УДК 004.853

Нейросетевая модель автоматизированного перевода

Т.А. Васяева, Ю.А. Золушкин, Т.В. Мартыненко, Е.А. Шуватова
ГОУВПО «Донецкий национальный технический университет»
vasyaeva@gmail.com, illuzium1999@gmail.com,
tatyana.v.martynenko@gmail.com, mauritia88@gmail.com

Аннотация

В статье рассмотрены подходы к реализации автоматизированных систем перевода, современные примеры таких систем и принципы их работы. Проанализированы актуальные проблемы автоматизированного перевода, показано, что нейросетевые модели имеют преимущества. Реализована нейросетевая модель автоматизированного перевода с использованием рекуррентных нейронных сетей. Для разработки и обучения нейронной сети использован язык программирования Python, Обучение модели и описанные эксперименты выполнены на корпусе данных WMT. Реализованная модель имеет архитектуру кодер-декодер.

Введение

Поскольку мир становится все более глобализированным, потребность в точном и эффективном переводе как никогда высока. Качественный перевод важен в различных областях:

1. Глобальный бизнес. В работе на международном уровне, позволяет легко переводить документы, контракты, сообщения, электронные письма, а также речь участников конференции или деловой встречи.
2. Туризм. Помогает туристам, приехавшим в другую страну ориентироваться на местности, понимая информационные объявления.
3. Онлайн-переводчики могут быстро и точно переводить тексты с сайтов, онлайн-чатов и социальных сетей.
4. Медицина. Используется при переводе медицинских документов и инструкций на разные языки.

Анализ предметной области

Исследования в области автоматизированного перевода, одного из направлений работы в области обработки естественного языка, ведутся очень давно. Первые модели и разработанные на их основе системы были реализованы еще в 1950-х годах. Решением задачи машинного перевода могут быть: правиловые системы (rule-based) (1950-е годы); статистические модели (statistics-based) (1960-2010); нейронные модели (neural-based) (2010-е годы - настоящее время).

В последние годы эффективным решением стал нейросетевой перевод, использующий возможности искусственного интеллекта. Рассмотрим некоторые из наиболее известных современных систем автоматизированного перевода и основные принципы их работы.

1. Google Translate – одна из самых популярных систем автоматизированного перевода; запущена в 2006 году и на сегодняшний день поддерживает более 100 языков; основана на статистическом подходе, также использует нейронную сеть для улучшения производительности и качества перевода.
2. Yandex.Translate – поддерживает более 90 языков и использует схожий подход, что и Google Translate, но использует свой собственный словарь, который базируется на большом количестве текстов на разных языках.
3. Microsoft Translator – поддерживает более 60 языков и также основывается на статистическом подходе. Чтобы улучшить качество перевода, использует технологию глубокого обучения и нейронные сети.
4. DeepL – относительно новая система автоматизированного перевода, запущенная в 2017 году; основана только на технологии глубокого обучения, использует нейронные сети; может переводить более чем 40 языков. В большинстве случаев перевод более качественный, в сравнении с системами, которые основываются на статистическом подходе.
5. Systran – является одной из более ранних систем автоматизированного перевода и основывается на гибридном подходе, который использует как статистические данные, так и лингвистические правила; поддерживает более 50 языков и используется многими компаниями и правительственными учреждениями для перевода текстов.

Постановка проблемы

Несмотря на хорошие результаты нейросетевого перевода в настоящее время продолжаются исследования по следующим направлениям:

1. Качество перевода. В настоящее время качество перевода нейросетевыми моделями существенно лучше, но не всегда идеально.

2. Многоязычность. Перевод на несколько языков, используя одну и ту же модель, является еще одной актуальной и нерешенной проблемой.

3. Данные. Доступность и качество параллельных текстов (текстовых данных, которые содержат пары с предложением на одном языке и его переводом на другом языке) являются довольно серьезной проблемой. Особенно если данные получены автоматически, например, субтитры или технические описания.

4. Нестабильность. Нейросетевые модели имеют тенденцию к переобучению и плохой устойчивости к изменениям в данных. Это приводит к потере качества перевода и необходимости постоянного переобучения модели.

