Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200600

Automatic Extraction of Lithuanian Cybersecurity Terms Using Deep Learning Approaches

Aivaras ROKAS [a,1], Sigita RACKEVIČIENĖ [b]
and Andrius UTKA [a]
[a] *Vytautas Magnus University, Lithuania*
[b] *Mykolas Romeris University, Lithuania*

Abstract. The paper presents the results of research on deep learning methods aiming to determine the most effective one for automatic extraction of Lithuanian terms from a specialized domain (cybersecurity) with very restricted resources. A semi-supervised approach to deep learning was chosen for the research as Lithuanian is a less resourced language and large amounts of data, necessary for unsupervised methods, are not available in the selected domain. The findings of the research show that Bi-LSTM network with Bidirectional Encoder Representations from Transformers (BERT) can achieve close to state-of-the-art results.

Keywords. Cybersecurity, terminology, automatic term extraction, deep learning, neural networks, embeddings

1. Introduction

Automatic term extraction is extensively used for the development of termbases and ontologies which are essential in translation, teaching/learning language for specific purposes, domain-specific knowledge acquisition, etc. In addition to well-established statistical, linguistic and hybrid methods, the state-of-the art automatic term extraction is performed by applying machine learning and deep learning systems. However, the latter methods are still under development and need extensive research, especially for under-resourced languages such as Lithuanian. This paper presents research results on the deep learning methods aiming to determine the most effective one for automatic extraction of Lithuanian terms from a specialized domain (cybersecurity) with restricted resources. To achieve the aim, the following objectives were set:

1. To compile a specialised corpus comprising documents on cybersecurity issues;
2. To develop the gold standard corpus (training, validation and test data) with manually labelled terminology;
3. To test various deep learning models (pre-processing of the data, automatic term extraction, and comparison of the results).

Since Lithuanian is a less resourced language, supervised and semi-supervised deep learning methods are most suitable for automatic extraction of Lithuanian terminology

[1] Corresponding Author: Aivaras Rokas; Vytautas Magnus University, K. Donelaičio st. 58, Kaunas, Lithuania; E-mail: aivaras.rokas@vdu.lt.

as unsupervised methods require very large amounts of data. Therefore, in this research, semi-supervised approach was chosen.

To our knowledge, this is the first attempt to apply deep learning approach for Lithuanian term extraction. Until now this method has been mostly used for English terminology [1], [2], [3], [4], [5].

2. Background of the Research

In our research, two types of networks are applied to terminology extraction: long short-term memory (LSTM) and Gated Recurrent Unit (GRU), as well as two types of embeddings: FastText and BERT. Below, the main features of the methods applied are discussed.

2.1. LSTM and GRU Networks

During the last decade, one of the most widely used deep learning methods has been LSTM networks, also applied for terminology extraction. In this natural language processing task, terminology extraction is seen as a sequence labelling problem, where sequence is understood as words in a sentence [1], [2], [3], [4], [5].

LSTM is a type of recurrent neural network (RNN), which uses a cell state and three gates and is able to avoid the long-term dependency problem, memorize data for a longer period of time, and is able to fix vanishing gradient problems which plague generic RNNs [6].

However, LSTM networks have their own shortcomings, for example, a simple LSTM cannot account for context from the future, only from the past. Therefore, for certain NLP tasks a bidirectional LSTM network is employed which is able to make use of both past and future inputs. A bidirectional LSTM has two LSTMs, one capturing the information from the past and another capturing the information from the future, thus potentially improving a generic LSTM network.

To ensure that tags stay consistent, a Conditional Random Fields (CRF) network can be implemented as well. CRF is a probabilistic method for marking and segmenting sequence data [7]. CRFs are able to predict tags using context and calculate the likelihood of transitioning from one tag to another.

A GRU network is yet another type of RNN which, compared to LSTM network, requires fewer parameters and less computational power. It uses only two gates (reset and update gate), whereas LSTM network uses three gates (input, output and forget gate). Therefore, GRU network potentially should be more suitable for applications where training data is scarce [8]. Similarly to LSTM, the GRU network can be potentially improved by utilizing a bidirectional GRU network and further enhancing it by combining it with the CRF network.

