Towards an evolutionary-based approach for natural language processing

Luca Manzoni, Domagoj Jakobovic, Luca Mariot, Stjepan Picek, Mauro Castelli

l.mariot@tudelft.nl

GECCO 2020, 8–12 July 2020
Task: given an initial sequence of $k$ words $w_1, \ldots, w_k$, complete the sentence by predicting the last word $w_{k+1}$

Exact or plausible prediction?

Original completion: table
Next Word Prediction (NWP)

- Task: given an initial sequence of $k$ words $w_1, \ldots, w_k$, complete the sentence by predicting the last word $w_{k+1}$
- *Exact* or *plausible* prediction?

Plausible prediction: *chair*

*Input*

```
the  cat  is  on  the
```
Next Word Prediction (NWP)

- Task: given an initial sequence of \( k \) words \( w_1, \ldots, w_k \), complete the sentence by predicting the last word \( w_{k+1} \).
- Exact or plausible prediction?

Plausible (?) prediction: tractor

Input:

```
the cat is on the
```
Task: given an initial sequence of \( k \) words \( w_1, \cdots, w_k \), complete the sentence by predicting the last word \( w_{k+1} \).

**Exact or plausible prediction?**

Plausible (?) prediction: tractor

We consider the setting of plausible word predictions with **Genetic Programming (GP)**.
To cast NWP as a learning task for GP we need to consider:

- **Input representation.** How can the input words be represented in a suitable way for GP?
- **Functional operators.** What operations can be performed on the representation of the words?
- **Output interpretation.** How can we decode the output of a GP individual and interpret it as a word?
GP Input: word2vec embedding

- **word2vec**: a NN-based model that learns a word embedding of a vocabulary over the vector space $\mathbb{R}^d$

![Diagram](image)

- Similar words $u, v$ are mapped to vectors $\vec{u}, \vec{v} \in \mathbb{R}^d$ with a high cosine similarity:

$$\text{sim}(\vec{u}, \vec{v}) = \frac{\sum_{i=1}^{d} \vec{u}_i \cdot \vec{v}_i}{\|\vec{u}\|_2 \cdot \|\vec{v}\|_2}$$
(1) The input words are converted to vectors through the word2vec embedding.
(2) The vectors of the input words are fed to the GP tree, and the output vector is evaluated at the root node.
(3) The output vector is converted to the most similar word occurring in the vocabulary learned by word2vec.
(4) Compute the similarity between the original (target) word and the word predicted by GP
Fitness Function

- The fitness is computed over a *training set* $S$ of sentences, all with the same number of words $k + 1$
- A *fitness case* is thus defined as a pair 
  \[ c = ((w_1, \cdots, w_k), w_{k+1}) \]
- Each word $w_i$ is represented by the vector $\vec{w}_i$ produced by the word2vec embedding
- *Fitness* of a GP individual $T$: similarity between target $\vec{w}_{k+1}$ and the output vector $\vec{p}_{k+1}$, averaged over all fitness cases
  \[ \text{fit}(T) = \frac{1}{|S|} \cdot \sum_{c \in S} \text{sim}(\vec{w}_{k+1}, \vec{p}_{k+1}) \]
Training Phase – Experimental Settings

Common Parameters:

▶ Dataset: Million News Headlines (MNH)
▶ Headlines length: 6 words (267 292 instances in MNH)
▶ word2vec embedding dimensions: \( d \in \{10, 15, 20, 25, 50, 100\} \)
▶ Training set size per GP run: 2672 (randomly selected from the 267 292 6-word headlines)

GP Parameters:

▶ Functional set: +, −, \(\times\), \(/\), \((\cdot)^2\), \(\sqrt{\cdot}\)
▶ Population size: 500 individuals
▶ Selection operator: steady-state with 3-tournament operator
▶ Mutation probability: \(p_m = 0.3\)
▶ Termination criterion: 100 000 fitness evaluations
▶ Number of independent runs: 30
Is GP learning a language model?

