# Recurrent Neural Network based Language Models for Spelling and Grammatical Error Correction

Sina Ahmadi
Paris Descartes University

## Objectives

- Building recurrent neural networks models for the task of error correction
- Investigating sequence-to-sequence models
- Applying attention mechanism to the correction model
- Demonstrating that neural machine translation models can be successfully applied to any language

## Introduction

**Automatic spelling and grammar correction** is the task of automatically correcting errors in written text.

- This cake is *basicly* sugar, butter, and flour. [→ basically]
- *i*'m *entirely* awake. [→ {I, wide}]

In traditional methods, correction systems were modeled based on the linguistic nature of errors. However, because of the notorious complexity and irregularity of human language, more models that performed better were needed. **Statistical Machine Translation** systems have been used successfully in this context, in particular as a result of the increasing availability of **manually annotated corpora**. More recently, the use of **neural networks** has delivered significant gains for mapping tasks between pairs of sequences. In this presentation, we are particularly interested in **recurrent neural network models** which provides a memory-like functionality by making use of all the previous inputs.

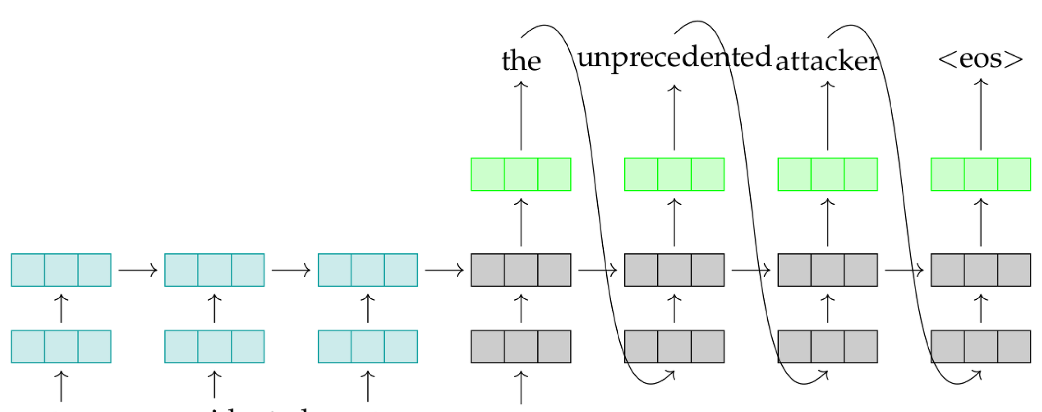



Figure: An Encoder-Decoder model

The ability to correct errors accurately will improve the reliability of the underlying applications and better processing of unedited texts on the Web.

## Neural networks as a probabilistic model

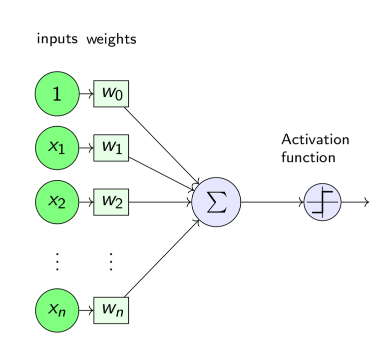


Figure: A Perceptron

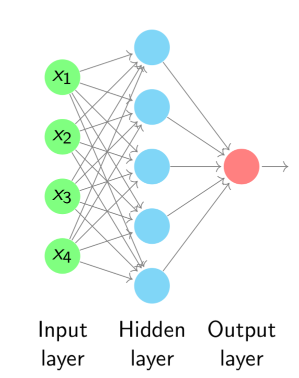


Figure: A multilayer Percepton

- Mathematical model of the biological neural networks
- Computes a single output from multiple real-valued inputs:

$$z = \sum_{i=1}^{n} w_i x_i + b \quad (1)$$

- Putting the output into a non-linear function:

$$tanh(z) = \frac{e^{2z}-1}{e^{2z}+1} \quad (2)$$

- Back-propagates in order to minimize the loss function $H$:

$$\theta^* = \underset{\theta}{argmin}\mathbf{H}(\hat{y} - y) \quad (3)$$

We assume the task of error correction as a type of **monolingual machine translation**. Therefore, given a potentially incorrect source sentence $S$ with its correction equivalent $T$, we look for the highest probable target candidate $\hat{T}$ as follows:

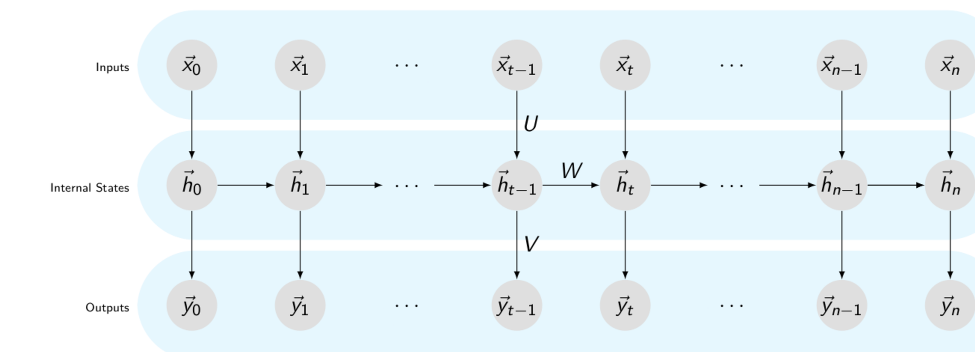
$$\hat{T} = \underset{T}{argmax} P(T|S;\theta) \quad (4)$$

## Some NLP challenges in Machine Translation

1. Large input state spaces → **word embedding**
2. Long-term dependencies
   - Constraints: **He** did not even think about **himself**.
   - Selectional preferences: I ate salad with **fork** NOT ~~rake~~.
3. Variable-length output sizes
   - This strucutre have anormality → **30** characters
   - This structure has an abnormality. → **34** characters

## Methods

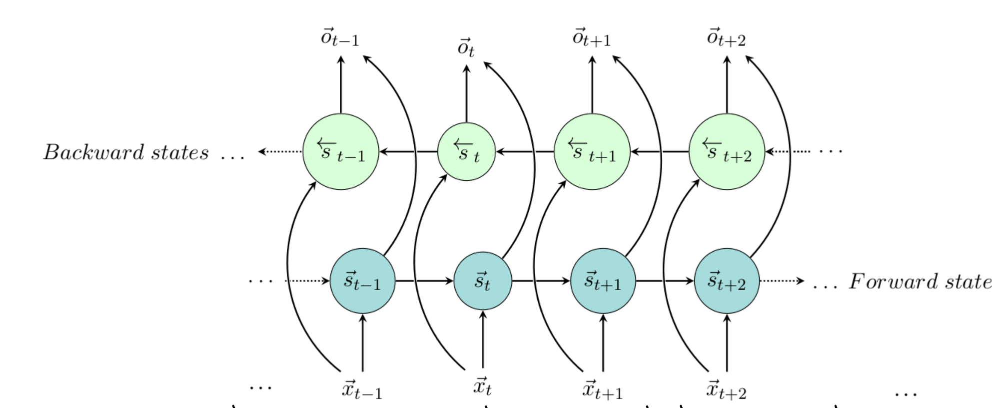
- **Recurrent Neural Network**



$$h_t = tanh(Wx_t + Uh_{t-1} + b) \quad (5)$$
$$\hat{y}_t = softmax(Vh_t) \quad (6)$$

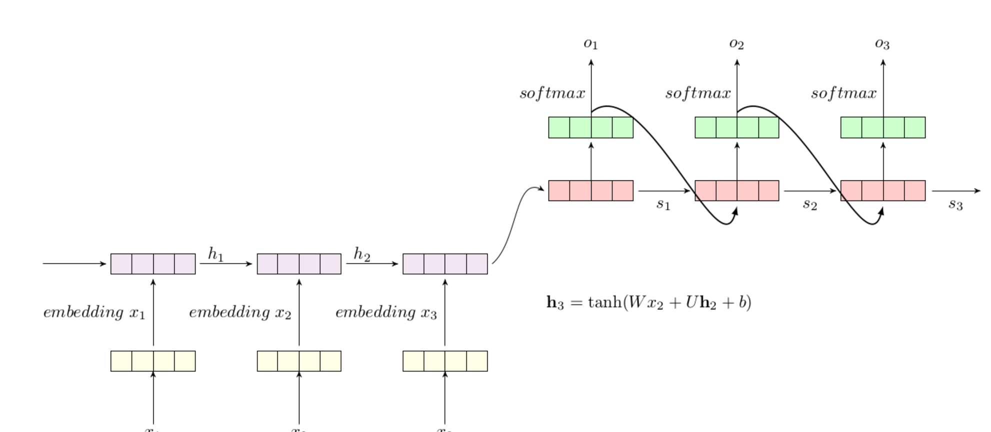
$W$, $U$ an $V$ are the parameters of our network we want to learn.

