UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

MIT Big Data Science

COS/CSC 801 - Deep Learning

Johann Pienaar - u24050505

Neacail Hilhorst - u16174977

Victor Dippenaar - u14221838

October 28, 2022

Data Augmentation Approaches for Legal Document Analytics

DECLARATION OF ORIGINALITY

UNIVERSITY OF PRETORIA

The University of Pretoria places great emphasis upon integrity and ethical conduct in the preparation of all written work submitted for academic evaluation.

While academic staff teach you about referencing techniques and how to avoid plagiarism, you too have a responsibility in this regard. If you are at any stage uncertain as to what is required, you should speak to your lecturer before any written work is submitted.

You are guilty of plagiarism if you copy something from another author's work (e.g. a book, an article or a website) without acknowledging the source and pass it off as your own. In effect you are stealing something that belongs to someone else. This is not only the case when you copy work word-for-word (verbatim), but also when you submit someone else's work in a slightly altered form (paraphrase) or use a line of argument without acknowledging it. You are not allowed to use work previously produced by another student. You are also not allowed to let anybody copy your work with the intention of passing if off as his/her work.

Students who commit plagiarism will not be given any credit for plagiarised work. The matter may also be referred to the Disciplinary Committee (Students) for a ruling. Plagiarism is regarded as a serious contravention of the University's rules and can lead to expulsion from the University.

The declaration which follows must accompany all written work submitted while you are a student of the University of Pretoria. No written work will be accepted unless the declaration has been completed and attached.

- Victor Dippenaar (u14221838)
- Neacail Hilhorst (u16174977)
- Johann Pienaar (u24050505)

Declaration

1. I understand what plagiarism is and am aware of the University's policy in this regard.
2. I declare that this assignment report is my own original work. Where other people's work has been used (either from a printed source, Internet or any other source), this has been properly acknowledged and referenced in accordance with departmental requirements.
3. I have not used work previously produced by another student or any other person to hand in as my own.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or her own work.

SIGNATURE: Johann Pienaar DATE: 28 Oct 2022

SIGNATURE: ______________________ DATE: ____________

SIGNATURE: ______________________ DATE: ____________

Data Augmentation Approaches for Legal Document Analytics

Introduction and Problem Description

This report sets out to investigate the extraction of contract elements through the use of deep learning, as proposed by Chalkidis and Androutsopoulos (2017). The report will attempt to replicate the findings of the original paper and contrast the differences between the results obtained by the original authors with the results reported here.

The monitoring of contract elements entails a lot of tedious manual inspection and overhead on the part of law firms, contractors, law enforcement, and other stakeholders. The automated extraction of particular contract elements based on the specific use case has the potential to reduce the manual labour requirements which currently underpin this work. Examples which highlight the importance of extracting contract elements include:

- Legislative changes and imminent expiration of contracts are important elements that law firms communicate to their clients.
- Agreed payments and deliverables are kept track of by contractors.
- Law enforcement might flag contracts which pertain to certain parties, have a high contract value, or are defined by other features.

The authors previously considered contract element extraction using Logistic Regression and linear Support Vector Machines classifiers which were provided with hand-crafted features, word embeddings, and 25-dimensional part-of-speech tag embeddings. The authors further improved their results by implementing manual post-processing rules (Chalkidis, Androutsopoulos, and Michos 2017).

The authors found that the LSTM-based deep learning approaches investigated for contract element extraction, including a BILSTM-LR, BILSTM-LSTM-LR, and BILSTM-CRF architectures, performed better than the best-performing linear sliding-window machine learning classifiers. The deep learning approaches were provided with 200-dimensional word embeddings, 25-dimensional part-of-speech tag embeddings, and 5-dimensional token shape embeddings. These works hold additional benefits compared to the authors' previous works as well as other state-of-the-art contract element extraction methods which rely on hand-crafted features, patterns, known entity lists, or manual processing rules which are tailored based on the specific contracts.

The experimental results indicated that the addition of an LSTM or CRF layer on top of the BILSTM layer improved the classifier, especially when considering contract elements with few training examples. The BILSTM-LSTM-LR model misclassified fewer tokens, while the BILSTM-CRF performed better when the extraction of entire contract elements was evaluated.

