Legal Knowledge and Information Systems
M. Araszkiewicz and V. Rodríguez-Doncel (Eds.)

doi:10.3233/FAIA190328

Application of Character-Level Language Models in the Domain of Polish Statutory Law

Aleksander SMYWIŃSKI-POHL [a], Krzysztof WRÓBEL [b], Karol LASOCKI [a] and Michał JUNGIEWICZ [a]

[a] *AGH University of Science and Technology, Krakow, Poland*
[b] *Jagiellonian University, Krakow, Poland*

Abstract. Polish statutory law so far is distributed as PDF, HTML and text files, where the structure of the rules and the references to internal and external regulations is provided only implicitly. As a result, automatic processing of the regulations in legal information systems is complicated since the semi-structured text needs to be converted to a structured form. In this research, we show how character-level language models help in this task. We apply them to the problems of detecting the cross-references to structural units (e.g. articles, points, etc.) and detecting the cross-references to statutory laws (titles of laws and ordinances). We obtain 98.7% macro-average F1 in the first problem and 95.8% F1 in the second problem.

Keywords. character-level language models, cross-reference recognition, language modelling, legal text processing, Polish law

1. Introduction

The Polish statutory law is available for everyone in the system called the Internet System of Legislative Acts (Internetowy System Aktów Prawnych – ISAP[1]). The system contains all the bills, ordinances, rulings of the Constitutional Tribunal and international agreements adopted from 1918, the year Poland has regained its independence, until now. All acts in the system are distributed as PDF files. Some of the metadata of the acts are provided directly on the web page dedicated to the individual document, but the actual content of the document is structured only visually.

Our goal is automatic structuring of the body of the Polish legislative acts. Much of this structure may be processed with regular expressions (especially the structural units of acts, since their patterns are very rigid). In this research, we concentrate on two issues that are harder to tackle with simple, rule-based techniques: detection of cross-references to structural units of the acts, such as articles, paragraphs, and points and detection of cross-references to titles of legislative acts. We apply the same algorithm to both problems, namely we use a recurrent neural network and a character-level language model (cLM).

[1] `http://isap.sejm.gov.pl`

2. Related Work

To our knowledge, there are very few works on detecting legal cross-references using machine learning methods. Almost all of the systems described in literature use a rule-based approach.

One approach to the problem based on machine learning methods is described in [1] and was tested on Japanese texts. The novelty of this work is two-fold. It lies in using machine learning for legal references resolution. Secondly, the authors claim their innovation is resolving references not only to document targets but also to sub-document parts. Their system achieves 80.06% in the F1 score for detecting references, 85.61% accuracy for resolving them, and 67.02% in the F1 score for end-to-end setting task on the Japanese National Pension Law corpus.

A more classical approach for detecting cross-references in legal texts is [2]. The authors used certain NLP patterns to build a rule-based system. These patterns were developed on Luxembourg's legislation, written in French. The system was tested on the Personal Health Information Protection Act (PHIPA) by the Government of Ontario, Canada, written in both French and English and on several Luxembourgish legislative texts.

As a rule-based baseline system for detecting references in Polish legal texts we used one developed as a part of the SAOS (Court Judgement Analysis System) project [3]. The general schema of the algorithm is to first tokenize the text, and then extract certain ranges of the tokens as candidates for references. After that, based on some rules and regular expressions, it looks for fragments of text that contain legal references and splits them into classes.

On the other hand, the application of language-model based algorithms for various NLP tasks seems to be a standard approach at least in the last 2 years. In the past, in tasks such as text classification or named entity recognition, *words* were treated as the main units of processing [4]. The vector space model (VSM) was one of the formalisms best suited for providing a coherent representation of words for ML algorithms. They used to be represented as one-hot encoded vectors, where the size of the vector equals the size of the vocabulary.

A relatively recent solution to the problem of limited vocabulary is word embeddings (WE) – dense vector representations of words. These embeddings are obtained in an unsupervised manner, thus they are easily adaptable to new languages and problems. The most successful methods are based on neural networks and factorization of co-occurrence matrices. Popular systems, such as word2vec [5], GloVe [6] and fastText produce [7] so-called static WE, since the representation is independent of the word context. As such it limits the expressiveness of the models since the vectors are unable to capture polysemy. The „traditional" word embeddings also face a problem of the composition of multiple words into one vector – the vectors might be linearly combined (e.g. averaged) or units for multi-word entities have to be defined separately.

Contextual word embeddings are the latest representation able to solve that problem. The most recent systems: ELMo [8], BERT [9] and Flair [10] encode not only the word in question but also its surroundings. Moreover, Flair and ELMo do not employ tokenization, since they use character-based or byte-pair-encoding (BPE) based embeddings. This allows for computing dense representation for unrestricted spans of text.

