Combining Similarity and Transformer Methods for Case Law Entailment

Juliano Rabelo
Alberta Machine Intelligence Institute,
University of Alberta
Edmonton, AB, Canada
rabelo@ualberta.ca

Mi-Young Kim
Department of Science, Augustana
Faculty, University of Alberta
Camrose, AB, Canada
miyoung2@ualberta.ca

Randy Goebel
Dept. of Computing Science and
Alberta Machine Intelligence Institute,
University of Alberta
Edmonton, AB, Canada
rgoebel@ualberta.ca

ABSTRACT

We tackle the complex problem of determining entailment relationships between case law documents, one of the tasks in the Competition on Legal Information Extraction and Entailment (COLIEE). With input of an entailed fragment from a case coupled with a candidate entailing paragraph from a noticed case, our approach relies on four main components: (1) extraction of similarity measures between the two pieces of text; (2) application of a transformer-based technique on the input text; (3) applying a threshold-based classifier; and (4) post-processing the results considering the *a priori* probability determined by the data distribution on the training samples and combining the results of (1) and (2). Our experiments achieved an F-score of 0.70 on the official COLIEE test dataset, ranking first among all competitors for that task in the 2019 competition.

CCS CONCEPTS

• **Information systems → Content analysis and feature selection**; **Similarity measures**; **Clustering and classification**; *Document topic models*; *Information extraction*; Specialized information retrieval.

KEYWORDS

legal textual entailment, document similarity, binary classification, imbalanced datasets

ACM Reference Format:
Juliano Rabelo, Mi-Young Kim, and Randy Goebel. 2019. Combining Similarity and Transformer Methods for Case Law Entailment. In *Seventeenth International Conference on Artificial Intelligence and Law (ICAIL '19), June 17–21, 2019, Montreal, QC, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3322640.3326741

1 INTRODUCTION

Every day, large volumes of legal data are produced by law firms, law courts, independent attorneys, legislators, and many other sources. In that scenario, management of legal information becomes manually intractable, and requires the development of tools which automatically or semi-automatically aid legal professionals in handling the information overload. In the COLIEE competition, four facets of that challenge are presented: case law retrieval, case law entailment, statute law retrieval and statute law entailment. Here we provide the details of our approach to the task of case law entailment, evaluate the results achieved and comment on future work to further improve our model.

The initial approaches for open-domain textual entailment focused on shallow text features, evolved to the use of word embeddings, logical models and machine learning in general, and more recently the literature shows that deep learning based approaches have shown impressive results in a wide range of textual entailment benchmarks.

Past approaches for legal text entailment have relied on machine learning techniques with specific feature extraction, or on the use of some distributed vector representation of the text pieces, association rules and semantic knowledge representation. Re-sampling techniques have also been used to overcome the data imbalance problem. A similar problem to the case law entailment, also covered in COLIEE, is statute law entailment. For that task, the consistently best performance accross all past COLIEE editions has been achieved by a combination of condition/conclusion/exception detection rules and negation handling.

The method for case law entailment presented in this paper combines similarity based features which rely on multi-word tokens instead of single words, in an attempt to capture deeper semantics from the text, and further exploits the BERT framework [13], fine-tuned to the task of case law entailment on the provided training dataset. Similar to our approach to COLIEE 2018 [27], a post-processing step is used and considers *a priori* probabilities, being applied at the end of the pipeline (see section 3 for details). This method achieved the best F-score among all competing teams on the COLIEE 2019 edition, as shown in section 4. As future work, we plan on investigating more structure-aware techniques (such as dependency trees comparison), perform training of BERT and other transformer-based tools with case law specific data, and deepen our analysis of the application of "embeddings-like" representations for this problem.

The rest of this paper is organized as follows: in Section 2 we briefly review open domain textual entailment and the special case of case law entailment. Section 3 describes our approach. Section 4 describes our experiments with an analysis of the results. In Section 5 we conclude the paper and comment on future work.