This report sets out to replicate the experimental results reported in Chalkidis and Androutsopoulos (2017). The 5-dimensional token shape embedding data was not made publicly available. The authors were contacted in an attempt to obtain these embeddings, however, this proved to be unsuccessful. Consequently, a 5-dimensional feature vector was crafted using PCA from the original 14 dimensional hand-crafted feature vector described in (Chalkidis, Androutsopoulos, and Michos 2017). The code used to obtain the comparative results is made available in a private GitHub repository[1].

More recently, the first publicly available word embeddings trained on a large number of legal corpora were made available as Law2Vec. Published results have demonstrated improved model performance using domain-specific embeddings since the word context is better represented and less noise is introduced (Chalkidis and Kampas 2019).

The application of pre-trained large language models such as BERT has also received attention in the literature of late (Chalkidis, Fergadiotis, Malakasiotis, Aletras, et al. 2020). It was found that fine-tuning or pre-training BERT from scratch using domain-specific corpora improved model performance on legal text classification and sequence tagging tasks. Legal-BERT was released as a family of BERT models for the legal domain on HuggingFace[2].

Dataset Description

Chalkidis, Androutsopoulos, and Michos (2017) released the largest publicly available dataset for contract element extraction. The dataset contains roughly 3500 gold contract element annotated contracts and 750000 unlabelled contracts, all written in the English language. The contracts were encoded to ensure privacy regulations are adhered to. This analysis is concerned with the training and testing of contract element extraction methods through the use of labelled contracts. The labelled contracts are further divided into two subdirectories, namely elements contracts and clauses contracts. The clauses contracts were annotated for clause headings, whereas the elements contracts were annotated for contract title, contract parties, start date, effective date, termination date, contract period, contract value, governing law, jurisdiction, and legislation refs.

Both the elements and clause contracts have been split into a train and test set by the original authors. The contract encoding assigns each unique token a unique integer identifier. The unique integer identifiers are mapped to the relevant word and part-of-speech tag embeddings in the encoded vocabulary file. The word and part-of-speech tag embeddings were obtained by applying the skip-gram word2vec model to the unlabelled contracts and 50000 POS-tagged contracts in English. The token shape embeddings were not made publicly available, and direct communication with the original authors indicated that these embeddings would be difficult to come by.

The labelled contracts take the form of a series of token identifiers annotated with

[1] https://github.com/jpienaar-tuks/COS801

[2] https://huggingface.co/nlpaueb

their labelled relevant categories, for example, TOKEN_XXX[CATEGORY] where the token identifier can be mapped to the relevant word, POS tag, and token embedding. Linebreaks have been included as specific tokens with a zero value in their feature representation.

Approaches/Model/Methods/Algorithms description

The authors of Chalkidis and Androutsopoulos (2017) found that LSTM-based models showed promising performance when applied to named entity extraction (NER). They make use of three LSTM-based extraction methods namely BILSTM-LR, BILSTM-LSTM-LR and the BILSTM-CRF. For the purposes of our experiments, we will attempt to get similar results using the same methods with slight variations of on their configurations.

