Vidler, Tony, McGarry, Kenneth and Baglee, David (2023) Text Mining Legal Documents for Clause Extraction. In: The 19th International Conference on Data Science (ICDATA'23), 24-27 Jul 2023, Las Vegas, USA.

Downloaded from: http://sure.sunderland.ac.uk/id/eprint/16508/

Usage guidelines

Please refer to the usage guidelines at http://sure.sunderland.ac.uk/policies.html or alternatively contact sure@sunderland.ac.uk.

Text Mining Legal Documents for Clause Extraction

Tony Vidler
School of Computer Science
Faculty of Technology
University of Sunderland, UK
bh85am@student.sunderland.ac.uk

Ken McGarry
School of Computer Science
Faculty of Technology
University of Sunderland, UK
ken.mcgarry@sunderland.ac.uk

David Baglee †
School of Engineering
Faculty of Technology
University of Sunderland, UK
david.baglee@sunderland.ac.uk

***Abstract*—Natural Language Processing (NLP) solutions for legal contracts have been the preserve of large law firms and other industries (e.g., investment banks), especially those with large amounts of resources, having both the volume and range of legal documents and manpower to label the training data. The findings suggest that it is possible to use a smaller volume of training contacts and still generate results that are within an acceptable range. Our results show that just 120 training contracts trained on a pre-trained language model can generate results that are within 10% of the same model trained on 3.3 times the volume. In conclusion, smaller law firms could benefit from machine learning NLP solutions for clause extraction.**

***Index Terms*—NLP, Text Mining, Legal Clauses, Deep Learning, BERT.**

I. Introduction

Legal Documents are common in both business and personal worlds, used to create a written legally binding contract between two or more parties. While some of these can have very standard templates, such as Credit Card Terms and Conditions (which become a legal agreement once the card is used), others will be very bespoke, such as the building and running of a nuclear power station. The majority of existing legal documents have been created by a lawyer in a legal firm or from in-house lawyers (large firms would have their legal department), created using a word processor, printed and physically signed. The legal wording created has historically been unique to that contract, where the writing style of the lawyer has come into play, which has resulted in a wide variety of clause texts available for each legal clause [1].

Early software solutions to extract the legal clauses from the documents have been mainly rules-based, requiring specialised teams to review large volumes of documents to look for variations in each clause type and write complex rules to extract these terms from other documents [2]. As machine learning and artificial intelligence have developed over the years, new software solutions have been developed for the legal industry. One of these machine learning technologies, Natural Language Processing (NLP) is becoming common in several software solutions [3]. They have been created to allow law firms to utilise electronic copies of their legal document to find and extract the clause text, which can be used in activities such as Legal Research, Electronic Discovery, Contract Review and Document Automation [4].

† Corresponding author.

In Fig 1 we show the process of generating the training and test documents:

1) Select some documents at random, e.g., 50, and find and label the required clauses in these documents
2) Split the labelled documents into training/testing, e.g., 40/60%, and include the remaining unlabelled documents in the testing dataset
3) Run training using the training dataset, i.e., the labelled documents
4) Evaluate the testing metrics, only using the labelled portion of the testing dataset:
   a. If the metrics are low/unsatisfactory, then select another number of documents, and with the assistance of the predictions from the previous training/testing, label those clauses. Go to step ii.
   b. If the metrics are satisfactory, then the model is complete, so end process.

Natural Language Processing (NLP) is being used in the legal services sector [5], covering the five main processes that legal firms are interested in, which are:

- Legal Research: the process of finding relevant information from legal documents to support legal decision making [6].
- Electronic Discovery: the process of finding relevant files, then finding information from them, which is used to support several use-cases
- Contract Review: the process of reviewing and amending contracts [7]
- Document Automation: the process of creating new legal documents, by utilising existing legal documents. [8]
- Legal Advice: the process where legal advice is provided based on existing legal documents and laws Term Extraction would be required for all of these activities, i.e., to find and extract the relevant legal text within the legal documents [9].

The main commercial products that are available, some of these products have been in existence for a long time, such as the case LexisNexis from the early 1970s which has built a huge database of content (claimed to be 30TB), Thomson Reuters or Bloomberg Law, which all have subscription-based services to access their content. However, newer legal tech firms in the market are capturing market share by offering smarter technologies, such as Machine Learning and NLP to

