UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft
Faculty for Computer Science, Electrical Engineering and Mathematics
Department of Computer Science
Research Group Youssef & co

Review on Neural machine

—

translation system

—

jointly trained to align and translate

by
Youssef Ameur

Paderborn, August 14, 2020

Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen worden ist. Alle Ausführungen, die wörtlich oder sinngemäß übernommen worden sind, sind als solche gekennzeichnet.

Paderborn, 15.08.2020
Ort, Datum

Youssef Ameur
Unterschrift

Abstract. Recently many approaches to statistical machine learning have been presented, that are purely based on neural networks. Most of these neural machine translation models, consist of an encoder and a decoder, which respectively encodes an input sentence into a fixed-length-vector, and decodes this representation into a correct translation. The method we discuss in this paper, assumes that the fixed-length-vector is an obstacle to further improving the performance of encoder-decoder architecture based translation systems. To overcome this limitation an extension is added to the model, so that it is allowed to (soft-)search parts of the source sentence that are relevant to predicting a target word, without forming these parts into hard segments. The performance results of the proposed approach will be computed for an English to French translation task and we will see how they compare to state-of-the-art phrase based translation systems. A qualitative analysis that has been done, will further prove that the assumption made holds true.

Contents

1 Introduction **1**

2 Formal Notations **2**

3 Related Work **3**
3.1 Rule based machine translation 3
3.2 Statistical Machine translation 3
3.3 Neural machine translation 4

4 Approach **5**
4.1 RNN-encoder-decoder design pattern 5
4.2 Learning to Align and Translate 6
4.2.1 Encoder 6
4.2.2 Decoder 6
4.2.3 Overall view 7

5 Comparison **8**
5.1 Experiment Setting 8
5.1.1 Data Set 8
5.1.2 Training 8
5.1.3 Evaluation with BLEU 9
5.2 Results 9
5.2.1 Quantitative analysis 9
5.2.2 Qualitative analysis 10

6 Discussion **12**
6.1 Advantages of NMT 12
6.2 Disadvantages of NMT 12

Bibliography **13**

1 Introduction

Different strategies have been introduced to the domain of machine translation, some are rule-based, others statistical, and a newly emergent one, is the neural machine translation.
In this new approach, contrary to others (Koeh et al. [DHHK13], Simard et al. (2007) [SUIK07]), that use multiple subcomponents, the model relies purely on one large trained neural network to output a correct translation to a given sentence.

A widely used design pattern for the neural network architecture is the encoder-decoder architecture, which has been presented in the work of Cho et al.(2014) [CvMB4b], it consists of two recursive neural networks, namely an encoder, that encodes an source sentence into a fixed-length-vector, and a decoder, that from the encoder's output computes a translation.
This method has shown promising results [CvMB4b], nevertheless we notice a significant decline in its performance when dealing with long sentences.
A number of ways have been proposed to better the performance of neural machine translation systems ([Gra14], [WSC+16]), by adding an attention mechanism. However the translation quality degradation remained in regards to long sentence. In [CBB16] it is speculated that the main cause is the restriction of the encoder to output a fixed-length-vector. In the case of long sentences, especially ones that are longer than the sentences in the training corpus, it is difficult to encode all the relevant information contained in them.

In this paper, a possible solution to the underlined issue is presented in [CBB16], it consists in removing the fixed-length-vector, and allowing the encoder-decoder model, after translation of each word to look for a set of positions in the original sentence that have the most information relevance. And later on, we will show how the performance of the sugessted model compare to other translation systems.

Finally, we will discuss the benefits of such method, some of the obstacles that the neural machine translation still face, and what room is there for further improvement.

2

Formal Notations

- **NN** : Neural Network.
- **RNN** : Recurrent Neural Network.
- **BiRNN** : Bidirectional Recurrent Neural Network.
- **PBMT** : Phrase Based Machine Translation.
- **RBMT** : Rule Based Machine Translation.
- **SMT** : Statistical Machine Translation.
- **NMT** : Neural Machine Translation.
- **LSMT** : Long Short Term Memory.

3 Related Work

3.1 Rule based machine translation

One of the oldest paradigms to machine translation is rule based machine translation or RBMT for short. This model relies on linguistic informations about the source and target languages. These linguistic information are found in either uni-, bi-, or multilingual dictionaries, which cover the main semantic, morphological, and syntactic regularities of each language. Based on these rules a translation for the input sentence is generated. A particular well performing MT system relying on rule based translation is called Apertium [FGRN$^+$11]

With such approaches it is well within reach to create translations for any language pair that have no common texts. A second general advantage that is offered, is domain independency. Seeing that the rules are mainly linguistic, they are not related to any specific domain. Thus it is "usable" for a wide range of texts.