ICAIL '19, June 17–21, 2019, Montreal, QC, Canada
© 2019 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-6754-7/19/06.
https://doi.org/10.1145/3322640.3326741

2 RELATED WORK

Textual entailment is a logic task in which the goal is to determine whether one sentence can be inferred from another. In the more general case, the task consists of categorizing an ordered pair of sentences into one of three categories: "positive entailment" occurs when one can use the first sentence to prove that a second sentence is true. Conversely, "negative entailment" occurs when the first sentence can be used to disprove the second sentence. Finally, if the two sentences have no correlation, they are considered to have a "neutral entailment". In COLIEE, teams are challenged with the task of classifying two case law textual fragments possessing a "positive entailment" relationship or not (i.e., they have "neutral entailment").

In the following subsections, we will discuss related research on textual entailment in general and techniques developed specifically for case law entailment.

2.1 Open-domain Textual Entailment

Textual entailment is useful as a task per se or as a component in larger applications. For example, question-answering systems may apply textual entailment techniques to identify an answer from previously stored databases [3], [24]. Textual entailment may also be used to enhance document summarization (e.g., being used to measure sentence connectivity [14], or as additional features to the summary generation [22]). Due to the recent increased interest on textual entailment research, publicly available benchmarks exist to evaluate such systems (e.g., [5], [30]).

Early approaches for open-domain textual entailment relied heavily on exploiting surface syntax or lexical relationships [10], and then a wide range of tools, such as word embeddings, logical models, graphical models, rule systems and machine learning were applied [1]. A modern research trend for open-domain textual entailment is an application of general deep learning models, such as ELMo [26], BERT [13] and ULMFit [17].

These methods build on the approach introduced in [12], which showed how to improve document classification performance by using unsupervised pre-training of an LSTM [15] followed by supervised fine-tuning for specific downstream tasks. The pre-training is done on very large datasets, which do not need to be labeled and are intended to capture general language knowledge (usually, the pre-training is modelled as a language modeling task). Then, supervised learning is used as a fine-tuning step, thus requiring a significantly smaller labeled dataset, aiming at adjusting the weights of the final layers of the model and making it suitable for the specific task. These models have achieved impressive results in a wide range of publicly available benchmarks of different common natural language tasks, such as RACE (reading comprehension) [20] , COPA (common sense reasoning) [29], CoLA (linguistic acceptability) [31] and RTE (textual entailment) [11] to name a few.

2.2 Case Law Textual Entailment

The specific task of assessing textual entailment for case law documents is quite new. The first COLIEE edition which included this task was in 2018 [18], and the two best performing approaches are described below.

Chen et al. [9] proposed the application of association rules for the problem. They applied a machine learning-based model using Word2Vec embeddings [23] and Doc2Vec [21] as features. This approach faces two main problems: the lack of sufficient training data to make the models converge and generalize, and the computational cost of training which increases exponentially on the size of the dataset. To overcome this scalability issue, they proposed two association rule models: (1) the basic association rule model, which considers only the similarity between the source document and the target document, and (2) the co-occurrence association rule model, which uses a relevance dictionary in addition to the basic model.

Another approach [27] worth mentioning approached the task as a binary classification problem, and built feature vectors comprised of the measures of similarity between the candidate paragraph and (1) the entailed fragment of the base case, (2) the base case summary and (3) the base case paragraphs (actually a histogram of the similarities between each candidate paragraph and all paragraphs from the base case). Those feature vectors are used as input to a Random Forest [6] classifier and the results are post-processed to consider *a priori* knowledge (similarly to what we have done for this COLIEE edition - see Section 3). To overcome the problem of severe data imbalance in the dataset (with less than 3% of the examples being true entailment relationships), the dominant class was under-sampled and the rarer class was over-sampled by SMOTE sample synthesis [8]. This approach ranked first place in the case law entailment task of COLIEE 2018.

