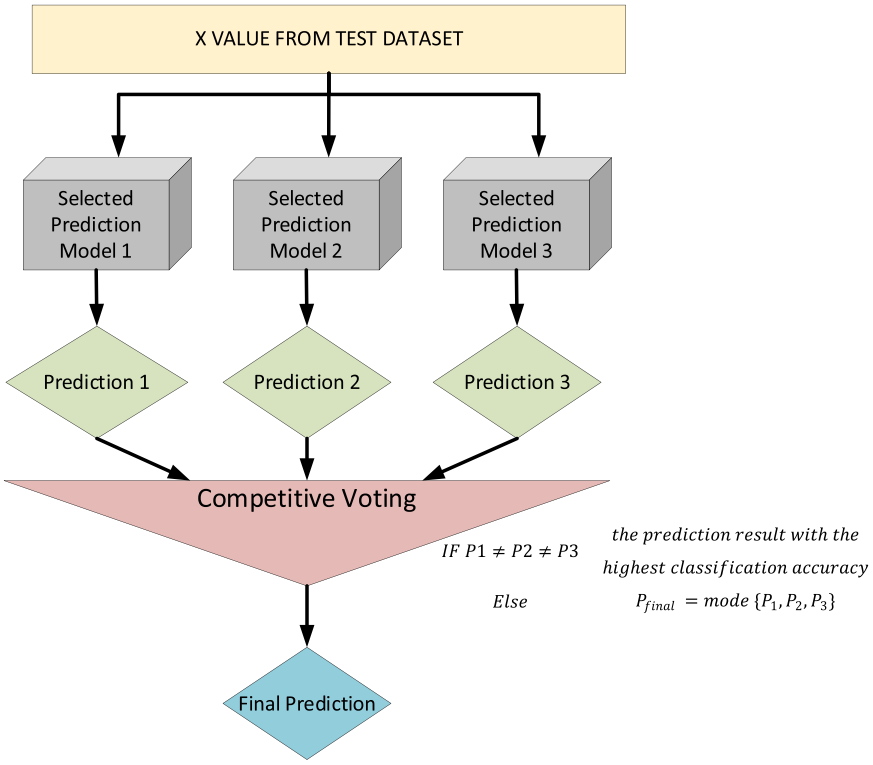

Fig. 6. Competitive voting process.

Table 9
Test dataset results of sentence type classification with ensemble method.

Model	Accuracy	Precision	Recall	F1 score
Model 5	0.84	0.83	0.81	0.81
Model 11	0.83	0.80	0.78	0.78
Model 12	0.87	0.83	0.85	0.83
Competitive voting	0.89	0.87	0.85	0.86

Table 10
Test dataset results of related party classification with ensemble method.

Model	Accuracy	Precision	Recall	F1 score
Model 5	0.77	0.72	0.67	0.69
Model 11	0.73	0.72	0.68	0.67
Model 12	0.80	0.79	0.69	0.73
Competitive voting	0.83	0.80	0.74	0.76

relatively small training dataset demonstrated the potential of automated contract analysis. If we compare our findings with a similar study in this area, they are very close to Pham and Han's (2023) study (mean weighted average F1 is around 0.9) that also aims to extract risk-related information from the contracts as an input for risk management decision-making.

Some key findings that may help development of similar models by other researchers within this domain include:

1. Comparison of Performance of Traditional vs. Deep Learning Models: While traditional models such as TF-IDF with SVM (Model 5) demonstrated strong baseline performance, the deep learning models (especially BERT) exhibited superior results. This is likely due to the greater capacity of deep learning models to capture complex language structures and dependencies, which are prevalent in legal and contractual texts.
2. Comparison of Custom Embeddings vs. Pre-trained Embeddings: The performance gap between Model 11 (RNN with GloVe) and Model 10 (RNN with custom embeddings) suggests that pre-trained embeddings, which capture broader linguistic structures, can sometimes outperform domain-specific embeddings. GloVe's general understanding of semantic relationships across a large corpus provided better overall accuracy, indicating that while custom embeddings may capture domain-specific nuances, they may not always generalize as well to a variety of sentence structures. The decision to use custom embeddings was critical in addressing the unique challenges of understanding contract language. Still, as noted, there are potential benefits in combining both custom and pre-trained embeddings to leverage the strengths of each.