The main challenge, that the mentioned method faces, is the lack of availability of good dictionaries. In addition building a new one can be highly expensive. Moreover the RBMT model is composed of multiple components, each responsible for a specific task. For example in Apertium [FGRN$^+$11], lexical processing, part-of-speech tagging, and structural transfer are all handeled by different parts of the system. Which stands in the way of achieving stable results.

3.2 Statistical Machine translation

A second approach that was successful in the machine translation field, is a statistical approach. It starts by deriving a statistical model out of bilingual text corpora, and then based on it, a translation is generated. In the field of SMT, a method that has yielded good results is phrase based approach [KOM03]. However this method has its own share of problems.
For example in phrase based statistical machine translation, a source sentence is segmented into several phrases, this can lead to inaccurate translations such as gender agreement, mainly when dependencies go beyond the length of the phrases.
Another problem that these systems face, is that similar to the RBMT systems, they usually are composed of several intricate subcomponents, that have to be tuned separately.

Many solutions to the long dependency problem have been proposed, in many papers ([SVL14], [Sch12]) it includes the use of neural networks.
In these propositions, however, the neural network was introduced only as a subcomponent to an already existing translation system. For instance, in [Sch12], the main task of the neural network was to compute a score for a source and target sentence and it was used as an additional attribute in the phrase based machine translation system. In [SVL14], on the other hand, the neural network is responsible for re-ranking candidate translations.

While the introduction of the neural network to translation systems has improved their performance, this answer has its own limitation, and did not address the second problem resulting from the components division.

3.3 Neural machine translation

NMT can be seen as a complete new approach when compared to the previously presented ideas in 3.1 & 3.2, with respect to structure. As mentioned before, opposed to the multitude of subcomponents that make up the statistical and rule based machine translation systems, in NMT the translation is generated by one neural network.

A basic structure, for this model, is the encoder-decoder architecture that we will go over in more detail in the next chapter 4
In [KCGB$^+$14] this architecture was extended with LSMT units, to enable the model to learn where to shift its attention in the input sentence. This architecture is the core of the Google Neural Machine Translation system. [WSC$^+$16]

In the next chapter, the typical NMT models will be presented, and the changes proposed to further improve their performance will be explained.

4 Approach

4.1 RNN-encoder-decoder design pattern

In this section we go briefly over the recurrent neural network encoder-decoder architecture. A simplified representation of it, is shown in Figure 4.1.

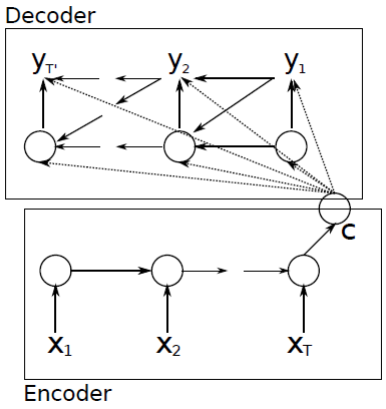

Figure 4.1: Illustratrion of the RNN Encoder-Decoder from [KCGB+14]

The RNN-encoder-decoder model, consists of two recurrent neural networks, one having the function of an encoder that maps a variable length input sentence into a fixed length vector and the second, the function of a decoder, that out of this representation generated by the encoder outputs a correct translation.

The encoder maps a consecutively a sequence of vectors $x = (x_1, \ldots, x_{T_x})$ to a vector c, where,

$$c = q(h_1, \ldots, h_{T_x}) \tag{4.1}$$

with q being a non-linear function and $h_t \in \mathbb{R}^n$ represent a hidden state at time t. Changes to these states are calculate in function of the previous hidden state and the input sequence read at time t.

The decoder, basically, generates words of the target sentence, based on both, the previously predicted target word and the vector output by the encoder.
Put differently the decoder computes the probability of a translation y by decomposing the joint probabilities into the ordered conditionals:

$$p(y) = \prod_{t=1}^{T} p(y_t|\{y_1, \ldots, y_{t-1}\}, c) \tag{4.2}$$

Each of the conditional is computed in dependence of, a hidden state s_t of the decoder, c the context vector and the last predicted word:

$$p(y_t|\{y_1, \ldots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c) \tag{4.3}$$

4.2 Learning to Align and Translate

In the work of Cho et al.(2014) [CBB16] it is speculated that the fixed-length-vector is an obstacle to further improvement. We describe in this section the changes that were brought to the architecture described in section 4.1 in order to overcome this limitation.