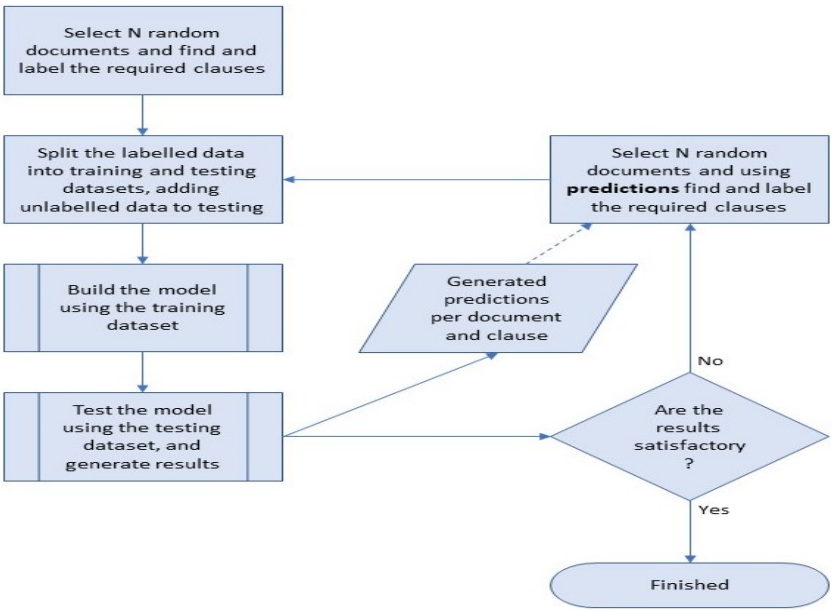

Fig. 1. Overall system operation of data throughput and transformations

improve the accuracy and precision of searches, utilising their clients' legal document repositories.

II. Related work

Previous work has focused on measuring text similarity, proposing a method that combines different measurements, which includes sentence structure, word-to-word and word order similarities [10]. Sentence structure similarity involves parsing the sentence, which includes Parts-of-Speech tagging, Grammar tagging and Named Entity Recognition. The next step is to use the parsed and tagged sentence and generate a semantic representation graph. The graph captures the structure information of the sentence. Word-to-word similarity involves finding similarities between words in the two sentences being compared, the more similar the words, the higher the score. Word order similarity involves finding similar words in the same order.

Key to the process of text extraction of the text elements and how they can be automated for legal contracts [11]. This involved the creation of a labelled dataset of approx. 3,500 English contracts, which have been tagged with 11 types of elements (e.g., contract title, party, governing law). Their dataset has been encoded, so that each word (token) is represented by an integer number (e.g., termination is represented by 3156), and any words not in the vocabulary are represented as UNK. Each token in the labelled dataset is they followed by an element tag. One of the reasons for their work is that contract element extraction is currently a mostly manual exercise, which can be tedious and expensive. They look at Named Entity Recognition and how it relates to their research. One key point is that while NER can find certain entities (dates, amounts, etc.), it doesn't necessarily determine the type of date (e.g., contract start or termination date) or amount types (e.g., monthly rent payments or collateral fees).

Deep learning for NLP using a comparative study of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) [12] was performed by Yin. For the RNN they look at two types, these being the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), which use gating mechanisms that were developed to aid some limitations of the RNN. CNN's can be considered hierarchical and RNNs as sequential architectures. Other types of networks that can be utilised for NLP include Bidirectional RNN, Deep (Bidirectional) RNN and Recursive Neural Networks (RCNN).

A recent development is the Transformer – a deep learning model that uses self-attention mechanisms, a technique that mimics cognitive attention [13]. Transformers were designed and developed by a team of eight Google researchers working on 'Google Brain' or 'Google Research'. The Transformer dispenses with recurrent (RNN) and convolution (CNN) neural networks and is based solely on attention mechanisms. Transformers are designed to process sequential input data and process the entire input all at once and can provide context for any position in the input sequence, and get information about far-away tokens. Transformers are now used in a number of pre-trained language models, which include Generative Pre-trained Transformer (GPT) 2, GPT-3, BERT, RoBERTa, ALBERT. These models are being used to perform a variety of NLP tasks such as language translation, named entity recognition, document generation, and question-and-answering.

The Bidirectional Encoder Representations from Transformers (BERT) model, which utilises the Transformers and has both pre-training and fine-tuning [14]. Unlike other models proposed to date that are unidirectional, where pre-training data is processed from left-to-right, BERT is designed to be

a bidirectional model as its pre-training fuses the left and right context. A key feature is that unidirectional models, such as GPT, are used to learn general language representations that restrict the power of pre-training, such as left-to-right architectures where each token can only attend previous tokens in the self-attention layers of the Transformer, when processing sentence level tasks or token level tasks such as question-and-answering. BERT removes this constraint using a Masked Language Model (MLM) in pre-training, where random tokens are masked, with the objective to predict the masked tokens based on their context.

The model architecture used by BERT is a multi-layer bidirectional Transformer encoder, which is based on the original Transformer. Input can handle a single and a pair of sentences (for question-and-answering) as a one-token sequence. It uses WordPiece embeddings with a 30,000-token vocabulary, and uses two classification tokens; [CLS] as the first token of a sequence, and [SEP] to separate two sentences in a token sequence [15]. Training BERT involves two activities, these being Pre-Training and Fine-Tuning.