The most recent studies show [11] that such models can solve a large number of problems: language modeling, named entity recognition, machine translation, text gen-

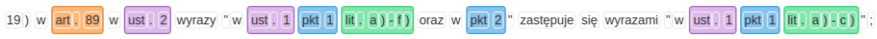

Figure 1. An example of references to structural units appearing in the Polish statutory law.

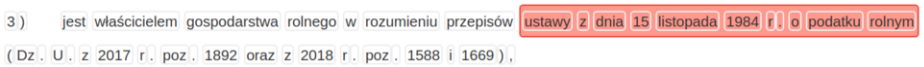

Figure 2. An example of a reference to a legislative act appearing in Polish statutory law.

eration, text summarization, natural language inference, and question answering – with very little or even no manually annotated data for the downstream task. Yet we haven't found any paper that uses contextual WE for the problems we tackle.

3. Problem Description

3.1. Cross-References to Structural Units

The problem of cross-reference to structural units of statutory law is depicted in Figure 1. The example comes from an amending act, which are typically packed with all types of references. We call these references *cross-references to structural unites*, since they point to particular, structural fragments of laws, such as chapters, articles, paragraphs, points, letters, indents as well as particular sentences.

The cross-references to structural units in the Polish statutory law can be roughly divided into two groups: those that are used in the amending bills, where the sequence of units almost always starts with an article[2], which is further placed within a particular law, and those that are more common in non-amending bills, when the top-level element may be any valid unit. In the second case, the higher-order units are indicated implicitly as the units the reference appears in.

We define the problem of detecting cross-references to structural units as the detection of the exact span of the reference and as a qualification of the span as one of the following (13) types: *article*, *point*, *paragraph*, *letter*, *indent*, *chapter*, *division*, *branch*, *title*[3], *book*, *part*, *subchapter*, and *sentence*. However, since the rule-based tool devised to detect the cross-references in the Polish law detects only 3 types of references: *article*, *paragraph*, and *point*, to make the comparison fair we only provide the results for these three categories.

3.2. Cross-References to Statutory Laws

The problem of *cross-references to legislative acts* is depicted in Figure 2. Usually, the title of an act starts with *ustawa* (bill) or *rozporządzenie* (ordinance), followed by date of publication and ends with a detailed location of the act, allowing for its unambigu-

[2]Rare cases of amendments include a chapter which is completely removed or added and an amendment in the title of the law.

[3]No to be confused with the title of the law.

ous identification. Although the priming word is always present, date might be omitted and title does not have to be followed by the location details. These two features make detection of act titles a challenging problem. We define the problem of detecting cross-references to legislative acts as the detection of the exact span of the title.

4. Applied Algorithms

4.1. Character-Level Language Model

We use Flair toolkit [10] to train the cLM and compute the contextual embeddings. The cLM allows for obtaining embeddings of any fragment of text. To achieve the best results two language models are trained: forward and backward.

The training of cLM starts with preparation of the corpus, definition of the character dictionary and determination of the training parameters. The loss function is cross-entropy, which translates to perplexity (exponent of cross-entropy).

One of the most important training parameters is the size of the internal state of RNN. The authors of Flair use 1024 or 2048 [10], resulting in 2048 or 4096 components in the final embedding. This is a large number in comparison to popular static WEs that range from 100 to 300 in size. On the other hand, the dictionary is much smaller since it only includes a limited subset of Unicode characters. The default learning rate is set to 20, which decreases with the training process. In our experiments, when training the cLMs the size of the internal state of the RNN was set to 2048.

4.2. Cross-References Detection as NER

Flair also includes a module which performs Named Entity Recognition (NER). The text is split into tokens and the contextual embeddings of the tokens are computed based on the cLM one character after the token (for the forward model) and one character before the token (for the backward model). The vectors are concatenated and they are used as the representation of the token in a biLSTM network. There is a Conditional Random Field (CRF) layer at the top, which performs the final assignment of the tags to the tokens. This model was used directly in both experiments, since the detection of both types of cross-references may be treated as a NER-like problem.

5. Data

The features of the corpus used to train the cLM are given in the second column in Table 1. The number of tokens is not very large, compared to typical corpora used to train language models, yet thanks to its domain specificity, we have achieved good perplexity (92,4) training for 3 days on one node with two K40 GPUs. To prepare the documents for the problems, we have collected approximately 10 acts from each year, starting in 1994 and ending in 2018, resulting in 243 documents.