5. Скорость. Нейросетевые модели по-прежнему обладают довольно высокими требованиями к вычислительным ресурсам и времени. Работа с такими моделями может быть крайне медленной, что не всегда приемлемо в реальном мире.

6. Локализация. Нейросетевой перевод достаточно плохо работает с сочетаниями языков, содержащих локальные диалекты или специфические термины.

В связи с этим, дальнейшие исследования в области построения языковых моделей и реализации на их основе систем автоматизированного перевода являются актуальными и перспективными.

Постановка задачи

Задача машинного перевода формально описывается так: у нас есть *source* предложение – это предложение на языке, с которого мы должны перевести и *target* предложение – это предложение на языке, на который мы должны привести.

$$source = x_1, x_2, \dots, x_n\ ,$$
$$target = y_1, y_2, \dots, y_n.$$

Задача машинного перевода найти наиболее вероятную последовательность на *target* языке при условии входящей последовательности на *source* языке.

$$\widehat{target} = \underset{target}{argmax}\, P(target\ |source, \theta).$$

Target предсказанной модели перевода это argmax по всем *target* то есть по всем переводам, которые вообще могут соответствовать предложению *sourc*e. То есть самый вероятный перевод предложения:

$$P(target \mid source\ ) = P(y_1, y_2 \dots y_m|source) =$$
$$P(y_1\ |source\ ) * P(y_2|y_1, source) \dots$$
$$P(y_m|\ y_1, \dots, y_m - 1, source)$$

Необходимо получить переводы *source* предложения на *target* язык, который может сгенерировать наша модель и выбирать из них с максимальной вероятностью. То есть тот, который вероятнее всего является действительно переводом предложения *source* на *target*.

Подготовка набора данных

Одним из основных аспектов, влияющих на качество моделей машинного перевода, является выбор набора данных (датасета). Рассмотрим наиболее современные и открытые датасеты параллельных текстов, использующиеся при создании моделей обработки естественного языка (табл. 1).

Таблица 1 - Сравнительные характеристики современных датасетов параллельных текстов

Название датасета	Разработчик	Объем
WikiMatrix	Meta Research	135 миллионов параллельных предложений
CCMatrix	Meta Research	4.5 миллиарда параллельных предложений
News-Commentary	Barrault et al.	109 текстов
WMT	Bojar et al.	42.3 миллионов параллельных предложений
MASSIVE	Amazon	1 миллион текстов

Следует отметить, что нацеленность датасета MASSIVE [1] на парадигму MMNLU (massively multilingual natural-language understanding) делает набор данных чрезмерно тяжеловесным.

WikiMatrix [2] наиболее часто применяются при обучении систем машинного перевода между разными языками без необходимости переходить на английский язык. WMT [3] является наиболее часто используемым для построения моделей машинного обучения, содержит тексты на русском и английском языках, и использован для исследований в данной работе.

Разработка нейросетевой модели перевода

Перед разработкой модели, необходимо выполнить следующие этапы: токенизация; нормализация; фильтрация; векторизация.

В [4] последовательно рассмотрены перечисленные этапы обработки естественного языка, выполнено их описание, реализация и тестирование. Также в статье [4] проведен анализ методов преобразования слов, таких как Embedding и One-hot-encoding. Для решения задачи нейронного переводчика было выбрано векторное представление слов Embedding.

Получение векторных представлений слов, на корпусе параллельных данных WMT 2020, с использованием word2vec описано в [5]. В работе [5] выполнена реализация на языке программирования Python с использованием облачного сервиса на основе Jupyter Notebook – Google Colab.

Нейросетевая модель для перевода, это языковая модель с ограничениями, т.е. модель не просто генерирует нормальные предложения, а они являются переводом конкретного предложения. Наиболее распространенными моделями являются модели кодер-декодер (рис. 1), которые обычно используют рекуррентную нейронную сеть (RNN).

