ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

Exploring the feature space of character-level embeddings

Ivano Lauriola[1,2]*, Stefano Campese[1], Alberto Lavelli[2], Fabio Rinaldi[2,3], and Fabio Aiolli[1]

1- University of Padova - Dept of Mathematics
Padova - Italy

2- Fondazione Bruno Kessler
I-38123 Trento, Italy

3- Dalle Molle Institute for Artificial Intelligence Research - IDSIA
CH-6928 Manno, Switzerland

Abstract. Recently, character-level embeddings have become popular in the Natural Language Processing community. These representations provide a description of a word which depends solely on its inner structure, i.e. the sequence of characters. Convolutional and recurrent neural networks are the undisputed protagonists in this context, and they represent the state of the art for many character-level applications. In this work, we firstly compare different neural architectures against adaptive string kernels in simplified scenarios. Then, we propose a hybrid ensemble that injects structural kernel-based features into a neural architecture, providing an efficient and scalable solution. An all-around experimental assessment has been carried out on several string datasets, including biomedical entity recognition and sentiment analysis.

1 Introduction

One of the emerging challenges in the Natural Language Processing (NLP) community is the development of models able to learn a suitable representation (or embedding) of words, sentences, and documents. Several representations have been proposed in the last decade for this purpose, as is the case of popular word-embeddings [1], where words with the same meaning have a similar representation. Word-embeddings have revolutionised most of the existing NLP applications, and they are often placed inside a plethora of sentence-level downstream tasks, such as Entity recognition [2] or Part-of-Speech tagging [3]. However, word-embeddings have several drawbacks. Firstly, they cannot represent out-of-vocabulary words. Secondly, ordinary word-based representations do not take into account the internal structure of a word, i.e. the sequence of characters that compose it. This inner structure could provide useful information in several tasks and domains. For instance, in the biomedical field, proteins and chemical compounds have a particular structure that is a fundamental source of information [4]. To this end, character-level representations, such as the embeddings learned by fastText [5], have been recently explored. These methods learn a representation of a word which strictly depends on the sequence of characters.

Headed by Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs), neural architectures have been widely used to learn character-level embed-

*The work of I. Lauriola is partially supported by grant CR30I1_162758 of the Swiss National Science Foundation.

dings, with applications in question answering [6], biomedical Named Entity Recognition [7], and machine translation [8]. Kernel methods represent a possible alternative, and they have been already used for similar purposes [9]. Each method has its own peculiarities and drawbacks. For instance, Neural Networks (NNs) require a huge amount of training data and specialized hardware, whereas kernel methods do not scale well with large datasets.

Having said that, this paper exposes a two-fold contribution. Firstly, we analyze the quality of character-level embeddings developed by different approaches, i.e. RNN, CNN, and adaptive string kernels, showing remarkable results. Secondly, we propose a hybrid architecture, here named Extreme Spectrum Machine (ESM), which merges explicit string kernels with random weights (i.e. untrained) NNs. We evaluate our methods on several Biomedical Named Entity Recognition (BNER) and Sentiment Analysis (SA) datasets.

2 Background

While CNN and RNN are nowadays mighty popular and they do not need an exhaustive introduction, kernel methods are briefly described here, focusing on spectrum kernels and kernels combinations.

Kernel methods rely on a modular architecture which comprises two elements, a general purpose optimization engine, and the kernel [10]. The kernel is a semidefinite positive function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that defines the similarity between pairs of inputs through their dot-products in an implicit feature space $\mathcal{K}$, i.e. $k(\boldsymbol{x}, \boldsymbol{z}) = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{z}) \rangle$. The embedding function $\phi : \mathcal{X} \to \mathcal{K}$ maps data from the input space $\mathcal{X}$ to the kernel space $\mathcal{K}$. The kernel allows us to solve tasks with different data-types and structures, such as graphs, trees, and sequences or strings.

2.1 Spectrum kernels

Let Σ be the set of possible symbols and $\Sigma^* = \bigcup_p \Sigma^p$ be the set of strings of arbitrary length. Several string kernels exist in the literature, such as the popular *spectrum* [10] kernel, whose feature space (or embedding) $\phi^p : \Sigma^* \to \mathbb{Z}_+^{|\Sigma^p|}$ counts any possible contiguous sub-sequence of length p. The u-th feature computed on an input string $\boldsymbol{x}$ is defined as $\phi_u^p(\boldsymbol{x}) = |\{(v_1, v_2) : \boldsymbol{x} = v_1 u v_2\}|$, $u \in \Sigma^p$, where $v_1 u v_2$ denotes the concatenation of strings v_1, u, and v_2. The p-spectrum kernel is defined as the dot-product between p-spectrum embeddings, that is $k^p(\boldsymbol{x}, \boldsymbol{z}) = \langle \phi^p(\boldsymbol{x}), \phi^p(\boldsymbol{z}) \rangle$.

