Multilingual Automatic Term Extraction in Low-Resource Domains

Tan Le Ngoc, Fatiha Sadat
Université du Québec à Montréal / Montreal, Quebec, Canada
201, avenue du Président-Kennedy, H2X 3Y7 Montréal
le.ngoc_tan@courrier.uqam.ca, sadat.fatiha@uqam.ca

Abstract

With the emergence of the neural networks-based approaches, research on information extraction has benefited from large-scale raw texts by leveraging them using pre-trained embeddings and other data augmentation techniques to deal with challenges and issues in Natural Language Processing tasks. In this paper, we propose an approach using sequence-to-sequence neural networks-based models to deal with term extraction for low-resource domain. Our empirical experiments, evaluating on the multilingual ACTER dataset provided in the LREC-TermEval 2020 shared task on automatic term extraction, proved the efficiency of deep learning approach, in the case of low-data settings, for the automatic term extraction task.

Introduction

There are a lot of researches over the past decades in the Automatic term extraction (ATE) task. However, it remains very challenging. Kageura and Marshman (2019) defined terms as lexical items that represent concepts of a domain. This definition of the concept depends on the specific domain. That makes it more difficult to extract all relevant terms based on the fundamental nature of the terms.

Recent work on sequence-to-sequence neural networks-based models proved the efficiency for multiple NLP applications. To deal with this linguistics aspect, neural networks-based approaches use continuous-space representations of words, word embeddings, in which words that occur in similar context tend to be close to each other in representational space (Mikolov, Yih, and Zweig 2013; Mikolov et al. 2017). The benefits of using neural networks to deal with sparse problem are useful. The accurate term extraction systems play an important role in tasks-driven NLP, especially to handle the out-of-vocabulary terms, infrequent terms, single-word and multi-word terms, etc. In this paper, we describe our proposed approach based on recurrent neural networks sequence-to-sequence, with bidirectional LSTM (Hochreiter and Schmidhuber 1997) model, to deal with term extraction for low resource domain.

The remainder of this paper is organized as follows: Section 2 presents background and related works in the automatic term extraction (ATE). Section 3 describes the proposed approach. Section 4 presents the experiments and the evaluations. Finally, Section 5 gives conclusions and perspectives.

Related Work

The current approaches on ATE can mainly be categorized into rule-based, graph-based, statistical-based and deep learning-based.

In the rule-based approach, ATE systems rely on several features of term lengths, Part-of-Speech (POS) tags and POS patterns (Stanković et al. 2016). However, this approach has not applied for all domains and faces erroneous propagation from POS tagging and parsing to define all possible POS patterns, due to complexity of language-dependent structure (Zhang, Gao, and Ciravegna 2016).

In the graph-based approach, the documents are transformed as a graph. Nodes represent words in the documents. The connexions between nodes represent the co-occurrence between words and have an edge weigth. There are a variety of graph-based methods such as TextRank, TopicRank, SingleRank and PositionRank (Wan and Xiao 2008; Zhang, Petrak, and Maynard 2018; Florescu and Caragea 2017).

In the statistical-based approach, ATE systems are trained based on statistics features extracted from text documents, such as n-gram statistics, term frequency, position of a word and co-occurrence of the relevant terms. Several methods are applied such as TF-IDF, the co-occurrence of the candidate keywords or key phrases, Naive Bayes, Support Vector Machine (Uzun 2005; Zhang et al. 2006). However, statistical-based methods need a large amount of datasets with high quality and are often dependent on the domain. Another problem, during the prediction, consists of the missing of the relevant terms due to their low frequency.

In the deep learning-based approach, ATE systems are basically based on neural networks architecture. Wang, Liu, and McDonald (2016) proposed a framework using both Convolutional Neural Network (CNN) and Long-Short Term Memory (LSTM). Kucza et al. (2018) applied sequence labeling method to extract relevant terms. (Gao and Yuan 2019) proposed a novel term extraction method based on span classification and span ranking on the top of CNN architecture.

Our Proposed Approach

In this paper, we consider the automatic term extraction as sequence labeling. Our approach consists of two main parts: (1) Preprocessing of raw data and (2) Building of the neural network-based term extraction model.

In the first step of the pipeline of the system, the raw corpus is cleaned and tokenized by removing stop words or non Latin-alphabetic symbols. The next step consists of representing the cleaned data into the distributed vectors, also called embeddings (Collobert and Weston 2008). Figure 1 illustrates the architecture of automatic term extraction system based on bidirectional LSTM neural networks.