4.2.1 Encoder

The encoder in the proposed scheme compute a sequence of annotations. So that the annotations do not only summarize the preceding words, but the following ones as well, the use of bidirectional recurrent neural network (BiRNN) for this task was suggested in [TLL+16].

The BiRNN implements two RNNs, a forward RNN $\overrightarrow{f}$ that reads the input sequence as it was given from x_1 to x_{T_x}, and a backward one $\overleftarrow{f}$ that reads the sequence in reverse order from x_{T_x} to x_1. These RNNs respectively generate a forward and backward sequence of hidden states $\overrightarrow{h_1}, \ldots, \overrightarrow{h_{T_x}}$ and $\overleftarrow{h_1}, \ldots, \overleftarrow{h_{T_x}}$
By concatenating both the hidden states (forward and backward) at a certain index j, we obtain an annotation $h_j = [\overrightarrow{h_j^T}; \overleftarrow{h_j^T}]$ that summarizes the words in both directions of the word x_j in the sentence.
It is important to note, that due to the RNN tendency to better represent recent inputs, the annotation's h_j focus, will be shifted around the j-th word.

In the next section we will see, how these annotations in combination with an alignment model, will be used to compute a context vector by the decoder, which will help generate a translation.

4.2.2 Decoder

Since the fixed-length-vector is being removed, the conditional probability of the words to predict becomes as follows:

$$p(y_i|y_1, \ldots, y_{i-1}, x) = g(y_{i-1}, s_i, c_i) \tag{4.4}$$

Note here, that in contrast to equation 4.3, 4.4 differs at the level of the context vector used, instead of a single context vector c, for each target word y_i a unique vector c_i is computed.

The sequence of annotations $(h_j,\ldots,h_{T_x})$ output by the encoder, are summed in combination with weights α_{ij} to compute the context vector.
The weight itself, is determined based on an alignment model, that depicts how well an input and an output at certain positions (that may differ) match, and the previous hidden state s_{i-1} (state before emiting y_i)
The alignment in this model is not considered to be a latent variable, it only allows the gradient of the cost function to be backpropagated through.

4.2.3 Overall view

To summarize the context vector c_i represent the expected annotation with probability α_{ij}, that is the probability of a word y_i to be aligned with or translated from a word x_j out of the source sentence.
The i-th context vector c_i then, is the expected annotation over all the annotations with probabilities α_{ij}.

Naturally, this suggests the implementation of an attention mechanism to the decoder, and in doing so, it is not necessary anymore for the encoder to encode all the information in the source sentence. The decoder will selectively retrieve information spread throughout the sequence of annotations

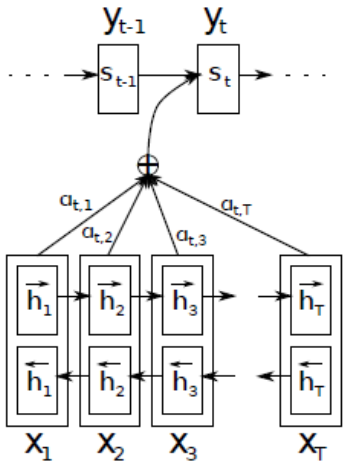

Figure 4.2: Illustration of the model generating the t-th target word from [KCGB+14]

5 Comparison

To get a clear idea on the performance improvement that the changes to the model have brought, we compare the BLEU scores[1] of the model presented in [CBB16] and a previous one that uses fixed-length-vectors [CvMB4b].
Both these models rely on the encoder-decoder architecture. In the first [CvMB4b] the encoder outputs a fixed-length-vector (referred to as; RNNenc), and the second [CBB16] uses the joint alignment and translation approach (referred to as; RNNsearch) The difference lies at the levels of the encoder output, and at the level of the decoder when processing this representation of the source sentence.

5.1 Experiment Setting

5.1.1 Data Set

The data set, used to conduct the test, is the bilingual parallel corpora provided by ACL WMT'14[2] for the language pair English-French. It contains Europarl (61M words), news commentary (5.5M), UN (421M), and two crawled corpora (of 262.5M words combined), totaling 850M words.
A method, by Axelrod [Axe14] is then used to reduce the data size to 348M words.
This data set did not contain any monolingual corpus.