It has to be noted that BERT as the most basic model in the BERT series was used in this study. There are other models such as RoBERTa that ranks high in text classification task evaluation by improving the text encoding of BERT, ELECTRA that uses two contesting neural networks to improve its model performance, DeBERTa as a new algorithm for virtual contesting training and LEGAL-BERT where English legal texts are added to the corpus as a supplementary training set (Fu et al., 2023). Thus, the findings given above should be interpreted considering that performance values are based on only the 12 models where other BERT models were not considered. Moreover, commercial LLMs, including ChatGPT could also provide a solution for the risk-based contract review problem. Several researchers like Wong et al. (2024) and Mialon et al. (2023) argue that LLMs pre-trained in the general domain are not directly applicable for domain-specific tasks such as contract review. Direct application of LLMs to process construction contracts may even lead to unprofessional incorrect outputs due to hallucination (Huang et al., 2023). In this study, when ChatGPT was directly applied as a text classifier the precision, recall, F1 score and accuracy values were found as 0.1478, 0.1579, 0.1324 and 0.3284, respectively which are significantly lower than fine-tuned models, reinforcing the previous researchers' arguments on better performance

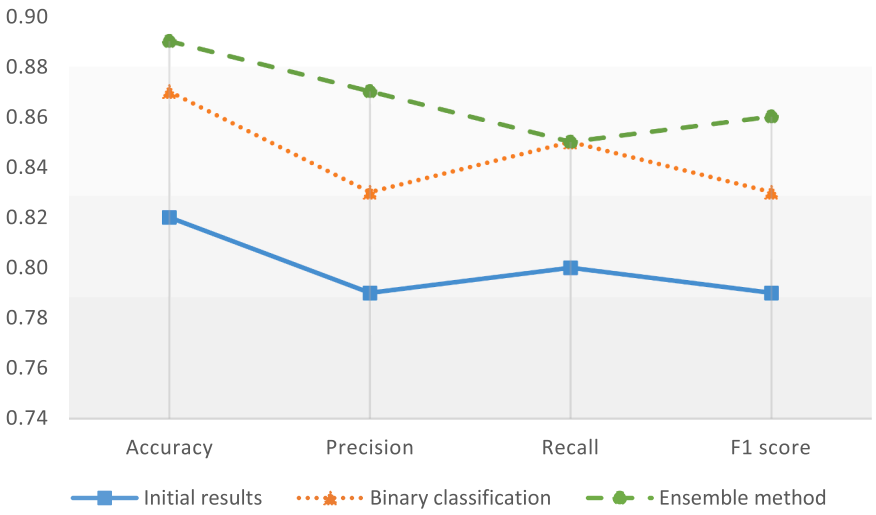

Fig. 7. Improvements in sentence type classification.

