Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 4754–4766
Marseille, 20-25 June 2022

Complex Labelling and Similarity Prediction in Legal Texts: Automatic Analysis of France's Court of Cassation Rulings

Thibault Charmet[1] **Inès Cherichi**[2] **Matthieu Allain**[2] **Urszula Czerwinska**[2]
Amaury Fouret[2] **Benoît Sagot**[1] **Rachel Bawden**[1]
[1]Inria, Paris, France [2]Cour de Cassation, France
firstname.lastname@{[1]inria,[2]justice}.fr

Abstract

Detecting divergences in the applications of the law (where the same legal text is applied differently by two rulings) is an important task. It is the mission of the French *Cour de Cassation*. The first step in the detection of divergences is to detect similar cases, which is currently done manually by experts. They rely on summarised versions of the rulings (syntheses and keyword sequences), which are currently produced manually and are not available for all rulings. There is also a high degree of variability in the keyword choices and the level of granularity used. In this article, we therefore aim to provide automatic tools to facilitate the search for similar rulings. We do this by (i) providing automatic keyword sequence generation models, which can be used to improve the coverage of the analysis, and (ii) providing measures of similarity based on the available texts and augmented with predicted keyword sequences. Our experiments show that the predictions improve correlations of automatically obtained similarities against our specially colelcted human judgments of similarity.

Keywords: Legal NLP, Similar Case Prediction, Classification, Summarisation

1. Introduction

The *Cour de Cassation* (Court of Cassation) is the highest court in the French judicial system for all civil and criminal matters. Its mission is to control the exact application of the law by lower courts (including courts of appeal), guaranteeing a unified interpretation of the law. It is crucial for them to be able to identify *divergences* in their own rulings, i.e. situations where several rulings apply the same legal text differently. Divergences can occur at three levels: within the Cour de Cassation, between trial courts and, more rarely, between a trial court and the Cour de Cassation. The quality of a judicial system depends on its capacity to minimise the existence of such divergences, of which about ten are currently identified each year.

Being the court of last resort for issues within its jurisdiction, it is crucial for the Cour de Cassation to be able to identify divergences that may arise within itself and, in particular, between its six chambers (First, Second and Third Civil Chambers, Social Chamber, Commercial, Economic and Financial Chamber, and Criminal Chamber). They can then implement jurisdictional and consultation mechanisms to ensure the unity of the law within the Court. The Cour de Cassation is also eager to identify divergences between rulings of the courts of appeal and lower courts.

The identification of divergences is complex, requiring strong legal analysis skills as well as a perfect mastery of the law and jurisprudence, which is why it currently relies exclusively on the human expertise of experts (magistrates and civil servants of the Cour de Cassation, analysis by academics in legal journals, etc.). An exhaustive detection of discrepancies would require the comparison of hundreds of thousands of decisions, which is impossible to achieve by human work alone. As a result, the current detection process does not allow for the identification of all divergences. Moreover, given the limited resources within the Cour de Cassation, the identification of divergences is often done by actors outside the *Cour*: law professors, commentators, lawyers, etc. This is what motivated a collaboration involving Cour de Cassation experts and specialists of natural language processing (NLP) aimed at developing automatic tools to help divergence detection.

An initial step in detecting divergences is to detect rulings (Fr. *arrêts*) that are similar in terms of their applicable legal reasoning, and this is the focus of the current paper. Similarity case matching (SCM) and legal text retrieval are well studied tasks in NLP for legal texts (Bhattacharya et al., 2019a; Rabelo et al., 2020; Rabelo et al., 2021; Xiao et al., 2019). Methods vary depending on the type of documents available, the structure of those documents and the type of similarity that is targeted, all of which are often specific to a given legal institution. Similarity prediction methods range from traditional frequency-based approaches such as TF-IDF (Kumar et al., 2011) to neural-based approaches (Mandal et al., 2017), and they rely on different sorts of input information (depending also on the availability), including the original rulings and synthesised information such as metadata (Yoshioka and Song, 2019) and summaries (Tran et al., 2019; Rossi and Kanoulas, 2019).

In the Cour de Cassation, the manual detection of similar rulings is carried out in a similar way: given a ruling, experts retrieve similar rulings by comparing two types of manually created summary that may be associated with a ruling: (i) a synthesis of the ruling

in relation to each potential means of overturning it (Fr. *sommaires*) and (ii) a keyword sequence (see Figure 1) summarising each synthesis (Fr. *titrages*). While this aids retrieval considerably, there are multiple challenges with this purely manual approach: (i) the lack of coverage of such annotations (they are only available for about 10% of all rulings available) and (ii) the high level of variability in the choice, granularity and length of keyword sequences (keywords are written in natural language and are not pre-defined), making it more difficult to detect similar rulings based on them.

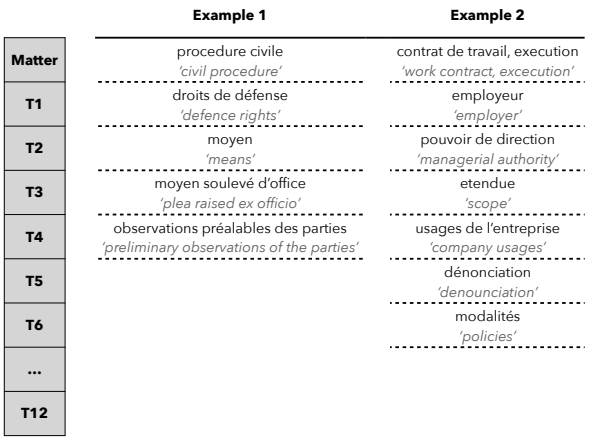


Figure 1: Two examples of keyword sequences. The first link is the matter, provided in all cases. Each keyword sequence is composed of 1 to 12 keywords, organised hierarchically from most general to most specific. English glosses are added here for readability.

In this paper, we therefore propose a number of experiments with the aim of assisting Cour de Cassation experts in their detection of similar cases to make the task easier and faster and their coverage more complete. We first propose an approach to automatically generate keyword sequences from syntheses using a neural machine translation (NMT) approach. The aim of this step is to be able to (i) in the long-term, generate keyword sequences for rulings that do not currently have them[1], and (ii) generate multiple, diverse keyword sequences for rulings in order to aid similar ruling retrieval. We then experiment with several text-based similarity measures, using the original rulings, the syntheses and the keyword sequences to predict the similarity of rulings. In this second step, we include experiments that use the predicted keyword sequence from the first step. We limit our scope to the most critical divergences for the Cour de Cassation, which are the divergences within the Cour de Cassation itself. We evaluate our ability to detect similar rulings based on manual annotations produced by experts at the Cour de Cassation, enabling us to judge the usability of our work. Our data is available for research purposes on request under a specific licence as detailed in the code repository.[2]

The remainder of the paper is structured as follows: After an overview of related work (Section 2), we describe the textual data on which our work is based, namely Cour de Cassation rulings, syntheses and keyword sequences (Section 3). Section 4 describes our methods for the two above-described steps: (i) the automatic generation of keyword sequences using NMT techniques (Section 4.1), and (ii) the similarity measures we chose to compare in our experiments (Section 4.2). We present our experiments in Section 5 and results in Section 6.

