Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220446

Linking Appellate Judgments to Tribunal Judgments - Benchmarking Different ML Techniques

Charles CONDEVAUX [a,1], Bruno MATHIS [b] Sid Ali MAHMOUDI [a]
Stéphane MUSSARD [a] and Guillaume ZAMBRANO [a]
[a] *CHROME, University of Nîmes, France*
[b] *Centre Européen de Droit et d'Economie, ESSEC Business School, France*

Abstract. The typical judicial pathway is made of a judgment by a tribunal followed by a decision of an appellate court. However, the link between both documents is sometimes difficult to establish because of missing, incorrect or badly formatted references, pseudonymization, or poor drafting specific to each jurisdiction. This paper first shows that it is possible to link court decisions related to the same case although they are from different jurisdictions using manual rules. The use of deep learning afterwards significantly reduces the error rate in this task. The experiments are conducted between the Commercial Court of Paris and Appellate Courts.

Keywords. legal document linking, document similarity, long document processing, named entity recognition, Siamese network

1. Introduction

Although a case may be transferred from one court to another, each tribunal (court of first instance), court of appeal or supreme court gives its own identifier to each decision referring to the same case. In most countries, any decision from a higher court does not necessarily identify the underlying judgment with a unique number, nor is it always associated with publicly available metadata. This is because tribunals and courts are independent from each other. Their information system is more geared toward the production of the decision than to the management of the relationship with the parties involved by the case.

France is implementing an open data program of all judicial decisions. We anticipate that re-users will feel the lack of associated metadata and will seek alternative means to construct the judicial pathway of a case.

The aim of this research is to find out the most efficient method to link first-instance decisions (by tribunals) to second-instance one (by courts of appeals) despite the possible absence of metadata and explicit references in the text body. In practice, this model should help determine, for every outgoing appellate decision, what its original judgment

[1]Corresponding Author: charles@condevaux@unimes.fr

was, to determine the pair <First-instance,Appeal> relating to the same case. For this purpose, different techniques can be used.

- Matching pairs with rules may be a first solution to solve the problem. However success highly depends on how dates, party names and jurisdiction names are written.
- Matching pairs using automatically extracted metadata (NER) and standard ML algorithms (SVM, Random Forests...) allows for better generalization, but performance remains dependent on the underlying extraction models and the text representation (bag of words, Word2vec, etc.).
- Matching pairs with transformer models:
  Transformers ([1]) applied on raw text should outperform the previous approaches. However, due to their limitations in processing long documents, the use of sentence embedding seems to be a good option (see [2]). Another one is to summarize the document, see for instance [3] and [4] for some applications to legal documents. This significantly reduces the length of the text and eases comparisons. Finally, the use of efficient versions of self-attention (i.e. local attention) makes it possible to handle long documents efficiently while ensuring low memory usage.

The aim of this paper is threefold:

1. Extract metadata that can be used to link court decisions.
2. Build a dataset of pairs of court decisions (that match and do not match) in order to further train some machine learning and transformers models.
3. Compare traditional machine learning algorithms (SVM, XGBoost, etc...) to transformer based models to gauge similarity between documents.

2. Related works

Recent papers relate to similarity between judicial documents with different perspectives.

The first category of papers seek to predict a judgment given previous ones on similar cases. Research was first focused on long-standing statistical models like SVM, see [5]. It moved later on to neural networks, see [6], and then on those specialized in similarity, such as Siamese networks. For instance, [7] introduces a model mixing one-shot learning with recurrent neural networks and an attention mechanism to predict, on the basis of similarity of facts, the polarity of a judicial outcome (confirmation or reversal). These results are robust for different types of embeddings.

The second category of papers study Similar Case Matching (SCM). One key use-case to lawyers is to assess consistency of case law, that is, to check that rulings from a supreme court do not diverge too much between similar cases. [8] analyze the similarity and the relevance of rulings of the Court of Justice of the European Union, where relevant cases are identified through their cosine similarity, Jaccard's similarity and words mover's distance based on different vectorization schemes. Recently, [9] proposed a similarity model in order to measure the divergence level among French Cour de Cassation rulings for similar cases.