The large Transformer models (e.g., BERT Large) have shown performance improvements over the smaller models, but future increases are more challenging due to GPU/TPU memory limitations. proposes techniques to reduce the number of parameters in a model, called A Lite BERT (ALBERT) [16]. The first technique is a 'factorised embedding parameterisation', by decomposing the vocabulary embedding matrix into two smaller matrices allowing the separation of the hidden layers size from the vocabulary embedding size, with the result that the hidden layers can be increased without increasing the number of parameters. The second technique is cross-layer parameter sharing, which prevents the parameters from growing with the depth of the network. As an example, the BERT Large and equivalent ALBERT Large model has 18 times fewer parameters [17].

III. Methodology

Pre-trained Language Models (BERT, DeBERTa and RoBERTa) will be used to determine the preferred model for the selected dataset, using the Question-and-Answering task, with separate testing performed to find the optimal number of EPOCHs. The main set of testing will be to determine the optimal number of training contracts to produce results that are within a reasonable range of the typical 80/20 percent training/testing datasets, the target being within 10% of the measurement metric.

The dataset selected is the Contract Understanding Atticus Dataset (CUAD) [18], which contains 510 clauses, with 25 document types and 41 labelled clauses. The dataset has been set up as a Question-and-Answer dataset, in the style of SQuAD.

For the testing in this project, the number of clauses used will be approx. 20% of the total available. From the list of 41, the following clauses have been selected:

1) Agreement Date: selected as its short clause, and usually in date format, for example: "March 1, 2012"
2) Anti-Assignment: selected as on average it has 1.7 answers per document, and has a longer-than-average answer
3) Document Name: selected as it's a common clause, and has a short answer, a typical answer would be "AGENCY AGREEMENT"
4) Effective Date: select as it's a common clause, and it has a mixture of answer styles, including both a date format "July 1, 2019" and text format, e.g. "This Agreement shall become effective as of the day and year first above written and shall govern the relations between the parties hereto thereafter, unless terminated as set forth in this Section 6."
5) Expiration Date: selected as while it's a date, it usually has a long text answer, e.g. "The initial term of this Agreement (the "Initial Term") shall commence on the Effective Date and shall continue for a period of ten (10) years thereafter."
6) Governing Law: selected as it's a common clause, but each answer is very unique
7) Parties: selected as there are multiple answers for each document, with an average of 5 answers per document
8) Renewal Term: selected as it appears only 34% of documents
9) A Total clause of the above 8 clauses

The measurement success metric the CUAD paper use is the Area Under Precision Recall curve (AUPR), which uses both the Precision and Recall metrics. Using Recall at a defined level, i.e., 80% they find the calculated Precision. The method to find these metrics can use one or more of the 20 best predictions for each clause, and potentially all 20 predictions. Using a sliding scale of confidence (from 100% to 0%) of each prediction, they move down the scale until the Recall has reached the defined level, and then take the corresponding Precision calculation. This compares at the text level.

For this testing, the predictions that have the top confidence value for each clause will usually be only one prediction, but we may have two or more that all have the same confidence value. To measure the success of the model at each training/testing run the F1-Score metric will be used, as it gives a balance between Precision and Recall. This compares at the token level.

Using the F1-Score and best predictions will have an impact on those clauses that have multiple answers, i.e., the Parties clause, which will show lower success results.

IV. Results

The F1-Score for each clause has been run against a set of epochs, the set being [4, 10, 20, 40, 60, 80]. There appears to be no correlation between the answer length or type to the metric, i.e., the Agreement Date has the shortest answer and typically in a date format shows its best F1-Score is at 40 epochs, while Document Name at the second shortest and text format is best at 4 epochs. With 40 epochs having 4 of the 8 clauses plus the Total (of the 8 clauses) as the best metric, 40

epochs will be used for all other tests. The model used was RoBERTa, using 80/430 training/testing contracts.

The results of the model testing are shown in Fig 2. The tests used 40 epochs, and training contracts in the set [50, 100, 200, 300]. The planned models to be tested were BERT, DeBERTa, RoBERTa, and are the official published base models.

Different Clauses appear to perform better in particular models, the Agreement Date works well with both BERT and DeBERTa depending on the number of training contracts used, while the Effective Date and Governing Law work best with the RoBERTa model. Totalling the best metric for each clause at each of the training contracts, as shown in the associated results table, BERT has 5 top F1-Scores, DeBERTa has 8.5, and RoBERTa has 22.5. DeBERTa and RoBERTa shared the top spot for Document Name at 50 training contracts.

The graphs show the BERT is behind the other two models at 50 and 100 training contracts, but gets close to DeBERTa and RoBERTa at 200 and 300 training contracts, but is still slightly behind. DeBERTa is slightly behind RoBERTa at 50 and 100 training contracts, but very close at 200 and 300 training contracts. These can be easily seen by viewing the Total clause, at 300 training contracts we have 55.0%, 53.2% and 55.8% for BERT, DeBERTa and RoBERTa respectively. The selected model will be RoBERTa, which has the best F1-Score for 8 of the 9 clauses at 50 training contracts, and 6 of the 9 clauses at 100 training contracts.

