# Accurate Supervised and Semi-Supervised Machine Reading for Long Documents

Guo Tianyi

2018.1.8

## Outline

## Outline

## Introduction

- Deep neural networks (DNNs) have provided promising results for a variety of reading comprehension and question answering tasks. (Weston et al., 2014; Hermann et al., 2015; Rajpurkar et al., 2016)
- While a basic sequence to sequence can perform these tasks (Sutskever et al., 2014), it has some disadvantages.
- The answer may be encountered early and need to be stored , leading to forgetting or corruption; attention can be added to the decoder to solve this problem (Hermann et al., 2015).
- Hierarchical reading models address this problem by breaking the document into sentences (Choi et al., 2017).

## Introduction

This paper introduced a simpler hierarchical model without this linguistic structure called Sliding-Window Encoder Attentive Reader(SWEAR), and use it as framework to explore semi-supervised learning for reading comprehension.

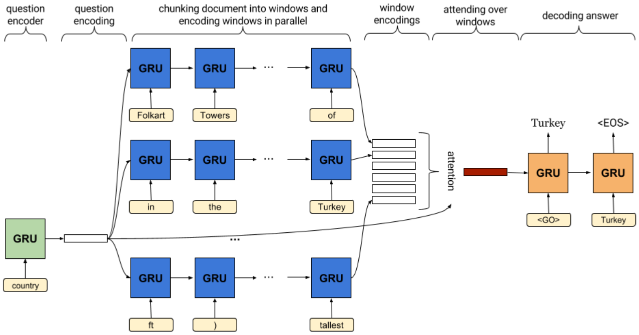


Figure: Sliding-Window Encoder Attentive Reader

## Introduction

The model SWEAR contains 4 parts:

1. **Encode question**
   Encode each question into a vector space representation.
2. **Chunk and encode document**
   Chunk each document into overlapping, fixed-length windows and, conditioned on the question representation, encodes each window in parallel.
3. **Attention**
   Attend over the window representations and reduces them into a single vector for each document
4. **Decode**
   The answer is decoded from this document vector

## Introduction

SWEAR outperforms the previous state-of-the-art on the supervised WikiReading task (Hewlett et al., 2016), improving Mean F1 to 76.8 from the previous 75.6 (Choi et al., 2017)

Constructed a **semi-supervised version of WikiReading** by downsampling the labeled corpus into a variety of smaller subsets, while preserving the full unlabeled corpus(i.e. Wikipedia).

Evaluated multiple methods of reusing unsupervised recurrent autoencoders in semi-supervised versions of SWEAR. in these models they are able to reuse all the autoencoder parameters without *fine–tuning*, meaning the supervised phase only has to learn to condition the answer on the document and query.

## Outline

## Problem Description — 5Supervised Version

$(Q, D)$ is a query–document pair, query sequence is $Q = (q_1, q_2, \cdots, q_{N_Q})$, document sequence is $D = (d_1, d_2, \cdots, d_{N_D})$, the task is to generate a new sequence of words that matches the correct answer $A = (a_1, a_2, \cdots, a_{N_A})$

- The dataset is WikiReading. Its document are Wikipedia articles and queries and answers are Wikidata properties and values.
- The dataset contains 18.58M instances divided into training, validation, and test with an 85/10/5 split.
- The answer is present verbatim in the document only 47.1% of the time, severely limiting models that label document spans.

## Problem Description — Semi–Supervised Version

In semi-supervised model, an additional corpus of documents without labeled (Q, A) pairs is available.
Taking advantage of the large size of the WikiReading dataset, we created a series of increasingly challenging semi-supervised problems with the following structure:

- **Unsupervised:** The entire document corpus (about 4M Wikipedia articles), with queries and answers removed.
- **Supervised:** Five smaller training sets created by sampling a random (1%, 0.5%, 0.1%) of the WikiReading training set, and taking (200, 100) random samples from each property in the original training set.