In addition to that specific task on case law entailment, past editions of COLIEE included a task on statute law entailment, whose goal is to identify entailment relationships between Japanese bar exam questions and relevant legal articles. The best performance on that task for all past COLIEE editions has been achieved by a combination of legal information retrieval and textual entailment approach, which exploits semantic information using a logic-based representation [19]. A meaning extraction process uses a selection of features based on a kind of paraphrase, coupled with a condition/conclusion/exception analysis of articles and queries, and also exploiting negation patterns extracted from the articles. The logic-based representation is then constructed as a semantic analysis, which is used to classify questions according to their difficulty level by analyzing the logic representation. If a question is in the "easy" category, the entailment answer is obtained in a straightforward manner from the logic representation; otherwise, an unsupervised learning method is applied.

3 OUR APPROACH

The task of case law entailment in COLIEE may be defined as follows: given a base case b and one fragment of text f contained within b, and a second case r which is relevant in respect to b, the task consists in determining which paragraph(s) of r entail f. More formally, given b, f and r as above (r represented by its paragraphs $P = \{p_1, p_2, ..., p_n\}$), we need to find the set $E = \{p_1, p2, ..., p_m \mid p_i \in P\}$ where $entails(p_i, f)$ denotes a relationship which is true when $p_i \in P$ entails the fragment f.

We treat that problem as a binary classification problem, by considering each paragraph p_i in r as a candidate, which must then be classified as entailing f or not. To do so, we created a classifier which processes each pair (p_i, f) and uses as features two measures of similarity and the output of BERT for a text entailment task: (1) a cosine measure [2] which uses multiple word tokens to represent the text fragments; (2) a cosine measure which considers only the noun-phrases of the text; and (3) the confidence level of the BERT framework for the text fragments. A score combining those values is computed and thresholds are empirically determined to generate the (partial) output, which is post-processed considering the *a priori* probability of the training dataset. More details on each of these components is given in the next subsections.

3.1 Multi-word Token Similarity

An immediate classic technique for measuring similarity between documents is the representation of the text as a bag of words, and then calculating the cosine distance between those document representations in the vector space. Usually, some auxiliary methods such as stop words removal, stemming and fuzzy string matching can be applied in support of that process. However, often the text tokenization considers each word as a token (i.e., punctuation marks and spaces are seen as token delimiters), thus completely neglecting the possibility that sentences formed by the same words in different order may have different meanings. For example, this method would consider as indentical the sentences (a) "the big dog jumped over the lazy cat" and (b) "the big cat jumped over the lazy dog."

In order to capture deeper semantics, we replaced the usual tokenization as follows: instead of considering each word as a token, our method actually tokenizes the input text considering punctuation marks and stop words as delimiters. We are thus able to (1) retain the words which carry semantics and (2) consider the appropriate modifiers. In the example above, our method would output the tokens "big dog jumped over", "lazy cat" and "big cat jumped over", "lazy dog" as tokens for sentences (a) and (b) respectively.

Once we identify the multi-word tokens, we applied stemming to the individual words just to avoid missing valid matches due to plurals or inflected verb forms, and then calculated the distance between the text fragments using the regular cosine distance. Similarly, we also extracted the noun-phrases of the text fragments using spaCy [16]. The final model uses a combination of all similarity metrics whose individual performance is shown in Section 3. See the details on how the metrics are combined on subsection 3.2.

3.2 Final Similarity Score Calculation

After running the components described above, we end up with two similarity scores associated with each (f, p_i) pair. We generate the final compound score for each pair by applying a simple weighted average of the two similarity scores, weighing the multi word-token similarity score as twice the weight of the noun-phrase similarity score (weights defined empirically after the analysis of each component individual result on this task - see Section 4). Our final score is given by the equation below:

$$score(f, p_i) = \frac{2 * MWT(f, p_i) + NP(f, p_i)}{3}$$

where MWT represents the calculated multi word-token similarity and NP represent the calculated noun-phrase similarity.

We then define a minimum similarity threshold. All candidates whose score is greater or equal to that threshold make to the partial results, which will be post-processed. The BERT contribution to the final model comes during the post-processing, described in Subsection 3.4.