2.2. Word Embeddings

In order to employ neural networks for text analysis, word embeddings are a necessary prerequisite. Word embeddings are "dense, distributed, fixed-length word vectors, built using word co-occurrence statistics as per the distributional hypothesis" [9: 2]. Word embeddings can capture semantic and syntactic information of words [10]. Training

word embeddings does not require a labelled dataset, but requires a substantial amount of unlabelled data. There are a variety of embeddings such as word2vec, GloVe, FastText, etc. However, word embeddings like word2vec and GloVe cannot deal with unknown or out-of-vocabulary words. FastText is an improved version of Mikolov's word2vec embedding [11]. It is able to learn morphology of words since it is based on the skip-gram model where each word is represented as a bag of n-gram characters and is able to handle unseen words. Therefore, we use FastText word embeddings in our experiments as FastText is more suited for languages such as Lithuanian with the rich vocabulary and complex morphology.

However, FastText has limitations: it creates a word vector based on all the sentences where it has occurred and does not consider different meanings which a word acquires in different contexts. This problem is solved by using contextual embeddings. Presently, the most widely used one is BERT [12]. It is a multi-layer bidirectional Transformer encoder, which is able to consider context and create a different vector for each contextual use of a word. It can potentially improve previously described networks that are using fixed embeddings like FastText. In our experiment, we compare neural networks using FastText with BERT to determine the best method for automatic terminology extraction of Lithuanian terms.

3. Experiment of Automatic Term Extraction

3.1. Datasets

For the purposes of the research, the specialised Lithuanian cybersecurity corpus was compiled. The corpus is intended to reflect the use of cybersecurity language in original and translated texts over a period of 20 years (1999-2019) and is composed of five main categories of texts grouped according to their genres:

1. Legal acts of the Republic of Lithuania: laws, resolutions of the government, orders of ministers on cybersecurity issues;
2. Administrative documents: reports of the National Cybersecurity Centre;
3. Translated EU legislation: EU secondary law acts (directives, regulations); communications of the Commission, opinions of the committees, etc.;
4. Translated international conventions: Convention on Cybercrime;
5. Academic papers: textbooks, scientific papers and books on cybersecurity;
6. Informational publications for the general public on cybersecurity.

Thus, the corpus reflects the use of cybersecurity terms both in national and international settings. The size of the corpus is over 2 mil. words (2,363,618) [13].

As a semi-supervised deep learning approach was chosen for the research, it was necessary to compile the gold standard for the training of deep neural network models used in the experiment. A very small-scale corpus of the selected documents (66,706 words) was compiled for the given purpose and 1,258 cybersecurity terms were manually annotated. The following annotation criteria were formulated: a) linguistic criterion (only nominal units were annotated – nouns, noun phrases, abbreviations, combinations of noun phrases and abbreviations, e.g., *saugumas* 'security', *integruotasis saugumas* 'security by default', *IRT produktas* 'ICT product'); b) conceptual criterion (only nominal units holding relevant terminological value, i.e. denoting concepts of or related to cybersecurity domain, were annotated, e.g. *kibernetinė grėsmė* 'cyber threat',

kibernetiniai išpuoliai 'cyberattacks', *informuotumas apie kibernetinį saugumą* 'cybersecurity awareness').

In this research, the gold standard data were annotated using the BIESO annotation format [14].

3.2. Pre-Processing of Data

In the initial stage of the experiment, pre-processing of cybersecurity corpus and gold standard corpus was conducted. The following pre-processing tasks were performed: file conversion to plain text format, character encoding change, word tokenization, stop-word list development, and text formatting.

In order to train the deep neural network, the gold standard dataset was divided into 3 parts: 70% for training, 20% for validation and 10% for testing.

In this research, word embeddings (that capture syntactic and semantic information of a word) generated by the skip-gram method of FastText and BERT-base multilingual contextual embeddings from Google were applied to the deep neural network [12], [15]. They were selected to better represent rare words [16]. In order to have more effective FastText word embeddings, the dataset was supplemented by the entire Lithuanian Wikipedia database which contains 27,907,392 million words.