2. Related Work

The use of NLP technologies for the legal domain is a thriving area of study, with several applications being targeted (Zhong et al., 2020), including legal judgement prediction (Aletras et al., 2016), legal question answering (Monroy et al., 2009), legal summarisation (Bhattacharya et al., 2019b) and similar case matching (Bhattacharya et al., 2019a). Many, but not all, of the papers cited in this section deal with English data—our work, however, is on French.

Similar Case Matching Our main goal is to identify rulings that are similar and therefore relevant for the detection of divergences. Related tasks are similar case matching (SCM) and legal case retrieval, which can be seen as two perspectives on the same problem: given a case, detect which other cases are either similar or relevant, where the notion of similarity or relevance is dependent on the needs of the legal experts.[3]

A number of shared tasks have been organised around these topics: COLIEE (Rabelo et al., 2020; Rabelo et al., 2021), which for several years has proposed a case retrieval shared task, whereby related cases must be found from an entire dataset, and CAIL (Xiao et al., 2019), which proposes an SCM task, whereby the most similar pair of cases are to be identified out of a triplet of cases. Although the underlying idea of the task is the same, the task of case retrieval, which is more similar to our situation, is arguably more challenging since it involves mining cases from a considerably larger pool of cases (potentially thousands compared to three for the triplet-based datasets).

Previous work can be grouped into roughly two categories: (i) network-based methods, which rely on comparing documents based on their references/citations to previous cases and/or to each other (Minocha et al., 2015), and (ii) text-based methods, which involve comparing the textual content of documents, using traditional measures such as TF-IDF (Kumar et al., 2011)

[1] For some rulings, this will also require the automatic generation of syntheses, or an adaptation of the method to generate keyword sequences directly from the original rulings. We leave this to future work.

[2] https://github.com/rbawden/Similarity-cour-de-cassation

[3] Note that the needs of the legal experts can differ considerably depending on the source of the data and their role.

or more advanced techniques involving topic modelling and neural networks (Mandal et al., 2017). The most recent shared tasks show that neural approaches achieve superior results (Xiao et al., 2019). Both categories of approach depend heavily on the type of information available in the document (presence or not of references between documents and the degree of internal structure to documents), and this is determined by the individual practices of law courts, making it difficult to generalise approaches across data from different sources (Bhattacharya et al., 2019a).

Legal Text Summarisation Given that legal documents are often long and contain a large amount of information that is not directly useful for comparing content, it is frequent for SCM and case retrieval models to rely on available meta information (including topics) (Yoshioka and Song, 2019) as well as summarisation (Tran et al., 2019; Rossi and Kanoulas, 2019). Both of these types of information can be seen as a way of synthesising the content of the original document.

Legal text summarisation (Kanapala et al., 2019; Bhattacharya et al., 2019b) is a task in itself, with many works focusing on extractive summarisation (Farzindar and Lapalme, 2004; Hachey and Grover, 2006; Bouscarrat et al., 2019), whereby the summary is composed of selected sentences from the original text, although abstract summarisation has also attracted some interest (Bhattacharya et al., 2019b). As mentioned by (Bhattacharya et al., 2019b), domain-independent summarisation techniques can also be applied (Widyassari et al., 2020), although their performance will suffer unless adapted to the legal domain. The development of legal-specific language models such JuriBERT (Douka et al., 2021) can help adaptation to the particularities of the domain in these cases.

Again, the difficulty of the task is linked to which types of information are available. For example, topics and keywords may not always be available as an additional source of information (as in the case of our keyword sequences). For the information to be rich enough to be useful for similar case detection, given that nuances may be very fine, often it is necessary to categorise documents according to ultra-fine categories, which can also pose problems for automatic keyword classification (Chalkidis et al., 2019) (Tuggener et al., 2020)

3. Cour de Cassation Data

Our source of data is the Jurinet database, which includes all decisions of the Cour de Cassation since 1990, as well as all those published in the monthly bulletins since 1963. As an initial step before being able to identify divergences in the application of the law, rulings (*arrêts* in French) that are similar in terms of their applicable legal reasoning are currently manually detected by experts. It would be impractical for them to do this based on the original rulings, which are long and complex. They therefore do this with the help of two means of analysis: i) a synthesis (*sommaire* in French) for each identified means of overturning a given ruling (there can be several) and (ii) a keyword sequence (*titrage* in French) summarising each synthesis.

Syntheses A synthesis summarises the ruling from the point of a view of one identified means of potentially overturning it. The synthesis is far shorter than the original text (see Table 1 for average text lengths).

Keyword Sequences Each synthesis is associated with a sequences of keywords, organised hierarchically, such that the keywords become increasingly specific (two examples are given in Figure 1). The first element of the sequence, the matter (*matière* in French), is provided for all rulings, and is then followed by a certain number of keywords (in practice up to 12).

Text type	#examples	Avg. #words	#unique words
Ruling	147,729	1023	662,668
Synthesis	182,359	83	145,000
Keyword sequence	182,359	19	39,988

Table 1: Statistics on the analysed rulings (post-cleaning–see Section 5).

Currently, only around 20% (∼150k) of the rulings have been analysed by experts (annotated with at least one synthesis and keyword sequence), which severely limits the scope of case retrieval. What is more, keyword sequences are partly subjective and therefore subject to variability in terms of the choice of keywords (which are not strictly pre-defined) and their level of granularity (some experts choose to include more details and therefore provide longer sequences). As shown in Figure 2, the number of unique keywords at each level of the keyword sequence is often very high and also highly variable; the number of possibilities first increases as the level of granularity increases (up to the 4th link in the sequence) and then decreases (from the 5th link) due to the fact that there are fewer keyword sequences of that length.

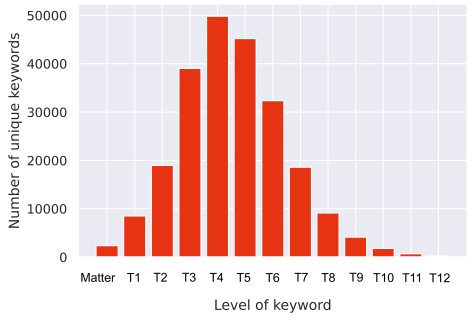


Figure 2: The number of unique keywords at each level of the keyword sequences.