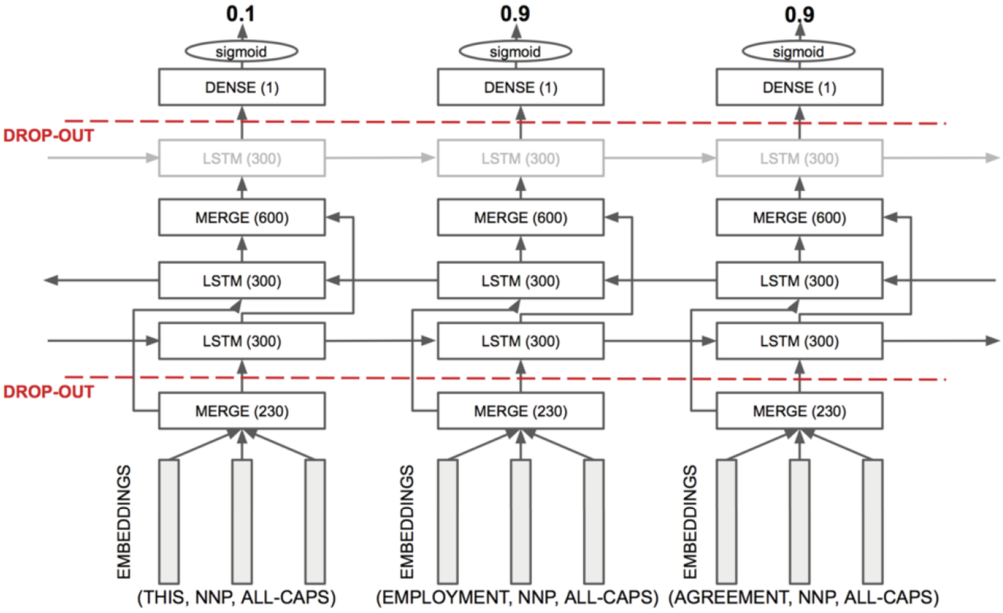

Figure 1: Graphical representation of an BILSTM-(LSTM)-LR (Chalkidis and Androutsopoulos 2017)

The first extraction method that we incorporated is the BILSTM-LR. This is the same as Figure 1, with two minor differences:

- In BILSTM-LR the top LSTM layer is not included.
- We used a sequence width of 11 as recommended by Chalkidis, Androutsopoulos, and Michos (2017) instead of three as shown in the illustration

The selection of hyperparameters was not automated for this experiment. Instead, we manually configured these parameters based on the work already done by Chalkidis and

Androutsopoulos (2017) and from our own experiments. The LSTM chains are configured to have 300-dimensional hidden states. Chalkidis and Androutsopoulos (2017) found that increasing the dimensionality negatively affected the speed at which the experiments were executed without producing noticeable improvements. We experimented with the dropout rate to see the effect. It was interesting to note the effect that the dropout rate had on the performance. We found that a dropout rate of 20% worked well. A future improvement could be to take an automated approach to see what hyper-parameters provide the best-performing solution.

The second extraction method used was the BILSTM-LSTM-LR. The main difference between the BLSTM-LR and the BILSTM-LSTM-LR is that the latter has an additional LSTM layer included in it. This can be seen in the top half of Figure 1. It has been noted that the inclusion of additional LSTM layers improves the ability to perform certain NLP-related tasks. This stacked LSTM layer is set to be unidirectional, as it is more computationally efficient than bidirectional LSTM layers (Wu et al. 2016). The hyper-parameters were configured to be the same as the BILSTM-LR method.

The final extraction method considered for this experiment is the BILSTM-CRF. This differs from the BILSTM-LSTM-LR model such that the upper LSTM chain and the linear regression layers get replaced(in Figure 1 this is represented by the upper LSTM chain, the dense boxes and sigmoid functions), by a conditional random field (CRF). CRFs have shown to work well when paired with LSTMs (Yao et al. 2014). They have improved contextual awareness that allows it to consider the predictions of the tokens before and after it. This is particularly useful for NER, as it allows it to know when a given entity stops and a new one begins. The BILSTM-CRF was configured with the same hyperparameters as the other two extraction methods.

The authors also made use of a 5-dimensional token shape embedding which represents the following possible token shapes:

- Uppercase alphabetic characters which could include hyphens or periods.
- Lowercase alphabetic characters which could include hyphens and periods.
- Uppercase alphabetic first character followed by lowercase alphabetic characters which could include hyphens or periods.
- Digits which could include commas or periods.
- Linebreaks.
- Any other token.

As mentioned previously, the token shape embeddings were not made publicly available. For this reason, we used PCA to generate a 5-dimensional token shape embedding based on the handcrafted features made available as part of Chalkidis, Androutsopoulos, and Michos (2017). This is due to the similarity of the original handcrafted features and the token shape embeddings. The PCA operation preserves 77% of the explained variance. To illustrate the similarity between the token shape embeddings and the handcrafted features, the original handcrafted features detailed the following:

- Three binary features for alphabetic tokens with all upper, all lower, and mixed case.
- Binary feature for alphanumeric tokens containing numbers.
- Seven binary features used to indicate the length of each token.
- Binary feature for numeric tokens.
- Binary feature for special character tokens.
- Binary feature for stopwords.

Chalkidis, Androutsopoulos, and Michos (2017) recommended extracting so-called pseudo-extraction zones starting 20 tokens before and ending 20 tokens after the target tokens for their logistic regression and support vector machines sliding window classifiers. This approach was adapted and used in our LSTM-based approaches.

