applied sciences

MDPI

Article

Preliminary Results on Different Text Processing Tasks Using Encoder-Decoder Networks and the Causal Feature Extractor

Adrián Javaloy [1] and Ginés García-Mateos [2,*]

[1] Max Planck Institute for Intelligent Systems, 72076 Tübingen, Germany; adrian.javaloy@tuebingen.mpg.de
[2] Department of Computer Science and Systems, University of Murcia, 30100 Murcia, Spain
* Correspondence: ginesgm@um.es; Tel.: +34-868-888-530

Received: 28 July 2020; Accepted: 18 August 2020; Published: 20 August 2020

Abstract: Deep learning methods are gaining popularity in different application domains, and especially in natural language processing. It is commonly believed that using a large enough dataset and an adequate network architecture, almost any processing problem can be solved. A frequent and widely used typology is the encoder-decoder architecture, where the input data is transformed into an intermediate code by means of an encoder, and then a decoder takes this code to produce its output. Different types of networks can be used in the encoder and the decoder, depending on the problem of interest, such as convolutional neural networks (CNN) or long-short term memories (LSTM). This paper uses for the encoder a method recently proposed, called Causal Feature Extractor (CFE). It is based on causal convolutions (i.e., convolutions that depend only on one direction of the input), dilatation (i.e., increasing the aperture size of the convolutions) and bidirectionality (i.e., independent networks in both directions). Some preliminary results are presented on three different tasks and compared with state-of-the-art methods: bilingual translation, LaTeX decompilation and audio transcription. The proposed method achieves promising results, showing its ubiquity to work with text, audio and images. Moreover, it has a shorter training time, requiring less time per iteration, and a good use of the attention mechanisms based on attention matrices.

Keywords: natural language processing; deep neural networks; causal encoder; bilingual translation; speech-to-text; LaTeX decompilation

1. Introduction

Deep neural networks (DNN) are going through a golden era, demonstrating great effectiveness and high ubiquity to be used in different areas of research, specifically in natural language processing (NLP) tasks. Presently, hardware capabilities have been multiplied and dataset sizes have also grown to the point of having millions of entries in problems such as bilingual translation, audio transcription or LaTeX decompilation. For example, recently Yang et al. [1] presented an interesting survey of the state of the art in bilingual translation or, in general, neural machine translation (NMT) problems. The existing approaches are divided into recurrent and non-recurrent models; between them, the Transformer model by Vaswani et al. [2] achieved remarkable improvements by exploiting the idea of fully attention-based models. Some works, such as the ConvS2S model by Gehring et al. [3], address NMT problems in a fully convolutional approach, obtaining results that are comparable to the state of the art. This methodology has also been applied to speech recognition in audio, such as the work by Kameoka et al. [4]. Concerning the problem of LaTeX decompilation, it can also be understood as an NMT task, in this case from image to text [5]. Deng et al. [6] proposed a convolutional solution to this problem using a hierarchical attention mechanism called coarse-to-fine, which produced significant improvements over previous systems with simpler attention models.

However, this progress in DNN applied in NMT has also translated into a more competitive research environment, promoting some bad habits that have been built over the years. As stated by Lipton and Steinhardt [7], these include claiming hypothesis as true even when they have not been proved, not clearly differentiating between speculations and facts, or flooding their written works with unnecessary mathematical formulas with the aim of showing expertise.

In this paper, a novel type of encoder recently proposed [8], called Causal Feature Extractor (CFE), is assessed for different NLP tasks. It is based on the causal convolutional neural networks introduced by Oord et al. [9], and it is used as the encoder in an encoder-decoder architecture, a commonly used model in MNT tasks. Specifically, this encoder-decoder model is applied to a variety of NLP problems which have something in common: all of them take a sequence as input, and output another sequence that depends on the input. Thus, the main goal of this work is to test the new encoder in different types of input, making use of statistical tests, giving them a strong basis that supports all the conclusions based on the obtained results. A particularity of the selected MNT tasks is that they have different types of input, while the output is always a text sequence: in bilingual translation problem (in our case, from English to German) the input is a text; in LaTeX decompilation, the input is given by the image of an equation; and in audio transcription, the input is a one-dimensional audio signal which is transformed into a spectrogram. CFE is able to work in all these cases, achieving promising results.