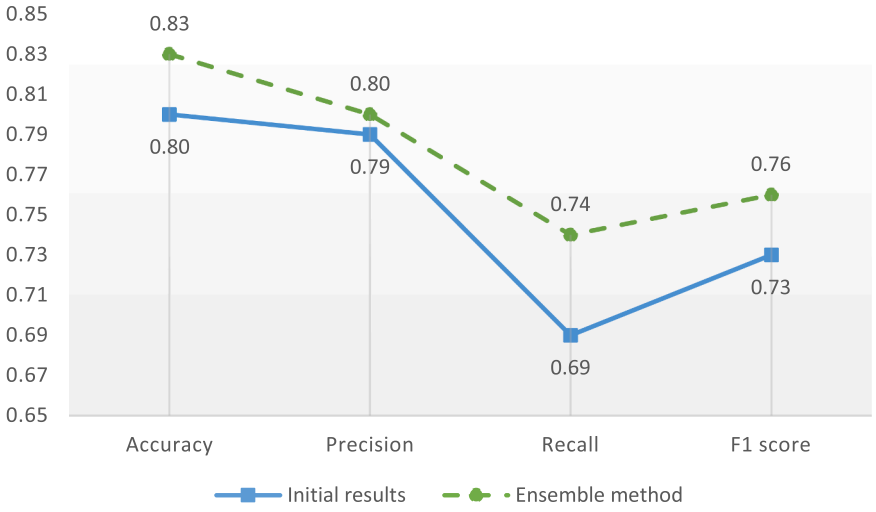

Fig. 8. Improvements in related party classification.

of fine-tuned BERT models over ChatGPT in contract review. On the hand, there are studies in the contract management domain such as reported by Gao et al. (2024), particularly about analyzing long construction contract texts where LLMs are found to be more user-friendly due to their text generation capabilities to provide answers to specific questions meeting the consulting needs of practitioners, which was not considered as a performance criterion in our study. The performance comparison between fine-tuned BERT models and any other LLM like GPT-4 depends on several factors, including the specific task, dataset, and evaluation metrics.

6. Conclusions

Despite the vital role of construction contracts in determining the risks, rights, and obligations assigned to contracting parties, an exhaustive analysis of lengthy contract documents during tight bidding schedules remains a persistent challenge. Manual expert reviews require extensive time and effort while being prone to oversight risks that can result in avoidable disputes and failures. There is an evident need for automated systems to rapidly analyze construction contracts and accurately detect problematic clauses upfront. This research proposed a solution leveraging recent advances in NLP and ML to categorize contract sentences for automated analysis. The research methodology followed sequential steps initiating with the compilation of labeled datasets, followed by training of an array of ML models, and finalized with performance enhancements via binary classification and ensemble method. Over 3000 sentences derived from FIDIC books and an actual construction contract were manually categorized across two taxonomies of sentence types and related parties. With the validated datasets established, 12 models combining diverse vectorization techniques and algorithms were implemented.

These findings carry significant theoretical and practical implications. The findings demonstrate that applying NLP and ML to contract review process to highlight sentences related with risk and responsibility can significantly expedite the risk management process. The comparative evaluation of various NLP techniques and ML algorithms provides insights into their capabilities which may be useful for researchers aiming to develop models for similar tasks. Models like BERT, which excel at understanding contextual language, are particularly well-suited for classification of sentences. The competitive voting ensemble method shows promise as a tool for improving classification accuracy, making it a valuable approach for practitioners aiming to implement automated contract review systems in real-world settings.

Practically, the proposed approach can considerably help construction firms by expediting contract reviews. The ability to rapidly classify clauses and detect risk exposures would allow contractors to assess risks

and make informed bidding decisions within tight timeframes. Using our model, contract review documents can be automatically prepared and used by Risk Management Teams to populate risk registers, assess the level of risk and finally, recommend contingency values based on the level of risk and responsibility. Adoption of AI-based automated contract review can significantly enhance risk management process in large contracting firms. Risk management process can be fully automated by using our automated contract review model as an input to an "intelligent risk registerer" that can use our model's outputs on risk and responsibility to locate different types of risks (such as political, economic etc.) and associated risk owners in risk checklist templates, which is recommended for future research.

This study also has certain limitations. The training data relied solely on FIDIC books, necessitating model re-development for alternative contract types. Subjective manual labeling and category choices may not reflect the general risk perceptions in the industry, limiting the generalizability of findings. Future work should focus on creating richer training datasets across diverse contractual formats to generalize model applicability. While this research focuses on using supervised ML, future studies can employ rule-based techniques and reinforcement learning for comparative purposes. As discussed in the literature review section, it is critical to identify ambiguity in a text written in natural language. Although FIDIC books are made up of well-defined sentences, similar models based on other forms of contracts could incorporate ambiguity detection modules to flag unclear clauses, which requires further research. Another future research can be about comparison of the performance of our recommended ensemble model incorporating NLP and ML algorithms with other LLMs such as RoBERTa, ELECTRA, DeBERTa and LEGAL-BERT and GPT, where comparison should include several criteria not only performance metrics such as accuracy, but also level of user engagement, training time and other utility metrics.