3.3. Experimental Setup

In preparation for the experiment, the following methods were analysed: Bidirectional Long Short-Term Memory with CRF (Bi-LSTM-CRF), Bi-LSTM, LSTM, as well as Bidirectional Gated Recurrent Unit with CRF (Bi-GRU-CRF), Bi-GRU and GRU. The experiments by other researchers revealed that the most suitable method to our task would be the Bi-LSTM-CRF [1], [17], [18]. The Bi-LSTM method can "take into account an effectively infinite amount of context on both sides of a word and eliminates the problem of limited context that applies to any feed-forward model" [17: 357], and the CRF layer can take into account the surrounding tags so that predictions stay consistent.

In order to determine the most optimal model, the experiment was carried out in the following stages:

- Firstly, various baseline LSTM and GRU networks were tested using Adam optimizer and FastText embeddings;
- Secondly, each of the best baseline LSTM and GRU networks were tested with various optimizers;
- Thirdly, the best model was compared with a model that has been trained using BERT contextual embeddings to test if contextual embeddings can further improve our model.

Baseline networks were tested using the following hyperparameters: batch size 32, hidden dimensions 100, word vector dimension 100, number of epochs 100, dropout 0.5. These hyperparameters were selected through experimentation of various values and combinations. For example, the increasing the number of hidden layers improves the test error, while a small number of hidden dimensions would lead to underfitting. A low dropout value would yield insignificant results, while a too high a dropout value would result in under-learning.

3.4. Results

In this section, we present the results of our terminology extraction tests performed applying LSTM and GRU networks.

3.4.1. Baseline Tests

In order to identify which of LSTM and GRU baselines perform the best, we have tested 8 baselines: LSTM, LSTM-CRF, Bi-LSTM, Bi-LSTM-CRF, GRU, GRU-CRF, Bi-GRU, and Bi-GRU-CRF.

Table 1. Results of baseline LSTM models

No.	Model	Precision	Recall	F1
1.	LSTM	63.3 %	60.7 %	62.0 %
2.	LSTM-CRF	68.2 %	66.6 %	67.4 %
3.	Bi-LSTM	70.7 %	67.5 %	69.1 %
4.	Bi-LSTM-CRF	73.5 %	67.5 %	70.3 %

Table 2. Results of baseline GRU models

No.	Model	Precision	Recall	F1
1.	GRU	64.5 %	61.7 %	63.1 %
2.	GRU-CRF	70.1 %	61.5 %	65.8 %
3.	Bi-GRU	68.5 %	67.3 %	67.9 %
4.	Bi-GRU-CRF	70.9 %	67.5 %	69.2 %

The results provided in Table 1 and Table 2 reveal that Bi-LSTM-CRF model performed best achieving F1 score of 70.3 %. The second position was taken by Bi-GRU-CRF which fell short only by 1.1 %. Bi-LSTM took the third position and fell short from Bi-LSTM-CRF by 1.2 %. The worst performing models proved to be generic LSTM reaching only 62.0 % and generic GRU reaching 63.1 %.

3.4.2. Bi-LSTM-CRF and Bi-GRU-CRF Tests with Various Optimizers

The efficiency of neural network training greatly depends on optimisation strategies. The Bi-LSTM-CRF and Bi-GRU-CRF models were tested using the following optimizers: Adam [19], SGD [20], AdaDelta [21], RMSprop [22], Adagrad [23]. It is important to note that the learning rate for each optimizer was set to 0.001, except for Adagrad and SGD for which the learning rate was set to 0.01.

The findings provided in Table 3 and Table 4 reveal that the two best variations of Bi-GRU-CRF and Bi-LSTM-CRF are the ones with RMSprop and AdaDelta optimizers respectively. The highest scores in all three categories (precision, recall and F1) were reached by Bi-LSTM-CRF with AdaDelta optimizer with 5.2 % increase, when compared to the best baseline test.