2. Materials and Methods

2.1. Encoder-Decoder Architecture

Presently, the predominant technique for implementing deep neural networks in the field of MNT is the encoder-decoder architecture. A great number of variations have been proposed in the literature, offering solutions that make up the state of the art in different tasks [10]. It consists of two parts that collaborate with each other. On the one hand, there is an encoder that, from the input vector $\mathbf{X} = x_1, x_2 \dots x_l$, generates an intermediate vector $\mathbf{Z} = z_1, z_2 \dots z_l$ of the same length as $\mathbf{X}$, where each column z_i describes the characteristics of the environment around the i-th value of the input. On the other hand, the decoder acts sequentially and, at each instant t, it takes the i-th input of the intermediate vector, z_i, and the output produced by itself at the previous instant, y_{t-1}. It computes the output at the current instant, y_t, thus forming the output vector $\mathbf{Y} = y_1, y_2 \dots y_{l'}$, with length l', that can be different from l. The output ends when the decoder produces a special "end of string" symbol.

An interesting complementary technique working in conjunction with the encoder-decoder architecture is the attention mechanism, which was introduced by Bahdanau et al. [11]. It is an effective method that allows the decoder to decide the most interesting parts of the input, i.e., what parts of $\mathbf{Z}$ are used at each instant. This technique has proven to be effective on problems such as audio textual interpretation [12] and other problems [13]. Figure 1a shows a graphical overview of the encoder-decoder architecture with an attention mechanism.

In our case, the attention model is a fully connected neural network (FCNN) with 1 hidden layer and l output values, where l is the size of the input. The input to this part is the intermediate code of the encoder, $\mathbf{Z}$, and the vector of hidden states of the decoder in the previous step, h_{t-1}.

Additional techniques are used to improve the effectiveness of the system, such as dropout [14,15] (randomly removing some neurons with a given probability), weight normalization [16] (regularizing the weights of the neuron layers), gradient clipping [17] (limiting the norm of the gradient to a maximum value), and random search of the hyperparameters [18] (performing different executions of the process to find the optimal configuration of the hyperparameters of the network).

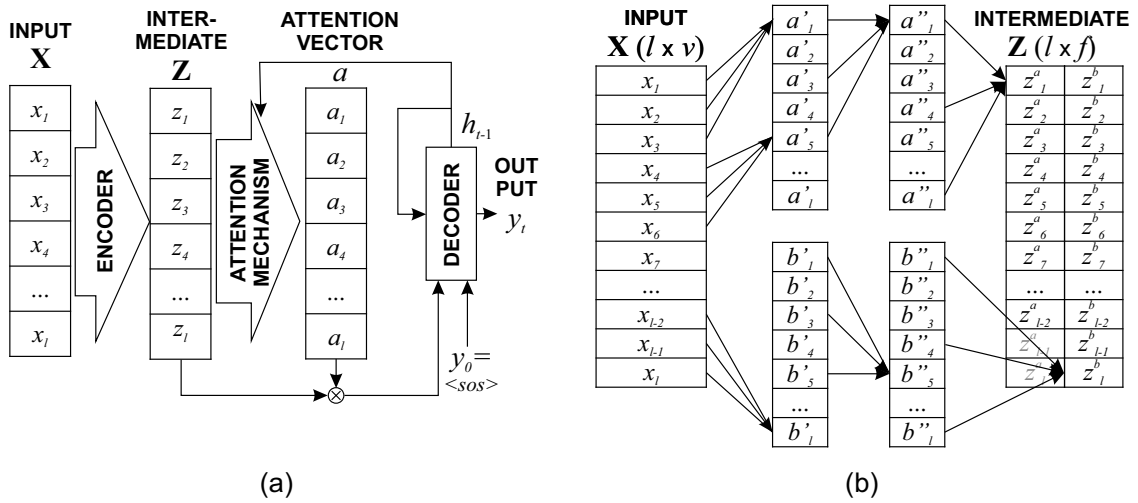

Figure 1. Scheme of the proposed neural network architecture. (**a**) Global scheme of the encoder-decoder architecture including the attention mechanism. The input matrix, **X**, which contains l vector elements ($x_1, x_2 \dots x_l$), is transformed into an intermediate code, $\mathbf{Z} = z_1, z_2 \dots z_l$. Then, the attention mechanism selects the importance, a, of each tuple at each time, t. Using both values, the decoder produces the output at each instant, y_t. The hidden state of the decoder at the previous instant, h_{t-1}, is fed into the attention mechanism and into the decoder. <*sos*> means "start of sequence". (**b**) Outline of the proposed Causal Feature Extractor for the part of the encoder, in this case with 3 layers. The input is matrix **X**, with l vectors of size v. There are two independent sub-nets (upper and lower), each of which generates $f/2$ features for each input vector. They are convolutional NNs which are causal (a uses the previous values, and b uses the later values) and dilated (the step in the 1st layer is 1, in the 2nd layer 2, and in the 3rd layer 4). The output is the code **Z**, with f features for each input vector.

