Legal Information Retrieval with Generalized Language Models

ILPS Participation to COLIEE 2019

Rossi, J.; Kanoulas, E.

Legal Information Retrieval with Generalized Language Models

ILPS Participation to COLIEE 2019

Julien Rossi
j.rossi@uva.nl
University of Amsterdam
Amsterdam, Netherlands

Evangelos Kanoulas
e.kanoulas@uva.nl
University of Amsterdam
Amsterdam, Netherlands

ABSTRACT

This paper describes a new method to identify text pairwise relevance, in the context of the Case Law retrieval task from COLIEE 2019. This method combines text summarizing and a generalized language model in order to assess pairwise relevance. With still lots of possibilities for improvement and optimization, it achieves a competitive performance in the setting of the Retrieval for the Noticed Cases.

CCS CONCEPTS

• **Information systems → Information retrieval**; **Retrieval models and ranking**; **Relevance assessment**; *Content analysis and feature selection.*

KEYWORDS

legal text, case retrieval, document representation, ranking, neural language models, BERT

ACM Reference Format:
Julien Rossi and Evangelos Kanoulas. 2019. Legal Information Retrieval with Generalized Language Models: ILPS Participation to COLIEE 2019. In *Proceedings of COLIEE 2019 workshop: Competition on Legal Information Extraction/Entailment (COLIEE 2019).* ACM, New York, NY, USA, 4 pages.

1 INTRODUCTION

We focus on the Task 1 of the COLIEE 2019 competition: the Legal Case Retrieval Task. In this task, a given court case is considered as a query to retrieve supporting cases (also named 'noticed cases') for the query case. A noticed case supports the decision taken in the query case, although the final decision itself is irrelevant in our retrieval. What matters is the proximity of the legal themes that are tackled by the query case and the noticed cases. Each query case is given a collection of potential supporting cases (also named 'candidate cases'), these collections are provided labeled for the training dataset, and unlabelled for the unknown test dataset.

We translate this retrieval task into a ranking problem, which we formulate as a pairwise relevance classification problem, given the binary labels "Noticed" or "Not noticed". We form a pairwise semantic latent representation of the query case and the candidate case, that we submit to a Multi-Layer Perceptron. The ranking of documents is based on the score for the positive class. We study as well how to use these scores to produce the optimum search result with regards to Precision, Recall and F1-Measure.

In very recent years, Generalized Language Models have provided State of the Art performances in many tasks related to Natural Language Processing, by providing general pre-trained systems on top of which practitioners could build classifiers for specific tasks. This approach is the opposite of ad-hoc specific systems designed for specific tasks.

The central problem is the projection of words and documents into a latent vector space in which usual vector operations, such as the cosine similarity, would mirror semantic proximity as observed by a human reader. Word2Vec[9] could generate a fixed representation per word, given a corpus, but it could not account for the different context any word might be used in. Contextual embeddings, such as CoVE[8] and ELMo[10] solved for that weakness, and introduced a collection of task-specific models. While ULMFit[5] was the first to introduce the idea of having two separate training phases: namely a pre-training phase, where a Language Model is being learnt, followed by a fine-tuning phase, where a system built on top of the feature generation is trained on a task. OpenAI GPT[11] continued that idea based on the concepts behind Transformers[14] for feature generation, and unsupervised pre-training[3] as a pre-training phase. BERT[4] added a bidirectional context for word features, while adhering to the two-step principle of pre-training and fine-tuning. Even more recently, OpenAI GPT 2[12] and MT-DNN[7] continued to improve on most of the NLP tasks.

In this paper, we formulate our pairwise relevance task as a downstream task for which a pre-trained BERT system is getting fine-tuned.

2 METHODOLOGY

2.1 Text Preprocessing

We consider the corpus as the complete collection of query cases, and unique candidate cases from the training dataset, with an estimated size of around 10000 unique documents, a total of 26 million tokens from a vocabulary of 370000 unique terms.

BERT uses WordPiece[15] tokens as inputs. Existing pre-trained BERT models accept inputs up to 512 WordPiece tokens. By design, WordPiece will subdivide each token into multiple tokens, we consider in our preparation that this will double the number of tokens observed under a standard Punkt[6] tokenizer as implemented in NLTK[2]. Our observation of the training dataset, in Figure 1, shows that we have to consider that all documents will be longer than 512 tokens, even by a factor of up to 2 0.