CRediT authorship contribution statement

Erol Huseyin: Writing – review & editing, Supervision, Conceptualization. **Birgonul Mustafa Talat:** Writing – review & editing, Supervision, Project administration, Funding acquisition. **Dikmen Irem:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization. **Eken Gorkem:** Writing – original draft, Formal analysis, Data curation, Conceptualization.

Declaration of Competing Interest

There are no competing interests.

Acknowledgments

This work was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) [grant number 217M471].

Data Availability

Data will be made available on request.

References

Akintoye, A.S., MacLeod, M.J., 1997. Risk analysis and management in construction. Int. J. Proj. Manag. 15 (1), 31–38. https://doi.org/10.1016/S0263-7863(96)00035-X.

Al Qady, M., Kandil, A., 2010. Concept relation extraction from construction documents using natural language processing. J. Constr. Eng. Manag. 136 (3), 294–302. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000131.

Arora, C., Sabetzadeh, M., Briand, L., Zimmer, F., 2015. Automated checking of conformance to requirements templates using natural language processing. IEEE Trans. Softw. Eng. 41 (10), 944–968. https://doi.org/10.1109/TSE.2015.2428709.

Bird, S., Loper, E., Klein, E., 2009. Natural Language Processing with Python. https://www.nltk.org/.

Bojanowski, P., Grave, E., Joulin, A., Mikolov, T., 2016. Enriching word vectors with subword information. ArXiv. https://doi.org/10.48550/arXiv.1607.04606.

Chakrabarti, D., Patodia, N., Bhattacharya, U., Mitra, I., Roy, S., Mandi, J., Roy, N., Nandy, P., 2018. Use of artificial intelligence to analyse risk in legal documents for a better decision support. TENCON 2018 - 2018 IEEE Region 10 Conference, pp. 0683–0688. https://doi.org/10.1109/TENCON.2018.8650382.

Chalkidis, I., Androutsopoulos, I., 2017. A deep learning approach to contract element extraction. In: Wyner, In.A., Casini, G. (Eds.), Legal Knowledge and Information Systems. IOS Pres, pp. 155–164. https://doi.org/10.3233/978-1-61499-838-9-155.

Chalkidis, I., Androutsopoulos, I., & Michos, A. (2017). Extracting contract elements. ICAIL '17: Proceedings of the 16th Edition of the International Conference on Artificial Intelligence and Law, 19–28. https://doi.org/10.1145/3086512.3086515.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1, pp. 4171–4186. https://doi.org/10.18653/v1/N19-1423.

Explosion. (2022). en_core_web_lg-3.4.0 (Version 3.4.0). https://github.com/explosion/spacy-models/releases/tag/en_core_web_lg-3.4.0.

Fu, Y., Xu, C., Zhang, L., Chen, Y., 2023. Control, coordination, and adaptation functions in construction contracts: A machine-coding model. Autom. Constr. 152. https://doi.org/10.1016/j.autcon.2023.104890.

Galser, I., Scepankova, E., Matthes, F., 2018. Classifying semantic types of legal sentences: Portability of machine learning models. In: Palmirani, In.M. (Ed.), Legal Knowledge and Information Systems. IOS Press, pp. 61–70. https://doi.org/10.3233/978-1-61499-935-5-61.

Gao, Y., Gan, Y., Chen, Y., Chen, Y., 2024. Application of large language models to intelligently analyze long construction contract texts. Constr. Manag. Econ. 1–17. https://doi.org/10.1080/01446193.2024.2415676.