2.2. Causal Feature Extractor

In the encoder-decoder architecture, both the encoder and the decoder are independent modules that can be implemented in different ways. For example, they can consist of Convolutional Neural Networks (CNN) [19], which is a typical selection method in images. In the case of sequential data, Long Short-Term Memories (LSTM) [20] are more frequently found.

As mentioned before, the purpose of this study is to analyze the feasibility of a new type of layer for the encoder, called Causal Feature Extractor (CFE) [8]. This method is inspired by the Dilated Convolutional Neural Networks and the Causal Convolutional Neural Networks, introduced by Oord et al. [9]. The proposed model, depicted in Figure 1b, is built under three main ideas:

- First, in order to extend the receptive field of the convolutions without requiring large kernels, several convolutional neural layers are stacked, and each one has two times the dilation of the previous one. That is, in the first layer, the convolution for position t depends on $t, t-1, t-2 \dots$; in the second layer, it depends on $t, t-2, t-4 \dots$; in the third layer, $t, t-4, t-8 \dots$, and so on.
- Second, with the aim of making a better use of the attention mechanisms in comparison with CNNs, these stacked convolutional layers are turned into causal convolutions, meaning that the output at one position will depend on the inputs previous or next to that position, but never both. This is the same idea as the Causal CNN proposed by Oord et al. [9].
- Third, considering that the use of causal layers means the misuse of one part of the input, two stacks of causal convolution layers are used, each one taking into account a different direction of the input (the previous or the subsequent input values). The same idea of bidirectionality has also been applied to LSTMs [21].

The main hyperparameters that define the structure of the CFE encoder are the CNN kernel width, the desired receptive field, and the number of features to generate, f. The first hyperparameter

indicates the width of the kernels of the convolutions. Along with the second parameter, they determine the number of layers of the CNN. For example, if the kernel width is 5 and the desired receptive field is 20, then there would be 3 convolutional layers (since the dilations are multiplied by 2, the receptive fields of the 1st, 2nd and 3rd layers would be 5, 10 and 20, respectively). Other hyperparameters that are used during the training process are the size of the batches applied in the input, the way of normalizing the weights of the convolutions, the maximum norm of the gradient, and the dropout rate applied to the neurons; there is also the possibility of including or not the position of the input values in the encoder.

In the previous work [8], CFE was applied in the encoder of a text normalization problem (i.e., given a text with symbols, producing a text without symbols as it should be read by a text-to-speech system), achieving a good effectiveness in this NLP problem. The accuracy of the result ranged from 83.5% to 96.8% depending on the training datasets. In the present paper, it is further applied to audio and images; in the second case, the concept of causality considers an order of the pixels from top to bottom and from left to right.

2.3. Language Processing Tasks

The proposed CFE encoder can be applied to any task that requires transforming an input sequence into an output sequence. Thus, the experiments have been focused on the three following well-known computational linguistic problems:

- *Text translation* or *bilingual translation*. This is one of the first and most studied problems in machine NLP, so it is an interesting test bed for the proposed method. Given a text in one language, the output is an equivalent phrase in another language. The difficulty of this task is that there may be words and idioms that do not have a direct translation, or phrases that can be translated into different ways, being all of them valid. The state-of-the-art system used for comparison is given by the Transformer model introduced by Vaswani et al. [2] which overcame the results of other popular machine translation systems such as the GMT model (used in Google Translate). It uses an encoder-decoder architecture and new iterations and improvements of the attention mechanism. We also included in the comparison an encoder-decoder model with LSTM networks in the decoder.

  In the experiments, we used the dataset for the translation from English to German provided in the ACL 2016 Conference on Machine Translation (http://www.statmt.org/wmt16/). The training set of this resource contains near two million parallel sentences (English-German), with a total about 48 million words in English and 45 million words in German. The validation set contains 3000 sentences, and the test set also 3000 sentences. The parameter used to measure the quality of the result is the well-known Bilingual Evaluation Understudy (BLEU) [22]. Another interesting parameter is the perplexity [23], that is used during the training process in the validation set to check the network progress. It is defined as 2 raised to the cross entropy of the empirical distribution of the actual data and the distribution of the predicted values, so that a lower value indicates a better result.