The results of the training contract testing are shown in the graphs below Fig 3, three of the clauses performed very well with the model, these being Agreement Date, Expiration Date and Governing Law, which all reached an F1-Score of 70% or higher at only 100 training contracts. These three clauses all have a high percentage of appearances in the testing dataset, and on average all had less than 1.1 answers per document.

The Anti-Assignment clause only reached an F1-Score of 60% at 100 training contracts and a high of 70% at 400 training contracts. The main reason for this is there are on average 1.7 answers per document, so the False Negatives were high as only the best answer was used. At 100 training contracts, the metrics are TP: 251, TN: 92, FP: 67, FN: 268. The Document Name clause only reached an F1-Score of 58% at 100 training contracts and a high of 65% at 340 contracts. The main reason for this is the answer to the Document Name can appear multiple times in a document, and the best prediction was located in a different token position to the labelled value.

The testing of the Training Contracts performed generated 20 predictions for each clause. These predictions are used to label additional data. For this research, the dataset has all contracts labelled, but in a real-world scenario, only a small portion may be labelled, and additional data may also need to be labelled to continue training and testing. Comparing the predictions to the actual labelled answers and utilising the Recall metric, this data can be used to show what percentage of the actual answers can be found within these predictions. The results of the top 5, 10 and 20 predictions and recall metrics as shown in Fig 4.

For the majority of the clauses, high recall is achieved at just 50 training contacts, and the top 5, 10 and 20 prediction recall metrics are very similar across the range of training contracts, e.g., Agreement Date at 100 training contracts have values of 82.7%, 84.2% and 87.3% respectively, therefore the required answer can be found 8 out of 10 times in these predictions, and usually within the top 5 if not the top answer. The predictions can be used to label additional data, albeit after a minimum number of training runs, which in this example is 50.

Table I and II shows the comparison of results on four different methods:

1) Text Comparison using the predictions above a defined confidence level, to reach the Recall level
2) Token Comparison using the predictions above a defined confidence level, to reach the Recall level
3) Text Comparison using the predicted answers with the best confidence level
4) Token Comparison using the predicted answers with the best confidence level

The Total Clause in table I shows a Precision of 67.3% and 6.9% for the Text and Token comparison respectively, showing that the Text comparison is matching incorrect text. Both have low confidence levels, showing that the majority of the predictions were required to achieve the recall level of 80%. The Total Clause in table I shows a Precision of 77.7% and 68.5% respectively, again showing the Text comparison is matching incorrect text. The confidence level is higher, as only the best prediction was used.

In Fig 5 and table III show the metrics for each of the four calculation methods, for the Total clause. Method 2 has a high volume of False Positives as the confidence level reaches 0% to reach the 80% recall, meaning that each of the 20 predictions was used in the calculations.

Using our preferred metric, method 4 'Token Comparison using the predicted answers with the best confidence level' we have an F1-Score of 53.4%, however, we can point to the exact clause location in the document using method 4, while method 1 could be pointing to a completely different clause, there are less False Positives to filter out of the results, as method 4 has a confidence level of 57% while method 1 only has a level of 5.0%.

The model testing, using 50, 100, 200 and 300 training contracts, showed that potentially good results were available at lower training contracts. After the main training contracts, it became clear that 120 training contracts generated results that almost matched the main objective, being within 10% of the results at 80% (400) of training contacts, which are shown in table IV. Exploring alternative approaches for the Parties and Document Name clauses was interesting, and generated improved results for the individual clauses, but didn't change the overall Total. Using the alternative approaches would be for each end user to determine.

An alternative for the Parties clause may be better. Rather than label each party name, party alias, party role, etc., label the whole party clause, which could be for each party, or all parties. It is likely that using the Token start and end position