The third perspective focuses on document and information retrieval. Retrieval-oriented models are able to extract similar elements from a large corpus of legal docu-

ments (see [10] for a transformer-based technique and [11] for an ontology-based one). [12] made a literature review about legal information retrieval systems. They found that CNN models may outperform all other models including BERT, however their performance remains questionable for SCM.

The *jurisdictional linkage* may be both associated to the first and the third categories. At least in France, there is no process to build the pathway of a case through different jurisdictions. In this research, judgment dates, and names of parties, lawyers and jurisdictions are key similarity factors, while in works mentioned in the second category, facts and citations are major similarity factors.

3. Methodology: *jurisdictional linkage*

Solving a jurisdictional linkage problem requires several steps. It is necessary to collect a corpus of compatible documents, to process them to build a sufficiently large training dataset and then to apply a set of machine learning methods to solve a matching task.

3.1. The corpora

The first-instance corpus is made of 360,000 decisions from the Tribunal de Commerce de Paris, dated from 2000 to 2010. It includes a majority of judgments proper, which can be subject to an appeal. On average, one out of ten judgments is subject to an appeal. The corpus also includes interim orders, divestment orders, and other procedural acts, which are also considered as judicial decisions.

The second-instance corpus is made of appellate decisions that include the word sequence "Tribunal de Commerce de Paris", which refers to the biggest jurisdiction of judgment among the 130 or so commercial tribunals in France. This represents some 13,000 decisions. The vast majority of them follow a first-instance judgment. The rest follow a decision from Court of cassation, whose corpus is out of scope of this study.

Less than a quarter in our corpus explicitly designate their underlying trial court judgment by their number. Fortunately, every appellate decision does identify the date of judgment, the jurisdiction of judgment, and the parties (claimants and defendants), mentions that are mandatory by law.

3.2. Dataset creation

Both corpora being available in a version deprived of metadata, a NER model is used to extract such metadata in the first place. Most of these are located in the header or in the upper part of the text body, when no structured header is present. The following labels are created:

- Number ('numéro de répertoire général')
- Date of judgment
- Jurisdiction of judgment
- Party (claimant or defendant)

Two datasets are built, one of 638 appellate decisions, the other made of 1496 judgments from several areas of law, not only in the commercial area. They are manually an-

notated with these labels. Each decision may be annotated with several claimants and/or several defendants.

The data labelling is done on the Kairntech platform [13] and statistics are reported in Table 1.

Label	First-instance	Appeal
Number of decisions	1496	638
Date of appellate decision	N/A	606
Date of judgment	2468	604
Party	4483	2086
Id of appellate decision	N/A	553
Id of first-instance decision	1438	230
Appellate court	N/A	611
Tribunal	1487	614
Total number of annotations	9876	5304

Table 1. Manual annotations per label (the annotations on the Jurisdiction label are split by a simple rule between Appellate Court and Tribunal)

3.3. Post-processing and normalization

Since NER is an extractive and not a generative task, some labels require prior normalization. As dates can be written in either literal or numeric form, a small BART model [14] is trained to convert dates to *yyyy-mm-dd* format. For this purpose a dataset is artificially created mixing numerical format, literal format, punctuation, typos and noise. Thousand of examples are generated, i.e:

- Original string: "*18 Juin 2010*"
- Noisy string: "*e le 18 jiun, 2O1O...*"
- Label: "*2010-06-18*"

As the NER task also gives the location of the extracted sequence in the document, the Seq2Seq generative task is performed on an extended segment of a few words before and after it to correct some truncated predictions.

This normalization allows to classify documents by date and to extract the subset of 13,000 court of appeal decisions that are potential candidates for the linking process with the Paris Commercial Court.

3.4. Matching pairs with rules

The objective is to find the maximum number of matching pairs for a minimal effort.