- *LaTeX decompilation*. This problem, which is useful in tasks such as digitization of scientific texts, can also be seen as a particular case of automatic translation. In this case, the input is an image containing a mathematical formula, and the output is a LaTeX command that must produce the same formula as generated by a LaTeX engine. It combines computer vision and neural machine translation, so it is interesting for studying the effectiveness of the proposed CFE model into images. As before, the solution is not necessarily unique, since multiple LaTeX commands can produce the same result.

  The current state-of-the-art model used for comparison is the system presented by Deng et al. [6]. Again, it is based on an encoder-decoder architecture; the encoder consists of two steps, a CNN

and a recurrent network, while the decoder is a recurrent network. The method introduces a specific attention mechanism called coarse-to-fine attention. The experiments have been done with the dataset available in [6], which contains over 103,000 training samples, 9300 validation samples and 10,300 test samples. Some of these samples are shown in Figure 2. The accuracy measures are also the BLUE and the perplexity.

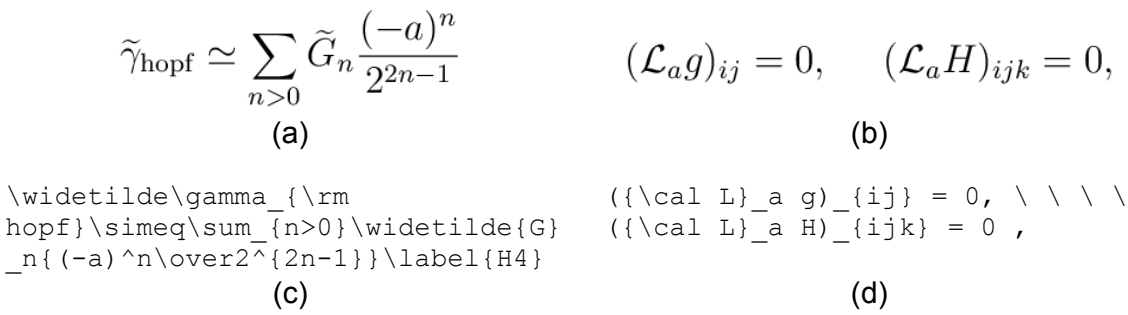

Figure 2. Two sample images of the dataset for the LaTeX decompilation task. (**a**) and (**b**) Input images. (**c**) and (**d**) Output LaTeX commands corresponding to the images. Information extracted from the public dataset: http://lstm.seas.harvard.edu/latex/.

- *Audio transcription*. The task of audio transcription is another well studied problem, which can also be understood as a type of translation, from audio to text. In this way, the main types of input have been analyzed: text, audio, and images. This problem is used both in online services and in out-of-line transcription of multimedia content. The defining characteristic, with respect to the other problems, is the possible existence of noise in the audio.

  The state of the art of this problem is given by models that do not follow an encoder-decoder architecture, but techniques based on hidden Markov models. Nevertheless, there are good encoder-decoder transcription systems which can be used for comparison. In particular, we used the Listen-Attend-Spell model from Chan et al. [12] to compare the results of the proposed CFE. The dataset is the AN4 set from CMU (http://www.speech.cs.cmu.edu/databases/an4/), which contains more than 1000 recordings of dates, names, numbers, etc. Concreting, the training set includes 1018 samples and the test set 140. The accuracy measures are the word error rate (WER) defined as the correctly identified words over the total, and the perplexity.

3. Results and Discussion

3.1. Experimental Setup

For the execution of the experiments, OpenNMT (https://opennmt.net/) was used. It is an open source ecosystem for neural machine translation using Python. We used the implementation based on PyTorch (https://pytorch.org/) deep learning framework. Apart from the library functions, it also offers useful implementations of some recent methods for different problems. In the bilingual translation problem, it includes the Transformer method Vaswani et al. [2], and an alternative encoder-decoder model using LSTM in the encoder. For the LaTeX decompilation problem, the model called Im2Text Deng et al. [6] was used for the comparison; and in the speech-to-text problem, the Listen-Attend-Spell model by Chan et al. [12].

The computer used in the experiments is a PC with an Intel(R) Core(TM) i7-5930K processor with 12 threads (6 with hyperthreading) at a frequency of 3.50 GHz; it has 3 NVIDIA GeForce GTX1080 GPUs and 600 Gb of SSD hard disk, although only one GPU is used in each execution.