- Idea: compare the best GP individuals at the first and last generation, and GP with a random predictor

> **GP First/Last Generations**

> **GP vs. Random predictor**

- **Main finding**: The GP evolutionary process is able to learn, to a certain extent, a representation of the MNH dataset.
What is the influence of the word2vec embedding?

- **Idea:** compare the best GP individual with the "trivial" predictors that always generate the first or the last word.

- **GP vs. First predictor**
- **GP vs. Last predictor**

- **Main finding:** Lower embedding dimensions work better. For higher ones, the GP behavior approaches the trivial predictors.
Testing Phase – Experimental Settings

- Selected the best GP tree out of 30 runs for each dimension

|   | 10 | 15 | 20 | 25 | 50 | 100 |
|---|----|----|----|----|----|-----|
| size | 27 | 38 | 39 | 48 | 36 | 27 |

- Each selected tree was tested over a random sample of 10,000 6-words headlines from the MNH dataset

- As in the training phase, the task was to predict the sixth word by reading the first five in input

- For each sentence, we computed the similarity between the predicted and the original word
Example of tree evolved by GP

Example of best individual evolved by GP for embedding dimension $d = 10$:

\[
\begin{align*}
&+ \\
&\quad + \\
&\quad\quad + \\
&\quad\quad\quad + \\
&\quad\quad\quad\quad + \\
&\quad\quad\quad\quad\quad + \\
&\quad\quad\quad\quad\quad\quad + \\
&\quad\quad\quad\quad\quad\quad\quad \cdot^2 \\
&\quad\quad\quad\quad\quad\quad\quad w_3 \\
&\quad\quad\quad\quad\quad\quad\quad \sqrt{} \\
&\quad\quad\quad\quad\quad\quad\quad - \\
&\quad\quad\quad\quad\quad\quad\quad\sqrt{} \\
&\quad\quad\quad\quad\quad\quad\quad\times \\
&\quad\quad\quad\quad\quad\quad\quad w_2 \\
&\quad\quad\quad\quad\quad w_0 \\
&\quad\quad\quad\quad w_4 \\
&\quad\quad\quad w_1 \\
&\quad\quad w_4 \\
&\quad w_0 \\
& w_2
\end{align*}
\]
Testing Results

Distributions of similarity between predicted and original word over the test set:
Examples of test headlines completed by the best GP individual for embedding dimension $d = 10$:

| Predicted headline                              | Original            |
|------------------------------------------------|---------------------|
| Regional education to fund youth **preschool**  | allowance           |
| Aerial footage of flooded Townsville **houses** | homes               |
| Greens renew call for tax **changes**           | review              |
| Napthine to launch new Portland **rail**        | marina              |
| 4 charged over 10000 jewellery **robberies**   | heist               |
| Vanstone defends land rights act **overhaul**   | changes             |
| Community urged to seek infrastructure **funds**| funding             |
| Govt. pressured on company tax **bureaucracy**  | rates               |
| Petition urges probe into abattoir **maintenance** | closure           |
| Rain does little for central **towns**         | Victoria            |
Learning vs. Exact Prediction:

- GP usually predicts a different word than the original one
- Not necessarily a drawback: a sentence can have many different meaningful completions
- GP can navigate the word2vec embedding and predict words that are aligned with the semantics of the sentence

Dimensionality and Fitness:

- The embedding dimension has a significant impact on the GP performance: the higher the dimension, the lower the fitness
- Neural networks-based models usually employ embeddings with hundreds of dimensions
Future Directions

- Use vector-oriented operators as GP functionals (e.g., rotations)
- *Probabilistic generation*: use an ensemble of GP trees to induce a probability distribution on the word to predict
- Extend the approach to *text generation* (e.g. by using a *sliding window approach*)
- Co-evolve a population of GP *generators* and a population of GP *discriminators*, to distinguish real words from GP ones
Thank you for your attention!