While the goal of the project was to replicate the results of the original authors, there are a few things we did that were not included in the author's original approach. Firstly. For our project, we used the CRF layer implementation of tensorflow addons[3]. This is a change from the work of the original authors who used the older KERAS-CONTRIB code. We also experimented with using the mean for unknown word embeddings, this did not prove successful and was not included in the final approach. The token shape embeddings were created from a hand-crafted feature vector. This was out of necessity as they were not included in the original data set, nor could the authors provide us with them. While the hyper parameters were set manual we did experiment with dropout, learning rate, and batch size values, but we did end up following the same approach set in the paper. This provided a more accurate benchmark, but there is room for improvement in this area. Finally, we did not consider the clause headings, since our focus was on the contract elements directory.

Experimental Description of the Results

The performance of the classifiers is evaluated using the precision, recall and F1 score metrics. The authors performed two groups of experiments evaluating the classification per token as well as per entire contract element. The per token classification results reported by Chalkidis and Androutsopoulos (2017) are illustrated in Table 1. The macro averaged values have been updated to exclude clause headings, such that the results can be compared directly with our results.

Initially, the model was replicated using a dropout percentage of 10% and a batch size of 32 across all contract elements. The models were provided with word embeddings, part-of-speech tag embeddings, and generated pseudo token shape embeddings as input features. The models were executed on an AMD Ryzen 5 5600X with 32 GB RAM and an Nvidia GeForce 1660Ti GPU. The comparative model results are summarized in Table 2. The values indicate the difference between the results obtained as part of this report and the results reported by the original authors. As such negative values

[3] https://www.tensorflow.org/addons/api_docs/python/tfa/layers/CRF

Table 1: Precision(P), Recall (R) and F1 Scores Measured Per Token Reported By Chalkidis and Androutsopoulos (2017)

ELEMENT	BILSTM-LR			BILSTM-LSTM-LR			BILSTM-CRF		
TYPE	P	R	F1	P	R	F1	P	R	F1
Title	0.95	0.93	0.94	0.94	0.95	0.95	0.96	0.95	0.95
Parties	0.97	0.92	0.95	0.97	0.94	0.95	0.98	0.92	0.95
Start	0.91	0.97	0.94	0.93	0.97	0.95	0.92	0.98	0.95
Effective	0.98	0.92	0.95	0.97	0.96	0.97	0.95	0.89	0.92
Termination	0.65	0.90	0.75	0.70	0.92	0.79	0.65	0.93	0.77
Period	0.44	0.82	0.57	0.47	0.86	0.59	0.55	0.85	0.65
Value	0.74	0.55	0.63	0.74	0.63	0.68	0.72	0.60	0.66
Gov.	0.98	0.98	0.98	0.99	0.98	0.98	0.99	0.97	0.98
Jurisdiction	0.90	0.89	0.89	0.90	0.88	0.89	0.90	0.88	0.88
Legisl.	0.82	0.95	0.88	0.83	0.94	0.88	0.82	0.94	0.87
Macro-average	0.83	0.88	0.85	0.84	0.90	0.86	0.84	0.89	0.86

indicate an improvement in a given metric over the results reported by the original authors. Inversely a positive value indicates a decrease in a given metric compared to the results reported by the original authors. It is clear that the performance of our model was inferior given the approach and hyperparameters. The dropout was subsequently altered, and it was found that results improved at a value of 10%. This can be verified from Table 4 in the Appendix.

Table 2: Comparative Model Performance Results: Initial

ELEMENT	BILSTM-LR			BILSTM-LSTM-LR			BILSTM-CRF		
TYPE	P	R	F1	P	R	F1	P	R	F1
Title	9.2%	-3.1%	3.4%	10.6%	-2.2%	5.2%	11.3%	-0.9%	5.0%
Parties	4.6%	-0.3%	2.6%	5.8%	-0.6%	2.2%	17.2%	0.8%	9.3%
Start	19.9%	-1.8%	11.3%	26.2%	-2.3%	15.2%	26.9%	-0.9%	16.5%
Effective	79.5%	-3.0%	64.1%	76.9%	2.0%	63.9%	76.7%	-6.4%	61.4%
Termination	32.6%	-7.4%	26.3%	34.0%	-4.1%	26.6%	65.0%	93.0%	77.0%
Period	14.4%	-3.6%	13.0%	10.9%	0.5%	8.2%	19.7%	8.3%	16.7%
Value	10.8%	-15.9%	-3.8%	11.7%	-9.9%	0.8%	72.0%	60.0%	66.0%
Gov.	17.5%	-0.1%	9.5%	14.3%	-0.3%	7.0%	37.2%	-1.4%	22.1%
Jurisdiction	15.5%	-0.5%	7.7%	20.2%	-5.2%	9.2%	90.0%	88.0%	88.0%
Legisl.	65.6%	-0.8%	60.1%	65.9%	-1.2%	59.1%	81.7%	94.0%	87.0%
Macro Avg	27.0%	-3.7%	19.4%	27.7%	-2.3%	19.7%	49.8%	33.5%	44.9%