In this work, the Multiple Kernel Learning (MKL) [11] framework has been used to learn linear non-negative combinations of spectrum kernels, with form: $k_{\boldsymbol{\mu}}(\boldsymbol{x}, \boldsymbol{z}) = \sum_r \mu_r k_r(\boldsymbol{x}, \boldsymbol{z}),\ \|\boldsymbol{\mu}\|_1 = 1 \wedge \mu_r \geq 0$, where k_r are the base kernels, and $\boldsymbol{\mu}$ is the weights vector which parametrizes the combination. The EasyMKL [11] algorithm has been used to learn the optimal combination of base spectrum kernels.

2.2 A qualitative comparison

Different methods emphasize different aspects of a problem, producing different embeddings. Spectrum features represent the number of occurrences of a fixed length

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

sub-string. Clearly, this representation is based on local information provided by substructures rather than long term dependencies, and the same sub-string gives the same information when appearing either as prefix or as suffix. Similarly, CNN also focuses on local information, where a sliding window crosses the input sequence and processes the observed sub-string. CNN could be apt in tasks where the presence of sub-structures is important. For instance, in the task of extracting sentiments and polarities from tweets, the mere presence of smiles in the text could drive the classification. On the other hand, recurrent architectures try to catch temporal dependencies between characters and elements of a sequence rather than the presence of a (processed) sub-structure. Recurrent architectures are eligible in problems where long/short relations between characters or sub-structures are important. An intuitive example is the biomedical entity recognition task, where prefixes and suffixes play crucial but different roles.

3 Extreme Spectrum Machine

Hereby, we introduce the Extreme Spectrum Machine (ESM), a simple but effective neural architecture which combines NN and spectrum kernels. The network consists of 3 elements. The first is the spectrum embedding that extracts the explicit spectrum of a word (or short text). We remind that the feature space of the p spectrum kernel has dimension $|\Sigma^p|$, and the number of active features for an input string $\boldsymbol{x} \in \Sigma^*$ is bounded by its length $\|\boldsymbol{x}\|_0 \leq \|\boldsymbol{x}\|_1 = |\boldsymbol{x}| - p + 1$. In other words, the spectrum embedding has a huge dimensionality and it is extremely sparse. A random projection is then applied to reduce the embedding space, making the explicit spectrum embedding tractable. The resulting embedding is defined as $\hat{\phi}^p(\boldsymbol{x}) = \langle \boldsymbol{W}, \phi^p(\boldsymbol{x}) \rangle$, where $\boldsymbol{W} \in \mathbb{R}^{d \times |\Sigma^p|}$ is the random projection matrix, and $\hat{\phi} : \Sigma^* \rightarrow \mathbb{R}^d$ is the ESM embedding function. The dimension d of the projection is an hyper-parameter. After the compression, a learnable dense layer is placed on the top of the ESM, performing the classification.

The computation of the initial p-spectrum embedding has linear complexity with the sequence length, whereas the remainder of the network just requires a couple of matrix multiplications, making the system extremely fast in training and inference. Consequently, the ESM can be easily applied to low-resource scenarios and to tasks with a lack of available training data (such as specific languages). Moreover, the ESM can be easily placed inside other sentence-level architectures, such as the popular transformers, without significantly increasing the number of total parameters. However, in this work we simply introduce the ESM framework, showing its potentiality and effectiveness in spite of its simplicity. Future directions, such as the learnability of $\boldsymbol{W}$ through a language model objective, the inclusion of multiple kernels, and the injection of the framework inside a sentence-level architecture are beyond the scope of this work.