3.3 BERT Confidence Score

Another component of our method used BERT [13], a framework designed to pre-train deep bidirectional representations by jointly conditioning on both left and right context in all layers. This leads to pre-trained representations which can be fine-tuned with only one additional output layer on downstream tasks, such as question answering, language inference and textual entailment, but without requiring task-specific modifications.

BERT is pre-trained on a large dataset (the goal being make it acquire general language "knowledge") and can be fine-tuned on relatively small, specific datasets (the goal being to make it learn how to combine the previously acquired knowledge in a specific scenario). This makes BERT a good fit for this task, since we do not have a large dataset available for training the model. To make use of BERT in our experiments, we fine tuned it using our training dataset (except for the validation dataset, which was used as the test set for the generated model, and a development dataset, required by the BERT fine tuning process, which was about the size of our validation dataset). To process the official COLIEE test dataset, we used the full training dataset to fine-tune the model.

BERT has achieved impressive results on well-known benchmarks such as GLUE [30], MultiNLI [32] and SQuAD [28], and could potentially be used by itself in this textual entailment task. While it produced good results in our experiments, it was clear that a combined approach would be more beneficial (see Section 4).

The manner in which we combined the above components in our final system is described below.

3.4 Post-Processing

By analyzing the *a priori* probabilities of the dataset, we see that the majority of the cases have exactly one entailing paragraph among all candidates. So we establish that, for each case, we will return at least one candidate, even if its score is lower than the threshold. Moreover, we also establish that at most 2 answers should be returned for each case, no matter how many candidates scored higher than the threshold. One of those entries in the final result will be necessarily the best BERT entry among the candidates for that case, provided it has a confidence level of at least min_{bert}. The pseudo-code below shows the details of our combined heuristics:

Notice that, if $len(\mathcal{R}) == max_{per\_case}$, "include()" will remove the candidate whose score is lowest to make sure the following condition holds: $len(\mathcal{R}) <= max_{per\_case}$.

3.5 Other Frameworks for Language Processing

We performed experiments with two other natural language processing frameworks: the Universal Sentence Encoder [7] and ULMFit [17]. The Universal Sentence Encoder is a model designed for encoding sentences into embeddings intended to be applied in downstream tasks via transfer learning, and it has been shown to surpass the performance of word-level embeddings (e.g., Word2vec

Algorithm 1 Post-processing

```
function POST-PROCESS($\mathcal{T}$, $min_{sim\_score}$, $min_{bert}$, $max_{per\_case}$)
    for each $case \in \mathcal{T}$ do
        $C \leftarrow get\_candidates(case, min_{sim\_score})$
        $true\_count \leftarrow \min(max_{per\_case}, len(C))$
        if $true\_count > 0$ then
            $\mathcal{R} \leftarrow get\_n\_best(C, true\_count)$
        end if
        $b \leftarrow get\_best\_bert\_entry(C, min_{bert})$
        if $b \notin \mathcal{R} \wedge b.confidence >= min_{bert}$ then
            $\mathcal{R}.include(b, max_{per\_case})$
        end if
    end for
    return $\mathcal{R}$
```

Table 1: Summary for the Case Law Entailment Task Dataset

Property	Value
Number of base cases	181
Total paragraphs in the related cases	5,814
Total true entailing paragraphs	202 (3.47%)
Avg. entailing paragraphs per base case	1.11
Stddev of entailment paragraphs counts	0.38

[23], Glove [25], Fasttext [4]). We generated embeddings for each entailed fragment f and each respective candidate paragraph p_i, and then calculated the cosine of the angle formed by the vectors of f and p_i.

ULMFit is somewhat similar to BERT in that it relies on training a general model in a large dataset, and then relies on transfer learning and fine tuning to specific downstream tasks by training on smaller datasets. In our experiments, neither of those tools provide good results. Like BERT, ULMFit reported great results in many natural language related tasks. In our available timeframe, We could not investigate if those tools would be effective, but we plan to perform a deeper analysis in subsequent work (see Section 5).

4 EXPERIMENTAL RESULTS

Here we summarize the results achieved with our experiments, but, prior to that, we present an overview of the COLIEE dataset and analyze the results achieved.