The effect of using pseudo-extraction zones was investigated, as described in the method section. The testing dataset provides an XML-like markup for so-called "golden extraction zones". Using these zones during testing it was found that the class balance in these golden extraction zones were markedly different at 3.7%±3.5% and that this had a substantial impact on the models' performance. Models were found to have high precision scores, but very low recall and F1 scores, indicating a high rate of false negatives. It was therefore decided to use the same pseudo-zone extraction technique on the testing data as was used on the training data (instead of the golden extraction zones). This resulted in a class balance of 16%±4.7%, which was much more similar to the testing data and significantly improved the models' performance. The marked effect on the model performance is summarised in Table 5 in the Appendix. The macro averaged F1 score across all three LSTM-based models were found to improve, however it should be noted that the performance of the BILSTM-CRF model was still inferior to the original paper at this point.

To ensure that the model wasn't learning localised patterns in batches during back-propagation, we also investigated the effect of randomising the dataset rows. The effects of this was negligible and can be seen in the appendix in Table 6.

The authors mention that the batch size hyperparameters were determined through cross-validation, but do not mention the final values of these parameters. Previous results made use of a batch size of 32. In attempting to improve the final BILSTM-CRF model, the batch sizes were altered and improved performance was found with a batch size of 64 for the start date, effective date, termination date, and contract value, as well as a value of 128 for the legislative references. This improved the performance of the BILSTM-CRF model, such that all of the LSTM-based models exhibited improved performance compared to the results reported by the original authors. This is likely due to the use of the pseudo-extraction zones and improved library performance. The results are illustrated by Table 3. It is interesting to note that the most significant contract element extraction improvements were to the termination date, period and contract value. The original authors hypothesized that these elements recorded relatively poorer results due to the smaller amount of training data.

Table 3: Comparative Model Performance Results: Final

ELEMENT	BILSTM-LR			BILSTM-LSTM-LR			BILSTM-CRF		
TYPE	P	R	F1	P	R	F1	P	R	F1
Title	-1.6%	-2.1%	-1.9%	-1.8%	-0.9%	-0.8%	1.8%	0.3%	0.5%
Parties	0.7%	-0.9%	0.4%	1.4%	-0.4%	0.0%	3.1%	6.1%	4.8%
Start	-5.1%	-2.4%	-3.7%	-3.4%	-1.8%	-2.6%	-4.6%	-0.8%	-2.7%
Effective	1.1%	2.8%	2.1%	-0.3%	6.5%	3.8%	3.8%	0.3%	2.1%
Termination	-27.9%	-8.6%	-20.7%	-23.3%	-4.6%	-15.9%	-18.8%	-5.0%	-13.4%
Period	-47.4%	-4.8%	-32.0%	-45.4%	-0.5%	-30.4%	-37.1%	-0.6%	-23.7%
Value	-16.6%	-16.2%	-16.8%	-17.8%	-5.8%	-10.7%	-18.4%	-6.6%	-10.7%
Gov.	-0.7%	-0.2%	-0.5%	0.3%	-0.6%	-0.7%	2.4%	11.7%	7.4%
Jurisdiction	-5.5%	0.9%	-2.6%	-5.8%	-1.0%	-3.3%	-0.2%	-3.0%	-2.6%
Legisl.	-15.3%	0.8%	-7.7%	-13.1%	-3.1%	-8.6%	-12.0%	10.8%	-1.3%
Macro Avg	-11.8%	-3.1%	-8.3%	-10.9%	-1.2%	-6.9%	-8.0%	1.3%	-4.0%

