CLAIMS

1. A computer-implemented method, the method comprising:

receiving a first plurality of data strings, the first plurality of data strings including a first set of positive example data strings each representative of a target data category of a named entity recognition task;

identifying a sub-string of characters from the first plurality of data strings;

obtaining a first population of candidate programs at least partially based on the first plurality of data strings, the sub-string being represented as a single unit in a candidate program of the first population of candidate programs;

generating a second population of candidate programs by conducting an iterative genetic operation on the first population of candidate programs, the iterative genetic operation including calculating a fitness score for each candidate program of the second population of candidate programs using a fitness function and the first plurality of data strings, the fitness function evaluating a match rate of a candidate program with the first plurality of data strings including evaluating a length of the candidate program with respect to an average length of all the first set of positive example data strings; and

extracting a second plurality of data strings from a data stream using a first candidate program of the second population of candidate programs.

2. The method of claim 1, wherein:

the first population of candidate programs include a first number of candidate programs;

the second population of candidate programs include a second number of candidate programs; and

the second number decreases from the first number.

3. The method of claim 2, wherein the second number decreases from the first number following one or more of an exponential decay algorithm, a linear decay algorithm, or a staggered decay algorithm.

4. The method of claim 3, further comprising setting a minimum number of candidate programs for the second population.

5. The method of claim 1, wherein:

the first plurality of data strings include a second set of negative example data strings, each of the second set of negative example data strings representative of a data category that negates the target data category; and

the fitness function evaluates a first number of positive example data string of the first set of positive example data strings that is exactly matched by a candidate program and a second number of negative example data string of the second set of negative example data strings that is exactly matched by the candidate program.

6. The method of claim 1, wherein:

the first plurality of data strings include a second set of negative example data strings, each of the second set of negative example data strings representative of a data category that negates the target data category; and

the fitness function evaluates a first number of characters that a candidate program matches from the first set of positive example data strings and a second number of characters that the candidate program matches from the second set of negative example data strings.

7. The method of claim 1, further comprising:

obtaining a third plurality of data strings, the third plurality of data strings being a subset of the second plurality of data strings; and

generating a second candidate program by conducting the iterative genetic operation on the second population of candidate programs using the third plurality of data strings.

8. The method of claim 7, wherein:

the first candidate program has a highest fitness score among the second population of candidate programs;

the conducting the iterative genetic operation on the second population of candidate programs using the third plurality of data strings generates a third population of candidate programs; and

the second candidate program has a highest fitness score among the third population of candidate programs.

9. The method of claim 8, wherein the second candidate program has a fitness score that is higher than a fitness score of the first candidate program calculated using at least one of the first plurality of data strings and the third plurality of data strings.

10. The method of claim 1, further comprising:

grouping the first plurality of data strings into a first group of data strings and at least one second group of data strings; and

separately conducting the iterative genetic operation on the first population of candidate programs and using each of the first group of data strings or the at least one second group of data strings.

11. The method of claim 1, wherein the iterative genetic operation includes a crossover operation and a mutation operation.

12. The method of claim 1, wherein a candidate program of each of the first population of candidate programs or the second population of candidate programs is a regular expression.

13. A system, comprising:

one or more processors; and

one or more computer-readable memories coupled to the one or more processors and having instructions stored thereon that are executable by the one or more processors to perform acts including:

receiving a first plurality of data strings, the first plurality of data strings including a first set of positive example data strings each representative of a target data category of a named entity recognition task;

identifying a sub-string of characters from the first plurality of data strings;

obtaining a first population of candidate programs at least partially based on the first plurality of data strings, the sub-string being represented as a single unit in a candidate program of the first population of candidate programs;

generating a second population of candidate programs by conducting an iterative genetic operation on the first population of candidate programs, the iterative genetic operation including calculating a fitness score for each candidate program of the second population of candidate programs using a fitness function and the first

plurality of data strings, the fitness function evaluating a match rate of a candidate program with the first plurality of data strings including evaluating a length of the candidate program with respect to an average length of all the first set of positive example data strings; and

extracting a second plurality of data strings from a data stream using a first candidate program of the second population of candidate programs.

14. The system of claim 13, wherein:

the first population of candidate programs include a first number of candidate programs;

the second population of candidate programs include a second number of candidate programs; and

the second number decreases from the first number.

15. The system of claim 13, wherein the acts include:

obtaining a third plurality of data strings, the third plurality of data strings being a subset of the second plurality of data strings; and

generating a second candidate program by conducting the iterative genetic operation on the second population of candidate programs using the third plurality of data strings.

16. The system of claim 13, wherein the acts include:

grouping the first plurality of data strings into a first group of data strings and at least one second group of data strings; and

separately conducting the iterative genetic operation on the first population of candidate programs and using each of the first group of data strings or the at least one second group of data strings.

17. A non-transitory computer readable storage medium storing computer program instructions capable of being executed by a computer processor, the computer program instructions defining acts of:

receiving a first plurality of data strings, the first plurality of data strings including a first set of positive example data strings each representative of a target data category of a named entity recognition task;

identifying a sub-string of characters from the first plurality of data strings;

obtaining a first population of candidate programs at least partially based on the first plurality of data strings, the sub-string being represented as a single unit in a candidate program of the first population of candidate programs;

generating a second population of candidate programs by conducting an iterative genetic operation on the first population of candidate programs, the iterative genetic operation including calculating a fitness score for each candidate program of the second population of candidate programs using a fitness function and the first plurality of data strings, the fitness function evaluating a match rate of a candidate program with the first plurality of data strings including evaluating a length of the candidate program with respect to an average length of all the first set of positive example data strings; and

extracting a second plurality of data strings from a data stream using a first candidate program of the second population of candidate programs.

18. The storage medium of claim 17, wherein:

the first plurality of data strings include a second set of negative example data strings, each of the second set of negative example data strings representative of a data category that negates the target data category; and

the fitness function evaluates a first number of positive example data string of the first set of positive example data strings that is exactly matched by a candidate program and a second number of negative example data string of the second set of negative example data strings that is exactly matched by the candidate program.

19. The storage medium of claim 17, wherein:

the first plurality of data strings include a second set of negative example data strings, each of the second set of negative example data strings representative of a data category that negates the target data category; and

the fitness function evaluates a first number of characters that a candidate program matches from the first set of positive example data strings and a second number of characters that the candidate program matches from the second set of negative example data strings.

20. The storage medium of claim 17, wherein the acts include:

obtaining a third plurality of data strings, the third plurality of data strings being a subset of the second plurality of data strings; and

generating a second candidate program by conducting the iterative genetic operation on the second population of candidate programs using the third plurality of data strings.