

УДК 004.81  
АВТОМАТИЧЕСКАЯ ГЕНЕРАЦИЯ ЭКВИВАЛЕНТНЫХ ТЕКСТОВ

Е. Д. ГУМЕННИКОВ

Научный руководитель И. А. МУРАШКО, д-р техн. наук, доц.  
Гомельский государственный технический университет им. П. О. Сухого  
Гомель, Беларусь

Задача автоматической генерации эквивалентных текстов заключается в автоматическом создании текста на основе предоставленного образца, обладающего теми же семантическими и стилистическими свойствами, что и оригинал. Решение подобной задачи может быть полезно во множестве отраслей, таких, например, как разработка чат-ботов, обучающих лингвистических программ. Исходными данными для такой программы служит фрагмент текста – образец. Выходными данными же будет текст, эквивалентный предоставленному, но содержащий другие формулировки.

Решение данной задачи путем применения стандартной рекуррентной сети имеет ограничение, связанное с краткосрочностью памяти сети, в предложенной структуре же семантическая, стилистическая и другая информация о тексте оригинала и генерируемого текста кодируется в виде вектора, что позволяет обучить сеть так, чтобы она предоставляла удовлетворительные результаты.

Для решения данной задачи предлагается система, базирующаяся на применении нескольких рекуррентных нейронных сетей. В начале работы системы исходный текст по слову подается на вход первой сети, также на ее входы подаются выходы предыдущих итераций. Цель данной сети – получить вектор численных значений, характеризующий данный исходный текст. Полученный результат подается на вход простого персептрона, чтобы получить единственное значение, хранящее в себе первое слово сгенерированного эквивалента. Далее вектор-характеристика исходного текста и значение первого слова, полученное от простого персептрона, подаются на вход другой рекуррентной сети. Данная сеть на каждой итерации генерирует очередное слово и вектор-характеристику складывающегося текста. На каждой итерации происходит процесс постобучения, в этом процессе ошибкой будет считаться отклонение полученного вектора характеристик от вектора характеристик исходного текста; если же ошибка велика, начинается процесс постобучения на основе градиентного спуска. После этого итерация повторяется до приемлемого результата. Значения весов нейронов, затронутые постобучением, не сохраняются, они нужны лишь в рамках текущей генерации.

Таким образом, предложенная архитектура превосходит стандартную *RNN*-сеть, сохраняя ее простоту. За счет уменьшения общего количества нейронов достигается улучшение производительности относительно *LSTM*.