Additional tests were performed to evaluate the importance of the token shape and POS tag embeddings, as investigated by a subsequent paper released by the authors (Chalkidis, Fergadiotis, Malakasiotis, and Androutsopoulos 2021). The effects of excluding the token shape embeddings and POS tag embeddings are provided in Tables 7 and 8 in the Appendix respectively. It is evident that the POS tag embeddings do not have a significant impact on the performance of the model and tend to slightly worsen the macro averaged model F1 score across all three LSTM-based models. Similarly, the token shape embeddings derived from the hand-crafted features do not significantly impact the performance of the models. It is interesting to note that the use of the token shape embeddings improves the BILSTM-CRF model.

Conclusion/Discussion/Future work

This report describes the replication of the work originally put forth by Chalkidis and Androutsopoulos (2017). Three LSTM-based models were developed, these include an BILSTM-LR, BILSTM-LSTM-LR, and BILSTM-CRF model. The models were trained on encoded gold contract annotated contracts. The token shape embeddings used by the original authors were not made publicly available. For this reason, 5-dimensional pseudo token shape embeddings were generated from the hand-crafted features utilised by the logistic regression and support vector machines sliding window classifiers (Chalkidis, Androutsopoulos, and Michos 2017). Our results showed improved model performance across all three LSTM-based models. We found improved F1 scores for the BILSTM-LR, BILSTM-LSTM-LR, and BILSTM-CRF models improved by 8.3%, 6.9%, and 4.0% respectively.

The improved model performance is likely due to the use of the pseudo-extraction zones which considers 20 tokens before and after each contract element. The performance of the pseudo token embeddings obtained from the hand crafted features could not be directly compared to the original token shape embeddings. It was however confirmed that the use of the POS tag embeddings hampers the model performance. Additionally, it was found that the token shape embeddings worsened the performance of the BILSTM-LR and BILSTM-LSTM-LR models, however this improved the BILSTM-CRF model. These observations are in line with the work of Chalkidis, Fergadiotis, Malakasiotis, and Androutsopoulos (2021).

Given these findings, it would be recommended to train the BILSTM-LR and BILSTM-LSTM-LR models solely on the word embeddings, whereas it is recommended to train the BILSTM-CRF model on both word embeddings and token shape embeddings with a batch size of 64 for the start date, effective date, termination date, and contract value, and a batch size of 128 for the legislative references.

Future work in the field may include enriching the label set by using a single classifier across contract elements. Since the LSTM-based models recorded the worst performance for contract elements with the fewest training examples, data augmentation techniques might also be explored. Finally, NLP techniques for legal contract element extraction could be extended to non-English languages. This might be achieved by fine-tuning a multi-lingual pre-trained large language model on domain-specific legal corpora, similar to what has been demonstrated by Legal-BERT - however with a focus on non-English languages.

Bibliography

Yao, Kaisheng et al. (2014). “Recurrent conditional random field for language understanding”. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 4077–4081.

Wu, Yonghui et al. (2016). “Google’s neural machine translation system: Bridging the gap between human and machine translation”. In: *arXiv preprint arXiv:1609.08144*.

Chalkidis, Ilias and Ion Androutsopoulos (2017). “A Deep Learning Approach to Contract Element Extraction.” In: *JURIX*, pp. 155–164.

Chalkidis, Ilias, Ion Androutsopoulos, and Achilleas Michos (2017). “Extracting contract elements”. In: *Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law*, pp. 19–28.

Chalkidis, Ilias and Dimitrios Kampas (2019). “Deep learning in law: early adaptation and legal word embeddings trained on large corpora”. In: *Artificial Intelligence and Law* 27.2, pp. 171–198.

Chalkidis, Ilias, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, et al. (2020). “LEGAL-BERT: The muppets straight out of law school”. In: *arXiv preprint arXiv:2010.02559*.

Chalkidis, Ilias, Manos Fergadiotis, Prodromos Malakasiotis, and Ion Androutsopoulos (2021). “Neural Contract Element Extraction Revisited: Letters from Sesame Street”. In: *arXiv preprint arXiv:2101.04355*.