4.1 Dataset Analysis

The training dataset has 181 base cases, each with its respective entailed fragment in a separate file. For each base case, a related case represented by a list of paragraphs is given, from which the paragraph(s) that entail the base-case-entailed fragment must be identified. Table 1 summarizes the dataset properties. The golden labels for this dataset were disclosed upfront.

To run our experiments, we created a validation dataset by separating a portion of the original training dataset. We needed to keep the training dataset as big as possible so we could train the BERT model. To do so, we randomly selected 164 cases for training and 17 cases as our validation dataset. Table 2 summarizes the validation dataset properties.

Table 2: Summary for the Case Law Entailment Task Validation Dataset

Property	Value
Number of base cases	17
Total paragraphs in the related cases	698
Total true entailing paragraphs	17 (2.43%)
Avg. entailing paragraphs per base case	1.00
Stddev of entailment paragraphs counts	0.00

Table 3: Summary for the Case Law Entailment Task Test Dataset

Property	Value
Number of base cases	44
Total paragraphs in the related cases	1,448
Total true entailing paragraphs	45 (3.10%)
Avg. entailing paragraphs per base case	1.02
Stddev of entailment paragraphs counts	0.7106

Table 4: Summary of Results on the Validation Dataset

Component	Precision	Recall	F-score
Multi word-token similarity	0.4615	0.3529	0.4000
Sentence-based embeddings	0.1739	0.4705	0.2539
BERT	0.5555	0.5882	0.5714
Noun chunks similarity	1.0000	0.1764	0.3000

The official COLIEE test dataset was initially released without the golden labels, which were only disclosed after the competition results were published. That dataset's properties are summarized on Table 3.

From Tables 1, 2 and 3, we can see that the properties of the three datasets are similar, which is appropriate for our experiments. There was also a fourth dataset, which is the development dataset. That dataset is required for the BERT fine-tuning process and is about the same size of the validation dataset. We omit its details here because it is only used internally by the fine-tuning process.

4.2 Validation Dataset Results

Table 4 summarizes the results achieved by each component of our model in isolation on the validation set:

The similarity-based components require a threshold, above which a case is considered as an entailing paragraph. Those parameters were empirically set to 0.27 (Multi word-token similarity), 0.78 (Sentence embeddings similarity) and 0.33 (Noun chunks similarity), which were the best parameters for the validation dataset (see Figure 1 for more details). BERT produces the probability of each class (entailment, not entailment), so we consider the cases in which the entailment class probability is greater than the not entailment class probability.

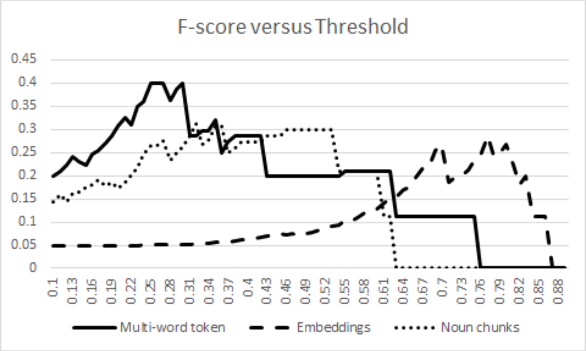

Figure 1: Variation of the F-score according to the threshold for the similarity-based methods

Despite the small volume of training data, BERT had the best performance considering the components in isolation, as can be seen on Table 2. For that reason, we chose to always include BERT's best entry for each case in our combined model. However, the performance of BERT by itself is considerably lower than what it achieves in other datasets, as we mentioned in section 3.3. This might be due to the lack of sufficient training data, or the fact that, different from "regular" textual entailment tasks, case law entailment requires some background information which might not be present on the sentences themselves. For example, GLUE has textual entailment samples such as ("No Weapons of Mass Destruction Found in Iraq Yet," "Weapons of Mass Destruction Found in Iraq") whose label is "not entailment." Entries in COLIEE, on the other hand, are in general not so straightforward. For example, consider the entry below, labelled as an entailment case in the COLIEE training dataset:

> *Entailed fragment*: "Given that the Respondent remains a security risk whom the Minister has not had sufficient opportunity to investigate to determine the extent of his involvement with the LTTE, his release into the community will cause real and non-speculative irreparable harm. Moreover, the Respondent is a danger to the public if released on minimal terms and conditions including a patent lack of supervision in the community by a surety. Releasing the Respondent directly prevents the Minister's efforts to ensure his availability for future proceedings - including a possible admissibility hearing; furthermore, it prevents him from carrying out his mandate of protecting the public and undermines the safety and security of the Canadian public."
>
> *Entailing paragraph*: "Irreparable harm would occur if the Respondent is released as she would not appear nor be available for removal from Canada. This would prevent the Minister from fulfilling his statutory obligations."

This might suggest that BERT's performance could be improved by training it on a large corpus of legal documents, so it could acquire the background knowledge necessary to process that kind of information. Despite of that, using the current strategy of only fine-tuning the model using part of the training dataset, the BERT module achieved the best results of all modules considered in isolation (precision: 0.5555, recall: 0.5882, F-score: 0.5714). The errors of each module in the validation dataset are shown on Table 5.

Table 5: False negative errors in our validation dataset. Results for the similarity based methods used the best thresholds for each one, mentioned in the beginning of this subsection.

Case	Cand.	BERT	Multi-Word	Emb.	Noun chunks
165	054	TRUE	FALSE	FALSE	FALSE
166	017	TRUE	FALSE	FALSE	FALSE
167	008	FALSE	FALSE	FALSE	FALSE
168	069	TRUE	FALSE	FALSE	FALSE
169	018	TRUE	FALSE	FALSE	FALSE
170	017	TRUE	TRUE	TRUE	TRUE
171	005	FALSE	FALSE	TRUE	FALSE
172	029	FALSE	TRUE	FALSE	TRUE
173	033	TRUE	TRUE	FALSE	FALSE
174	028	FALSE	FALSE	FALSE	FALSE
175	015	TRUE	FALSE	FALSE	FALSE
176	013	FALSE	TRUE	FALSE	TRUE
177	018	TRUE	TRUE	TRUE	TRUE
178	032	FALSE	FALSE	TRUE	FALSE
179	014	FALSE	FALSE	FALSE	FALSE
180	041	TRUE	TRUE	TRUE	TRUE
181	016	TRUE	FALSE	FALSE	FALSE

The experiments above showed that the approaches had some degree of complementarity, especially considering the multi word-token similarity and BERT. For that reason, we decided to devise a combined approach giving more importance to those components, as described in Section 3. The performance of the combined approach on the validation dataset was: precision: 0.6087, recall: 0.8235, F-score: 0.7000.

4.3 Test Dataset Results

We submitted the combined approach to the official COLIEE test dataset by varying only the threshold on the final combined score (we set the threshold to 0.25, 0.30 and 0.40). Table 6 shows the results achieved by all teams on that test dataset (each team may submit up to three results). Our method achieved the highest F-score among all teams, with a similar F-score in comparison with the results on the validation dataset, albeit a significant difference in the precision and recall scores.

As previously mentioned, our F-score was similar to the one achieved on the validation dataset, but the precision and recall scores varied. That may indicate the validation dataset was not representative enough or that our model parameters were somewhat over fit to that dataset. Something we planned to do but could not perform in the available competition timeframe was a 10-fold cross validation, which would take considerable more time because we would need to fine tune BERT using 10 different training datasets. This is proposed as one aspect of future work in Section 5. We also suggest the potential of experiments for each individual component