## Outline

## Supervised Model Architecture — Preliminaries and Notation

| Symbol | Meaning |
|---|---|
| $V$ | Vocabulary |
| $w$ | Word |
| $e_w$ | Word embedding |
| $e_D$ | Document vector sequence |
| $e_Q$ | Question vector sequence |
| $e_A$ | Answer vector sequence |

Table: Symbols

Encoder:RNN encoders(GRU)

$$h_t = f(x_t; h_{t-1}; \theta)$$

## Supervised Model Architecture — Sliding Window Recurrent Encoder

The core of the model is a sequence encoder that operates over sliding windows in a manner analogous to a traditional convolution.

Before encoding the document, it slides a window of length $l$ with a step size $s$ over the document and produce $n = \lfloor \frac{N_D - l}{s} \rfloor$ document windows.

It produces a sequence of sub-documents $(D_1, D_2. \cdots, D_n)$, each sequence contains a subsequence of $l$ words from the original document $D$.

## Supervised Model Architecture — Sliding Window Recurrent Encoder

The model encodes each window conditioned on a question encoding.

$$h^q = Enc(e^Q; \theta_Q)$$

where $h^Q$ is the last hidden state and $\theta_Q$ represents the parameters of the question encoder.

Initialized with this question encoding, the model employs another RNN to encode each document window as

$$h^w_{i,0} = h^q$$
$$h^w_i = Enc(e^{D_i}; \theta_W)$$

where $h^w_{i,0}$ is the initial hidden state, $h^w_i$ is the last hidden state, and $\theta_W$ represents the parameters of the window encoder.

## Supervised Model Architecture — Sliding Window Recurrent Encoder

SWEAR attends over the window encoder states using the question encoding to produce a single vector $h^d$ for the document, given by

$$p_i \propto \exp(u_R^T \tanh(W_R[h_i^w, h^q]))$$
$$h^d = \sum_i p_i h_i^w$$

$p_i$ is the probability window, $i$ is relevant to answering the question. $W_R$ and $u_R$ are parameters of the attention model.

## Supervised Model Architecture — Answer Decoding

Given the document encoding $h_d$, an RNN decoder (Dec) generates the answer word sequence:

$$\begin{aligned} h_0^a &= h^d \\ h_t^a &= Dec(h_{t-1}^a; \omega_A) \\ P(a_t^* = w_j) &\propto \exp(e_j^T \tanh(W_A h_t^a + b_A)) \\ a_t^* &= \arg\max_j (P(a_t^* = w_j)) \end{aligned}$$

where $h_0^a$ is the initial hidden state and $h_t^a$ is the hidden vector at time $t$. $A = \{a_1, a_2, \cdots, a_{N_A}\}$ is the sequence of answer words generated.

The training objective is to minimize the average cross-entropy error between the candidate sequence $A$ and the correct answer sequence $A$.

## Supervised Model Architecture — Supervised Results

| Model | Mean F1 |
|---:|:---:|
| Placeholder seq2seq(HE16) | 71.8 |
| SoftAttend(CH17) | 71.6 |
| Reinforce(CH17) | 74.5 |
| Placeholder seq2seq(CH17) | 75.6 |
| SWEAR(w/zeros) | 76.4 |
| SWEAR | **76.8** |

Table: Results for SWEAR compared to top published results on the WikiReading test set.

## Supervised Model Architecture — Supervised Results

| | HE16 Best | SWEAR |
|---|---|---|
| **Categorical** | **88.6** | **88.6** |
| **Relational** | 56.5 | **63.4** |
| **Date** | 73.8 | **82.5** |

Table: Mean F1 for SWEAR on each type of property compared with the best results for each type reported in Hewlett et al. (2016), which come from different models. Other publications did not report these sub-scores.