5.1.2 Training

The two models were trained with the help of a shortlist made of the 30.000 most frequent words in each language, that was concatenated using the tokenization process from the open source mt packages, moses.
Each of these models was trained twice, using a minibatch stochastic gradient descent algorithm with Adadelta [Zei12]. The systems were trained with corpus that contain sentences of length up to 30 words (i.e. RNNenc-30, RNNsearch-30) and once more with sentences of length up to 50 words (i.e. RNNenc-50, RNNsearch-50).

[1] A metric that evaluates Machine Translation systems [PRWZ02]
[2] https://www.statmt.org/wmt14/translation-task.html

5.1.3 Evaluation with BLEU

BLEU, which stands for Bilingual evaluation understudy, is an algorithm for automatic evaluation of machine Translation. It computes a score (between 0 and 1, with 1 being the best possible result) by considering n-grams overlap between a source and target sentence.

BLEU, as a method of evaluation, has some flaws, namely;
-Meaning of the sentence is not taken into account.
-Sentence structure is not directly considered.
-Morphologically rich languages are not handled.

Nevertheless, this method is often relied upon, because of its ubiquitousness. More precisely, this means that the algorithm is helpful when comparing two different models of translation on the same task.

5.2 Results

5.2.1 Quantitative analysis

The results of the test can be seen on the graph in Figure 5.1. In this instance the test was conducted on the full test set which include sentences having unknown words to the model
The improvement is clear, especially, in the second case where the model presented in section4.2 is trained with longer sentences.

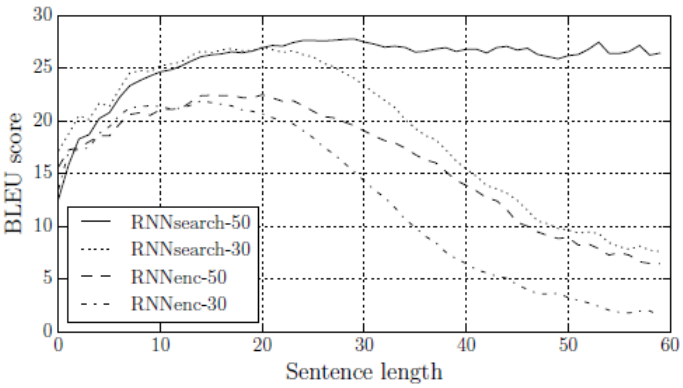

Figure 5.1: The BLEU scores of the generated translations on the test set with respect to the lengths of the sentences from ([CBB16])

In Table 5.1 the scores are presented again, this time for both instances, namely, for the test set in which the sentences contain unknown words (second column) and for test set with sentences without unknown words.
We can see that the RNNsearch model outperforms, the conventional RNNenc, and even the phrase-based translation system "moses" [KHB+07], when the training period was extended until no further improvement on the developement was noticeable (noted in the table as RNNsearch-50*) and removed the option to generate [UNK] token upon encountering unknown words.

Model	All	No UNK
RNNenc-30	13.93	24.19
RNNsearch-30	21.50	31.44
RNNenc-50	17.82	26.71
RNNsearch-50	26.75	34.16
RNNsearch-50*	28.45	36.15
Moses	33.30	35.63

Table 5.1: From [CBB16]

5.2.2 Qualitative analysis

Let us now, look at some examples of a translation generated by the system:

The agreement of the European economic area was signed in August 1992.

L'accord sue la zone économique européenne a été signé en août 1992.

Although adjectives and names are ordered differently between the language pair English/French we can clearly see how the system made the correct prediction; The phrase [European economic area] was correctly translated into [zone économique européenne].

The improvement brought by the structural changes is also observable, when translation the first word of the sentence. [The] can be translated into [le], [la], [l'] or [les] in French depending on the upcoming word. In the proposed architecture the model is allowed to look at both words and then make its prediction. This method i.e. soft-alignment, is as well beneficial regarding source and target sentences with different lengths.

Here, in this second example, we consider the performance of the model in case of long sentences by putting side to side the translations generated by the conventional NMT system (RNNencdec) and by the improved model (RNNsearch).

This kind of experience is part of Disney's efforts to "extend the lifetime of its series and build new relationships with audiences via digital platforms that are becoming ever more important," he added.

Translation generated by RNNencdec:

Ce type d'expérience fait partie des initiatives du Disney pour "prolonger la durée de vie de ses nouvelles et de développer des liens avec les lecteurs numériques qui deviennent plus complexes.

Translation generate by RNNsearch:

Ce genre d'expérience fait partie des efforts de Disney pour "prolonger la durée de vie de ses séries et créer de nouvelles relations avec des publics via des plateformes numériques de plus en plus importantes", a-t-il ajouté.