4 Experimental assessment

A large empirical comparison has been carried out to assess different character-level embeddings in 2 NLP tasks, which are Biomedical Named Entity Recognition (BNER), and Sentiment Analysis (SA). A total of 5 BNER datasets have been used for the evaluation. These datasets (described in [4]) contain tokens or n-grams extracted from biomedical documents through a dictionary look-up approach. The task is a binary classification problem, which consists of identifying if an input string is or not a biomedical entity[1], such as proteins, cellular components, or chemical compounds. Two datasets for the SA task have been included in the assessment. These datasets, here named Twitter-SA[2] and Sentiment-140[3], contain short texts from Twitter. The (binary) label of each tweet is the polarity of the message. A random sub-sampling has been applied to these two datasets to alleviate the computational cost of NNs and MKL. The characteristics of these datasets are shown in Table 1. No pre-processing has been applied.

dataset	training	test	$\|\Sigma\|$	seq. length	max length
BioNLP13CG-cc	1824	646	52	$5.75_{\pm 2.72}$	16
BioNLP13CG-chem	5365	1656	71	$5.37_{\pm 3.55}$	24
BioNLP13CG-species	1847	369	63	$5.18_{\pm 2.37}$	21
BioNLP13PC-cc	1939	1016	53	$7.00_{\pm 4.42}$	34
BioNLP13PC-chem	7974	4734	77	$8.18_{\pm 6.72}$	57
Twitter-SA	3000	1000	152	$63.74_{\pm 38.60}$	167
Sentiment-140	10000	1000	94	$74.36_{\pm 36.75}$	212

Table 1: Dataset description. Statistics have been computed on the training sets.

4.1 Model selection

Training sets have been split into training (80%) and validation (20%) sets. The same training/validation split has been used in every experiment. The baselines and methods compared are CNN, RNN, LSTM, SVM, MKL, and the proposed ESM. The hyper-parameters of CNNs that have been selected in validation are the number of convolution layers $\{1, 2\}$, the number of filters $\{128 \dots 512\}$, and their dimension $\{3, 5, 7\}$. In the case of RNNs and LSTMs, the hyper-parameters are the recurrent dropout $\{0, 0.2, 0.5\}$, the number of recurrent layers $\{1, 2\}$ and their dimensions, with $\{128 \dots 512\}$ neurons. The spectrum degree $p \in \mathcal{P} = \{1 \dots 6\}$ and the dimension of the random projection $\{128 \dots 1024\}$ for the ESM have been also selected in validation.

At the top of the networks, a dense layer has been stacked for doing the classification. Its dimension has been selected in $\{64 \dots 512\}$. The Adam optimizer (1e-4 lr) has been used, with cross-entropy loss. The batch size has been set to 32. For each neural architecture (CNN, RNN, and LSTM, ESM), the best configuration, i.e. the one which achieves the lowest validation loss, has been applied to the test set. The same procedure has been repeated 5 times, reporting average scores and standard deviation.

In the case of SVM, the validation covers the $C \in \{10^i : i = -1 \dots 4\}$ value and the spectrum degree $p \in \mathcal{P}$. All spectrum kernels in $\mathcal{P}$ have been combined by means of the EasyMKL algorithm. EasyMKL has a regularization hyper-parameter, λ, with possible values in $\{0, 0.1 \dots 1\}$, The code and the datasets are freely available[4].

[1] These datasets describe a simplified version of the original NER task.

[2] https://www.kaggle.com/c/twitter-sentiment-analysis2

[3] https://www.kaggle.com/kazanova/sentiment140

[4] https://github.com/sirCamp/extreme_spectrum_machine

4.2 Evaluation

Results of the evaluation are show in Table 2. Not surprisingly, simple RNN outperforms the LSTM in 4 out of 5 entity recognition tasks. These datasets consist of short strings, where there is no need to model long-range dependencies. However, results are overthrown in the case of (much longer) tweets. On the other hand, kernel methods, i.e. SVM and MKL, expose useful insights to improve our approach. The SVM with a single string kernel achieves lowest results on average, whereas the MKL is showing an exceptional performance on several tasks.

dataset	CNN	RNN	LSTM	SVM	MKL	ESM
B.13CG-cc	$92.3_{\pm.8}$	$92.3_{\pm.1}$	$92.1_{\pm.5}$	92.0	93.0	**93.2**$_{\pm.6}$
B.13CG-chem	**88.0**$_{\pm.8}$	$86.2_{\pm.5}$	$85.5_{\pm.6}$	84.5	85.3	$86.8_{\pm.8}$
B.13CG-species	$90.2_{\pm2.5}$	$89.9_{\pm.5}$	$90.2_{\pm.5}$	88.7	**90.5**	**90.5**$_{\pm.1}$
B.13PC-cc	$93.0_{\pm.16}$	$94.0_{\pm.4}$	$92.0_{\pm.4}$	92.2	94.0	**95.8**$_{\pm.8}$
B.13PC-chem	**89.8**$_{\pm2.9}$	$89.4_{\pm.2}$	$88.6_{\pm.5}$	88.6	**89.8**	$88.5_{\pm.2}$
Twitter-SA	$71.0_{\pm.6}$	$62.8_{\pm.3}$	$72.6_{\pm.8}$	76.6	**77.6**	$75.1_{\pm.9}$
Sentiment-140	$59.2_{\pm.9}$	$57.0_{\pm.2}$	$58.7_{\pm.7}$	68.0	**71.9**	$70.0_{\pm.5}$

Table 2: Accuracy scores computed on the test sets. Best results are highlighted.