## Supervised Model Architecture — Supervised Results

| Doc length | pct | seq2seq | SWEAR | imp |
|---|---|---|---|---|
| $[0, 200)$ | 44.6 | 79.7 | 80.7 | 1.2 |
| $[200, 400)$ | 19.5 | 76.7 | 77.8 | 1.5 |
| $[400, 600)$ | 11.0 | 74.5 | 76.3 | 2.3 |
| $[600, 800)$ | 6.6 | 72.8 | 74.3 | 2.1 |
| $[800, 1000)$ | 4.3 | 71.5 | 72.8 | 1.8 |
| $[1000, max)$ | 14.0 | 64.8 | 65.9 | 1.7 |

Table: Comparison of Mean F1 for SWEAR and a baseline seq2seq model on the WikiReading test set across different document length ranges. pct indicates the percentage of the dataset falling in the given document length range. imp is the percentage improvement of SWEAR over baseline.

## Outline

## Semi-Supervised Model Architecture — Recurrent Autoencoders for Unsupervised Pre-training

In Recurrent Autoencoder (RAE), the output sequence is replaced with the input sequence, so learning minimizes the cross-entropy between the reconstructed input sequence and the original input sequence. Encoder and decoder cells share parameters $\theta_U$.

- **Variational Recurrent Autoencoder**
  The Variational Recurrent Autoencoder (VRAE), introduced by Fabius et al. (2014), is a RAE with a variational Bayesian inference step where an unobserved latent random variable generates the sequential data.
- **Window Autoencoders**
  The paper takes advantage of the SWEAR architecture by training autoencoders for text windows, as opposed to the standard document autoencoders.

## Semi-Supervised Model Architecture — Baseline: Initialization with Autoencoder Embeddings

The baseline approach to reusing an unsupervised autoencoder in SWEAR is to initialize all embeddings with the pre-trained parameters and fix them. The paper called this model SWEAR-SS (for semisupervised).

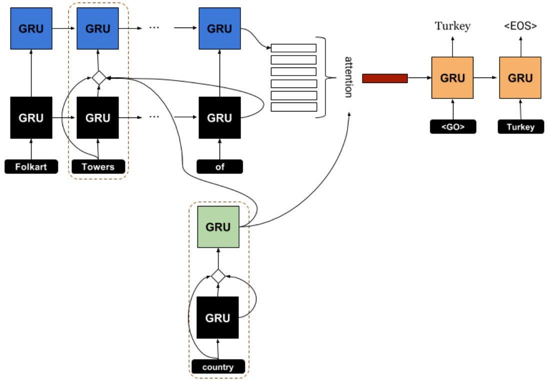
Initializing the encoders and decoder with autoencoder weights hurts performance.

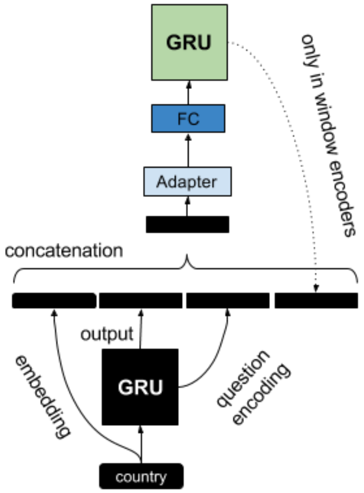
## Semi-Supervised Model Architecture — Review Models

This baseline approach to semi-supervised learning has significant disadvantages in our problem setting. The paper observed catastrophic forgetting. Second, conditioning the window encoders on the question eliminates the possibility to train window representations offline and causes a significant overhead during testing.

- **Multi-Layer Reviewer (SWEAR-MLR)**
  Fix the pretrained autoencoder RNN as the first layer and introduce a second, trainable *reviewer layer*.
- **Progressive Reviewer (SWEAR-PR)**
  Review the outputs of the encoders using a separate RNN that is decoupled from the window size.