For the configuration of the hyperparameters of the networks, two alternatives were tested: a manual adjustment of the parameters; and a random search of the hyperparameters space. In the second case, 30 random combinations of the hyperparameters were tested in a reduced execution of

1 h for each test, selecting the combination with the least error. The resulting structure of the networks using both methods is presented in Table 1. As indicated in this table, in all the cases the encoder is a CFE network, the decoder is a recurrent neural network (RNN), and there can be a dense neural network (or bridge) between them or not.

Table 1. Hyperparameters of the encoder-decoder networks used in the three problems of interest. Bridge: add a dense layer between encoder and decoder. Global attention: score function used in the attention model. Position encoding: add position information in the encoding. RNN layers: number of layers in the RNN of the decoder. RNN size: number of units in each layer of the RNN. CNN kernel width: size of the convolution filters in CFE. Receptive field: selected receptive field for the CFE. Normalization: method used to normalize the gradients. Batch size: size of the batches used in the training. Max grad. norm.: maximum allowed norm of the gradient. Dropout: dropout rate used. Learning rate decay: value applied to reduce the learning rate.

Hyperparameter	**Text Translation**		**LaTeX Decompilation**		**Audio Transcription**	
Method	**Manual**	**Random**	**Manual**	**Random**	**Manual**	**Random**
Bridge	no	yes	no	yes	no	yes
Global attention	general	concat	general	dot	general	dot
Position encoding	no	yes	no	no	yes	no
RNN layers	3	4	2	1	2	1
RNN size	512	238	500	414	500	126
CNN kernel width	5	3	5	11	5	7
Receptive field	20	13	20	16	20	17
Normalization	tokens	sents	sents	sents	sents	tokens
Batch size	64	16	20	12	16	9
Max grad. norm.	1	14.12	20	28.68	20	3.79
Dropout	0.3	0.71	0.3	0.2	0.3	0.52
Learning rate decay	0.5	0.654	0.5	0.576	0.5	0.465

Finally, to validate the statistical significance of the results, the approximate randomization test of Riezler and Maxwell [24] was applied. This test is used to prove that the outputs produced by two prediction systems are statistically distinguishable.

3.2. Accuracy Performance Results

Table 2 summarizes the results obtained by the encoder-decoder networks using CFE configured with both methods, manual and random search, and the alternative state-of-the-art methods, for the three problems of interest.

Table 2. Experimental results for the proposed CFE model and other architectures. BLEU/WER: accuracy measures for the test set, bilingual evaluation understudy (text translation and LaTeX decompilation) and word error rate (audio transcription), respectively; ACC and PER: accuracy and perplexity obtained for the validation set, respectively. The total number of iterations applied for each problem in the training process is indicated.

Problem	**Method**	**BLEU/WER %**	**ACC %**	**PER**
Text translation	CFE Manual	23.34	62.73	13.59
75,000 iterations	CFE Random	27.80	63.02	13.39
	LSTM	36.48	69.64	11.52
	Transformer	34.95	67.78	14.30
LaTeX decompilation	CFE Manual	77.57	96.55	1.24
25,000 iterations	CFE Random	75.82	96.56	1.24
	Im2Text	80.46	96.78	1.13
Audio transcription	CFE Manual	53.12	60.13	8.84
60,000 iteration	CFE Random	55.05	59.13	22.4
	Lis.-Att.-Spell	43.14	70.98	5.37

Figure 3 contains a graphical representation of the training process for these problems, showing the cross entropy of the models throughout the iterations applied.