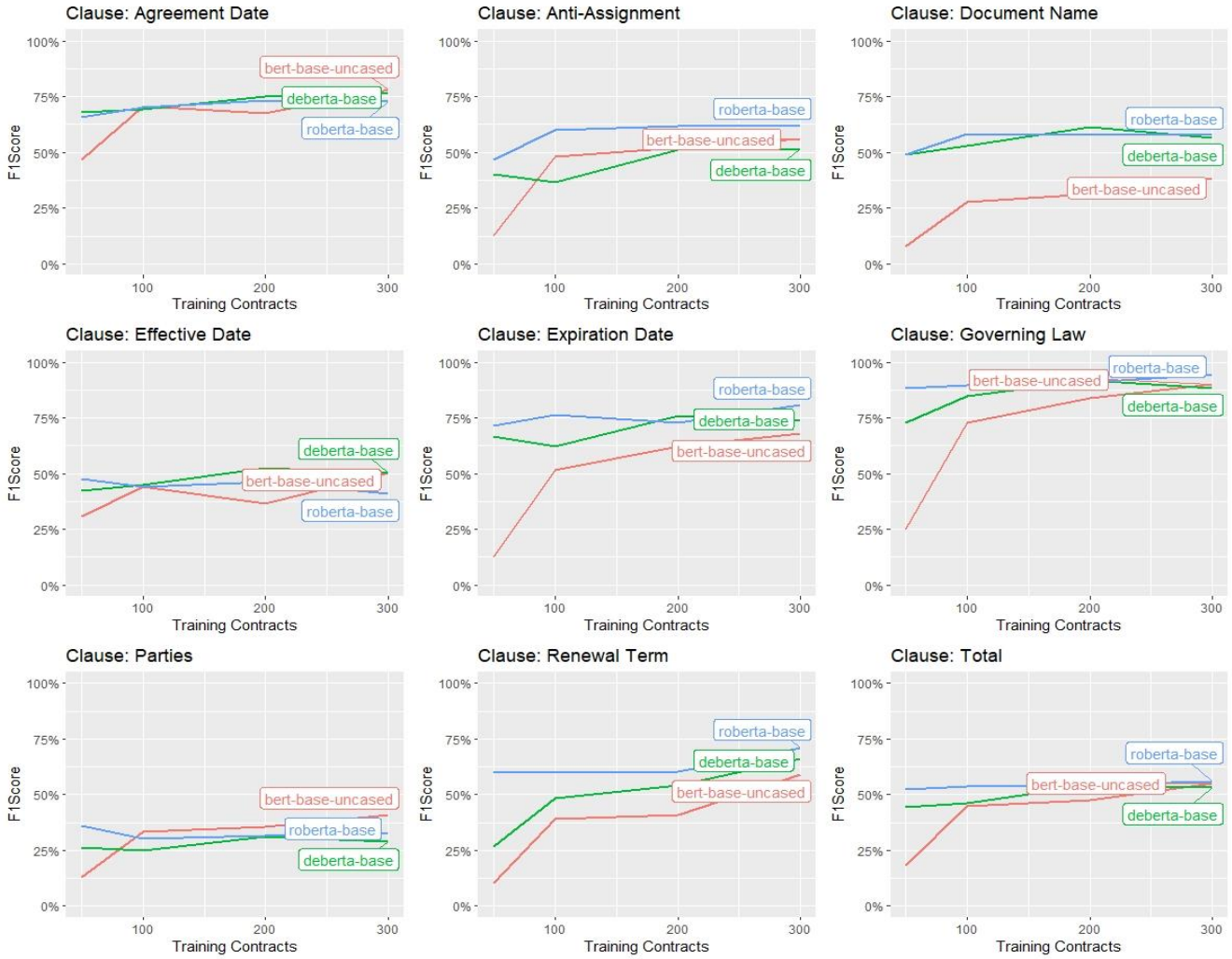

Fig. 2. Model testing results on bert-base, deberta and roberta-base

to validate the labelled answer, rather than using the Text commonality, is the ethically correct decision, for two reasons, it shows that the model is finding the correct clause text, and it required to show the lawyer the exact position of the predicted answer.

V. Conclusions

The research perform has demonstrated that legal clause extraction is possible with training performed on a smaller dataset than would traditionally be used. This would allow the smaller legal firms (either standalone or as a group of small firms) to use machine learning in the form of pre-trained language models. This has the benefit of them potentially bidding for larger items of work, that they would traditionally not have been able to perform due to the limited resources available to them. The experimental work demonstrated, the more training data available results in a better model, which generates improved predictions.

The research has also shown that the predictions of test data can be used to create additional labelled training data. For future work we shall review smaller models, testing a smaller base model may produce improved results. We will enhance the pre-training, which mainly used BookCorpus and English Wikipedia, with the inclusion of a wide range of legal documents. Such as UK Legislation (Legislation.gov.uk, 2022); EU Legislation (Europa.eu, 2022); US Contracts (Sec.gov, 2014) and US Case Law which has 360 years of US case law, containing nearly 7 million unique cases.

Acknowledgment

We would like to thank the two anonymous reviewers for their comments for improving this paper.

References

[1] V. Aggarwal, A. Garimella, B. V. Srinivasan, A. N, and R. Jain, "ClauseRec: A clause recommendation framework for AI-aided contract authoring," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and