Having multiple keyword sequences for a given ruling would be a way of facilitating retrieval of rulings, although this is costly to do manually. The motivation of our work is to assist the experts in their identification of similar rulings by providing automatic methods of:

1. Generating keyword sequences, which can be used (i) to increase the coverage of the experts' analysis by providing recommendations of keyword sequences, and (ii) to generate multiple and more diverse keyword sequences for those rulings that are already analysed.

2. Calculating ruling similarity (in terms of legal reasoning), using (amongst other texts) automatically produced keyword sequences.

4. Automatic Detection of Similar Documents and Automatic Titling

Inspired by the Cour de Cassation's own manual procedure and by related work in automatic case retrieval (see Section 2), we seek to improve the detection of similar documents by relying on not only the original rulings but also the associated summaries (syntheses and keyword sequences). In Section 4.1, we describe our approach to the automatic generation of keyword sequences from syntheses. This step has two goals: (i) being able to use keyword sequences on rulings for which no keyword sequences are manually provided, and (ii) being able to provide additional keyword sequences to rulings with manually given keyword sequences, thereby improving the recall of keyword sequence-based similarity metrics. In Section 4.2, we describe the similarity measures compared. This section also includes a description of the manual annotations produced by the Cour de Cassation experts for the evaluation of our similarity models.

4.1. Keyword Sequence Generation

We begin by describing our MT-inspired approach (Section 4.1.1), the dataset on which we trained and evaluated our models (Section 4.1.2) and the custom automatic metrics we used (Section 4.1.3).

4.1.1. MT-inspired Generation

We choose to use an MT-inspired approach, treating syntheses as source texts and producing keyword sequences as target texts. There are two possible modelling choices for the production of the keyword sequences (shown in Figure 3):

1. Predict whole keywords at a time, each keyword being a separate item in the target vocabulary (e.g. `droits␣de␣défense`, `moyen␣soulevé␣d'office`, etc.);

2. Insert keyword boundaries via pseudo-tokens (`<t>`) between keywords and treat the sequence as an otherwise unstructured sequence of tokens (e.g. `droits de défense <t> moyen soulevé d'office`, etc.).

We choose to use the second approach because the first one would result in a more severe data sparsity issue. On a related note, it enables us to use subword segmentation (Sennrich et al., 2016), a standard pre-processing step in NMT used to encourage generalisation over vocabulary items. An additional advantage is that this makes the source and target vocabularies more similar, making it possible to test source-target vocabulary sharing, which is a common choice in NMT. It also makes the approach compatible with using pretrained language models in the future to boost performance. Finally, it is an approach that enables novel keywords to be generated (the first approach limits the keywords to those already seen during training), providing more diversity in the keyword sequences generated.
Given that all rulings are associated with a matter, we also condition generation on the matter, including it as a prefix on the source-side, separated from the synthesis by the pseudo-token `<t>`.

4.1.2. Keyword Sequence Dataset

We automatically created a keyword sequence dataset by extracting ⟨synthesis, keyword sequence⟩ pairs from the Jurinet database described in Section 3, and split it into train, dev and test sets (see Table 2).

Dataset	#⟨synthesis, keyword sequence⟩ pairs
Train	159,836
Dev	1,833
Test	20,690

Table 2: Keyword sequence dataset statistics.

Data Preprocessing We apply manually designed cleaning rules to the initial data to handle encoding problems, convert HTML tags, remove unwanted artefacts introduced by the court software and unknown characters. All text is lower-cased to counterbalance inconsistencies in the court data. For the generation of keyword sequences, we applied subword segmentation to both the input syntheses and the output keyword sequences by applying SENTENCEPIECE (Kudo and Richardson, 2018) with the BPE strategy (Sennrich et al., 2016), testing vocabulary sizes from 4k to 32k.

4.1.3. Keyword Sequence Evaluation

The choice of automatic evaluation metric is important to gain proper insight into model performance. Given the high degree of variability in the keyword sequences, including in terms of granularity (i.e. linked to length), an accuracy based on an exact match between a predicted sequence and a reference sequence is inappropriate. We propose and describe below several different evaluation metrics at the corpus level, which all take into account level-wise comparisons: (i) accuracy scores for each level of the keyword sequence (from 1 to 12), (ii) an averaged accuracy score combining all levels and (ii) a custom weighted average accuracy to take into account the relative importance of levels.

Level-wise Accuracy Scores We compare models level by level by calculating an accuracy for each level of the sequence (1 to 12) over a whole corpus; i.e. how

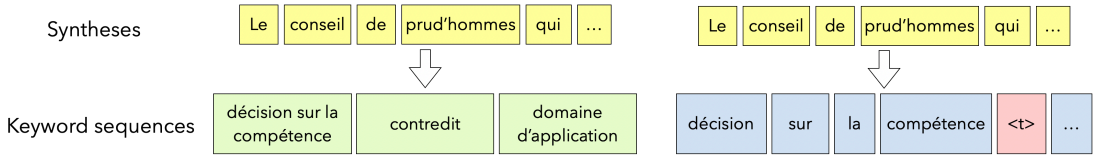


Figure 3: Comparison of the two possible MT-style approaches, with our proposed method on the right.

often the predicted keyword at a given level is the same as the reference keyword. This shows the decrease in prediction quality as we go from the first, coarse-grained levels to the last, more detailed and therefore sparse ones. As keyword sequences are of variable length, we only include a prediction in the calculation of the accuracy score for a level if either the prediction or the reference sequence has a keyword at that level.

Micro-averaged Accuracy We define this accuracy as the total number of correctly predicted keywords divided by the total number of keywords that were or should been predicted, only counting keywords where one exists for either the prediction or the gold sequence (a form of micro-average).

Weighted Accuracy Feedback from the experts indicated that the first levels of the keyword sequence were the most important and the higher-numbered levels less important, since the amount of granularity can be very variable and the consistency of annotation decreases as the level increases. We therefore also produce a weighted accuracy, where the weight distribution is given by the function $\frac{1}{3.85}x^{-0.8}$ (shown in Figure 4): larger weights are assigned to higher levels, the accuracy of the first keyword contributing 6.5 times more than the 12$^{\text{th}}$ one.[4]

4.2. Similarity Prediction

Our overall aim is to detect rulings that are similar to one another in a legal sense, in terms of how law has been applied. Therefore, while there already exist many NLP techniques for calculating the similarity between natural language texts, it is unclear whether these will capture the same type of similarity as we want here. As a result, we first selected a number of relevant (unsupervised) features, which we describe in (Section 4.2.1), in order to train a similarity assessment model based on them. Since no annotated corpus was available, we collected manual annotations of similarity by legal experts from the Cour de Cassation, as described in Section 4.2.2, to serve as a source of training and evaluation data (see Section 4.2.3).