In the absence of metadata, matching two documents requires a preliminary NER step. The inference of the model described above on the whole corpus allows us to get the underlying judgment number for the appeal decisions, and the judgment number for the judgments. Since these identifiers are unique, 2770 pairs are reliably built at the end of this step.

The case where the underlying judgment number is not available is also addressed. Many judgments are given on the same day in any jurisdiction, and examining the parties, either claimants or defendants, is a good way to disambiguate them. It is highly unlikely that the same party will be involved in several separate cases judged on the

same day and in the same jurisdiction. Even if some parties are natural persons dutifully pseudonymized in publicly available decisions and thus not suitable for disambiguation, legal persons can help disambiguate multiple judgments. The second proposed method therefore consists of matching date/jurisdiction/party triples, after eliminating pseudonyms and lower-casing legal persons. As several parties can match for the same judgment, the duplicates are finally removed.

This method remains largely imperfect since the parties, which cannot be normalized, are marred by numerous syntactic variants. Indeed, it only provides 44 more matching pairs, bringing our total to 2814. The method could be extended to the names of the lawyers, which are in principle available on the judgment and appeal decisions, and are non-pseudonymized. However, this approach is not applied because lawyers were not annotated in the first stage of the NER. We hypothesize that this would have increased our number of pairs by a few dozen at the cost of a significant additional annotation effort. The lawyers' names, while more homogeneous than the parties' names, are also subject to syntactic variants.

3.5. Matching pairs from meta-data

The manual rule-based program described above is used to build a dataset. Since document pairs are now known, the production of counter-examples is relatively simple and is done by randomly matching appellate with first-instance decisions. The task is defined as a binary classification task on tabular data made of two times 2770 rows and 5 columns (features) represented as strings:

- Date of appellate decision
- Date of judgment found in the appellate decision
- Jurisdiction of judgment found in the appellate decision
- Date of judgment found in the first-instance decision
- Jurisdiction of judgment found in the first-instance decision.

A binary classification task is then defined by concatenating the features of the document pairs. This task is then solved using standard machine learning algorithms such as SVM, logistic regression, multilayer perceptron (MLP) and decision-tree methods.

3.6. Matching pairs of documents with transformers

The classification task presented above can be adapted to a contemporary Transformer-based approach. Contrary to the experiments presented so far, these models do not need to define a set of features but are applied directly on the raw document if a sufficiently large annotated dataset is available. This has several advantages:

- It is not necessary to define manual rules
- The model will determine itself the necessary and discriminating tokens
- Its generalization is usually better and more robust given the high number of parameters and the pretraining process.

Since the proposed task is actually a similarity task, a Siamese neural network [15,16,17,18] approach is preferred, comparing the sentence embedding of the documents. A siamese network works on two parallel neural networks sharing their weights

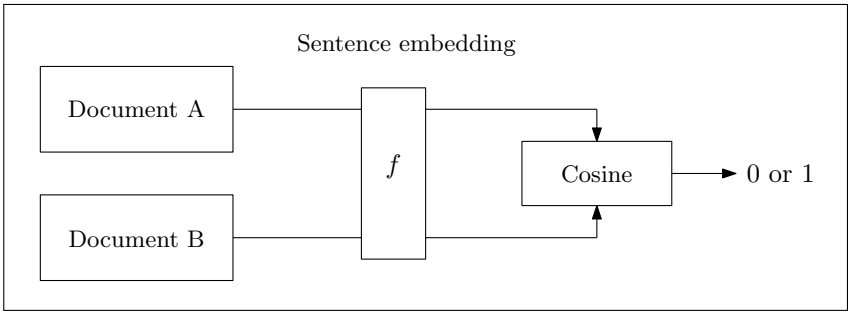

Figure 1. Architecture of a Siamese Neural Network with a Sentence Embedding (CamemBERT)

but working on different inputs. It allows to obtain a unique and static representation with a fixed length to ease comparisons. The architecture is presented in Fig 1.

The general architecture is similar to the one proposed by SBERT [19] but the underlying model is a CamemBERT [20] model warm-started from the official checkpoint and fine-tuned on 30Gb of legal data (MLM).