Grant, S., Kline, J.J., Quiggin, J., 2014. A matter of interpretation: Ambiguous contracts and liquidated damages. Games Econ. Behav. 85, 180–187. https://doi.org/10.1016/j.geb.2014.01.019.

Hayati, K., Latief, Y., Rarasati, A.D., 2019. Causes and problem identification in construction claim management. IOP Conference Series: Materials Science and Engineering, 469, 012082. https://doi.org/10.1088/1757-899X/469/1/012082.

Honnibal, M., Montani, I., Van Landeghem, S., & Boyd, A. (2020). spaCy: Industrial-strength natural language processing in Python (Version 3.4). https://spacy.io/.

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Liu, T., 2023. A survey on hallucination in large language models: principles, taxonomy, challenges, and open questions. ACM Trans. Inf. Syst.

Huertas, C., Juárez-Ramírez, R., 2012. NLARE, a natural language processing tool for automatic requirements evaluation. CUBE '12: Proceedings of the CUBE International Information Technology Conference, pp. 371–378. https://doi.org/10.1145/2381716.2381786.

Jung, Y., Hockenmaier, J., Golparvar-Fard, M., 2024. Transformer language model for mapping construction schedule activities to uniformat categories. Autom. Constr. 157, 105183. https://doi.org/10.1016/j.autcon.2023.105183.

Kim, Y., Lee, J., Lee, E.-B., Lee, J.-H., 2020. Application of Natural Language Processing (NLP) and text-mining of big-data to Engineering-Procurement-Construction (EPC) bid and contract documents. 2020 6th Conference on Data Science and Machine Learning Applications (CDMA), pp. 123–128. https://doi.org/10.1109/CDMA47397.2020.00027.

Landthaler, J., Waltl, B., Holl, P., Matthes, F., 2016. Extending full text search for legal document collections using word embeddings. In: Bex, In.F., Villata, S. (Eds.), Legal Knowledge and Information Systems. IOS Press, pp. 73–82. https://doi.org/10.3233/978-1-61499-726-9-73.

Lee, J., Yi, J.-S., Son, J., 2019. Development of automatic-extraction model of poisonous clauses in international construction contracts using rule-based NLP. J. Comput. Civ. Eng. 33 (3), 04019003. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000807.

Lee, J., Ham, Y., Yi, J.-S., Son, J., 2020. Effective risk positioning through automated identification of missing contract conditions from the contractor's perspective based on FIDIC contract cases. J. Manag. Eng. 36 (3), 05020003. https://doi.org/10.1061/(ASCE)ME.1943-5479.0000757.

Mendis, D., Hewage, K.N., Wrzesniewski, J., 2013. Reduction of construction wastes by improving construction contract management: a multinational evaluation. Waste Manag. Res. 31 (10), 1062–1069. https://doi.org/10.1177/0734242X13495724.

Mendis, D., Hewage, K.N., Wrzesniewski, J., 2015. Contractual obligations analysis for construction waste management in Canada. J. Civ. Eng. Manag. 21 (7), 866–880. https://doi.org/10.3846/13923730.2014.893907.

Mialon, G., Dessì, R., Lomeli, M., Nalmpantis, C., Pasunuru, R., Raileanu, R., Scialom, T., 2023. Augmented language models: a survey. arXiv Prepr. arXiv 2302, 07842.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv*. https://doi.org/10.48550/ARXIV.1301.3781.

Mok, W.Y., & Mok, J.R. (2019). Classification of breach of contract court decision sentences. Proceedings of the Third Workshop on Automated Semantic Analysis of Information in Legal Text (ASAIL 2019). http://ceur-ws.org/Vol-2385/paper7.pdf.

Moon, S., Shin, Y., Hwang, B.-G., Chi, S., 2018. Document management system using text mining for information acquisition of international construction. KSCE J. Civ. Eng. 22 (12), 4791–4798. https://doi.org/10.1007/s12205-018-1528-y.