4.2.1. Feature-based Similar Case Matching

We predict the similarity between pairs of cases by training multi-layer perceptron models to predict similarity scores based on a number of unsupervised features. These features are obtained from the comparison of pairs of rulings, syntheses or keyword sequences using one of the two following scores:

1. TF-IDF-based Similarity (TISIM) The idea underlying this feature is that case similarity can be captured through the presence of particular terms or sequences of terms specific to certain documents. A text is assigned a vector representation corresponding to learned TF-IDF weights based on the vocabulary of the texts used for training. The score is then computed by calculating the cosine similarity between a pair of vector representations (i.e. between a pair of rulings, a pair of syntheses or a pair of keyword sequences). The models are trained using `scikit-learn` (Pedregosa et al., 2011) and we vary different parameters to find the best settings for each type of text: the type of text used to train the model (rulings, syntheses or keyword sequences, the maximum n-gram size for vocabulary features (up to 3) and the maximum number of features (250k, 500k, 1M, 2M).[5] These experiments are detailed in Appendix B. The chosen models are as follows: (i) for the comparison of keyword sequences, it was best to train on keyword sequences using unigrams only and a maximum of 250k features, (ii) for the comparison of syntheses, it was best to train on syntheses using unigrams and bigrams and a maxuimum of 250k features and (iii) for the comparison of rulings, surprisingly, it was best to train on syntheses, using unigrams, bigrams and trigrams and maximum of 2M features.

2. Normalised Edit-distance Similarity (EDSIM) This score is defined as 1 minus the normalised edition distance between two texts, where the edit distance is computed at the keyword level for keyword sequences and at the token level[6] for syntheses and rulings:

$$\text{EDsim}\,(t_1, t_2) = 1 - \frac{D\,(t_1, t_2)}{\max\left(\text{len}(t_1), \text{len}(t_2)\right)},$$

where D is the Levenshtein distance and len the length of the text, i.e. the number of keywords/tokens.

Comparing Multiple Texts per Example In Section 4.1, we proposed to automatically generate keyword sequences from syntheses and can therefore use multiple keyword sequences per example to compute the above scores. When each example has one than one keyword sequence, we compute the similarity matrix

[4] This function was empirically selected as it gives ∼25% of the total weight to the first keyword and then decreasing weights to later ones in a way that reflects experts' views on their relative importance. It is normalised so that the sum of the weights for all 12 levels is (almost) exactly 1.

[5] We lowercase and remove accents.

[6] A token is a white-spaced delimited character sequence.

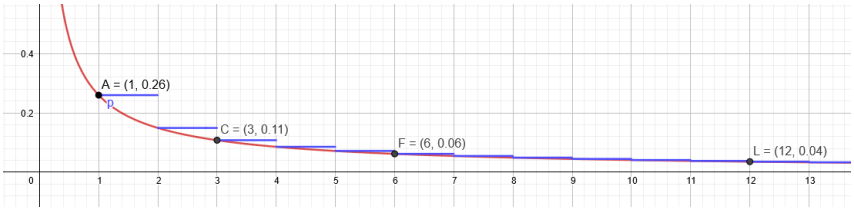


Figure 4: Custom weight distribution function for the calculation of the weighted average score.

between all available keyword sequences of example 1 and those of example 2 and then take either the mean score (AVG) or the maximum score (MAX).

4.2.2. Reference Ruling Similarity Corpus

Given the very specific nature of the similarity sought, we evaluate the different measures against gold standard annotations produced by experts at the Cour de Cassation. Our reference corpus comprises pairs of rulings that are to be classified according to their legal similarity. The annotation guide, with levels of similarity (from 0 to 3) was decided in discussion with Cour de Cassation experts: 0: no similarity, 1: weak similarity (same legal matter), 2: fair similarity (same legal matter and same domain), and 3: strong similarity (same legal question examined, even if the response is different). Crucially, we make sure that the rulings present in the Ruling Similarity Corpus are not part of the train and dev sets of the keyword sequence dataset.[7]

Selection of Example Pairs A major challenge in collecting annotations was to ensure that there were a sufficient number of positive examples included in the sample annotated, since the majority of rulings are not at all similar. We therefore pre-selected pairs of examples to be annotated for their similarity to maximise the chance of having a range of different similarities.
From the original Jurinet data, we selected a total of 780 pairs of rulings using three different methods to ensure a range of different similarities:

1. 1/3 from the select number of rulings already identified as being related (French *rapprochements*);
2. 1/3 from pairs that are from varied TISIM scores;
3. 1/3 from pairs whose keyword sequences are from varied EDSIM scores.

To avoid having to calculate all pair-wise similarities, these were calculated from a random subsample of 5000 examples. For the 2nd and 3rd partitions, we selected 100 examples from three score buckets manually defined such that the buckets are of similar size.[8]

Annotations Experts had access to the keyword sequences, the syntheses and to the complete rulings, although (from feedback), they relied mainly on the keyword sequences. There were a total of 16 experts distributed across the different chambers, with 2 annotations per pair. They annotated a total of 780 case pairs. The correlation coefficient between the two sets of annotations is as high as 0.929 (Pearson's r) and 0.809 (Cohen's κ), which shows that the task was well defined and the annotations are consistent.

4.2.3. Similar Case Matching Evaluation

We use Pearson's correlation to calculate the correlation between the predicted similarity scores and the human similiarity annotations.

5. Experimental Setup

5.1. Keyword Sequence Generation

Our keyword sequence generation models are Transformer models (Vaswani et al., 2017), trained using FAIRSEQ (Ott et al., 2019). We compare different model sizes: (i) BASE: 6 encoder and decoder layers, 8 attention heads, embedding dimension of 512 and feed-forward dimension of 2048) and (ii) MINI: 2 encoder and decoder layers, 2 attention heads, embedding dimension of 256 and feed-forward dimension of 1024.[9]

5.2. Similar Case Matching

As previously mentioned, we train multi-layer perception regressors based on the features described in Section 4.2.1. We train the model using `scikit-learn` and choose to use 3 layers each of dimension 48 and a maximum number of 2000 iterations following preliminary experiments.
Each feature is a similarity score based on the comparison of one of the three types of text (rulings, syntheses

[7] In practice, we first created the Ruling Similarity Corpus and then split the keyword sequence dataset such that any ruling found in the Ruling Similarity Corpus was in the keyword sequence dataset's test set. Since the former is much smaller than the latter, this did not create any significant bias in the distribution of rulings in the keyword sequence test set.

[8] For TISIM, bucket 2 contains ruling pairs whose TISIM score is between 0.4 and 0.6, buckets 1 (resp. 3) containing ruling pairs with lower (resp. higher) scores. For the average EDSIM score, bucket 2 contains ruling pairs whose score is between 0.2 and 0.5, buckets 1 (resp. 3) containing ruling pairs with lower (resp. higher) scores.