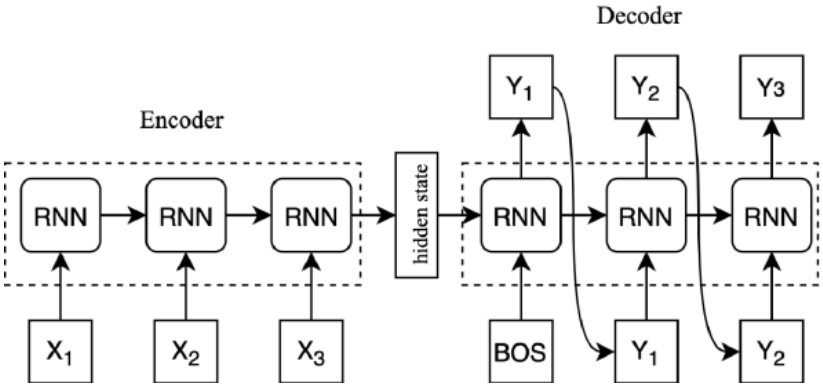

Рисунок 1 – Sequence-to-sequence модель для машинного перевода

Нейросетевая модель для перевода состоит из двух частей: энкодера и декодера. Энкодер (нейросетевая языковая модель) собирает информацию о предложении на source языке. Декодер использует информацию, собранную энкодером, чтобы сгенерировать предложение-перевод на target языке.

Входное/исходное предложение X вводится в кодер по одному слову за раз. Необходимо добавить маркеры начала последовательности (<sos>) и конца последовательности (<eos>) к началу и концу предложения соответственно. На каждом шаге входом в кодер RNN является как текущее слово, x_t, так и скрытое состояние из предыдущего шага, h_{t-1}, и кодер RNN выводит новое скрытое состояние h_t. Скрытое состояние в упрощенном понимании является векторным представлением предложения. RNN можно представить как функцию как от x_t, так и от h_{t-1}:

$$h_t = EncoderRNN(x_t, h_t - 1).$$

В качестве RNN может быть выбрана любая рекуррентная архитектура, такая как LSTM [6] или GRU.

$$X = \{x_1, x_2, \dots, x_T\},$$

где x_1=<sos>, x_2 = первое слово в предложении, и т. д.

Начальное скрытое состояние, h_0, обычно либо инициализируется нулями, либо значением, полученным при обучении.

Как только последнее слово x_T передано в RNN, скрытое состояние h_T используется в качестве вектора контекста, то есть $h_T = z$. Это векторное представление всего исходного предложения.

После получения вектора контекста z, необходимо начать его декодирование, чтобы получить целевое предложение. Как и в описанном выше случае, необходимо добавить маркеры начала и конца последовательности к предложению. На каждом шаге входом в декодер RNN является текущее слово y_t, а также скрытое состояние предыдущего шага s_{t-1}, где начальное скрытое состояние декодера s_0 представляет собой вектор контекста, $s_0 = z = h_T$, т. е. начальное скрытое состояние декодера является конечным скрытым состоянием кодера. Таким образом, подобно кодеру, декодер можно представить как:

$$s_t = DecoderRNN(y_t, s_t - 1).$$

В декодере нужно перейти от скрытого состояния к реальному слову, поэтому на каждом шаге необходимо использовать st для предсказания (пропуская его через линейный

слой), что является следующим словом в последовательности $\hat{y}_t$.

$$\hat{y}t = st.$$

Маркер <sos> используется для первого ввода в декодер y_1. Для последующих вводов $y_{t>1}$, в некоторых случаях необходимо использовать следующее слово в последовательности y_t, а иногда использовать слово, предсказанное декодером $\hat{y}_{t-1}$.

Данный поход соответствует обучению с подкреплением (teacher forcing).

Параметры для обучения нейронной сети приведены в табл. 2.

Реализация структуры нейронной сети на языке python представлена на рис. 2. Результаты обучения представлены на рисунках 3 и 4.