Table 6: Case Law Entailment Task Results

Team	Precision	Recall	F-score
ielab	0.3409	0.3333	0.3371
ielab	0.4545	0.4444	0.4494
ielab	0.2273	0.2222	0.2247
IITP	0.0455	0.0444	0.0449
IITP	0.6591	0.6444	0.6517
IITP	0.7045	0.6889	0.6966
JNLP	0.1364	0.1333	0.1348
JNLP	0.0682	0.0667	0.0674
JNLP	0.5909	0.5778	0.5843
TRCase	0.6818	0.6667	0.6742
TTCL	0.4000	0.8000	0.5333
TTCL	0.3780	0.6889	0.4882
TTCL	0.3882	0.7333	0.5077
UA	0.6364	0.7778	0.7000
UA	0.6296	0.7556	0.6869
UA	**0.6538**	**0.7556**	**0.7010**
UBLTM	0.1182	0.5778	0.1962
UBLTM	0.1273	0.6222	0.2113

in isolation from the official COLIEE test dataset, so we could verify whether the performances are similar to the ones we achieved on the validation dataset.

5 FINAL REMARKS AND FUTURE WORK

As final remarks, we can conclude from our experiments that:

- The similarity measure which considers the multi-word tokens can capture more semantics than usual word based similarity;
- BERT proved to be a powerful tool, requiring little fine tuning and achieving good results in a challenging task. On the experiments executed using our validation dataset, it achieved the best performance among all components considered in isolation;
- A combination of BERT and similarity-based techniques can provide improved results for case law entailment;
- The Universal Sentence Encoder performed much worse than BERT. That may indicate it needs training in the specific domain or that the best way to effectively use the generated embeddings would not be in a distance metric, but, for example, as inputs to a separate classifier, which would then learn how to correlate the representations and the expected labels.

We plan to extend this work by performing the following actions:

- Train BERT using a larger case law dataset and check whether it is capable of grasping law-related knowledge. There are publicly available case law corpus (e.g., Canadian Supreme Court Reports on https://scc-csc.lexum.com/scc-csc/en/nav.do) which can be used as input for that kind of training. Fine tuning the framework with a larger dataset of case law entailment sentences would be probably more effective, but there are not other case law entailment datasets known to the authors which could be used as input to the fine tuning process;
- Run a similar training procedure with tools such as the Universal Sentence Encoder and ULMFit and experiment to use their embeddings representations as input to supervised classifiers;
- Explore more structure-aware tools, such as dependency trees, (maybe in combination with Named Entity Recognizers) which might capture more subtle correlations between the text fragments;
- Extend the error analysis shown in Table 5 to consider what are the characteristics of the errors observed in each method, so that we can plan for a better combination of the methods;
- Perform a similar analysis using the test dataset, whose golden labels were released after the writing of this paper. We are especially interested on finding out why the recall and precision scores varied so much from the experiments on the validation dataset to the experiments on the test dataset.

ACKNOWLEDGMENTS

This research was supported by the Alberta Machine Intelligence Institute (AMII), and would not be possible without the significant support of Colin Lachance from vLex and Compass, and the teams from Ross Intelligence and Intellicon.

REFERENCES

[1] Ion Androutsopoulos and Prodromos Malakasiotis. 2009. A Survey of Paraphrasing and Textual Entailment Methods. *CoRR* abs/0912.3747 (2009). arXiv:0912.3747 http://arxiv.org/abs/0912.3747

[2] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. 1999. *Modern Information Retrieval.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[3] Asma Ben Abacha and Dina Demner-Fushman. 2019. A Question-Entailment Approach to Question Answering. *CoRR* abs/1901.08079 (2019). arXiv:1901.08079 http://arxiv.org/abs/1901.08079

[4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. *CoRR* abs/1607.04606 (2016). arXiv:1607.04606 http://arxiv.org/abs/1607.04606

[5] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP).* Association for Computational Linguistics.

[6] Leo Breiman. 2001. Random Forests. *Mach. Learn.* 45, 1 (Oct. 2001), 5–32. https://doi.org/10.1023/A:1010933404324

[7] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder. *CoRR* abs/1803.11175 (2018). arXiv:1803.11175 http://arxiv.org/abs/1803.11175

[8] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Int. Res.* 16, 1 (June 2002), 321–357. http://dl.acm.org/citation.cfm?id=1622407.1622416

[9] Y. Chen, Y. Zhou, Z. Lu, H. Sun, and W. Yang. 2018. Legal information retrieval by association rules. In *Twelfth International Workshop on Juris-informatics (JURISIN).*

[10] Ido Dagan and Oren Glickman. 2019. Probabilistic textual entailment: Generic applied modeling of language variability. (04 2019).