[9] We train each model until convergence, using the cross entropy loss, learning rate of 0.001, and the Adam optimiser (Kingma and Ba, 2017), dropout of 0.1 and a batch size 2048k tokens with gradient accumulation of 10. All models are trained on a single GPU (Quadro RTX 8000, 48GB). The best checkpoint for each model was chosen based on the dev set weighted accuracy.

or keyword sequences) and using one of the two similarity scores (TISIM or EDSIM). We generate features based on the gold (original) texts, but also on the predicted keyword sequences using the method described in Section 4.1.1, and we also combine these features. Given the small size of our Ruling Similarity Corpus, we perform cross-fold validation on the dataset with 20 random splits, each time taking 50% for the training data and 50% for testing (i.e. 395 for each). Note that we use the same setup regardless of the number of features being tested (including for a single feature).

6. Results

We present the results of two previously defined steps: (i) automatic titling of cases (Section 6.1) and (ii) prediction of case similarity (Section 6.2).

6.1. Keyword Sequence Generation

Table 3 shows the results for automatic keyword sequence generation. We compare models in terms of micro-averaged accuracy and weighted accuracy. We test two sizes of Transformer: BASE and MINI and for each one test the effect of sharing source and target vocabularies or not. We report the best model results for each combination (full results are in Appendix A). The best model is the BASE Transformer model with a BPE vocabulary size of 8k and sharing vocabularies for a weighted accuracy of 34.90%. We also experiment with a larger model in which the encoder and decoder are both pretrained on the French language model CamemBERT (Martin et al., 2020) and then fine-tuned on our training data. This model performs even better to reach a weighted accuracy of 35.84%.

			Accuracy (%)	
Arch.	BPE	Share	Micro-avg	Weighted
BASE	8k	✓	**32.29**	**34.90**
BASE	24k	×	30.35	33.10
MINI	16k	✓	29.71	31.35
MINI	8k	×	30.72	31.97
CamemBERT	32k	✓	**34.14**	**35.84**

Table 3: Results of the automatic generation of keyword sequences on the development set.

Figure 5 shows the level-wise accuracies for each model trained. The accuracy of the first keyword is the highest (71% for the best model) and the accuracy decreases as the keyword sequence level increases. This explains the relatively low averaged scores.

6.2. Similarity Prediction

Correlation for Individual "Gold" Features We begin by looking at individual features representing the automatic similarity scores between gold texts (rulings, syntheses and keyword sequences). We calculate the correlation of each of these features with the expert similarity scores. The results (Table 4) show

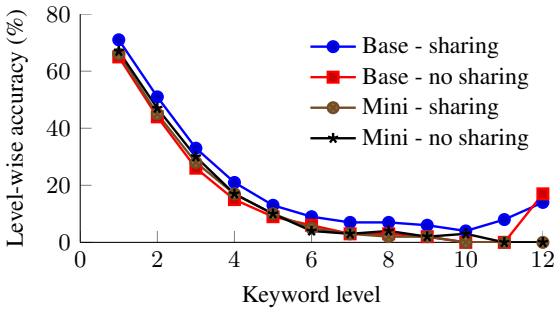


Figure 5: Level-wise comparison of accuracies across keyword sequence generation models.

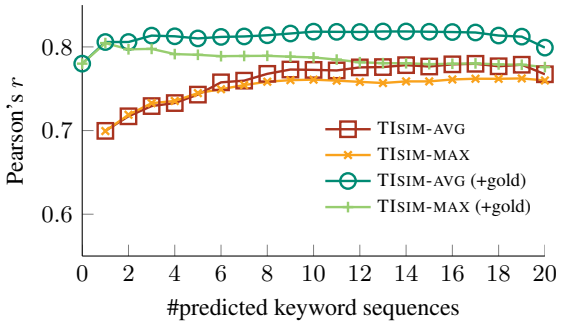
that TISIM-based features are better correlated than EDSIM-based features (see Appendix C for more detailed experiments). As per our intuitions, the texts that give the highest correlation with human similarity scores are the keyword sequences, followed by the syntheses, indicative of the fact that they better summarise the type of similarity sought. The correlations using the gold rulings are very weak and we hereafter choose not to further test these features.

Texts compared	TISIM	EDSIM
Rulings	0.04	0.11
Syntheses	0.74	0.49
Keyword sequences	0.78	0.64

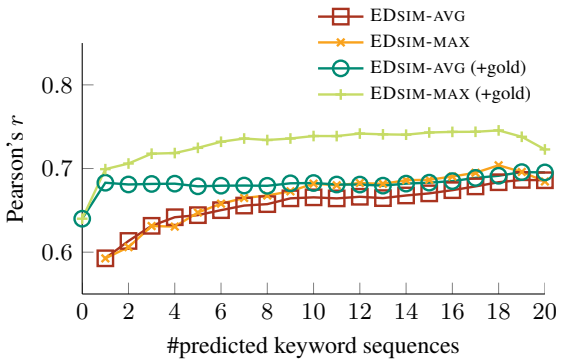
Table 4: Correlations (Pearson's r) of individual "gold" similarity features with expert similarity annotations.

Correlation for Predicted Keyword sequences We then test the correlation of individual features based on predicted keyword sequences by applying the best non-pretrained model from Section 6.1 (BASE-8k-sharing) to the syntheses to produce one or more predicted keyword sequences.[10] In practice, we use a beam size of 20 and vary the number of predictions k retrieved. We also experiment with adding the gold keyword sequence to the set of k predicted keyword sequences (i.e. $k+1$). The results are shown in Figure 6a (for TISIM) and Figure 6b (for EDSIM). Four observations can be made: (i) TISIM features again perform better than EDSIM ones, (ii) AVG is better for TISIM and MAX is better for EDSIM, (iii) including the gold keyword sequence in the keyword sequence set significantly helps in all cases, even when taking into account that $k+1$ keyword sequences are being used, and (iv) in most cases, including more predicted keyword sequences improves the correlation. This confirms our hypothesis that automatically producing additional keyword sequences can help. The best result (of 0.82) is achieved using TISIM-

[10] Interestingly, the use of CamemBERT predictions result in lower correlations with similarity judgements, despite them having a higher accuracy on keyword sequence prediction. We leave the investigation of this to future work.



(a) Using TISIM to calculate the pairwise feature score.



(b) Using EDSIM to calculate the pairwise feature score.

Figure 6: Correlation for predicted keyword sequences when varying the number of predicted keyword sequences per example (using either the MAX or AVG).

AVG, 3 predicted keyword sequences and the gold keyword sequence. However, importantly, good correlations can be achieved if this gold keyword sequence is not available: using the TISIM-AVG model and 9 predicted keyword sequences gives a correlation comparable to just using the gold keyword sequence.