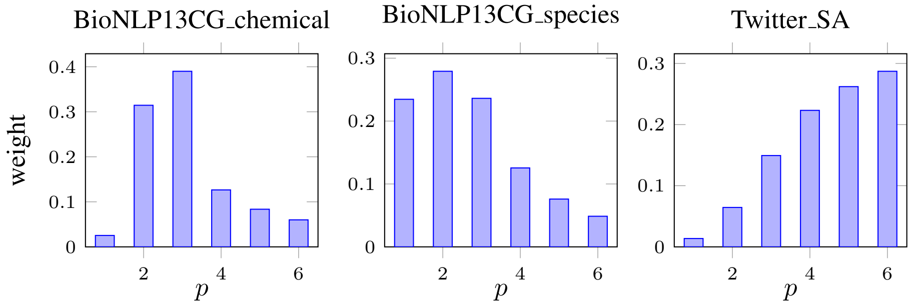

Fig. 1: The combination weights computed by EasyMKL.

To better explore the MKL results, the combination weights $\boldsymbol{\mu}$ learned by EasyMKL on different dataset are exposed in Fig. 1. Note that the highest contribution in biomedical datasets is provided by sub-sequences of length 2 and 3, that are sufficiently large to cover affixes, which are a good informative source. Reasonably, the weights distribution learned from tweets is focusing on larger sub-structures. This result clearly indicates that the combination of multiple spectrum features provides a considerable improvement in accuracy. Moreover, the combination mechanism plays a key role, and it is not trivial. This aspect will be taken into account in future extensions of the framework.

Finally, we compared our approach against fastText (FT) [5] as feature extractor on a few biomedical datasets. FT is a popular pre-trained model that mixes word-level and character-level information. However, FT achieves only +0.1 of accuracy on B.13CG-cc/PC-cc, and +0.2 on CG-species with the cost of an expensive pre-training on large corpora and 2 orders of magnitude of additional learnable parameters.

5 Conclusions

This work introduces the Extreme Spectrum Machine (ESM), a simple approach that engages structural spectrum features inside a neural network to learn character-level representations. The ESM has been compared against several neural networks and kernel methods, showing notable results on several datasets.

This work opens multiple interesting directions. Firstly, the embeddings produced by ESM can be easily injected into sentence-based architectures, such as the popular Transformer. This aspect, with an adequate pre-training can easily improve the effectiveness of ESM. Secondly, more complex kernel functions, such as the fixed-length sub-sequences, or the gap-weighted sub-sequences kernels, can be taken into account to improve the expressiveness of the embeddings. Finally, results from MKL suggest that combining multi-spectrum features is a key point for this approach, and this aspect will be investigated in depth.

References

[1] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[2] Maryam Habibi, Leon Weber, Mariana Neves, David Luis Wiegandt, and Ulf Leser. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33(14):i37–i48, 2017.

[3] Erick R Fonseca, João Luís G Rosa, and Sandra Maria Aluísio. Evaluating word embeddings and a revised corpus for part-of-speech tagging in Portuguese. *Journal of the Brazilian Computer Society*, 21(1):2, 2015.

[4] Ivano Lauriola, Riccardo Sella, Fabio Aiolli, Alberto Lavelli, and Fabio Rinaldi. Learning representations for biomedical named entity recognition. In *Proceedings of the 2nd workshop on Natural Language for Artificial Intelligence (NL4AI 2018)*, pages 83–94, 2018.

[5] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[6] David Golub and Xiaodong He. Character-level question answering with attention. *arXiv preprint arXiv:1604.00727*, 2016.

[7] Mourad Gridach. Character-level neural network for biomedical named entity recognition. *Journal of biomedical informatics*, 70:85–91, 2017.

[8] Jason Lee, Kyunghyun Cho, and Thomas Hofmann. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378, 2017.

[9] Ritambhara Singh and Yanjun Qi. Character based string kernels for bio-entity relation detection. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 66–71, 2016.

[10] John Shawe-Taylor, Nello Cristianini, et al. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[11] Fabio Aiolli and Michele Donini. EasyMKL: a scalable multiple kernel learning algorithm. *Neurocomputing*, 169:215–224, 2015.