For this reason, we introduce a summarization of the texts in the corpus, as implemented in gensim[13] using TextRank[1]. We

COLIEE 2019, June 21, 2019, Montreal, Quebec

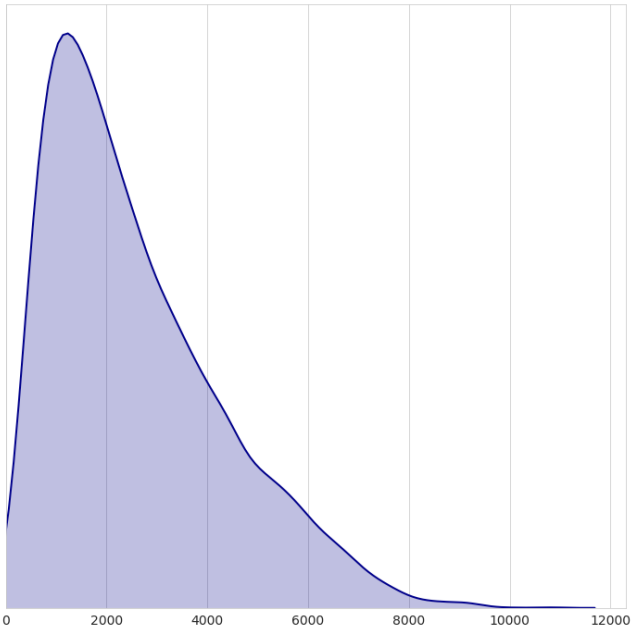

Figure 1: Distribution of the number of tokens per document.

choose to limit the size of the summary to 180 words, so a concatenated pair of texts will not be longer than 512 WordPiece tokens.

2.2 Documents Pairwise Embeddings

We instantiate a pre-trained BERT model (namely "Bert Base Uncased"), and use the hidden state of the 'CLS' token as the pairwise embedding for a pair of summarized query case and candidate case.

In this paper, we did not pre-train any further the BERT model with specific legal texts.

2.3 Relevance Assessment

We translate our pairwise relevance classification problem to the BERT fine-tuning for a downstream task of pairwise binary classification, by adding a Multi-Layer Perceptron that receives the pairwise embedding, and performs a binary classification.

We observe the distribution of the positive class over the dataset, we see that each query case has on average 4.8 noticed cases, and 90% of query cases have less than 10 noticed cases. The positive class is here the minority class, we balance the dataset by oversampling the positive class until a equal ratio is obtained for both classes.

We opted for a Mean Squared Error loss, and fine-tuned our model for 10 epochs. On our platform with one nVidia Tesla P40, with 24GB onboard RAM, the fine-tuning took 8 hours. For the binary classification task, we obtain an evaluation accuracy (on cases unseen during training) of 0.98, although we rely on the confusion matrix to assess the performance of the model, having in perspective that we will use the pairwise score for ranking purpose.

3 EXPERIMENTAL SETUP

The provided labeled dataset is split 75%/25% between training data and evaluation data. The dataset is split according to the cases, so that the cases in the evaluation dataset are not in the training dataset, so that it reflects properly the capacity of the system to generalize to unseen data. To be noted that, due to the setting of the task, both the case and the collection of candidates are unseen data[1].

After the fine-tuning step is finalized, the trained model computes the pairwise relevance score for each pair of query case and candidate case in the evaluation dataset, the score for the positive class is used to rank the candidate cases of each query case.

We proceed then to study the scores distribution, and evaluate optimum parameters for the selection of the candidate cases the system will return in response to the query case.

We observe the distribution of pairwise scores, with regards to the true class, in Figures 2 and 3. The distribution of the scores of the negative class is satisfactorily limited to the left, and we observe that the distribution of scores for the positive class has a more widespread mass. We expect this to have a negative impact on the recall of our system.

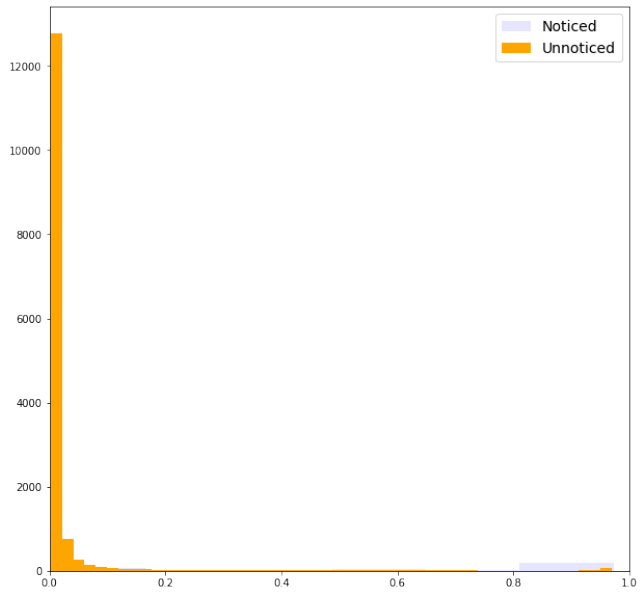

Figure 2: Distribution of Pairwise Relevance Score, for the Evaluation dataset. Negative class is highlighted