Long-Short Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) takes an input of a sequence of vectors $(x_1, x_2, ..., x_n)$ and produce an output of a sequence of vectors $(h_1, h_2, ..., h_n)$ to represent the information at each input step. LSTMs incorporate a memory cell which can protect and control the cell state. Several gates control the amount of information from the previous states which should be forgotten and updated the information from the inputs. Formally, the equations to be computed are as following:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{1}$$

$$c_t = (1 - i_t) \odot c_{t-1} \tag{2}$$

$$+ i_t \odot tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{3}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{4}$$

$$h_t = o_t \odot tanh(c_t) \tag{5}$$

where σ is the element-wise sigmoid function, and $\odot$ is the element-wise product. c_t and o_t are the cell state and the output at the step *t*, respectively.

Basically, a bidirectional LSTM is composed of a forward LSTM and a backward LSTM operate on a sequence in forward and backward directions.

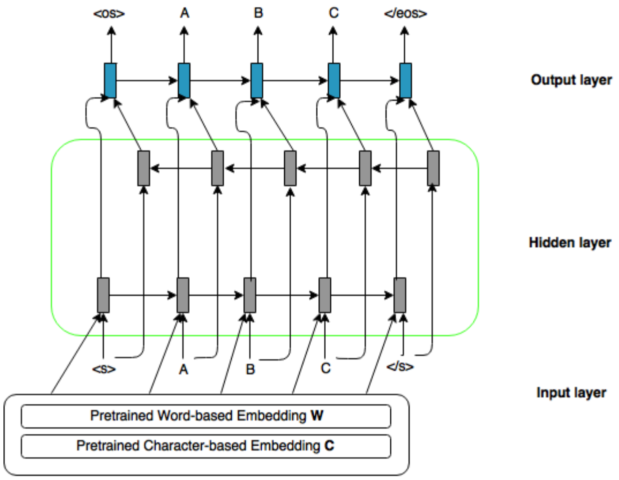
<os>
A
B
C
</eos>
Output layer
Hidden layer
<s>
A
B
C
</s>
Input layer
Pretrained Word-based Embedding **W**
Pretrained Character-based Embedding **C**

Figure 1: The system architecture of bidirectional LSTM neural networks-based automatic term extraction, with A, B, C represent relevant terms.

In this work, we concatenate the word embeddings and the character embeddings to represent the input sequences under vectors in a high dimensional vector space.

Experiments

First, we describe the statistics about the training datasets and the preprocessing steps. Next we provide the configuration about all the hyper-parameters used in our models. Finally, we present the experimental results with an error analysis.

Datasets

We use the training data, ACTER (Annotated Corpora for Term Extraction Research), in the TermEval 2020 shared task on monolingual automatic term extraction. The corpora contain over 100k manual annotations in English, French, and Dutch languages and four different domains such as corruption, dressage, heart failure, and wind energy. The statistics are presented in Table 1. The pretrained embeddings are trained on the raw training data for each appropriated language.

	English	**French**	**Dutch**
Corruption	489,191	475,244	470,242
Dressage	102,654	109,572	103,851
Wind energy	314,618	314,681	308,744
Heart failure	45,788	46,751	47,888

Table 1: Statistics of ACTER corpora with token count. Source: (Terryn, Hoste, and Lefever 2019)

Configuration

In this work, we apply the bidirectional LSTM to train our model for term extraction tasks in low-resource domain. We encode the raw textual document and the list of annotated terms provided by organizers with using pre-trained word embedding. We use the *nltk*[1] toolkit to preprocess the raw datasets by removing stop words and specific symbols, by tokenizing at word level and at character level in the appropriated languages, English, French and Dutch. We train each monolingual word embeddings with *word2vec* in the *gensim*[2] package (Table 2).

	English	**French**	**Dutch**
#embedding size	100	100	100
#vectors	8,227	8,611	8,967
#word index	21,899	26,799	32,100

Table 2: Statistics of pretrained word embeddings for our framework in English, French and Dutch

Then we split the preprocessed datasets into training and testing datasets (Table 3). We train our neural network-based model by using the *tensorflow*[3] package.

[1] https://www.nltk.org/

[2] https://radimrehurek.com/gensim/

[3] https://www.tensorflow.org/

	English	**French**	**Dutch**
#training data	9,570	20,196	13,691
#testing data	759	564	980

Table 3: Statistics of training and testing datasets (sentences) for our framework in English, French and Dutch

To experiment our framework, we apply the following hyper-parameters: 2-layer bi-directional Long Short-term Memory (LSTM) cells, embedding dimension of 100, 128 units in hidden layers in the feed-forward networks, optimizer with *Adam* optimizer (Kingma and Ba 2014), an initial learning rate of 0.001. We run 50 iterations (#max_epochs) with an early stopping based on the categorical cross-entropy scores for the validation set. We used 6-GPUs of NVIDIA GeForce GTX 2080 Ti 12Gb.