The annotationwas performed by 5 annotators with good knowledge of law (at least 5 years of studies in law) or linguistics (a master degree was required). We used the Inforex system [12] and followed a scheme where each document was annotated by two annotators and then a super-annotator resolved the conflicts. In fact the number of

Table 1. The statistics of the corpus used to train the language model (Acts) and the annotated sub-corpus.

Measure	Acts	Annotated
Number of tokens	9 776 676	396 963
Number of distinct lemmas	36 716	14 737
Number of sentences	371 082	14 737
Number of documents	1 892	243
Size in MBs	56	3.6
Average sentence length	26.3	26.9

Table 2. The F_1 score for the detection of the cross-references to structural units.

System	art	pkt	ust	micro	macro
rule-based	0.9454	0.9360	0.9364	0.9401	0.9393
cLM-based	**0.9797**	**0.9942**	**0.9874**	**0.9850**	**0.9871**

Table 3. The precision, recall and F_1 score for the detection of the cross-references to titles of legislative acts.

System	Precision	Recall	F_1
rule-based	**1.000**	0.6316	0.7742
cLM-based	0.9579	**0.9579**	**0.9579**

differences in annotations was very small and usually these were omitted or superfluous punctuation marks. The annotation of the data (the first round with two annotators and the second round with the super-annotator) took approximately 120 man-hours.

6. Experiments

We have split the annotated data (on the document level) into sub-corpora used for training of the model (*Train*), for tuning of the hyper-parameters (*Dev*) and for testing the model (*Test*) in ratio 70%/15%/15%. We have compared the performance of our model with SAOS extractors designed to perform the same task but in the domain of court rulings.

Table 2 contains the results for detection of cross-references to structural units. Our system achieves better results for all classes than the rule-based system. For articles, the performance is almost perfect. Table 3 contains the results for detection of cross-references to the titles of the legislative acts. The rule-based system has perfect precision, but its recall reaches only 63%. Our system is not completely precise (though 96% is a very decent result), but its recall is significantly higher (also 96%), thus the F_1 score is much better. Comparing to the first problem, it is apparent that the detection of titles is more challenging, but the system works very well.

7. Conclusions and Future Work

We have presented the results of the two experiments where we applied a cLM to the problems related to the processing of statutory law. The results of the experiments with the detection of cross-references obtained using that model are better than the results of a rule-based system. In all cases, the F_1 scores were above 95% showing that the models may be used practically.

In our future work, we will apply similar models to automatic detection and structuring of the amending acts, as well as to the detection of relations between cross-references to structural units.

Acknowledgments. This work was supported by the Polish National Centre for Research and Development – LIDER Program under Grant LIDER/27/0164/L-8/16/NCBR/2017 titled "Lemkin – intelligent legal information system". This research was also supported in part by PLGrid Infrastructure.

References

[1] O.T. Tran, N.X. Bach, M.L. Nguyen and A. Shimazu, Automated reference resolution in legal texts, *Artificial Intelligence and Law* **22** (2013), 29–60.

[2] N. Sannier, M. Adedjouma, M. Sabetzadeh and L. Briand, An automated framework for detection and resolution of cross references in legal texts, *Requirements Engineering* **22**(2) (2017), 215–237. doi:10.1007/s00766-015-0241-3.

[3] SAOS text mining extractor, GitHub, 2015.

[4] D. Jurafsky and J.H. Martin, *Speech and language processing*, 2009.

[5] T. Mikolov, K. Chen, G. Corrado and J. Dean, Efficient estimation of word representations in vector space, *arXiv preprint arXiv:1301.3781* (2013).

[6] J. Pennington, R. Socher and C.D. Manning, GloVe: Global Vectors for Word Representation, in: *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. `http://www.aclweb.org/anthology/D14-1162`.

[7] P. Bojanowski, E. Grave, A. Joulin and T. Mikolov, Enriching Word Vectors with Subword Information, *Transactions of the Association for Computational Linguistics* **5** (2017), 135–146.

[8] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee and L. Zettlemoyer, Deep contextualized word representations, in: *Proc. of NAACL*, 2018.

[9] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805* (2018).

[10] A. Akbik, D. Blythe and R. Vollgraf, Contextual string embeddings for sequence labeling, in: *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 1638–1649.

[11] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei and I. Sutskever, Language Models are Unsupervised Multitask Learners (2019).

[12] M. Marcińczuk, M. Oleksy and J. Kocoń, Inforex—A collaborative system for text corpora annotation and analysis, in: *Proceedings of the international conference recent advances in natural language processing, RANLP*, INCOMA Shoumen, 2017, pp. 473–482.