In agreement with the results presented in 5.2.1, this example confirms again that the RNNsearch model is more reliable.
In the underlined part of the sentence we can notice the deterioration of the translation, one obvious mistake is the missing quotes. In addition to the wrong translation of the part [Digital platforms] to [lecteurs numérique] instead of [platform numérique], which RNNsearch has correctly predicted.

6 Discussion

Although NMT systems are relatively new to the field of machine translation, in just a few years, results comparable to PBMT state-of-the-art systems have been achieved. On some aspects this new model has even surpassed the traditional one. Nevertheless, it still has some disadvantages, and there is yet room for improvement.

6.1 Advantages of NMT

The main gain of the presented model is, as we have seen in 5.2, is the performance upgrade when translating long sentences. This is due to the fact that the sentence is no longer encoded into a fixed length vector.
A second reason, regarding the translation quality upgrade, is that in a PBMT, different components (Translation, re-ranking, etc...) are trained separately and then combined using a scheme, in which a tuning algorithm assigns a distinct weight to each component. In opposition, in NMT all components are trained jointly, maximising the system's translation performance. This in turn leads to much more stable results.

6.2 Disadvantages of NMT

One aspect, in which NMT has bad performance results, is when dealing with out-of-domain data. This means that NMT systems are outperformed by PBMT if the input to be translated, is related to a specialized domain (e.g. Legal, Finance, etc...).
This is the case as well for rare words, because of the computational restraints. Most of these systems are trained only with dictionaries containing up to 50,000 words, this results in poor translations for highly-inflected languages and domains with a lot of named entities.

An additional drawback of the NMT systems is the difficulty of debugging. In other systems (e.g. RBMT and SMT) one can trace the process that yielded a specific translation, and remediate the problem, which is not the case for neural systems.

And finally, one big challenge for the NMT systems, is the amount of training data. This should be the case for any machine learning approach, however when only a small amount of parallel texts is available the NMT system perform clearly much worse than other systems.

Bibliography

[Axe14] Amittai Axelrod. Data selection for statistical machine translation, 2014. A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, University of Washington.

[CBB16] KyungHyun Cho, Yoshua Bengio, and Dzmitry Bahdanau. NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE, 2016.

[CvMB4b] Kyunghyun Cho, Bart van Merriënboer, and Dzmitry Bahdanau. On the properties of neural machine translation: Encoder–Decoder approaches, (2014b). In Eighth Workshop on Syntax, Semanticsand Structure in Statistical Translation.

[DHHK13] Nadir Durrani, Barry Haddow, Kenneth Heafield, and Philipp Koehn. Edinburgh's machine translation systems for european language pairs, 2013. School of Informatic, University of Edinburgh Scotland, United Kingdom.

[FGRN+11] Mikel L. Forcada, Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, and Francis M. Tyers. Apertium: a free/open-source platform for rule-based machine translation, 2011.

[Gra14] Alex Graves. Generating Sequences With Recurrent Neural Networks, 2014.

[KCGB+14] Bart van Merriënboer Kyunghyun Cho, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation, 2014.

[KHB+07] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation, 2007. Proceedings of the ACL 2007 Demo and Poster Sessions, pages 177–180,.

[KOM03] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical Phrase-Based Translation, 2003. proceedings of HLT-NAACL 2003, Main Papers , pp. 48-54.

[PRWZ02] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation, 2002. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 311-318.

[Sch12] Holger Schwenk. Continuous Space Translation Models for Phrase-Based Statistical Machine Translation, 2012. Proceedings of COLING 2012: Posters, pages 1071–1080.

[SUIK07] Michel Simard, Nicola Ueffing, Pierre Isabelle, and Roland Kuhn. Rule-based translation with statistical phrase-based post-editing, 2007. School of Informatic, University of Edinburgh Scotland, United Kingdom.

[SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to Sequence Learning with Neural Networks, 2014.

[TLL+16] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation, 2016. Department of Computer Science and Technology, Tsinghua University, Beijing, Noah's Ark Lab, Huawei Technologies, Hong Kong.

[WSC+16] Yonghui Wu, Mike Schuster, Zhifeng Chen, Mohammad Norouzi Quoc V. Le, yonghui, schuster, zhifengc, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, 2016.

[Zei12] Matthew D. Zeiler[1,2]. Adadelta: An adaptive learning rate method, 2012. [1]Google Inc., USA [2]New York University, USA.