However, the processing of legal documents imposes an additional constraint since transformer based models are for most of them trained for 512 tokens sequences and they cannot process long sequences due the quadratic complexity of self-attention. To measure which part of a decision is likely discriminating for the similarity task, four experiments are proposed on different parts of the document:

- First 512 tokens
- Last 512 tokens
- Full document (up to 16384 tokens)
- Summarized document (up to 512 tokens)

The data extracted by NER and used for the construction of the dataset are mostly concentrated in the first part of the document, so we can assume that the first 512 tokens are sufficient to reliably constitute the pairs. Since the decisions can be long, especially for the court of appeal, the last 512 tokens may not contain any discriminating information, lower performances are to be expected. In order to process the documents without truncating them, we rely on a conversion script to extrapolate from an existing model.[2]

The fourth experiment based on summarization is particular since it is necessary to develop a summarization model alongside. This process is done in two steps. First, a dataset of 70K pairs of decisions/summaries is extracted from Légifrance[3] and focused on the Cour de Cassation and the Conseil d'Etat (summaries are available only from supreme courts). A BARThez model [14,21] (Seq2Seq) is warm-started and fine tuned on the text in-filling and de-noising tasks on the 30Gb corpus of legal data. The model is then extrapolated (16384 tokens) thanks to the above script and fine-tuned this time on the summarization task. Although the jurisdictions processed are different from those of our matching task, the trained model generates concise summaries of the main elements that the judge motivates in his/her decision.

[2] https://github.com/ccdv-ai/convert_checkpoint_to_lsg
[3] https://www.legifrance.gouv.fr/

4. Experiments and results

Three types of experiments are presented, one related to the NER tasks and another one based on extracted metadata to which standard machine-learning models are fitted. The last one based on transformers that directly process the content of the documents. The experiments are performed with a five-fold cross-validation except for the NER task.

4.1. NER results

Two dataset configurations are tesed. The first one related to the tribunal only and the second one to the court of appeal. The model used is the legal CamemBERT presented above. Results are reported in Table 2.

F-measure	**First-instance**	**Appeal**
Date of Appellate decision	N/A	92.93
Date of Judgment	90.86	82.11
Parties	83.1	76.02
ID of Appeal	N/A	90.35
ID of First-instance	88.43	82.61
Appellate Court	N/A	91.12
Tribunal	96.32	88.26

Table 2. F-measure on the NER task.

4.2. Metadata-based classification

Since the task is defined as a classification problem on tabular data and the features are strings, an embedding method is needed to transform the inputs before solving the task.

For this, two methods are used, a Bag-of-words (BoW) method where the features are summed up and another one based on a pre-trained embedding model where the features are averaged for each document. To represent the pairs, those of the two documents are concatenated.

The experiments are conducted on 6 standard machine learning models for processing tabular data:

- Linear SVM
- SVM (rbf kernel)
- Logistic regression
- Multilayer perceptron (1e-4 learning rate, ReLU activation, 3 layers of size 100)
- XGBoost (500 estimators, max depth of 10, 0.1 learning rate)

Results are reported in Table 3.

We can directly observe that the performances vary widely depending on the representation method. The simplest to implement (BoW) gives higher results on average except for the random forest. From the results, providing (noisy) metadata is enough to correctly find over 95% of the pairs.

	Linear SVC	SVM	Logistic Reg.	MLP	XGBoost
BOW					
F-measure	88.48	96.23	87.14	96.06	95.08
Avg. variation	2.00	0.51	0.81	0.23	0.29
Embedding					
F-measure	71.53	71.60	73.40	75.43	86.02
Avg. variation	0.58	0.50	0.93	1.44	0.94

Table 3. F-measure of standard ML algorithms on the matching task.

4.3. Raw document based classification with transformers

Transformers with Siamese architecture allow to compare documents without providing metadata. Their use can thus be generalized quite easily as long as a labeled set of pairs is available. The summarization task is presented first because it is reused later and the best performing models remain the non-linear ones (SVM, MLP, XGBoost).