Moon, S., Chi, S., Im, S.-B., 2022. Automated detection of contractual risk clauses from construction specifications using bidirectional encoder representations from transformers (BERT). Autom. Constr. 142, 104465. https://doi.org/10.1016/j.autcon.2022.104465.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A.,

Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in python. J. Mach. Learn. Res. 12, 2825–2830.
Pennington, J., Socher, R., Manning, C., 2014. GloVe: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543. https://doi.org/10.3115/v1/D14-1162.
Pham, H.T., Han, S., 2023. Natural language processing with multitask classification for semantic prediction of risk-handling actions in construction contracts. J. Comput. Civ. Eng. 37 (6), 04023027.
Rameezdeen, R., Rodrigo, A., 2014. Modifications to standard forms of contract: the impact on readability. Constr. Econ. Build. 14 (2), 31–40. https://doi.org/10.5130/AJCEB.v14i2.3778.
Robeer, M., Lucassen, G., van der Werf, J.M.E.M., Dalpiaz, F., Brinkkemper, S., 2016. Automated extraction of conceptual models from user stories via NLP. 2016 IEEE 24th International Requirements Engineering Conference (RE), pp. 196–205. https://doi.org/10.1109/RE.2016.40.
Rosadini, B., Ferrari, A., Gori, G., Fantechi, A., Gnesi, S., Trotta, I., Bacherini, S., 2017. Using NLP to detect requirements defects: an industrial experience in the railway domain. In: Grünbacher, P., Perini, A. (Eds.), Requirements Engineering: Foundation for Software Quality, pp. 344–360.
Salama, D.M., El-Gohary, N.M., 2016. Semantic text classification for supporting automated compliance checking in construction. J. Comput. Civ. Eng. 30 (1), 04014106. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000301.
Shamshiri, A., Ryu, K.R., Park, J.Y., 2024. Text mining and natural language processing in construction. Autom. Constr. 158, 105200. https://doi.org/10.1016/j.autcon.2023.105200.
Shinyama, Y., 2019. PDFminer (Version 20191125). https://pypi.org/project/pdfminer/.
Shuai, B., 2023. A rationale-augmented NLP framework to identify unilateral contractual change risk for construction projects. Comput. Ind. 149, 103940. https://doi.org/10.1016/j.compind.2023.103940.
The Pandas Development Team, 2020. pandas-dev/pandas: Pandas. https://doi.org/10.5281/zenodo.3509134.
Wong, S., Zheng, C., Su, X., Tang, Y., 2024. Construction contract risk identification based on knowledge-augmented language models. Comput. Ind. 157–158, 104082. https://doi.org/10.1016/j.compind.2024.104082.
Yang, H., de Roeck, A., Willis, A., Nuseibeh, B., 2010. A methodology for automatic identification of nocuous ambiguity. Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pp. 1218–1226. https://www.aclweb.org/anthology/C10-1137.
Zait, F., Zarour, N., 2018. Addressing lexical and semantic ambiguity in natural language requirements. Fifth Int. Symp. . Innov. Inf. Commun. Technol. (ISIICT) 2018, 1–7. https://doi.org/10.1109/ISIICT.2018.8613726.
Zhang, J., El-Gohary, N.M., 2016. Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking. J. Comput. Civ. Eng. 30 (2), 04015014. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000346.
Zhang, Z., Liu, Y., Hu, Q., Zhang, Z., Liu, Y., 2020. Competitive voting-based multi-class prediction for ore selection. 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE), pp. 514–519. https://doi.org/10.1109/CASE48305.2020.9217017.
Zhou, H., Gao, B., Tang, S., Li, B., Wang, S., 2023. Intelligent detection on construction project contract missing clauses based on deep learning and NLP. Eng., Constr. Archit. Manag. https://doi.org/10.1108/ECAM-02-2023-0172.
Zhou, Z.-H., 2012. Combination methods. In Ensemble Methods (pp. 67–97). Chapman and Hall/CRC. https://doi.org/10.1201/b12207.