## Semi-Supervised Model Architecture — Multi-Layer Reviewer





## Semi-Supervised Model Architecture — Multi-Layer Reviewer

The question is first encoded by the autoencoder layer

$$\tilde{h}^Q = Enc(e^Q; \theta_U)$$

where both word embeddings($E$ and $e^Q$) and encoder($\theta_U$) are fixed and initialized with pretrained parameters.
Second, learnable RNN layer then takes the output of the autoencoder layer and corresponding input embeddings as input and produces the final question encoding

$$h^q = Enc(FC([e^Q, \tilde{h}^q, \tilde{h}_t]); \theta_Q)$$

$FC$ is a fully connected layer with ReLU activation function, and $\tilde{h}_t$ is the output of the autoencoder layer at time step $t$.

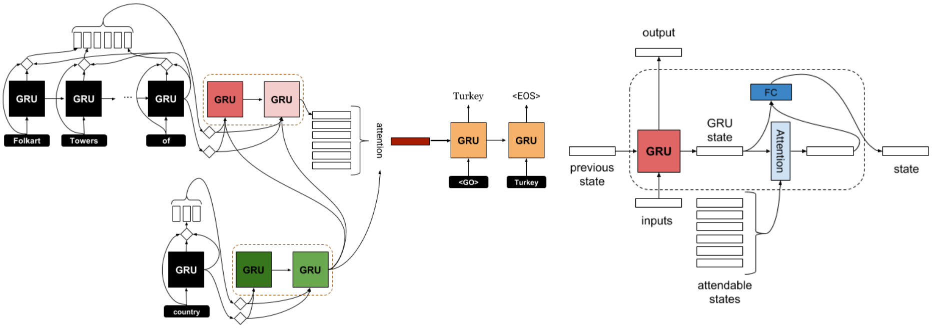
## Semi-Supervised Model Architecture — Multi-Layer Reviewer

The Similarly, windows are encoded in almost the same way, but instead of being initialized with the question encoder state, the state is an additional input to each unit in the reviewer layer.

$$\tilde{h}_i^w = Enc(e^{D_i}; \theta_U)$$

$$h_i^w = Enc(FC([e^{D_i}, \tilde{h}_i^w, \tilde{h}_t^w, \tilde{h}_q]); \theta_W)$$

## Semi-Supervised Model Architecture — Progressive Reviewer



## Outline

## Experimental Evaluation

| **Model** | **1%** | **0.5%** | **0.1%** |
|---|---|---|---|
| SWEAR | 63.5 | 57.6 | 39.5 |
| SWEAR-SS (RAE) | 64.7 | 62.8 | 55.3 |
| SWEAR-SS (VRAE) | **65.7** | **64.0** | **60.7** |

Table: Mean F1 results for SWEAR (fully supervised) and SWEAR-SS (semi-supervised) trained on 1%, 0.5%, and 0.1% subsets, respectively. Variants of SWEAR-SS indicate different sources of fixed encoder weights.

| **Model** | **100** | **200** |
|---|---|---|
| SWEAR | 25.0 | 33.0 |
| SWEAR-SS (VRAE) | 39.0 | 45.0 |

Table: Results for SWEAR and the best SWEAR-SS initialization (VRAE) trained on 100- and 200- per-property subsets, respectively.

## Experimental Evaluation

| **Model** | **Mean F1** |
|---|---|
| SWEAR-PR | **66.5** |
| dropout on input only | 65.4 |
| no dropout | 64.6 |
| shared reviewer cells | 63.8 |
| SWEAR-MLR | 63.0 |
| w/o skip connections | 60.0 |

Table: Results for semi-supervised reviewer models trained on the 1% subset of WikiReading.

## Outline

## Related Work

- Eunsol Choi, Daniel Hewlett, Alexandre Lacoste, Illia Polosukhin, Jakob Uszkoreit, and Jonathan Berant. 2017. Coarse-to-fine question answering for long document. In *Association for Computational Linguistics (ACL)*.
- Alexander Miller, Adam Fisch, Jesse Dodge, AmirHossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. *arxiv*.
- Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2440–2448. Curran Associates, Inc.
- ...