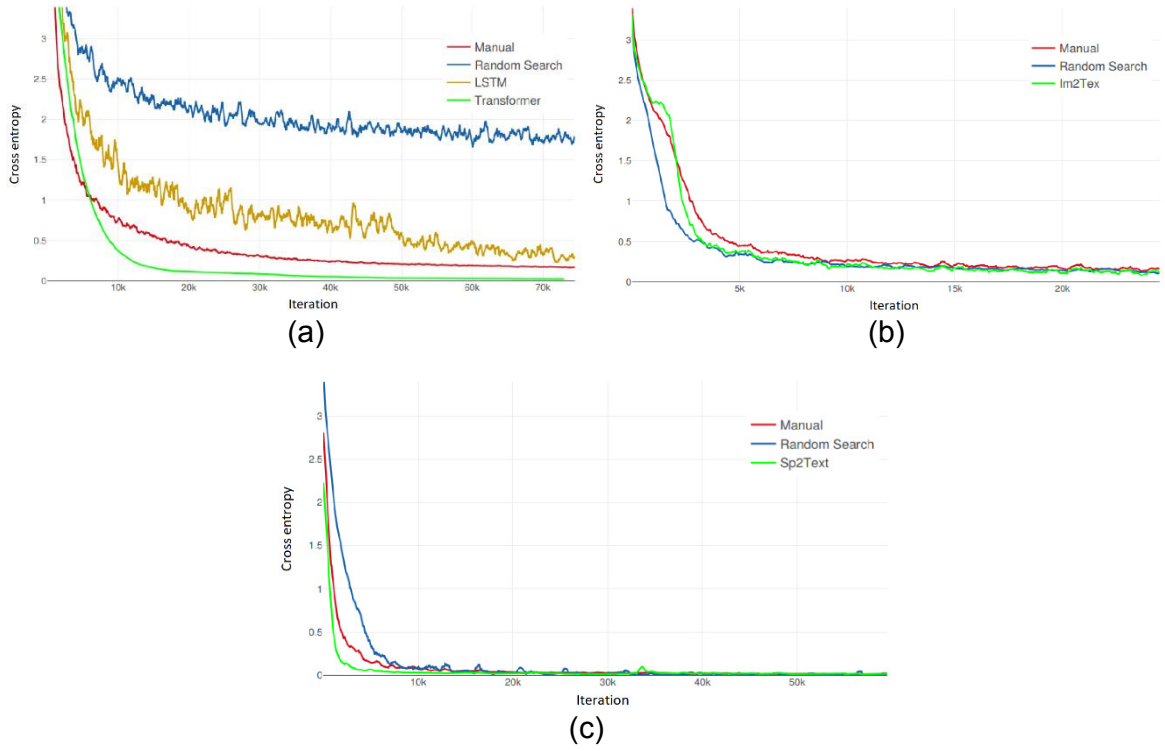

Figure 3. Evolution of the training errors (cross entropy) of the different models compared, for the three problems of interest. (**a**) Text translation. (**b**) LaTeX decompilation. (**c**) Audio transcription.

3.3. Discussion of the Results

In general, the proposed CFE encoder is able to achieve very promising results, near those that are in the state of the art. The evolution of the CFE models in Figure 3 shows a behavior that is very similar to the other systems used for comparison. In any case, these tests should be considered to be preliminary results, needing further experiments and improvements to achieve its full potential. For example, new adaptations could be studied for the decoder network, which was not the purpose of the present work.

These are the main findings of the experiments:

- The proposed CFE models are not able to overcome the results of the state-of-the-art methods used for comparison, as it can be seen in Table 2, although they are very close in many cases. These differences between methods have been confirmed by the approximate randomization tests, indicating that the differences of the predictors are statistically significant. However, it must be observed that these alternative methods are specifically designed for each problem, while the proposed method has shown to be generic, being able to work with text, audio and images, with minimal adaptations for each problem.
- In all the experiments, the number of iterations of the learning process was fixed for each problem (as indicated in Table 2). However, it has to be considered that the average time per iteration is not the same for all the methods. In fact, the proposed CFE encoder is approximately 1.7 times faster than the other alternatives. Thus, for a fixed learning time, the proposed solution could overcome the other methods in some cases. This can be observed in the validation measures (ACC and PER). For example, using the same learning time in the LaTeX decompilation task, CFE achieves an ACC of 96.5%, while Im2Text achieves 96.1%. In other words, Im2Text method needs around

70% more time to achieve its optimum result. A special case is the Transformer method for the problem of bilingual translation, whose average time per iteration is 4 times greater than the time of CFE; so, for the same training time, the performance achieved by CFE would be higher.

- It was observed that the proposed CFE encoder makes a better usage of the attention mechanisms [8]. The attention matrices obtained by CFE are *sharper* than those obtained for the other methods, i.e., they present a bigger different between the elements of interest and those that are not interesting for the decoder. This effect can be observed in the attention matrices shown in Figure 4 for the bilingual translation problem. This is a very positive aspect, since it indicates that future improvements of the proposed method could benefit more from the attention mechanisms.