[11] Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges Workshop.* Springer, 177–190.

[12] Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised Sequence Learning. *CoRR* abs/1511.01432 (2015). arXiv:1511.01432 http://arxiv.org/abs/1511.01432

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[14] Anand Gupta, Manpreet Kaur, Shachar Mirkin, Adarsh Singh, and Aseem Goyal. 2014. Text Summarization through Entailment-based Minimum Vertex Cover. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (*SEM 2014).* Association for Computational Linguistics and Dublin City University, Dublin, Ireland, 75–80. https://doi.org/10.3115/v1/S14-1010

[15] Sepp Hochreiter and JÃijrgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735 arXiv:https://doi.org/10.1162/neco.1997.9.8.1735

[16] Matthew Honnibal and Mark Johnson. 2015. An Improved Non-monotonic Transition System for Dependency Parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, 1373–1378. https://aclweb.org/anthology/D/D15/D15-1162

[17] Jeremy Howard and Sebastian Ruder. 2018. Fine-tuned Language Models for Text Classification. *CoRR* abs/1801.06146 (2018). arXiv:1801.06146 http://arxiv.org/abs/1801.06146

[18] Yoshinobu Kano, Mi-Young Kim, Masaharu Yoshioka, Yao Lu, Juliano Rabelo, Naoki Kiyota, Randy Goebel, and Ken Satoh. 2018. COLIEE-2018: Evaluation of the Competition on Legal Information Extraction and Entailment. In *Twelfth International Workshop on Juris-informatics (JURISIN)*.

[19] Mi-Young Kim and Randy Goebel. 2017. Two-step Cascaded Textual Entailment for Legal Bar Exam Question Answering. In *Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law (ICAIL '17)*. ACM, New York, NY, USA, 283–290. https://doi.org/10.1145/3086512.3086550

[20] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. 2017. RACE: Large-scale ReAding Comprehension Dataset From Examinations. *CoRR* abs/1704.04683 (2017). arXiv:1704.04683 http://arxiv.org/abs/1704.04683

[21] Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. *CoRR* abs/1405.4053 (2014). arXiv:1405.4053 http://arxiv.org/abs/1405.4053

[22] Elena Lloret, Óscar Ferrández, Rafael Muñoz, and Manuel Palomar. 2008. A Text Summarization Approach under the Influence of Textual Entailment. In *NLPCS - 5th International Workshop on Natural Language Processing and Cognitive Science*. 22–31.

[23] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. *CoRR* abs/1310.4546 (2013). arXiv:1310.4546 http://arxiv.org/abs/1310.4546

[24] Shiyan Ou and Zhenyuan Zhu. 2011. An Entailment-based Question Answering System over Semantic Web Data. In *Proceedings of the 13th International Conference on Asia-pacific Digital Libraries: For Cultural Heritage, Knowledge Dissemination, and Future Creation (ICADL'11)*. Springer-Verlag, Berlin, Heidelberg, 311–320. http://dl.acm.org/citation.cfm?id=2075271.2075319

[25] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543. http://www.aclweb.org/anthology/D14-1162

[26] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

[27] Juliano Rabelo, Mi-Young Kim, Housam Babiker, Randy Goebel, and Nawshad Farruque. 2018. Legal Information Extraction and Entailment for Statute Law and Case Law. In *Twelfth International Workshop on Juris-informatics (JURISIN)*.

[28] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*. 2383–2392. http://aclweb.org/anthology/D/D16/D16-1264.pdf

[29] Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

[30] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. *CoRR* abs/1804.07461 (2018). arXiv:1804.07461 http://arxiv.org/abs/1804.07461

[31] Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural Network Acceptability Judgments. *arXiv preprint arXiv:1805.12471* (2018).

[32] Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2017. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. *CoRR* abs/1704.05426 (2017). arXiv:1704.05426 http://arxiv.org/abs/1704.05426