Table 3. Results of five optimizers applied to Bi-LSTM-CRF

No.	Optimizer	Precision	Recall	F1
1.	Adam	73.5 %	67.5 %	70.3 %
2.	Stochastic gradient descent	69.0 %	55.4 %	61.3 %
3.	AdaDelta	78.5 %	72.7 %	75.5 %
4.	RMSprop	76.3 %	71.6 %	73.8 %
5.	Adagrad	71.3 %	59.3 %	64.7 %

Table 4. Results of five optimizers applied to Bi-GRU-CRF

No.	Optimizer	Precision	Recall	F1
1.	Adam	70.9 %	67.5 %	69.2 %
2.	Stochastic gradient descent	68.3 %	64.7 %	66.5 %
3.	AdaDelta	65.8 %	61.6 %	63.7 %
4.	RMSprop	78.2 %	68.4 %	73.3 %
5.	Adagrad	72.5 %	63.7 %	68.1 %

3.4.3. BERT

In the last stage of the experiment, the best model (Bi-LSTM-CRF with AdaDelta optimizer and FastText embeddings) was contrasted to Bi-LSTM network with BERT embeddings.

For our test with BERT, we used Adam optimization algorithm with weight decay as it is the default optimizer that BERT was trained on. The hyperparameters remained the same as in the previous networks. Our Bi-LSTM network trained with BERT embeddings reached precision of 79.4 %, recall 77.8 %, and F1 78.6 %. This is a 3.1 % F1 increase which is significant, especially with such a small training dataset. The initial review of the extracted terms shows that BERT is able to extract more previously unseen terms compared to Bi-LSTM-CRF. Overall, BERT seems to improve our model in every aspect.

During the experiment, we discovered that having trained our neural network using multilingual BERT embeddings with monolingual (Lithuanian) training data, the model has also trained itself on 103 other languages. This phenomenon is recorded by Pires et al., as well [24]. This is possible because originally multilingual BERT embeddings were trained on 104 different languages. Therefore, it was able to recognize and extract cybersecurity terms from all 104 languages that multilingual BERT supports despite the training data being annotated only with Lithuanian terms. This can potentially be very useful in bilingual and multilingual NLP tasks such as supervised or semi-supervised terminology extraction by reducing the amount of annotation data. In order to determine its effectiveness and reliability on other languages for terminology extraction, a more extensive testing is required.

4. Conclusions

The presented experiments confirm that deep learning models can be successfully applied to automatic extraction of Lithuanian domain specific terms and enable to achieve high precision, recall, and F1 scores even with very small annotated training data.

In the first stage of the experiment where the baselines of LSTM and GRU neural networks were tested, Bi-LSTM-CRF and Bi-GRU-CRF networks showed the best performance reaching F1 scores of 70.3 % and 69.2 %, respectively.

In the second stage, Bi-LSTM-CRF with AdaDelta optimizer achieved the best results with F1 of 75.5 %. Our results can be compared to Kucza et al. [10], who similarly tackled domain-specific term extraction using neural networks as a sequence labelling problem and with Bi-LSTM reached F1 score of 86.73 %. In this case, our best performing model in the second stage of the experiment (Bi-LSTM-CRF) fell short by 11.2 %. This rather big difference could be due to the much smaller amount of annotated terms: the dataset in [10] (GENIA and ACL RD-TEC 2.0) consisted of 78,567 annotated terms vs. our dataset with 1,258 annotated terms. In Kucza et al., [10] the experiment Bi-GRU outperformed their best performing LSTM model by 0.87 %, whereas in our tests, Bi-LSTM-CRF outperforms Bi-GRU-CRF by 1.1 %. In another experiment performed by Wang et al. [4], who similarly used a LSTM network for domain-specific term extraction, the best achieved result was 69.2 % on the ACL RD-TEC dataset which is 6.3 % less than our best performing Bi-LSTM-CRF network on the Lithuanian cybersecurity dataset.

The third stage of our experiment further improved the performance of Bi-LSTM model reaching F1 score of 78.6 %. This result was achieved using Bi-LSTM with BERT embeddings. Besides, our model using multilingual BERT embeddings, which was trained with monolingual data, managed to train itself on other 103 languages.

The results of our experiments suggest that for Lithuanian term extraction, the semi-supervised deep learning approach is a way to go. Although deep neural networks were trained on a very small amount of annotated data, the highest score almost reached 80 %. In order to achieve an even higher score, the quality and quantity of annotated data have to be increased. The automation of annotation of training data would greatly reduce the workload of annotators, thus reducing time consumption and increasing the amount of training data for deep neural networks. In bilingual and multilingual term extraction, multilingual BERT might be potentially helpful as it can reduce the amount of languages to be annotated. Therefore, BERT's multilingual capabilities should be more extensively explored. Also, other word embeddings such as ELMO, GPT-2, etc., and custom BERT embeddings should also be tested.