Combining Features In light of these results for the correlation of individual features and backed up by additional experiments that can be found in Appendix B, we decide not to report as main results features combinations involving EDSIM scores. We also choose to include the gold keyword sequence similarity as a separate feature rather than including it in the same feature as predicted keyword sequences as done above, after we found that scores are very marginally better (see Appendix C for details).

We therefore combine 3 types of features, each one representing a TISIM score per example pair: (i) gold keyword sequence, (ii) gold synthesis, and (ii) predicted keyword sequences (varying k and using AVG). The results in Figure 7 show that combining features is beneficial. The best results are achieved when combining all 3 features, even when using only one predicted keyword sequence. The best results are obtained using 20 predicted keyword sequences (0.854), but results are only marginally better than using 1 (0.846). These results show that syntheses provide complimentary information with respect to keyword sequences, additional

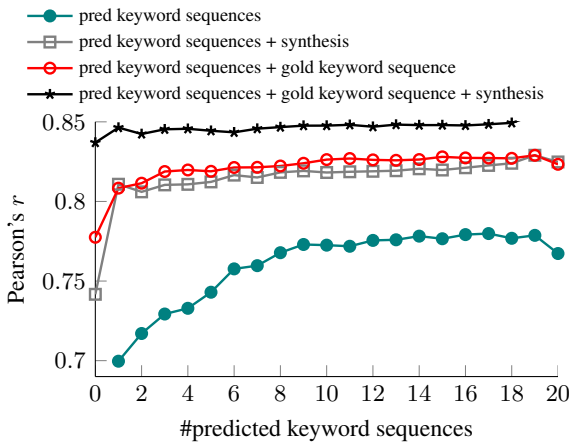


Figure 7: Correlation for feature combinations when varying the number of predicted keyword sequences.

predicted keyword sequences also improve the correlation, and if there are no gold keyword sequences, comparable scores can be achieved by using predicted keyword sequences.

7. Conclusion and Perspectives

The development of tools to help the work of legal experts in the Cour de Cassation is an important step in improving the coverage and accuracy of the detection of divergences. As well as the new dataset we provide, we presented two series of experiments (i) automatic keyword sequence generation using an MT-inspired approaches and (ii) ruling similarity detection, using both the initial documents and the multiple predicted keyword sequences from step (i). Our experiments showed that the predicted keyword sequences were highly beneficial in the calculation of ruling similarities and could compensate for the lack of gold keyword sequence to produce comparable scores if more predicted keyword sequences are used. This confirms our initial hypothesis and is likely to greatly improve the coverage of similar case detection within the French Cour de Cassation, which can currently only be done for rulings that have associated manual keyword sequences.

In future work, we hope to further investigate the impact of both the quality and diversity of predicted keyword sequences in light of our initial experiments with the CamemBERT model.

Acknowledgements

This project was partly funded by the "Lab IA," an initiative from the French Government's Department for Digital Affairs (DINUM) that aims to help French public institutions develop AI-related projects in collaboration with Inria. It was also partly funded by the two last authors' chairs in the PRAIRIE institute funded by the French national agency ANR as part of the "Investissements d'avenir" programme under the reference ANR-19-P3IA-0001.

8. Bibliographical References

Aletras, N., Tsarapatsanis, D., Preoţiuc-Pietro, D., and Lampos, V. (2016). Predicting judicial decisions of the European Court of Human Rights: a Natural Language Processing perspective. *PeerJ Computer Science*, 2(e93).

Bhattacharya, P., Ghosh, K., Pal, A., and Ghosh, S. (2019a). Methods for computing legal document similarity: A comparative study. In *Proceedings of the CEILI Workshop on Legal Data Analysis*, Madrid, Spain.

Bhattacharya, P., Hiware, K., Rajgaria, S., Pochhi, N., Ghosh, K., and Ghosh, S. (2019b). A comparative study of summarization algorithms applied to legal case judgments. In *Proceedings of the 41st European Conference on IR Research*, pages 413–428, Cologne, Germany.

Bouscarrat, L., Bonnefoy, A., Peel, T., and Pereira, C. (2019). STRASS: A light and effective method for extractive summarization based on sentence embeddings. In *Proceedings of the 57th Conference of the Association for Computational Linguistics: Student Research Workshop*, pages 243–252, Florence, Italy.

Chalkidis, I., Fergadiotis, E., Malakasiotis, P., Aletras, N., and Androutsopoulos, I. (2019). Extreme Multi-Label Legal Text Classification: A Case Study in EU Legislation. In *Proceedings of the Natural Legal Language Processing Workshop 2019*, pages 78–87, Minneapolis, Minesotta.

Douka, S., Abdine, H., Vazirgiannis, M., El Hamdani, R., and Restrepo Amariles, D. (2021). JuriBERT: A masked-language model adaptation for French legal text. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 95–101, Punta Cana, Dominican Republic.

Farzindar, A. and Lapalme, G. (2004). LetSum, an automatic legal text summarizing system. In *Proceedings of the Legal Knowledge and Information Systems. Jurix 2004: The 17th Annual Conference*, volume 120, pages 11–18, Berlin, Germany.

Hachey, B. and Grover, C. (2006). Extractive summarisation of legal texts. *Artificial Intelligence and Law*, 14(4):305–345.

Kanapala, A., Pal, S., and Pamula, R. (2019). Text Summarization from Legal Documents: A Survey. *Artificial Intelligence Review*, 51(3):371–402.

Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization.

Kudo, T. and Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium.

Kumar, S., Reddy, P. K., Reddy, V. B., and Singh, A. (2011). Similarity analysis of legal judgments. In *Proceedings of the Fourth Annual ACM Bangalore Conference*, pages 1–4, Bangalore, India.

Mandal, A., Chaki, R., Saha, S., Ghosh, K., Pal, A., and Ghosh, S. (2017). Measuring similarity among legal court case documents. In *Proceedings of the 10th Annual ACM India Compute Conference*, pages 1–9, Bhopal, India.

Martin, L., Muller, B., Ortiz Suárez, P. J., Dupont, Y., Romary, L., de la Clergerie, É., Seddah, D., and Sagot, B. (2020). CamemBERT: a Tasty French Language Model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Minocha, A., Singh, N., and Srivastava, A. (2015). Finding Relevant Indian Judgments using Dispersion of Citation Network. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1085–1088, Florence, Italy.

Monroy, A., Calvo, H., and Gelbukh, A. (2009). NLP for shallow question answering of legal documents using graphs. In *Computational Linguistics and Intelligent Text Processing*, pages 498–508. Springer Berlin Heidelberg.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. (2019). fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Rabelo, J., Kim, M.-Y., Goebel, R., Yoshioka, M., Kano, Y., and Satoh, K. (2020). A Summary of the COLIEE 2019 Competition. In *New Frontiers in Artificial Intelligence*, pages 34–49. Springer International Publishing.