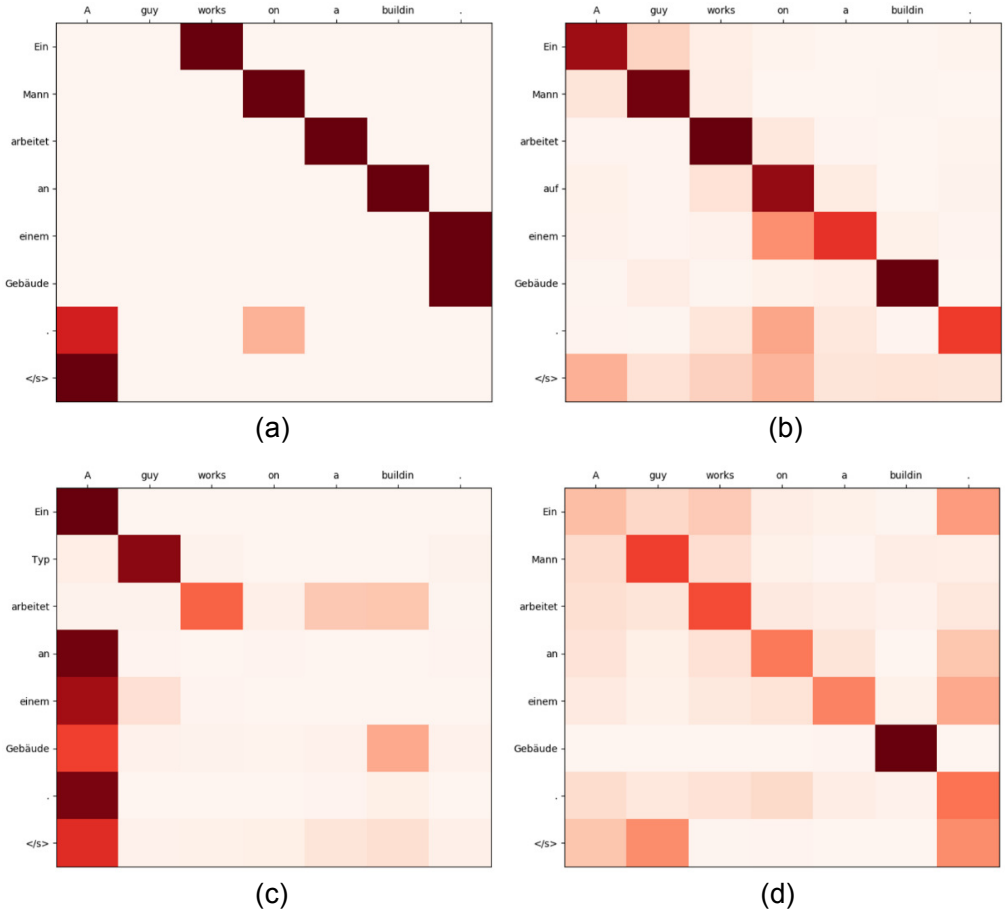

Figure 4. Attention matrices produced by the attention mechanism in the problem of bilingual translation, for the translation of the sentence (horizontal axis) "A guy works on a building". (**a**) Attention matrix for manual CFE, the output (vertical axis) is "Ein Mann arbeitet an einem Gebäude". (**b**) Attention matrix for random CFE, the output is "Ein Mann arbeitet auf einem Gebäude". (**c**) Attention matrix for Transformer model, the output is "Ein Typ arbeitet an einem Gebäude". (**d**) Attention matrix for LSTM, the output is "Ein Mann arbeitet an einem Gebäude".

4. Conclusions

In this paper, we analyzed the feasibility of a novel type of encoder, the Causal Feature Extractor, as a part of an encoder-decoder deep neural network, in different problems of machine neural translation. The results obtained are very promising, achieving a 63.0% accuracy in bilingual translation, 96.6% in LaTeX decompilation and 60.1% in audio transcription. However, the best solution is always the specifically designed system, that has been adjusted and fine-tuned by the corresponding research groups over the years, with improvements of 6.6%, 0.2% and 10.8% in the cited problems, respectively.

Therefore, the results obtained by our approach are close to that of other works that constitute the state of the art, especially in the image processing problem of LaTeX decompilation.

Furthermore, the proposed model has the inherent advantages of convolutional networks with respect to recurrent and LSTM networks. On the one hand, it is a generic architecture that can be adapted to a large number of scenarios, while the use of recurrent networks is more restricted. On the other hand, convolutional networks are known for being parallelizable and highly optimized for training using GPUs, so improving the implementation of this architecture should be much faster than recurrent networks. This was observed in the average execution times per iteration, which is considerably faster for CFE than for the specific models. Those solutions require on average 70% more time than the proposed approaches.

Clearly, there is still ample room for improvement in the application of CFE to the problems of natural language processing. For example, more complex attention mechanisms (such as multi-head attention or local attention) could be combined with the proposed CFE architecture. Also, elimination or relaxation of the use of dilations in the CFE architecture, which could be diluting the influence of the input data too much, could be beneficial. Finally, since the proposed CFE model is very generic, it could be interesting to analyze its application in other areas of computational learning.