Таблица 2. – Параметры обучения нейронной сети seq2seq

Название параметра		Значение
Оптимизатор		Adam
Количество эпох		40
Критерий (loss)		Кросс энтропия
Размер батча		64
Размер эмбединга	кодера	256
	декодера	256
Размер скрытого слоя		512
Дропаут	кодера	0.5
	декодера	0.5
Коэффициент обучения с подкреплением		0.5

```
Seq2Seq(
  (encoder): Encoder(
    (embedding): Embedding(148755, 256)
    (rnn): LSTM(256, 256, num_layers=2, dropout=0.5, bidirectional=True)
    (dropout): Dropout(p=0.5, inplace=False)
  )
  (decoder): Decoder(
    (embedding): Embedding(75949, 256)
    (rnn): LSTM(256, 512, num_layers=2, dropout=0.5)
    (out): Linear(in_features=512, out_features=75949, bias=True)
    (dropout): Dropout(p=0.5, inplace=False)
  )
)
```

Рисунок 2 – Структура нейронной сети

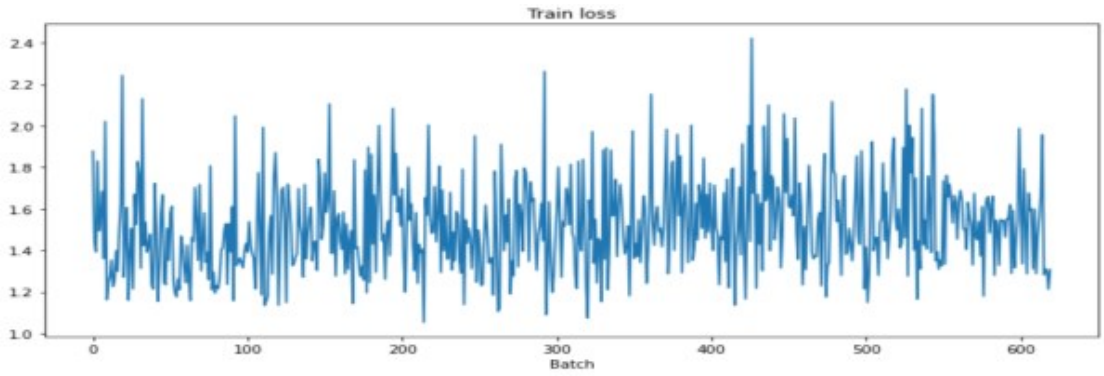

Рисунок 3 - График изменения ошибки во время обучения нейронной сети

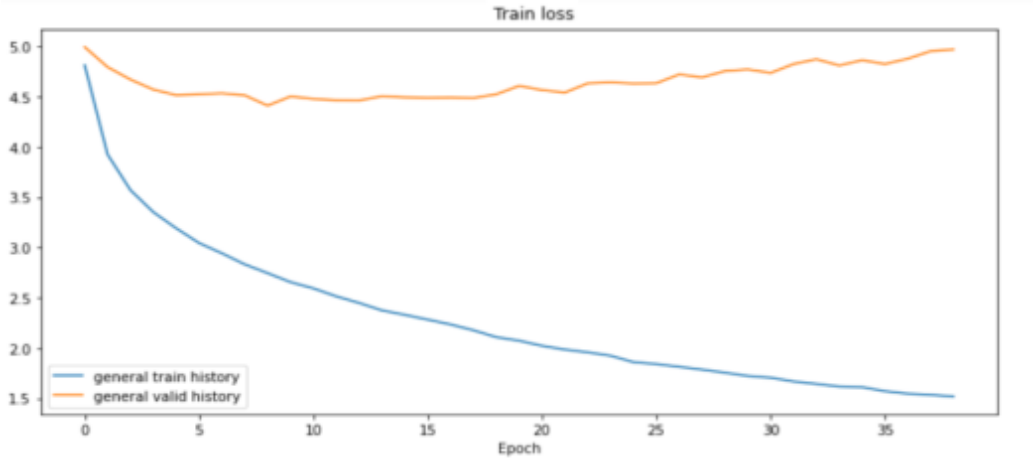

Рисунок 4 - График изменения ошибки обучения от количества эпох

Оценка качества полученного перевода осуществлялась с использованием метрики BLEU [7].