- **Bidirectional Recurrent Neural Network**



$$\overrightarrow{h_t} = tanh(\overrightarrow{W}x_t + \overrightarrow{U}\overrightarrow{h}_{t-1} + \overrightarrow{b}) \quad (7)$$
$$\overleftarrow{h_t} = tanh(\overleftarrow{W}x_t + \overleftarrow{U}\overleftarrow{h}_{t+1} + \overleftarrow{b}) \quad (8)$$

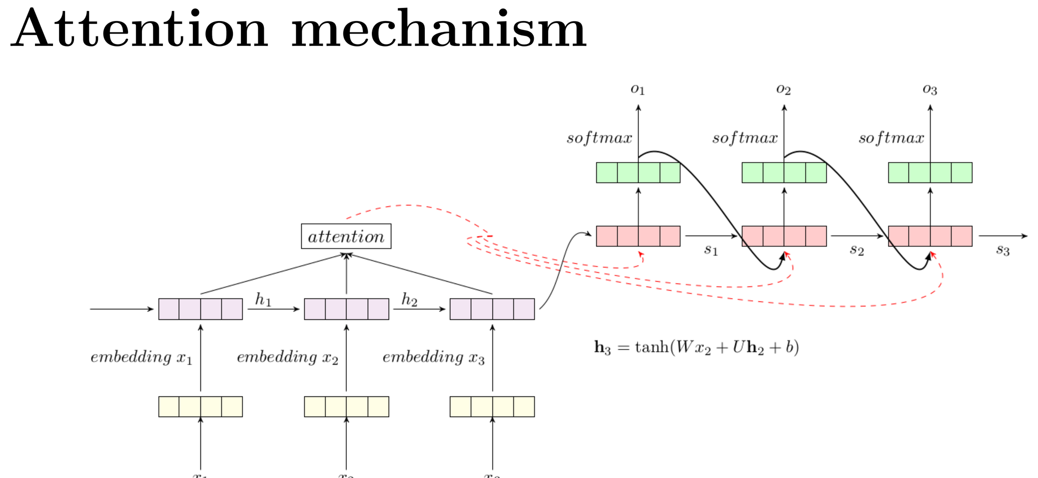
- **Sequence-to-sequence models**



$$h_t = LSTM(x_t, h_{t-1}) \quad (9)$$
$$c = tanh(h_T) \quad (10)$$

where $h_t$ is a hidden state at time $t$, and c is the context vector of the hidden layers of the encoder.

- **Attention mechanism**



$$c_t = \sum_{j=1}^{T} a_{tj} h_j \quad (11) \qquad \alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T} exp(e_{ik})} \quad (12)$$
$$e_{ij} = attentionScore(s_{i-1}, h_j) \quad (13)$$

## Experiments

In our research, we worked at character-level where each character in an input sequence is mapped to an real-valued number and then it is embedded. Our experiments on the QALB Arabic corpus containing 10 231 305 characters in its train set, shows the following results using MaxMatch $M^2$ metric:

| Model | $M^2$ scorer | | |
|---|---|---|---|
| | P | R | $F_{0.5}$ |
| Baseline | 1.0000 | 0.0000 | 0.0000 |
| RNN | 0.5397 | 0.2487 | 0.4373 |
| BiRNN | 0.5544 | 0.2943 | 0.4711 |
| Encoder-decoder | **0.5835** | **0.3249** | **0.5034** |
| Attention | 0.5132 | 0.2132 | 0.4155 |

Table: Evaluation results of the models using MaxMatch $M^2$ metric. Bold numbers indicate the scores of the best model.

## Conclusion and future studies

**Conclusion:**

- Modeling correction error for any language.
- Variant results using different metrics.
- Reducing precision in correction of long sentences.

**Future studies:**

- Models to be explored in more levels, e.g., word-level, phrase-level.
- Limiting the length of the sequences in training models.
- Using deeper networks with larger embedding size.
- Preventing over-learning of models by not training them over correct input tokens (action ="OK").

## Additional information

This poster summarizes a part of master's thesis of the author. For more details visit `http://sinaahmadi.github.io/`.

UNIVERSITÉ PARIS DESCARTES