We consider two policies for building a list of returned candidate cases:

- Rank based: we will choose to return only the Top-N results, N being determined empirically from the observations made on the hold-out evaluation dataset
- Score based: we will choose to return all results with a score higher than a threshold value, determined empirically from the observations on the hold-out evaluation dataset

[1] although not all candidate cases are unique to only one query case, but we expect the statement to hold overall

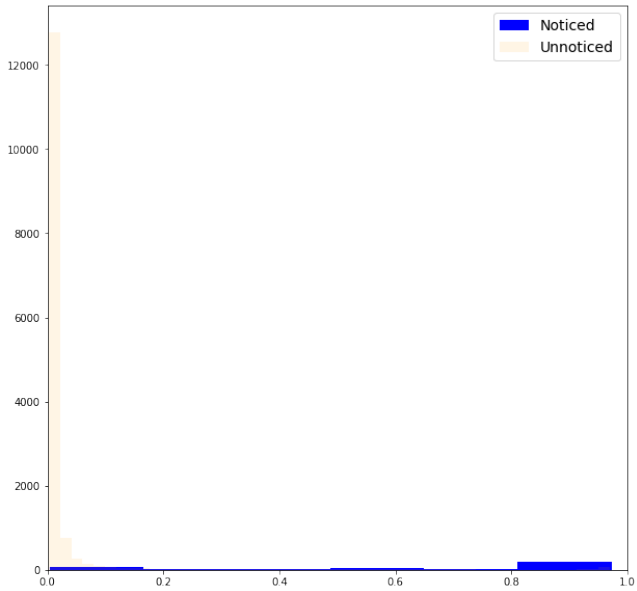

Figure 3: Distribution of Pairwise Relevance Score, for the Evaluation dataset. Positive class is highlighted

We opted for a Score-Based parameter for the BERT system, while our baseline is BM25 with a Rank-Based parameter. These parameters were used when predicting for the unseen and unlabeled test dataset. For BERT, we retained a value that was maximizing F1-score, and another value that was maximizing Precision.

4 RESULTS AND ANALYSIS

Our results are given in Table 1.

System		Evaluation Dataset			Test Dataset		
Name	Cut	P	R	F1	P	R	F1
BM25	6	0.47	0.55	0.51	0.47	0.52	0.49
BERT F1	0.946	0.72	0.47	0.57	0.68	0.43	0.53
BERT P	0.96	0.82	0.38	0.52	0.82	0.34	0.48

Table 1: Precision, Recall and F1-score.

The performance for the test set is the submitted result for the competition.

We observe that, for the selected metrics, the results on the unknown data (test dataset), based on parameters devised on a small portion of labeled data (evaluation dataset), are in line with the results on the evaluation dataset, which indicates a good capacity to generalize for our system.

As expected from the pairwise relevance score distribution, recall is negatively impacted by the widespread distribution of scores for the positive class, while precision suffers from the small proportion of pairs with a high score belonging to the negative class. The recall for both BERT systems is also lower than the BM25 baseline.

We consider the limitations of reducing a ranking problem to a pairwise relevance classification problem. The system learns to have a score on the right side of the decision threshold (0.5 in the binary classification setting), instead of learning that the score of any sample from the negative class should be lower than the score of any sample from the positive class. While this limitation is addressed by systems from the Learning to Rank family, this paper did not use the associated techniques.

We analyze further the submitted results, based on the disclosed golden labels. The unlabeled test dataset had, on average, 5.4 noticed cases per query case, which is slightly higher than 4.8 as observed for the labeled training dataset. We also plot the distribution of scores with regards to the True class in Figures 4 and 5, which we find almost identical to the distribution observed for the evaluation dataset. We conclude to the proper generalization of our system.