Appendices

Appendix A: Additional Results

Table 4: Model Performance Results: Dropout Variation

ELEMENT	BILSTM-LR			BILSTM-LSTM-LR			BILSTM-CRF		
TYPE	P	R	F1	P	R	F1	P	R	F1
Title	11.3%	-4.3%	4.0%	8.3%	-1.4%	4.2%	15.4%	-0.1%	7.8%
Parties	5.0%	0.5%	3.2%	3.7%	0.5%	1.6%	8.5%	5.5%	7.0%
Start	22.9%	-2.5%	13.2%	25.6%	-2.1%	14.8%	11.9%	29.4%	21.1%
Effective	79.1%	-4.6%	63.5%	79.4%	-0.2%	67.3%	76.9%	-7.3%	61.6%
Termination	23.3%	-7.5%	16.6%	29.9%	-5.9%	22.1%	36.8%	-5.4%	33.2%
Period	11.3%	-3.1%	9.8%	18.7%	0.0%	16.4%	29.6%	4.5%	26.4%
Value	9.2%	-10.5%	-2.2%	11.1%	-3.5%	3.4%	11.9%	1.2%	6.6%
Gov.	18.7%	-0.5%	10.1%	21.1%	-0.8%	10.9%	21.6%	-0.8%	11.6%
Jurisdiction	13.9%	2.1%	7.9%	17.4%	-5.0%	7.4%	18.6%	6.9%	12.0%
Legisl.	68.0%	-1.8%	63.5%	68.4%	-2.8%	62.6%	55.0%	1.0%	45.2%
Macro Avg	26.3%	-3.2%	19.0%	28.4%	-2.1%	21.1%	28.6%	3.5%	23.3%

Table 5: Model Performance Results: Pseudo-Zone Extractor

ELEMENT	BILSTM-LR			BILSTM-LSTM-LR			BILSTM-CRF		
TYPE	P	R	F1	P	R	F1	P	R	F1
Title	1.0%	-3.9%	-1.4%	-1.9%	-0.8%	-0.9%	8.3%	-2.3%	2.8%
Parties	1.3%	-1.5%	0.4%	1.4%	-0.3%	0.1%	5.5%	0.7%	3.1%
Start	-5.1%	-2.3%	-3.7%	-3.1%	-2.1%	-2.6%	-4.0%	-0.4%	-2.2%
Effective	1.6%	1.7%	1.8%	1.5%	3.3%	2.9%	95.0%	89.0%	92.0%
Termination	-30.3%	-7.4%	-21.4%	-26.0%	-4.2%	-17.1%	-25.8%	-5.1%	-17.3%
Period	-47.8%	-7.1%	-33.4%	-42.9%	-1.1%	-29.5%	-34.5%	15.5%	-13.2%
Value	-14.6%	-19.4%	-17.9%	-14.6%	-10.9%	-12.6%	-19.3%	2.2%	-4.8%
Gov.	-0.8%	-0.5%	-0.6%	0.0%	0.0%	-0.5%	2.6%	0.9%	1.8%
Jurisdiction	-6.4%	2.4%	-2.2%	-5.6%	-1.7%	-3.5%	-1.8%	12.6%	5.2%
Legisl.	-14.1%	-2.0%	-8.6%	-14.7%	-2.5%	-9.1%	-18.0%	94.0%	87.0%
Macro Avg	-11.5%	-4.0%	-8.7%	-10.6%	-2.0%	-7.3%	0.8%	20.7%	15.4%

Table 6: Model Performance Results: Shuffling Data

ELEMENT	BILSTM-LR			BILSTM-LSTM-LR			BILSTM-CRF		
TYPE	P	R	F1	P	R	F1	P	R	F1
Title	-0.3%	-3.4%	-1.8%	-2.9%	-0.3%	-1.1%	12.4%	0.7%	6.4%
Parties	0.8%	-0.5%	0.7%	1.1%	-0.3%	-0.1%	8.4%	-2.1%	3.2%
Start	-5.0%	-2.3%	-3.6%	-3.2%	-2.2%	-2.7%	92.0%	98.0%	95.0%
Effective	0.4%	3.6%	2.2%	-0.4%	5.3%	3.1%	95.0%	89.0%	92.0%
Termination	-28.3%	-7.4%	-20.3%	-22.8%	-5.8%	-16.2%	65.0%	93.0%	77.0%
Period	-47.3%	-2.2%	-30.6%	-46.8%	-0.2%	-30.9%	-12.2%	32.4%	6.0%
Value	-15.8%	-18.2%	-17.6%	-16.3%	-6.2%	-10.3%	72.0%	60.0%	66.0%
Gov.	-0.4%	-0.5%	-0.5%	0.1%	-0.7%	-0.8%	3.5%	-1.6%	1.0%
Jurisdiction	-5.1%	0.3%	-2.8%	-5.0%	-2.3%	-3.6%	2.2%	20.0%	11.4%
Legisl.	-15.5%	-1.0%	-8.8%	-13.4%	-3.6%	-9.0%	-14.5%	-0.8%	-8.6%
Macro Avg	-11.7%	-3.2%	-8.3%	-10.9%	-1.7%	-7.2%	32.4%	38.9%	34.9%