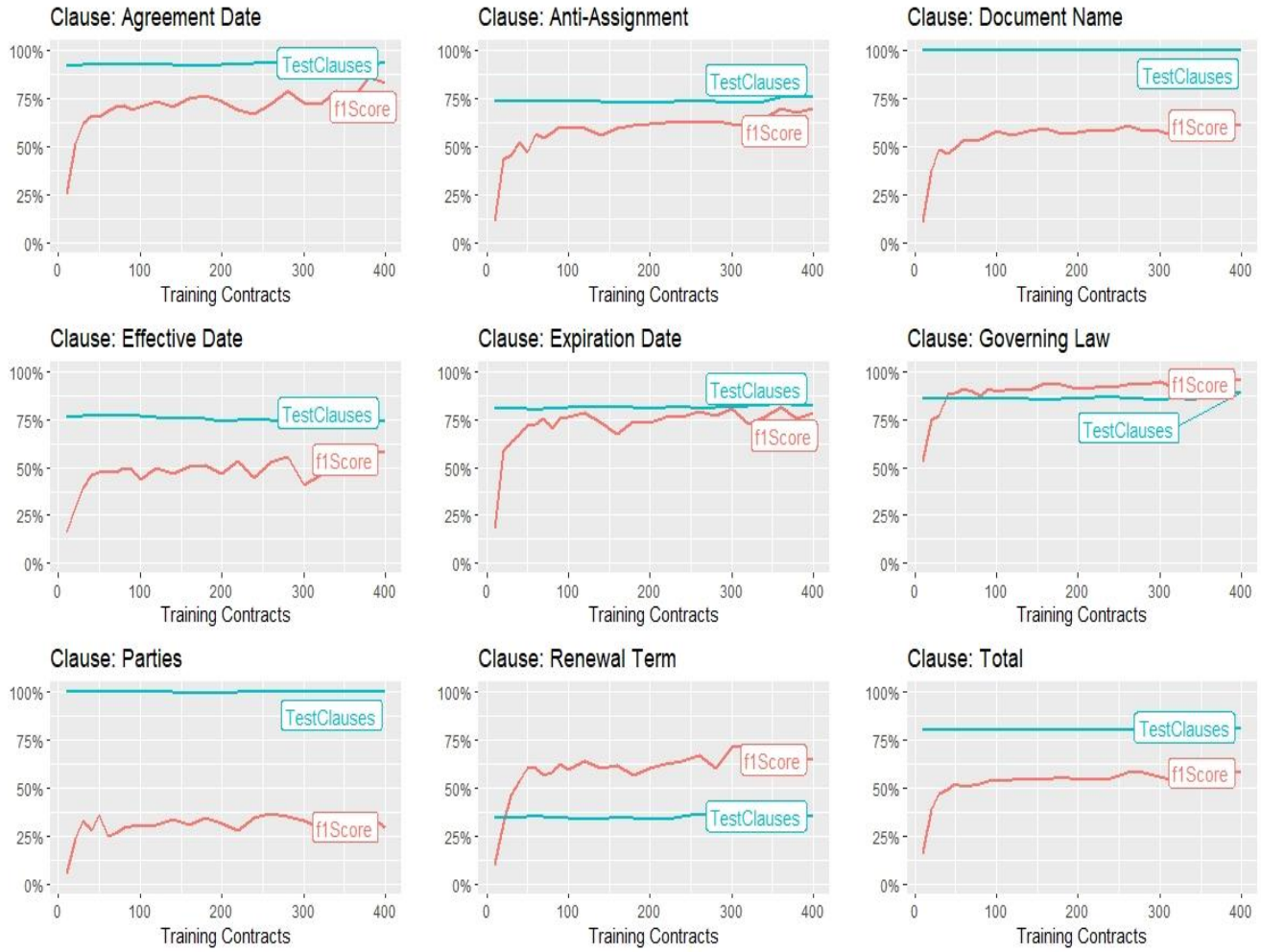

Fig. 3. Training Contracts testing results, showing F1-Score and the percentage of testing clauses available

Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 8770–8776. [Online]. Available: https://aclanthology.org/2021.emnlp-main.691

[2] Z. Wang, H. Song, Z. Ren, P. Ren, Z. Chen, X. Liu, H. Li, and M. de Rijke, "Cross-domain contract element extraction with a bi-directional feedback clause-element relation network," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 1003–1012. [Online]. Available: https://doi.org/10.1145/3404835.3462873

[3] S. A. Phand and J. A. Phand, "Twitter sentiment classification using stanford nlp," in *2017 1st International Conference on Intelligent Systems and Information Management (ICISIM)*, 2017, pp. 1–5.

[4] H. Zhang, B. Pan, and R. Li, "Legal judgment elements extraction approach with law article-aware mechanism," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 21, no. 3, dec 2021. [Online]. Available: https://doi.org/10.1145/3485244

[5] P. Shah, S. Joshi, and A. K. Pandey, "Legal clause extraction from contract using machine learning with heuristics improvement," *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, pp. 1–3, 2018.

[6] P. Spinosa, G. Giardiello, M. Cherubini, S. Marchi, G. Venturi, and S. Montemagni, "Nlp-based metadata extraction for legal text consolidation," 06 2009, pp. 40–49.

[7] S. P. Nayak and S. Pasumarthi, "Automatic detection and analysis of dpp entities in legal contract documents," *2019 First International Conference on Digital Data Processing (DDP)*, pp. 70–75, 2019.

[8] C. C. Dozier and T. Zielund, "Cross document co-reference resolution applications for people in the legal domain," in *Conference On Reference Resolution And Its Applications*, 2004.

[9] S. Joshi, P. Shah, and A. K. Pandey, "Location identification, extraction and disambiguation using machine learning in legal contracts," *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, pp. 1–5, 2018.

[10] M. Farouk, "Measuring text similarity based on structure and word embedding," *Cognitive Systems Research*, 04 2020.

