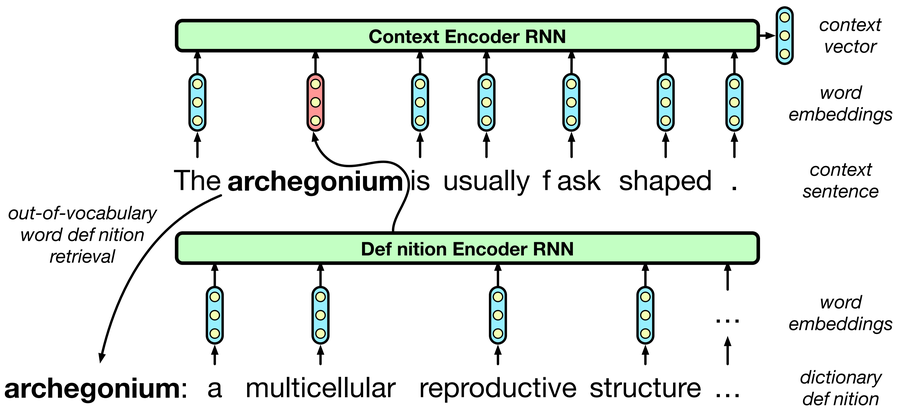
Learning to Compute Word Embeddings on The Fly

D. Bahdanau, T. Bosc, S. Jastrzebski, E. Grefenstette, P.Vincent, Y.Bengio

McGill

Introduction

The distribution of words in natural language is known to have a very long tail. Two most common approaches to deal with rare words are (a) replacing them with a special UNK token (b) using embeddings triained on vast corpora of raw text. We investigate a different approach: predicting embeddings of rare words on the fly using auxiliary information, such as e.g. spelling or (novel!) dictionary definitions.



Methods

For each word
- fetch all auxiliary information (speling, dictionary definitions, etc.)
- embed each "definition" using mean pooling or LSTM
- mean-pool all definition embeddings
- combine them with the word embedding, if available

Related Work

- Hill et al. (2016) trains a network to predict the headword embedding using embeddings of the head-word
- Long et al. (2016) use WordNet definition for knowledge base completion
- our work is different as we learn to read definitions for a downstream task in an end-to-end-way

Experiments

- we used spelling and/or definitions from WordNet
- we experimented with machine comprehension with on SQuAD, entailment recognition on SNLI and MultiNLI and language on One Billion Words dataset

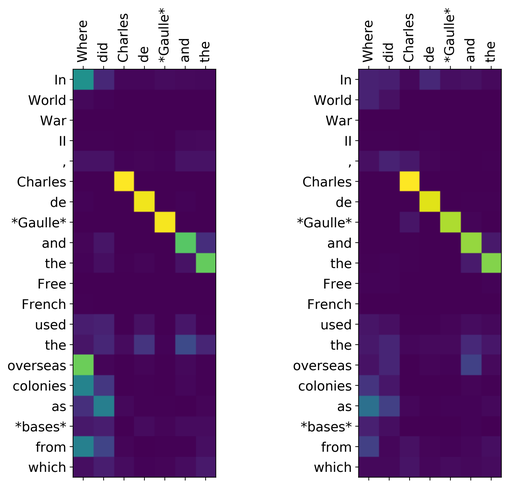
Results: Reading Comprehension

We evaluated our method using Stanford Question Answering Dataset (SQuAD), ~80k training examples. Our base model was Dynamic Coattention Network by Xiong et al. (2017).
Example definitions that helped: "overseas -> in a foreign country", "scientist -> a person with advanced knowledge of one or more sciences". Compared to GloVe, our model lacked knowledge that "historian" is a "profession", "Mark Twain" is an "author". It also could not handle the knowledge that "Earth" is a "planet", even when the definition was avavable, as "planet" was OOV for the dictionary reader.

Table 1: Exact match (EM) ratio for different models on SQuAD development and tests set. “dict” stands for dictionary, “MP” stands for mean pooling.

model	EM dev
baseline (B)	52.58
dict, MP, sum, no back-prop (D1)	56.27
dict, MP, sum (D2)	57.03
dict, MP, transform and sum (D3)	58.9
dict, LSTM (D4)	58.78
spelling (S)	61.94
spelling+lemmas (SL)	62.4
spelling+dict (SD)	63.06
GloVe (G)	64.19

Figure 2: The attention maps A_C of the models with (on the left) and without the dictionary (on the right). The rows corresponds to words of the context and the columns to the words of the question. One can see how with the help of the dictionary the model starts considering “overseas“ as a candidate answer to “where“.



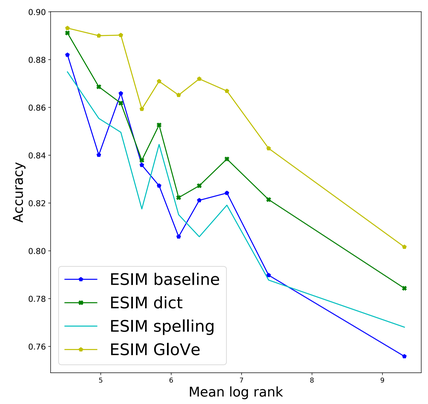
Results: Entailment Recognition

We used Stanford Natural Language Inference (SNLI) and Multi-Genre Natural Language Inference (MNLI) datasets (both around ~500k examples). Our base model is Enchanced Sequential Inference Model (ESIM). We were able to cover ~40% difference between the baseline model and the model using GloVe vectors.

	SNLI	MultiNLI	
		matched	mismatched
baseline	83.39/82.84	69.05/68.55	67.22/68.57
spelling	83.78/82.89	69.76/68.89	70.48/69.76
dict	84.88/84.39	71.39/71.45	71.65/70.7
GloVe	87.20/86.39	74.63/74.58	73.32/73.92

The embeddings computed on the fly from the dictionary by a model trained on MultiNLI:

As expected, the improvement on SNLI is higher for rare words:



Discussion

- with very little but highly relevant data (such as e.g. dictionary definitions) we can bridge the gap between training embeddings from scratch and using 840 billion words
- our method can be useful for narrow technical domains
- using external knowledge provides means of control by adding/editing definitions