4.3.1. Summarization task

The summarization task performed on 70k pairs of decisions/summaries from the Cour de Cassation and the Conseil d'Etat is trained on a specialized BARThez capable of processing sequences of 16384 tokens. The model is fine-tuned during 10 epochs, with an Adam optimizer, a learning rate of 8e-5, 10% of warmup steps, a linear learning rate decay, a batch size of 32 and a maximum generation length of 512 tokens. We use the summarization example script from HuggingFace[4] to train and evaluate. The model achieves 61.74/49.39/55.47 as Rouge1/Rouge2/RougeL scores [22] with a length penalty of 2 and 5 beams. Generated summaries for the matching task rely on the same hyperparameters.

4.3.2. Matching task

For the following tasks relying on the Siamese architecture, the model is trained during 15 epochs with an Adam optimizer, a learning rate of 5e-5, 500 warmup steps, a linear learning rate decay and a batch size of 32. The loss function used is a cosine similarity loss which computes the euclidian distance between the cosine similarity between `<s>` tokens from both documents and the associated label. Results are reported in Table 4.

	512 First tokens	512 Last tokens	Full document	Summary
F-measure	99.20	96.84	99.44	90.15
Avg. variation	0.17	0.54	0.11	0.78

Table 4. F-measure of transformer models on the matching task.

As expected, using the first tokens of the decision produces much better performances than using the last tokens, since discriminating data such as decision references, dates and lawyers' names are mostly located at the beginning. Processing up to 16384 tokens seems to show a marginal effect but this gain reduces the error rate by 30% (0.80% to 0.56%) which is significant for performances above 99%.

[4] https://github.com/huggingface/transformers/blob/main/examples/pytorch/summarization

The summary-based method provides disappointing results, possibly because summaries are very concise and therefore less informative, which makes the comparison of documents difficult. There is also some bias to train summaries on supreme courts rulings, which are oriented to doctrine, and infer them to lower-court decisions that focus more on facts.

The Siamese architecture is useful in production because a sentence embedding is learned during training. Thus, these embeddings can be pre-computed for each document, allowing inference on large volumes efficiently, the cosine distance being a simple normalized inner product.

5. Conclusion

We have presented a methodology to reconstitute the judicial path of a case. Through NER and simple predefined rules, a dataset of document pairs was built, allowing the task to be solved using machine and deep learning. Two methods have been proposed, one leveraging metadata, the other one focusing directly on the raw document. The experiments carried out have shown that the use of transformer based models, trained on the first part of the documents (512 first tokens), achieve very high performance and commit only few errors compared to the other methods presented and related to metadata. In practice, the use of Siamese architecture also has advantages in production. Because each decision can be represented by a pre-computed fixed length vector through sentence embedding, the comparison of a document against a large number of candidates can be done efficiently. On the other hand, methods exploiting metadata remain very dependent on the quality of the underlying NER model, the embedding method and the algorithm used to solve the task.

Acknowledgments

The authors would like to thank the French National Research Agency for funding the LAWBOT project ANR-20-CE38-0013: Deep Learning for Judicial Outcome Prediction.

References

[1] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is All you Need. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, et al., editors. Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc.; 2017. Available from: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[2] Westermann H, Savelka J, Walker VR, Ashley KD, Benyekhlef K. Sentence Embeddings and High-speed Similarity Search for Fast Computer Assisted Annotation of Legal Documents. CoRR. 2021;abs/2112.11494. Available from: https://arxiv.org/abs/2112.11494.

[3] Arpan Mandal SM Paheli Bhattacharya, Ghosh S. Improving Legal Case Summarization Using Document-Specific Catchphrases. In: JURIX, Legal Knowledge and Information Systems; 2021. p. 76-81.

[4] Aniket Deroy KG Paheli Bhattacharya, Ghosh S. An Analytical Study of Algorithmic and Expert Summaries of Legal Cases. In: JURIX, Legal Knowledge and Information Systems; 2021. p. 90-9.