Algorithm: bidirectional LSTM
Number of layers: 2 of 128 neurons
Optimization: Adam
Embedding size: 100
Learning rate : 0.001
Batch-size: 32
Number of epoch: 50
Drop-out rate: 0.5
Loss function: Categorical cross-entropy
Activation function: Softmax
Metric: Accuracy

Table 4: Hyper-parameters for our framework in English, French and Dutch

Experimental Results

In the automatic term extraction task, our model performance is calculated based on the most common evaluation metrics in Information Extraction domain such as Precision (P), Recall (R) and F1 score.

$$P = \frac{|\{\textit{relevant tokens}\} \cap \{\textit{found tokens}\}|}{\{\textit{found tokens}\}} \quad (6)$$

$$R = \frac{|\{\textit{relevant tokens}\} \cap \{\textit{found tokens}\}|}{\{\textit{relevant tokens}\}} \quad (7)$$

$$\textit{F1} = \frac{2 \times P \times R}{P + R} \quad (8)$$

where $\{\textit{found tokens}\}$ means the amount of predicted *tokens*, $\{\textit{relevant tokens}\}$ indicates the amount of *tokens* which are correctly annotated.

For evaluate our model, we performed the test on heart failure corpus provided by organizer. The relevant term outputs predicted by our models in English, French and Dutch are presented in Table 5 with and without stop words.

	English	**French**	**Dutch**
with stop words	2,219	2,031	2,861
without stop words	1,879	1,686	2,214

Table 5: Statistics of relevant term outputs predicted by our models in English, French and Dutch

According to the official final results provided by the shared task organizers, we ended up with 5 participating teams. Everyone submitted results for English (Tables 6 and 7), 3 teams submitted for French (Tables 8 and 9) and 2 for Dutch (Tables 10 and 11). Precision, recall, and F1 score were calculated twice: once including and once excluding Named Entities, but the ranking remains the same for the two modes of evaluation.

	Include NEs		
Team	**P**	**R**	**F1**
1.TALN-LS2N	0.3478	0.7087	0.4666
2.RACAI	0.4240	0.4027	0.4131
3.NYU	0.4346	0.2364	0.3062
4.e-Terminology	0.3443	0.1420	0.2010
5.NLPLab_UQAM	0.2145	0.1559	0.1806

Table 6: Evaluation for English datasets with name entities

	Exclude NEs		
Team	**P**	**R**	**F1**
1.TALN-LS2N	0.3258	0.7268	0.4499
2.RACAI	0.3857	0.4011	0.3933
3.NYU	0.4218	0.2512	0.3148
4.e-Terminology	0.3443	0.1554	0.2142
5.NLPLab_UQAM	0.2006	0.1597	0.1778

Table 7: Evaluation for English datasets without name entities

	Include NEs		
Team	**P**	**R**	**F1**
1.TALN-LS2N	0.4517	0.5155	0.4815
2.e-Terminology	0.3633	0.1350	0.1968
3.NLPLab_UQAM	0.1607	0.1118	0.1319

Table 8: Evaluation for French datasets with name entities

	Exclude NEs		
Team	**P**	**R**	**F1**
1.TALN-LS2N	0.4188	0.5088	0.4594
2.e-Terminology	0.3633	0.1437	0.2059
3.NLPLab_UQAM	0.1512	0.1120	0.1287

Table 9: Evaluation for French datasets without name entities

In sum, our NLPLab_UQAM team is ranked at the top in the automatic term extraction shared task for Dutch, with 18,74% and 18,64% F1 score for included name entities and for excluded name entities, respectively. However, our relevant terms outputs predicted by our models for English and

	Include NEs		
Team	**P**	**R**	**F1**
1.NLPLab_UQAM	0.1893	0.1856	0.1874
2.e-Terminology	0.2903	0.0957	0.1440

Table 10: Evaluation for Dutch datasets with name entities

	Exclude NEs		
Team	**P**	**R**	**F1**
1.NLPLab_UQAM	0.1807	0.1926	0.1864
2.e-Terminology	0.2903	0.1040	0.1531

Table 11: Evaluation for Dutch datasets without name entities

French are not good, due to several causes. We explain, in error analysis subsection, how our models have problems to identify and to predict correctly relevant terms in the appropriate languages.