Author Contributions: Conceptualization, A.J. and G.G.-M.; methodology, A.J. and G.G.-M.; software, A.J.; validation, A.J. and G.G.-M.; formal analysis, A.J.; investigation, A.J. and G.G.-M.; resources, A.J.; data curation, A.J.; writing—original draft preparation, A.J.; writing—review and editing, A.J. and G.G.-M.; visualization, A.J.; supervision, G.G.-M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Spanish Ministry of Science, Innovation and Universities, FEDER funds, under grant RTI2018-095855-B-I00 (G.G.-M.).

Acknowledgments: Adrián wants to acknowledge support from the Max Planck Institute for Intelligent Systems.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

ACC	Accuracy
BLEU	Bilingual evaluation understudy
CFE	Causal feature encoder
CNN	Convolutional neural networks
FCNN	Fully connected neural network
LSTM	Long short-term memory
MNT	Machine neural translation
NLP	Natural language processing
PER	Perplexity
WER	Word error rate
RNN	Recurrent neural network

References

1. Yang, S.; Wang, Y.; Chu, X. A Survey of Deep Learning Techniques for Neural Machine Translation. *arXiv* **2020**, arXiv:2002.07526.
2. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. Advances in Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
3. Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; Dauphin, Y.N. Convolutional sequence to sequence learning. *arXiv* **2017**, arXiv:1705.03122.
4. Kameoka, H.; Tanaka, K.; Kwaśny, D.; Kaneko, T.; Hojo, N. ConvS2S-VC: Fully Convolutional Sequence-to-Sequence Voice Conversion. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2020**, *28*, 1849–1863. [CrossRef]

5. Daudaravicius, V. Textual and Visual Characteristics of Mathematical Expressions in Scholar Documents. In Proceedings of the Workshop on Extracting Structured Knowledge from Scientific Publications (ESSP), Minneapolis, MN, USA, 6 June 2019; pp. 72–81.
6. Deng, Y.; Kanervisto, A.; Ling, J.; Rush, A.M. Image-to-markup generation with coarse-to-fine attention. In Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, Australia, 7–9 August 2017; pp. 980–989.
7. Lipton, Z.C.; Steinhardt, J. Troubling trends in machine learning scholarship. *Queue* **2019**, *17*, 45–77.
8. Javaloy, A.; García-Mateos, G. Text Normalization Using Encoder–Decoder Networks Based on the Causal Feature Extractor. *Appl. Sci.* **2020**, *10*, 4551. [CrossRef]
9. Oord, A.v.d.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv* **2016**, arXiv:1609.03499.
10. Shrestha, A.; Mahmood, A. Review of deep learning algorithms and architectures. *IEEE Access* **2019**, *7*, 53040–53065. [CrossRef]
11. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
12. Chan, W.; Jaitly, N.; Le, Q.; Vinyals, O. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 4960–4964.
13. Galassi, A.; Lippi, M.; Torroni, P. Attention, please! a critical review of neural attention models in natural language processing. *arXiv* **2019**, arXiv:1902.02181.
14. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
15. Baldi, P.; Sadowski, P. The dropout learning algorithm. *Artif. Intell.* **2014**, *210*, 78–122. [CrossRef] [PubMed]
16. Salimans, T.; Kingma, D.P. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016; pp. 901–909.
17. Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. In Proceedings of the International Conference on Machine Learning (ICML), Atlanta, GA, USA, 16–21 June 2013; pp. 1310–1318.
18. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
19. LeCun, Y.; Haffner, P.; Bottou, L.; Bengio, Y. Object recognition with gradient-based learning. In *Shape, Contour and Grouping in Computer Vision*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 319–345.
20. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
21. Zhou, P.; Shi, W.; Tian, J.; Qi, Z.; Li, B.; Hao, H.; Xu, B. Attention-based bidirectional long short-term memory networks for relation classification. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Berlin, Germany, 7–12 August 2016; pp. 207–212.
22. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. BLEU: A method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Philadelphia, PA, USA, 7–12 July 2002; pp. 311–318.
23. Nabhan, A.R.; Rafea, A. Tuning statistical machine translation parameters using perplexity. In Proceedings of the IRI-2005 IEEE International Conference on Information Reuse and Integration, Las Vegas, NV, USA, 15–17 August 2005; pp. 338–343.
24. Riezler, S.; Maxwell, J.T. On some pitfalls in automatic evaluation and significance testing for MT. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Ann Arbor, MI, USA, 29 June 2005; pp. 57–64.