Table 7: Model Performance Results: Disregard Token Shape Embeddings

ELEMENT	BILSTM-LR			BILSTM-LSTM-LR			BILSTM-CRF		
TYPE	P	R	F1	P	R	F1	P	R	F1
Title	-0.7%	-3.5%	-2.1%	-1.2%	-0.9%	-0.6%	2.3%	9.7%	5.7%
Parties	2.0%	-2.9%	0.1%	1.5%	-0.5%	0.0%	3.8%	-0.5%	1.7%
Start	-4.9%	-2.0%	-3.4%	-3.1%	-2.2%	-2.6%	-4.5%	-0.4%	-2.4%
Effective	1.1%	0.0%	0.6%	0.9%	4.0%	3.0%	2.3%	1.2%	1.8%
Termination	-28.2%	-7.8%	-20.4%	-24.8%	-5.8%	-17.2%	-20.6%	-2.9%	-13.5%
Period	-49.9%	-5.2%	-33.4%	-46.3%	0.3%	-30.3%	-33.1%	-0.8%	-22.0%
Value	-16.1%	-16.9%	-17.0%	-14.9%	-12.5%	-13.7%	-13.0%	-5.0%	-7.7%
Gov.	-0.2%	-0.8%	-0.5%	0.5%	-0.7%	-0.6%	3.6%	-0.7%	1.4%
Jurisdiction	-6.1%	2.5%	-2.0%	-5.2%	-2.0%	-3.6%	-1.7%	1.0%	-1.3%
Legisl.	-14.6%	-1.3%	-8.5%	-13.0%	-3.5%	-8.7%	-11.8%	1.2%	-6.3%
Macro Avg	-11.8%	-3.8%	-8.7%	-10.6%	-2.4%	-7.4%	-7.3%	0.3%	-4.3%

Table 8: Model Performance Results: Disregard POS Tag Embeddings

ELEMENT	BILSTM-LR			BILSTM-LSTM-LR			BILSTM-CRF		
TYPE	P	R	F1	P	R	F1	P	R	F1
Title	0.0%	-3.3%	-1.6%	-0.4%	-1.6%	-0.5%	0.7%	4.0%	1.9%
Parties	2.8%	-3.0%	0.4%	1.8%	-0.1%	0.4%	2.9%	5.5%	4.4%
Start	-4.8%	-2.4%	-3.6%	-3.0%	-2.1%	-2.5%	-3.2%	-0.7%	-2.0%
Effective	2.5%	-1.8%	0.4%	1.0%	3.6%	2.8%	2.9%	-0.4%	1.3%
Termination	-26.6%	-7.9%	-19.7%	-24.9%	-5.9%	-17.4%	-23.7%	23.5%	-0.9%
Period	-44.7%	-6.7%	-31.7%	-40.9%	-4.0%	-29.9%	-28.1%	-6.2%	-21.9%
Value	-15.8%	-15.8%	-16.2%	-15.1%	-15.8%	-15.6%	-18.0%	-4.1%	-8.9%
Gov.	0.0%	-0.9%	-0.4%	0.4%	-0.2%	-0.4%	1.1%	0.5%	0.8%
Jurisdiction	-5.5%	0.8%	-2.7%	-5.4%	-1.8%	-3.5%	-5.4%	17.2%	6.7%
Legisl.	-14.7%	-1.8%	-8.7%	-14.0%	-3.1%	-9.1%	-6.7%	-0.1%	-4.4%
Macro Avg	-10.7%	-4.3%	-8.4%	-10.1%	-3.1%	-7.6%	-7.8%	3.9%	-2.3%