Rabelo, J., Kano, Y., Kim, M.-Y., Yoshioka, M., and Satoh, K. (2021). Summary of the Competition on Legal Information Extraction/Entailment (COLIEE) 2021. In *Proceedings of the 8th International Competition on Legal Information Extraction/Entailment (COLIEE 2021)*, pages 1–7, Online.

Rossi, J. and Kanoulas, E. (2019). Legal information retrieval with generalized language models. In *Proceedings of the 6th Competition on Legal Information Extraction/Entailment (COLIEE 2019)*, Montreal, Quebec, Canada.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany.

Tran, V., Le Nguyen, M., and Satoh, K. (2019).

Building Legal Case Retrieval Systems with Lexical Matching and Summarization using A Pre-Trained Phrase Scoring Model. In *Proceedings of the 17th International Conference on Artificial Intelligence and Law*, pages 275–282, Montreal, Quebec, Canada.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. U., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008.

Widyassari, A. P., Rustad, S., Shidik, G. F., Noersasongko, E., Syukur, A., Affandy, A., and Setiadi, D. R. I. M. (2020). Review of automatic text summarization techniques & methods. *Journal of King Saud University - Computer and Information Sciences*.

Xiao, C., Zhong, H., Guo, Z., Tu, C., Liu, Z., Sun, M., Zhang, T., Han, X., Hu, Z., Wang, H., and Xu, J. (2019). CAIL2019-SCM: A Dataset of Similar Case Matching in Legal Domain. arXiv. 1911.08962.

Yoshioka, M. and Song, Z. (2019). HUKB at COLIEE 2019 information retrieval task-utilization of metadata for relevant case retrieval. *Proceedings of the 6th Competition on Legal Information Extraction/Entailment (COLIEE 2019)*.

Zhong, H., Xiao, C., Tu, C., Zhang, T., Liu, Z., and Sun, M. (2020). How does NLP benefit legal system: A summary of legal artificial intelligence. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5218–5230, Online.

9. Language Resource References

Tuggener, D., von Däniken, P., Peetz, T., and Cieliebak, M. (2020). LEDGAR: A large-scale multi-label corpus for text classification of legal provisions in contracts. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1235–1241, Marseille, France.

Appendices

A. Hyper-parameter search for keyword sequence prediction

In order to choose a good keyword sequence prediction model, as mentioned in Section 5, we test multiple scenarios in terms of (i) model size, (ii) BPE vocabulary size (jointly learned over syntheses and keyword sequences) and (iii) whether or not the encoder and decoder embedding matrices are shared. We report results according to weighted accuracy and micro-averaged accuracy in Table 5 for the base model and Table 6 for the mini model. We also report the results visually in Figures 8 and 9. The chosen model is the BASE model with a vocabulary of 8k and shared encoder-decoder embeddings.

		Accuracy (%)	
BPE	Share	Micro-avg	Weighted
4k	✓	31.80	34.12
4k	×	8.35	9.87
8k	✓	**32.29**	**34.90**
8k	×	15.28	17.62
16k	✓	33.02	34.36
16k	×	29.20	30.86
24k	✓	30.08	33.58
24k	×	30.35	32.48
32k	✓	31.31	34.29
32k	×	30.00	33.10

Table 5: Results of the automatic generation of keyword sequences on the dev set for the base model.

		Accuracy (%)	
BPE	Share	Micro-avg	Weighted
8k	✓	29.60	30.85
8k	×	**30.72**	**31.97**
16k	✓	29.71	31.35
16k	×	29.55	30.86
24k	✓	28.30	29.88
24k	×	29.77	31.44
32k	✓	28.65	30.84

Table 6: Results of the automatic generation of keyword sequences on the dev set for the mini model.

B. Hyper-parameter search for TF-IDF models

We test different scenarios for the training of the TF-IDF models in order to find the best setting for the comparison of each type of text: keyword sequences, syntheses and rulings. We vary:

- The training text on which the model is trained (keyword sequences, syntheses and rulings). We would expect the best training text type to be the same as the one on which the model is applied.

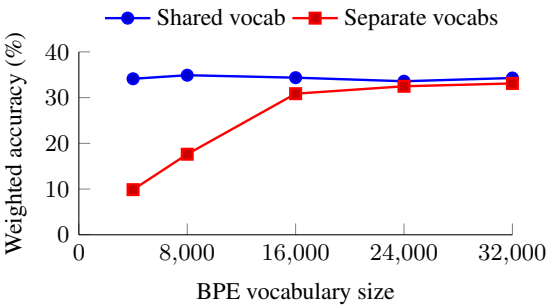


Figure 8: Weighted accuracies for different BPE vocab sizes, with and without vocabulary sharing during NMT training for the 'base' model.

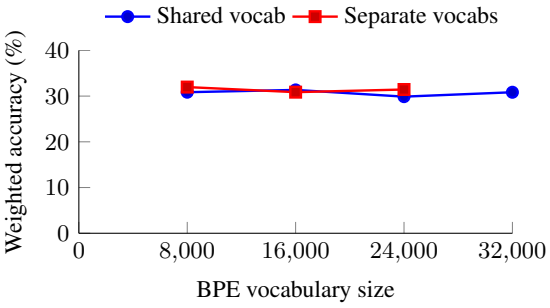


Figure 9: Weighted accuracies for different BPE vocab sizes, with and without vocabulary sharing during NMT training for the 'mini' model.

- The maximum n-gram size when defining the vocabulary features used (from 1 to 3)
- The maximum number of features to be used (250k, 500k, 1M, 2M)

All models are trained using `scikit-learn` (Pedregosa et al., 2011). When training on rulings, we use all those that do not correspond to documents in the test set. When training on either syntheses or keyword sequences, we use those that correspond to training documents. We preserve the keyword boundaries in keyword sequence files (<t> symbols) as they are important delimiters when calculating n-grams.
In order to make the results comparable with the experiments run in Section 5, instead of simply computing the Pearson correlation based on the raw scores, we train a multi-layer perceptron (using `scikit-learn`) on the individual features using cross-fold validation, as indicated in Section 5.
The results are shown in Figures 10a-10c.

- Figure 10a shows the results when applying the models to pairs of keyword sequences. Unsurprisingly, the best models are those trained on keyword sequences. There is little difference when varying n-grams and the maximum number of features and therefore, we choose to use the simplest model, i.e. 1-gram, 250k features maximum.

- Figure 10b shows the results when applying the models to pairs of syntheses. Again, the best results are obtained using the same texts to train as those used for testing (i.e. syntheses). The best results are achieved with a maximum n-gram size of 2 or 3, and similar results are achieved for all three vocabulary sizes. Given these very similar results, we choose to use the simpler of the models, namely with a maximum n-gram size of 2 and a maximum number of 250k features.