Заключение

Результатом работы является разработанная нейросетевая модель автоматизированного перевода. Выполнена математическая постановка задачи к переводу текстов. Выбран корпус данных для обучения модели. В работе использованы реализованные ранее [4] этапы обработки текста и реализация модуля получения векторных представлений слов [5]. Разработан модуль нейросетевого перевода, обучена модель для перевода текстов.

Литература

1. MASSIVE: A 1M-Example Multilingual Natural Language Understanding Dataset with 51 Typologically-Diverse Languages – Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, Prem Natarajan [Электронный ресурс]. – Режим доступа: https://arxiv.org/abs/2204.08582

2. WikiMatrix: Mining 135M Parallel Sentences in 1620 Language Pairs from Wikipedia –вы, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, Francisco Guzmán [Электронный ресурс]. – Режим доступа: https://arxiv.org/abs/1907.05791

3. Findings of the 2020 Conference on Machine Translation (WMT20) – Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz [Электронный ресурс]. – Режим доступа: https://www.statmt.org/wmt20/pdf/2020.wmt-1.pdf

4. Золушкин, Ю. А. Обработка естественного языка / Ю. А. Золушкин, Т. А. Васяева, А. А. Малицкая // Информатика, управляющие системы, математическое и компьютерное моделирование (ИУСМКМ-2021): Материалы XII Международной научно-технической конференции в рамках VII Международного Научного форума Донецкой Народной Республики к 100-летию ДонНТУ, Донецк, 26–27 мая 2021 года. – Донецк: Донецкий национальный технический университет, 2021. – С. 71-78.

5. Разработка векторных представлений слов для нейросетевой языковой модели / Ю. А. Золушкин, Т. А. Васяева, Т. В. Мартыненко, Е. А. Шуватова // Информатика, управляющие системы, математическое и компьютерное моделирование (ИУСМКМ-2022): Материалы XIII Международной научно-технической конференции в рамках VIII Международного Научного форума Донецкой Народной Республики, Донецк, 25–26 мая 2022 года. – Донецк: Донецкий национальный технический университет, 2022. – С. 219-223.

6. Manaswi N.K. RNN and LSTM. In: Deep Learning with Applications Using Python. Apress, Berkeley, CARecurrent Neural Networks(2018)

7. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W. J., BLEU: a method for automatic evaluation of machine translation (PDF). ACL-2002: 40th Annual meeting of the Association for Computational Linguistics. pp. 311–318.

Васяева Т.А., Золушкин Ю.А., Мартыненко Т.В., Шуватова Е.А. Нейросетевая модель автоматизированного перевода. *В статье рассмотрены подходы к реализации автоматизированных систем перевода, современные примеры таких систем и принципы их работы. Проанализированы актуальные проблемы автоматизированного перевода, показано, что нейросетевые модели имеют преимущества. Реализована нейросетевая модель автоматизированного перевода с использованием рекуррентных нейронных сетей. Для разработки и обучения нейронной сети использован язык программирования Python, Обучение модели и описанные эксперименты выполнены на корпусе данных WMT. Реализованная модель имеет архитектуру кодер-декодер.*

Ключевые слова: *автоматизированный перевод, нейросетевая модель, кодер-декодер, рекуррентные нейронные сети, глубокое обучение, машинное обучение, обучение с подкреплением, параллельные тексты, Python*

Vasyaeva T.A., Zolushkin Yu.A., Martynenko T.V., Shuvatova E.A. Neural network model of automated translation. *The article discusses approaches to the implementation of automated translation systems, modern examples of such systems and the principles of their work. The actual problems of automated translation are analyzed, it is shown that neural network models have advantages. Implemented neural network model of automated translation using recurrent neural networks. The programming language Python was used to develop and train the neural network. Model training and the described experiments were performed on the WMT data corpus. The implemented model has an encoder-decoder architecture.*

Keywords: *automated translation, neural network model, encoder-decoder, recurrent neural networks, deep learning, machine learning, reinforcement learning, parallel texts, Python*

Статья поступила в редакцию 25.02.2023
Рекомендуется к публикации профессором Скобцовым Ю.А.