Solving Biology Analogies With Deep Learning

Justin Xu, Han Lin Aung, Sajana Weerawardhena

Department of Computer Science - Stanford University

Problem

For the purposes of automating the generation of textbook learning material, we formulate a prediction algorithm to automatically answer analogy tasks in the context of a biology textbook. The problem that we attempt to solve is: given the input (a,b,c) of the analogy a:b::c:d, where a, b, c and d may either be words or phrases, predict the missing word/phrase d in the analogy.

Introduction

Analogy generation in the natural sciences, especially biology, has not been as actively researched but is a beneficial space for analogy generation especially in the realm of education. In biology, there are many relationships between different concepts represented by these terminologies. The generation of analogy can in fact greatly assist the education of those who are starting to learn biology.

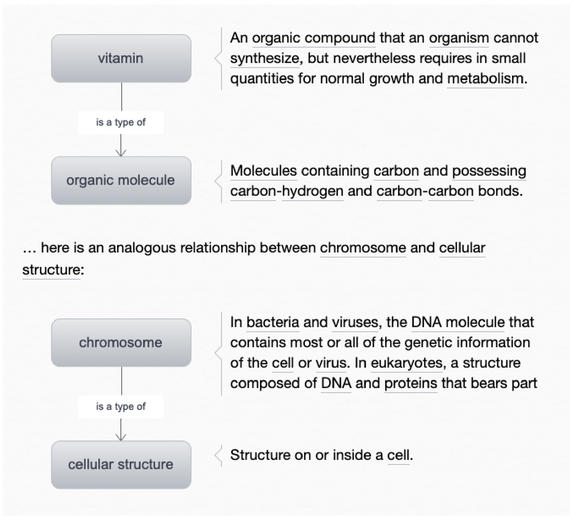

Figure 1: Analogy example

Data/Task

Given a:b::c:d, where a, b,c and d may be either words or phrases, predict the missing word/phrase d in the analogy.

Unigram Dataset:
The Unigram dataset consists of a total of 2145 single word analogies. One example analogy is mitochondrion:organelle::triose:monosaccharide as mitochondrion is a type of organelle just as triose is a type of monosaccharide.

N-gram Dataset:
The N-gram dataset consists of 77,645 analogies. One example is Carbon 14 atom:Radioactivity::C4 plant:C4 photosynthesis. A carbon 14 atom is known functionally for its radioactivity while C4 plants are known for its function of C4 photosynthesis.

Results

For the unigram dataset, we measured accuracy based on whether the top 10 predicted words appear in the gold standard and the average cosine similarity of the predicted embedding to the gold standard embedding.

	Accuracy	Cosine Sim
ELMo pretrained	**.235**	.54
Wikipedia GloVe	.105	.42
Biology GLoVe	.112	.424
Wikipedia FastText	.171	.548
Biology FastText	.0909	**.658**

Table 1: Unigram Vector Embedding scores based on accuracy and average cosine similiarity

For n-gram dataset, our accuracy measure is more fine-grained, as overall results are better. We measure if top 1, 2, 3, 4 predicted words appear in gold standard, raw accuracy score, and BLEU scores.

Approach

Vector offset model
Based on the vector offset model, we implemented analogy generation based on various pretrained embeddings and embeddings trained on the biology corpus

Seq2Seq and Seq2Vec
Implemented various flavors of seq2seq models (encoder-decoder) and seq2vec models(bi-LSTM encoder with linear layer)
Based on various flavors of seq2seq and seq2vec models, we also embedded BERT and ELMo embeddings to train our model.
Our full list of experiments are as follows:

Vector offset	Seq2Seq	Seq2Vec
GLoVe	Vanilla (with attention)	BERT
FastText	Char Encoder	bioBERT
ELMo	Char Encoder/decoder	ELMo
	Tri-daf (BiDAF extension)	
	ELMo	

Results (Cont.)

	Vanilla Seq2Seq	ELMo
Top 1 Acc	**0.502**	**0.502**
Top 2 Acc	**0.788**	0.784
Top 3 Acc	**0.930**	0.926
Top 4 Acc	**0.980**	0.968
Any Match Acc	**0.580**	0.579
Corpus BLEU	49.873	**54.08**

Table 2: Best Seq2Seq models results

	ELMo	BERT	BioBERT
Top 1 Acc	**0.507**	.277	.320
Top 2 Acc	**0.789**	.454	.516
Top 3 Acc	**0.930**	.572	.647
Top 4 Acc	**0.980**	.649	.722
Any Match Acc	**0.580**	.332	.381
Corpus BLEU	**56.35**	34.19	38.63

Table 3: Best Seq2Vec models results

Analysis

On the unigram dataset, our highest accuracy is **0.235** by ELMo vectors based on the character CNN whereas FastText scores highest in the average cosine similarity. These results show a promise in using more complex embeddings such as character and contextual embeddings instead of pure word embeddings like GLoVe.

Based on the n-gram dataset, seq2seq models and seq2vec models perform equally well on the given dataset. The best performing model based on raw accuracy and BLEU score is the model that incorporates ELMo embeddings, achieving **0.580** on accuracy and **56.35** BLEU score with the seq2vec model. What is surprising is that the vanilla seq2seq model performs better than its counterparts and even other seq2vec models including those that incorporate BERT embeddings.

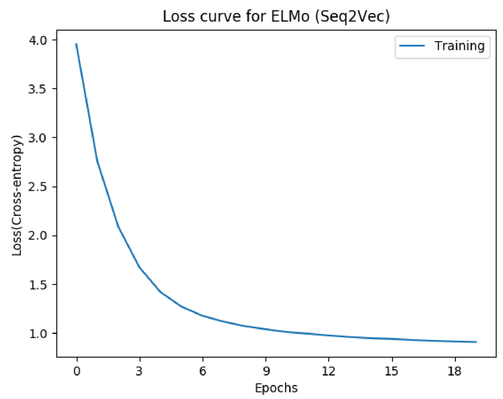

Figure 2: ELMO loss