Acknowledgements

The research is carried out under the project "Bilingual automatic terminology extraction" funded by the Research Council of Lithuania (LMTLT, agreement No. P-MIP-20-282). The project is also included as a use case in COST action "European network for Web-centred linguistic data science" (CA18209).

References

[1] Alzaidy R, Caragea C, Giles CL. Bi-LSTM-CRF Sequence Labeling for Keyphrase Extraction from Scholarly Documents. The world wide web conference; 2019 May 13; p. 2551-2557.

[2] Basaldella M, Antolli E, Serra G, Tasso C. Bidirectional LSTM Recurrent Neural Network for Keyphrase Extraction. Italian Research Conference on Digital Libraries 2018 Jan 25; p. 180-187. Springer, Cham.

[3] Kucza M, Niehues J, Zenkel T, Waibel A, Stüker S. Term Extraction via Neural Sequence Labeling a Comparative Evaluation of Strategies Using Recurrent Neural Networks. Interspeech 2018:2072-2076.

[4] Wang R, Liu W, McDonald C. Featureless Domain-Specific Term Extraction with Minimal Labelled Data. Proceedings of the Australasian Language Technology Association Workshop 2016 Dec; p. 103-112.

[5] Sahrawat D, Mahata D, Kulkarni M, Zhang H, Gosangi R, Stent A, Sharma A, Kumar Y, Shah RR, Zimmermann R. Keyphrase Extraction from Scholarly Articles as Sequence Labeling using Contextualized Embeddings. arXiv preprint arXiv:1910.08840. 2019 Oct 19.

[6] Vasilev I, Slater D, Spacagna G, Roelants P, Zocca V. Python Deep Learning: Exploring Deep Learning Techniques and Neural Network Architectures with Pytorch, Keras, and TensorFlow. Packt Publishing Ltd; 2019 Jan 16.

[7] Lafferty J, McCallum A, and Pereira FCN. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001), p. 282-289.

[8] Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555. 2014 Dec 11.

[9] Almeida F, Xexéo G. Word embeddings: A survey. arXiv preprint arXiv:1901.09069. 2019 Jan 25.

[10] de Sousa RC, Lopes H. Portuguese POS Tagging Using BLSTM Without Handcrafted Features. In Iberoamerican Congress on Pattern Recognition 2019 Oct 28 (pp. 120-130). Springer, Cham.

[11] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems 2013; p. 3111-3119.

[12] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. 2018 Oct 11.

[13] Rokas, Aivaras. 2020. Automatinis kibernetinio saugumo terminų atpažinimas / Automatic Extraction of Cybersecurity Terms. Master thesis. Vytautas Magnus University. Lithuanian.

[14] Mi C, Yang Y, Wang L, Zhou X, Jiang T. A Neural Network Based Model for Loanword Identification in Uyghur. Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018) 2018 May.

[15] Sarkar D. Text analytics with Python: a practitioner's guide to natural language processing. Apress; 2019 May 21.

[16] Brownlee J. Deep Learning for Natural Language Processing: Develop Deep Learning Models for your Natural Language Problems. Machine Learning Mastery; 2017 Nov 21.

[17] Chiu JP, Nichols E. Named entity recognition with bidirectional LSTM-CNNs. Transactions of the Association for Computational Linguistics. 2016 Jul;4:357-70.

[18] Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991. 2015 Aug 9.

[19] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. 2014 Dec 22.

[20] Bottou L. Stochastic gradient descent tricks. Neural networks: Tricks of the trade 2012; Springer, Berlin, Heidelberg. p. 421-436.

[21] Zeiler MD. Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701. 2012 Dec 22.

[22] Hinton G, Srivastava N, Swersky K. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. Cited on. 2012 Feb;14(8).

[23] Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. Journal of machine learning research. 2011;12(Jul):2121-59.

[24] Pires T, Schlinger E, Garrette D. How multilingual is Multilingual BERT? arXiv preprint arXiv:1906.01502. 2019 Jun 4.