[5] Aletras N, Tsarapatsanis D, Preoţiuc-Pietro D, Lampos V. Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective. PeerJ Computer Science. 2016;2:e93.

[6] Chalkidis I, Androutsopoulos I, Aletras N. Neural Legal Judgment Prediction in English. arXiv preprint arXiv:190602059. 2019.

[7] Condevaux C, Harispe S, Mussard S, Zambrano G. Weakly Supervised One-shot Classification using Recurrent Neural Networks with Attention: Application to Claim Acceptance Detection. In: JURIX 2019 32nd International Conference on Legal Knowledge and Information Systems. Madrid, Spain; 2019. Available from: https://hal.archives-ouvertes.fr/hal-02407405.

[8] Moodley K, Serrano PVH, van Dijck G, Dumontier M. Similarity and relevance of court decisions: A computational study on CJEU cases. In: JURIX; 2019. p. 63-72.

[9] Charmet T, Cherichi I, Allain M, Czerwinska U, Fouret A, Sagot B, et al. Complex Labelling and Similarity Prediction in Legal Texts: Automatic Analysis of France's Court of Cassation Rulings. In: LREC 2022 - 13th Language Resources and Evaluation Conference. Marseille, France; 2022. Available from: https://hal.inria.fr/hal-03663110.

[10] Vuong YTH, Bui QM, Nguyen HT, Nguyen TTT, Tran V, Phan XH, et al. SM-BERT-CR: a deep learning approach for case law retrieval with supporting model. Artificial Intelligence and Law. 2022 Aug. Available from: https://doi.org/10.1007/s10506-022-09319-6.

[11] Castano S, Falduti M, Ferrara A, Montanelli S. A knowledge-centered framework for exploration and retrieval of legal documents. Information Systems. 2022;106:101842. Available from: https://www.sciencedirect.com/science/article/pii/S0306437921000788.

[12] Zhong H, Xiao C, Tu C, Zhang T, Liu Z, Sun M. How does NLP benefit legal system: A summary of legal artificial intelligence. arXiv preprint arXiv:200412158. 2020.

[13] Geißler S. The Kairntech Sherpa – An ML Platform and API for the Enrichment of (not only) Scientific Content. In: Proceedings of the 1st International Workshop on Language Technology Platforms. Marseille, France: European Language Resources Association; 2020. p. 54-8.

[14] Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, et al. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics; 2020. p. 7871-80. Available from: https://aclanthology.org/2020.acl-main.703.

[15] Bromley J, Guyon I, LeCun Y, Säckinger E, Shah R. Signature Verification using a "Siamese" Time Delay Neural Network. In: Cowan J, Tesauro G, Alspector J, editors. Advances in Neural Information Processing Systems. vol. 6. Morgan-Kaufmann; 1993. .

[16] Chopra S, Hadsell R, LeCun Y. Learning a similarity metric discriminatively, with application to face verification. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 1; 2005. p. 539-46 vol. 1.

[17] Koch GR. Siamese Neural Networks for One-Shot Image Recognition; 2015. .

[18] Chicco D. In: Cartwright H, editor. Siamese Neural Networks: An Overview. New York, NY: Springer US; 2021. p. 73-94. Available from: https://doi.org/10.1007/978-1-0716-0826-5_3.

[19] Reimers N, Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics; 2019. Available from: https://arxiv.org/abs/1908.10084.

[20] Martin L, Muller B, Suárez PJO, Dupont Y, Romary L, de La Clergerie ÉV, et al. Camembert: a tasty french language model. arXiv preprint arXiv:191103894. 2019.

[21] Kamal Eddine M, Tixier A, Vazirgiannis M. BARThez: a Skilled Pretrained French Sequence-to-Sequence Model. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics; 2021. p. 9369-90. Available from: https://aclanthology.org/2021.emnlp-main.740.

[22] Lin CY. ROUGE: A Package for Automatic Evaluation of Summaries. In: Text Summarization Branches Out. Barcelona, Spain: Association for Computational Linguistics; 2004. p. 74-81. Available from: https://aclanthology.org/W04-1013.