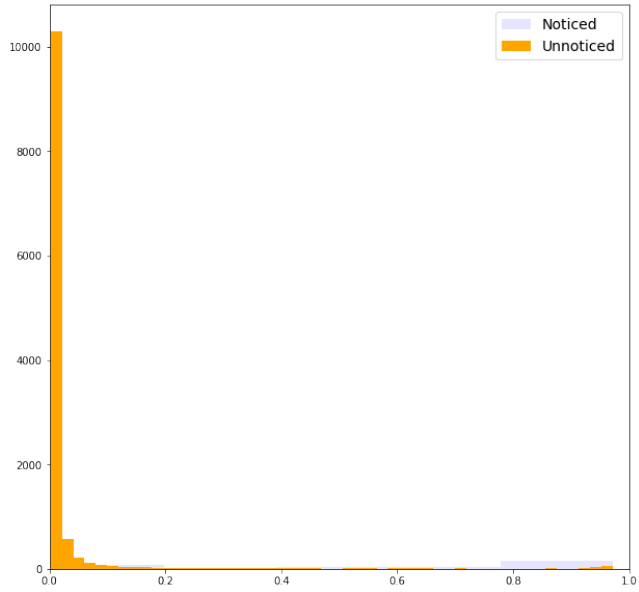

Figure 4: Distribution of Pairwise Relevance Score, for the Test dataset. Negative class is highlighted

5 CONCLUSION

We have demonstrated in this paper the ability of Generalized Language Models to perform properly, out of the box, in this particular setting of legal information retrieval. We did not establish a new State of the Art for this task, but we assume there are multiple opportunities for refinement and improvements.

We introduced the usage of summarization in order to keep the size of the input within the bounds imposed by BERT. We introduced the usage of BERT for the production of pairwise embeddings that are meaningful for a downstream task of relevance classification. Our policy for tuning the returned list of results have been introduced and has shown consistent results across two distinct datasets.

In our future work, we will certainly focus on refining and improving on the underlying BERT model, diversifying to other Generalized Language Models, as well as introducing advanced methods of ranking and training, such as Learning to Rank.

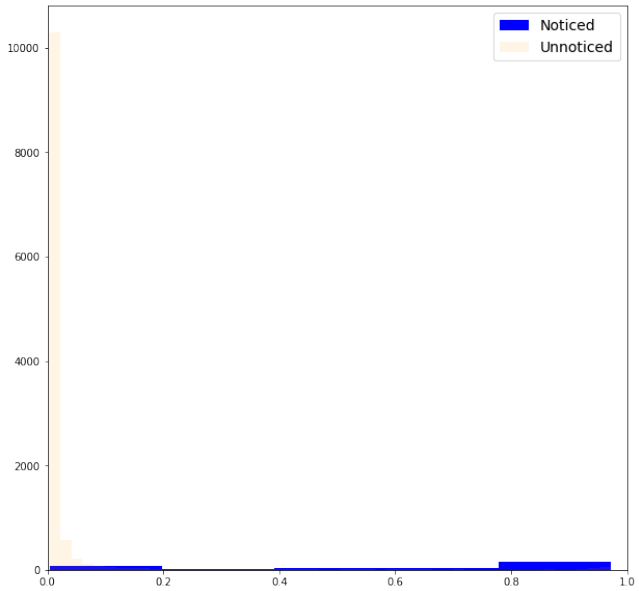

Figure 5: Distribution of Pairwise Relevance Score, for the Test dataset. Positive class is highlighted

REFERENCES

[1] Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. 2016. Variations of the Similarity Function of TextRank for Automated Summarization. *CoRR* abs/1602.03606 (2016). arXiv:1602.03606 http://arxiv.org/abs/1602.03606

[2] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

[3] 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

[4] 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

[5] 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

[6] Tibor Kiss and Jan Strunk. 2006. Unsupervised Multilingual Sentence Boundary Detection. *Comput. Linguist.* 32, 4 (Dec. 2006), 485–525. https://doi.org/10.1162/coli.2006.32.4.485

[7] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-Task Deep Neural Networks for Natural Language Understanding. *arXiv preprint arXiv:1901.11504* (2019).

[8] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in Translation: Contextualized Word Vectors. *CoRR* abs/1708.00107 (2017). arXiv:1708.00107 http://arxiv.org/abs/1708.00107

[9] 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

[10] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR* abs/1802.05365 (2018). arXiv:1802.05365 http://arxiv.org/abs/1802.05365

[11] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving Language Understanding by Generative Pre-Training. (2018). https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

[12] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019). https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[13] Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, Valletta, Malta, 45–50. http://is.muni.cz/publication/884893/en.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *CoRR* abs/1706.03762 (2017). arXiv:1706.03762 http://arxiv.org/abs/1706.03762

[15] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *CoRR* abs/1609.08144 (2016). arXiv:1609.08144 http://arxiv.org/abs/1609.08144