Error Analysis

We observe our framework is functional for any language in the automatic term extraction shared task. However, the obtained results are not good enough to find out all possible relevant terms due to several causes.

First, we notice the vocabulary size is not sufficient to deal with the new or unseen words from the testing datasets. The amounts of word index are 21,899, 26,799, 32,100 for English, French and Dutch, respectively (Table 2). Even using the pretrained embeddings could help to reduce the sparcity of the training data. This is the out-of-vocabulary challenge. Second, our models are trained with low resource settings. Third, we did not use any part-of-speech patterns in order to filter the predicted outputs. Consequently, that caused incorrect relevant terms outputs, including the prepositions or the determinants, decreased our performance evaluations. Some illustrations are presented as follows with stop words towards without stop words:

- (EN) abolished *the* inotropic effect - abolished inotropic effect
  ventricle *in* heart failure - ventricle heart failure
- (FR) évaluation *de la* fonction - évaluation fonction
  prise *en* charge *de l* ' insuffisance cardiaque suivi - prise charge ' insuffisance cardiaque suivi
- (NL) patiënten *met* hartfalen - patiënten hartfalen
  traject *te* voorkomen - traject voorkomen

Conclusion

In this paper, we have investigated the effects of using sequence-to-sequence neural networks-based models to deal with term extraction for low resource domain. Our empirical experiments proved the efficiency of deep learning approach, in the case of low-data settings, for the automatic term extraction task. The benefits of using neural networks to deal with sparse problem are useful. Future study should examine more domain-specific features in order to improve the accuracy of our model. In addition, we should investigate other neural network architectures for data augmentation in order to improve the system performance in the case of less-resourced languages.

References

Collobert, R., and Weston, J. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, 160–167. ACM.

Florescu, C., and Caragea, C. 2017. Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1105–1115.

Gao, Y., and Yuan, Y. 2019. Feature-less end-to-end nested term extraction. In Tang, J.; Kan, M.; Zhao, D.; Li, S.; and Zan, H., eds., *Natural Language Processing and Chinese Computing - 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9-14, 2019, Proceedings, Part II*, volume 11839 of *Lecture Notes in Computer Science*, 607–616. Springer.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Comput.* 9(8):1735–1780.

Kageura, K., and Marshman, E. 2019. *Terminology extraction and management*. Routledge Editor.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kucza, M.; Niehues, J.; Zenkel, T.; Waibel, A.; and Stüker, S. 2018. Term extraction via neural sequence labeling a comparative evaluation of strategies using recurrent neural networks. In *Interspeech*, 2072–2076.

Mikolov, T.; Grave, E.; Bojanowski, P.; Puhrsch, C.; and Joulin, A. 2017. Advances in pre-training distributed word representations. *arXiv preprint arXiv:1712.09405*.

Mikolov, T.; Yih, W.-t.; and Zweig, G. 2013. Linguistic regularities in continuous space word representations. In *hlt-Naacl*, volume 13, 746–751.

Stanković, R.; Krstev, C.; Obradović, I.; Lazić, B.; and Trtovac, A. 2016. Rule-based automatic multi-word term extraction and lemmatization. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 507–514.

Terryn, A. R.; Hoste, V.; and Lefever, E. 2019. In no uncertain terms: a dataset for monolingual and multilingual automatic term extraction from comparable corpora. *Language Resources and Evaluation* 1–34.

Uzun, Y. 2005. Keyword extraction using naive bayes. In *Bilkent University, Department of Computer Science, Turkey www. cs. bilkent. edu. tr/˜ guvenir/courses/CS550/Workshop/Yasin_Uzun. pdf.*

Wan, X., and Xiao, J. 2008. Single document keyphrase extraction using neighborhood knowledge. In *AAAI*, volume 8, 855–860.

Wang, R.; Liu, W.; and McDonald, C. 2016. Featureless domain-specific term extraction with minimal labelled data.

In *Proceedings of the Australasian Language Technology Association Workshop 2016*, 103–112.

Zhang, K.; Xu, H.; Tang, J.; and Li, J. 2006. Keyword extraction using support vector machine. In *international conference on web-age information management*, 85–96. Springer.

Zhang, Z.; Gao, J.; and Ciravegna, F. 2016. Jate 2.0: Java automatic term extraction with apache solr. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 2262–2269.

Zhang, Z.; Petrak, J.; and Maynard, D. 2018. Adapted textrank for term extraction: A generic method of improving automatic term extraction algorithms. *Procedia Computer Science* 137:102–108.