[11] I. Chalkidis, I. Androutsopoulos, and A. Michos, "Extracting contract elements," in *Proceedings of the 16th Edition of the International Conference on Artificial Intelligence and Law*, ser. ICAIL '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 19–28. [Online]. Available: https://doi.org/10.1145/3086512.3086515

[12] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of cnn and rnn for natural language processing," 2017. [Online]. Available: https://arxiv.org/abs/1702.01923

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett,

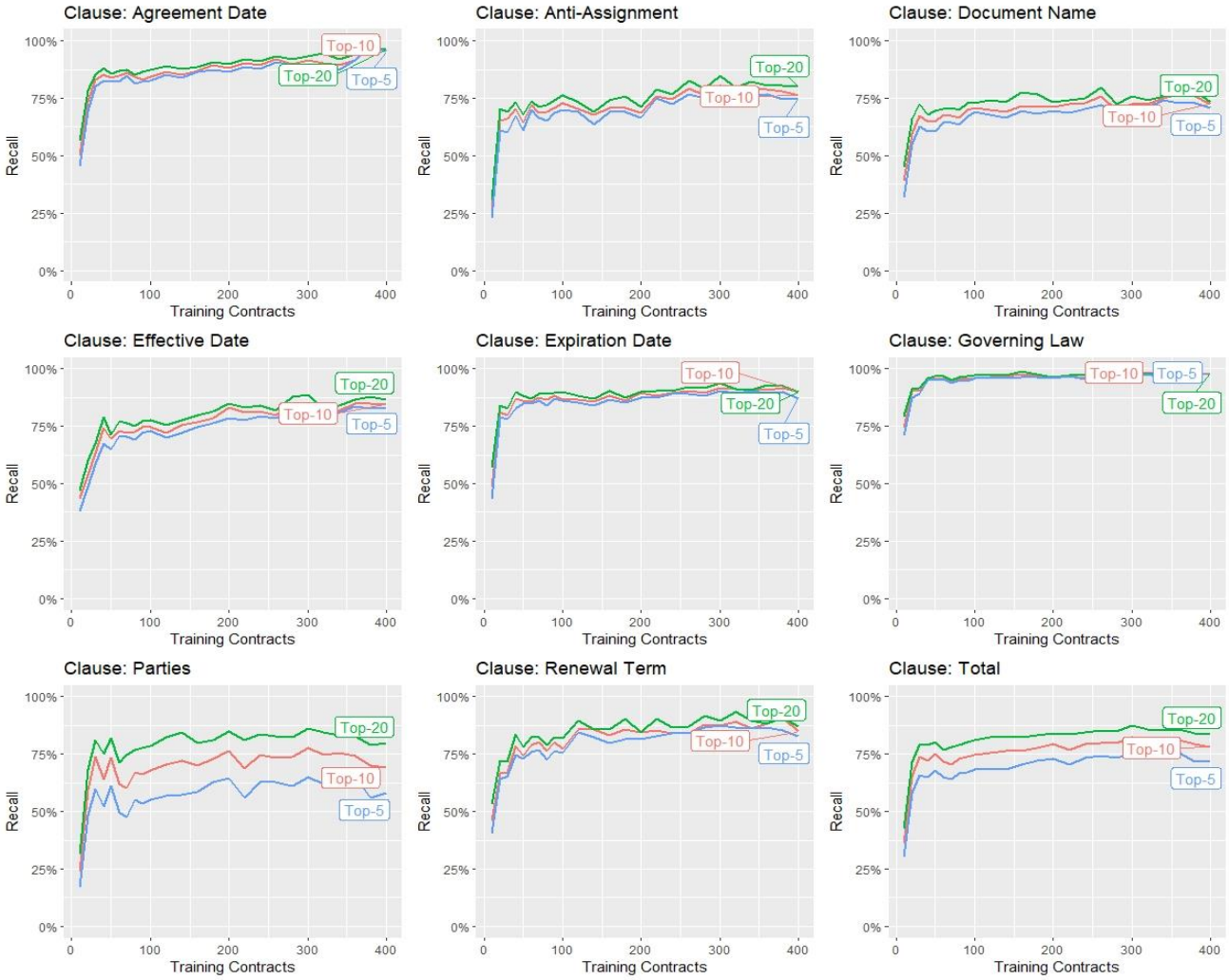

Fig. 4. Recall for top-5, top-10 and top-20

Eds., vol. 30. Curran Associates, Inc., 2017.

[14] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019, cite arxiv:1907.11692. [Online]. Available: http://arxiv.org/abs/1907.11692

[15] S. J. Mielke, Z. Alyafeai, E. Salesky, C. Raffel, M. Dey, M. Gallé, A. Raja, C. Si, W. Y. Lee, B. Sagot, and S. Tan, "Between words and characters: A brief history of open-vocabulary modeling and tokenization in NLP," *CoRR*, vol. abs/2112.10508, 2021. [Online]. Available: https://arxiv.org/abs/2112.10508

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding." J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 4171–4186.

[17] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=H1eA7AEtvS

[18] D. Hendricks, C. Burns, A. Chen, and S. Ball, https://www.atticusprojectai.org/cuad, 2015.