- Figure 10c shows the results when applying the models to pairs of rulings. The best results are actually obtained using syntheses to train. The best results are achieved using 3-grams and a maximum of 2M features. We choose not to increase these parameters more to keep computing cost reasonable.

The chosen models are summarised in Table 7

	Keyword sequences	Syntheses	Rulings
Train text	keyword sequences	syntheses	syntheses
Max n-gram	1	2	3
Max #feats.	250k	250k	2M

Table 7: Summary of chosen TF-IDF models for the pair-wise comparison of each type of document.

C. Additional Results for Similarity Prediction

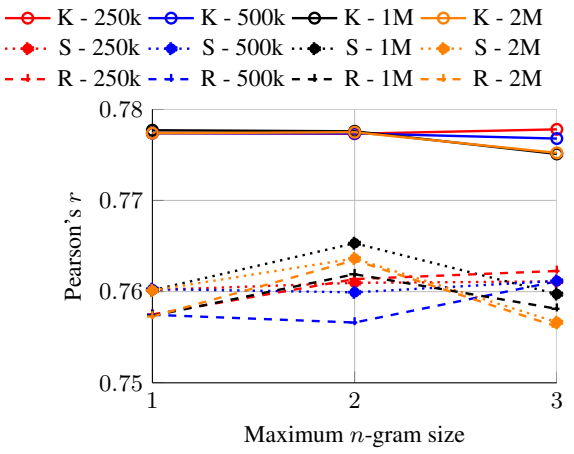
In Section 6.2, we mentioned that adding features using the edit distance similarity score did not help results. This is not necessarily surprising given that the EDSIM scores for individual features did not perform as well as TISIM scores. However features may in theory provide complementary information. In Table 8 we show that this is not the case in three scenarios:

1. Only syntheses are available;

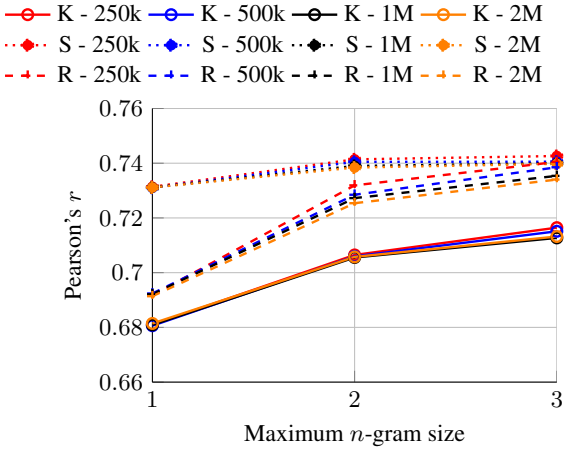
2. Only predicted keyword sequences are available (i.e. no gold keyword sequences), taking the 9 best predictions into account;

3. Gold keyword sequences are available, so we use the gold sequence as a separate feature and/or add it to the predicted keyword sequence set when computing the feature involving predicted keyword sequences.

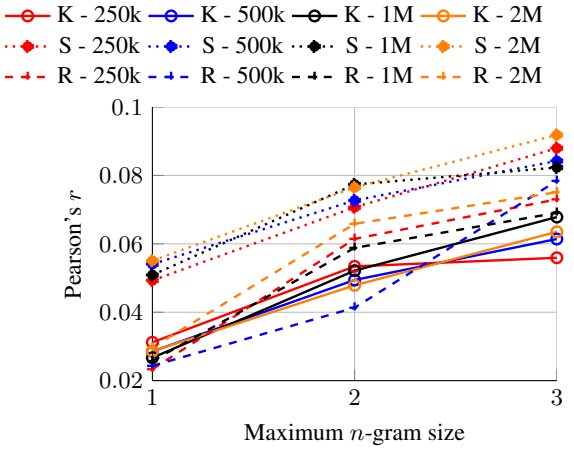
In all three scenarios, EDSIM scores perform worse than TISIM scores and when the two score types are combined, no additional gain is seen over TISIM scores.
Once we had established that it was better to use TISIM features only, we also discovered that it was better to use the similarity score of the gold keyword sequences as a separate feature rather than including



(a) Keyword sequences



(b) Syntheses



(c) Rulings

Figure 10: We trained multiple TF-IDF-based models, varying the term length n, the number of terms, and the text features on which TF-IDF scores are computed (K: keyword sequences, S: syntheses, R: rulings). We then evaluate all these models on our similarity-annotated test set (using a 20-fold cross-validation). The three figures shows the results of these different configurations when the TF-IDF model is applied to compute a TIsim feature comparing pairs of keyword sequences (a), syntheses (b) and rulings (c), i.e the correlation with similarity scores.

#feats	keyword sequences gold	#gold+pred	synth.	funcs.	r
(i) synthesis					
1	×		×	✓	TI	0.74
1	×		×	✓	ED	0.49
2	×		×	✓	TI, ED	0.74
(ii) Best keyword sequence model without gold					
1	×		0+9	×	TI	0.77
1	×		0+9	×	ED	0.66
2	×		0+9	×	TI, ED	0.77
(iii) Best keyword sequence model with gold					
2	✓		1+3	×	TI	0.82
2	✓		1+3	×	ED	0.71
4	✓		1+3	×	TI, ED	0.82

Table 8: Combination of EDSIM and TISIM features in three different scenarios.

#features	keyword sequences Gold	#gold+pred	Pearson's r
Just pred keyword sequences			
1	×	0+1	0.70
1	×	0+3	0.73
1	×	0+9	0.77
Gold+pred keyword sequences			
1	×	1+1	0.81
1	×	1+3	0.81
1	×	1+9	0.82
Gold and pred keyword sequences features			
2	✓	0+1	0.81
2	✓	0+3	0.82
2	✓	0+9	0.82
Both Gold and gold+pred keyword sequences separately			
2	✓	1+1	81
2	✓	1+3	82
2	✓	1+9	82

Table 9: Combination of predicted and gold keyword sequences in different setups. The "Just pred" section displays results when gold keyword sequences are not used. The "Gold" column indicates whether or not the gold keyword sequence is used as a separate feature. The "#gold+pred" indicates how many features are used to compute the feature involving predicted keyword sequences, starting with 0 or 1 depending on whether or not the gold keyword sequence is added to the predicted keyword sequence set, followed by the number of predicted keyword sequences used.

it in the average similarity score with the predicted keyword sequences. In Table 9, we show the different possible combinations for TISIM scores and averaging over multiple predicted keyword sequences where there are several of them. We can see that (i) the scores are slightly higher when including the gold keyword sequences as a separate feature ("Gold") rather than added to the predicted keyword sequence set ("gold+pred") and (ii) there is no additional gain in having two features, one to score gold keyword sequences and a separate one to score the gold and predicted keyword sequences.