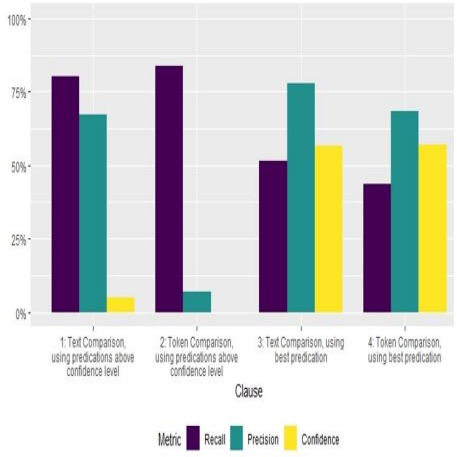

Fig. 5. Method comparison for the total set of clauses

TABLE I
ACCURACY COMPARISON BETWEEN TEXT AND TOKEN METHODS (1 AND 2).

	(1) Text Comparison				(2) Token Comparison		
Clause Name	**AUPR**	**Recall**	**Precision**	**Conf**	**Recall**	**Precision**	**Conf**
Agreement Date	78.17%	80.65%	70.09%	46.00%	88.17%	3.97%	0.00%
Anti-Assignment	65.65%	83.21%	7.31%	0.00%	81.02%	6.68%	0.00%
Document Name	85.41%	80.39%	86.32%	50.00%	73.53%	N/A	0.00%
Effective Date	37.30%	83.33%	3.44%	0.00%	76.19%	N/A	0.00%
Expiration Date	74.80%	81.93%	67.65%	33.00%	80.72%	65.05%	28.00%
Governing Law	94.15%	80.00%	97.50%	72.00%	80.00%	96.25%	67.00%
Parties	83.57%	81.95%	91.19%	3.00%	82.69%	47.07%	0.00%
Renewal Term	63.04%	87.10%	43.55%	13.00%	83.87%	43.08%	13.00%
Total	66.35%	80.22%	67.32%	5.00%	83.83%	6.88%	0.00%

TABLE II
ACCURACY COMPARISON BETWEEN TEXT AND TOKEN METHODS (3 AND 4).

	(3) Text Comparison			(4) Token Comparison		
Clause Name	**Recall**	**Precision**	**Avg Conf**	**Recall**	**Precision**	**Avg Conf**
Agreement Date	81.72%	77.55%	83.89%	73.12%	69.39%	84.50%
Anti-Assignment	46.72%	82.05%	60.51%	43.80%	80.00%	60.24%
Document Name	83.33%	83.33%	84.16%	56.86%	56.86%	84.31%
Effective Date	51.19%	43.43%	65.43%	45.24%	38.38%	64.89%
Expiration Date	77.11%	73.56%	82.78%	73.49%	70.93%	82.52%
Governing Law	87.78%	94.05%	89.53%	86.67%	92.86%	89.62%
Parties	30.02%	94.77%	22.32%	22.65%	81.46%	22.44%
Renewal Term	74.19%	47.92%	90.06%	70.97%	46.81%	90.06%
Total	51.33%	77.73%	56.73%	43.68%	68.46%	57.03%

TABLE III
COMPARISON BETWEEN TEXT AND TOKEN METHODS.

Method ID	**Calculation Description**	**Recall**	**Precision**	**Conf**	**TP**	**TN**	**FP**	**FN**
1	Text Comparison using the predictions above confidence level	80.2%	67.3%	5.0%	933	128	453	230
2	Token Comparison using the predictions above confidence level	83.8%	6.9%	0.0%	975	190	13189	188
3	Text Comparison using the best predicted answers	51.3%	77.7%	56.7%	597	123	171	566
4	Token Comparison using the best predicted answers	43.7%	68.5%	57.0%	508	123	234	655

TABLE IV
TRAINING CONTRACT RESULTS SHOWING THE F1-SCORE. 400 TRAINING CONTRACTS REPRESENTS 80%

	Training Contracts						
Clause Name	**40**	**80**	**120**	**160**	**200**	**300**	**400**
Agreement Date	66.1%	71.2%	74.0%	75.1%	73.4%	72.8%	83.0%
Anti-Assignment	52.3%	57.2%	59.8%	59.9%	61.7%	61.7%	70.1%
Document Name	46.5%	53.7%	56.1%	59.3%	57.7%	57.9%	61.0%
Effective Date	45.6%	49.8%	49.5%	50.1%	46.4%	40.9%	68.6%
Expiration Date	67.1%	70.5%	78.7%	67.5%	73.0%	80.8%	78.6%
Governing Law	87.8%	87.3%	90.6%	93.8%	90.8%	94.6%	95.5%
Parties	28.2%	29.2%	30.5%	30.5%	31.6%	32.7%	29.7%
Renewal Term	53.4%	57.9%	63.7%	62.0%	60.5%	71.1%	64.9%
Total	49.2%	52.1%	54.8%	54.5%	54.5%	55.8%	57.9%