Федеральное государственное бюджетное образовательное учреждение высшего образования

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**Департамент анализа данных и машинного обучения**

Пояснительная записка к курсовой работе

по дисциплине “Технологии анализа данных и машинного обучение”

на тему:

«Машинное обучение в задачах машинного перевода»

Выполнила:

студентка группы ПИ19-1в факультета информационных технологий и анализа больших данных

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Ким Е. А.

Научный руководитель:

ст.преп., Милованов Д.М.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2022 г

ОГЛАВЛЕНИЕ:

[ВВЕДЕНИЕ 3](#_Toc103996332)

[ОСНОВНАЯ ЧАСТЬ 4](#_Toc103996333)

[Глава 1. Краткая история машинного перевода и её развитие 4](#_Toc103996334)

[Глава 2. Предварительная подготовка текстовых данных 7](#_Toc103996335)

[Глава 3. Уменьшение кол-ва примеров и разделения на примеры для обучения и примеры для валидации и токенизация 9](#_Toc103996336)

[Глава 4. Создание алгоритма обучения модели 11](#_Toc103996337)

[Глава 5. Оценка модели c помощью BLEU 12](#_Toc103996338)

[Глава 6. Нормализация данных и работа алгоритма регрессии на них Глава 7. Стандартизация данных и работа алгоритма регрессии на них 12](#_Toc103996339)

[ЗАКЛЮЧЕНИЕ 13](#_Toc103996340)

# ВВЕДЕНИЕ

Коммуникация является одной из самых важных сфер общественной жизни человека, в наше время есть огромное количество способов связи с людьми, но до сих пор самым большим ограничением является языковой барьер и, хотя людьми были придуманы способы минимизировать влияние языка, но свести к нулю до сих пор не удалось. Язык любой страны развивается точно так же, как и технологии и по этой причине в наше время не существует алгоритма или искусственного интеллекта, которой мог бы развиваться понимать языки и помогать настолько эффективно, что людям бы перестало необходимо изучать речь другого государства. Цель данной курсовой работы разработать модель обучения машинного перевода для понимания машинного обучения в задачах машинного перевода.

# ОСНОВНАЯ ЧАСТЬ

## Глава 1. Краткая история машинного перевода и её развитие

На рисунке 1 изображено как развивался машинный перевод. Системы перевода до SMT рассмотрим очень коротко, так как они крайне неэффективны и были очень редко используемы.

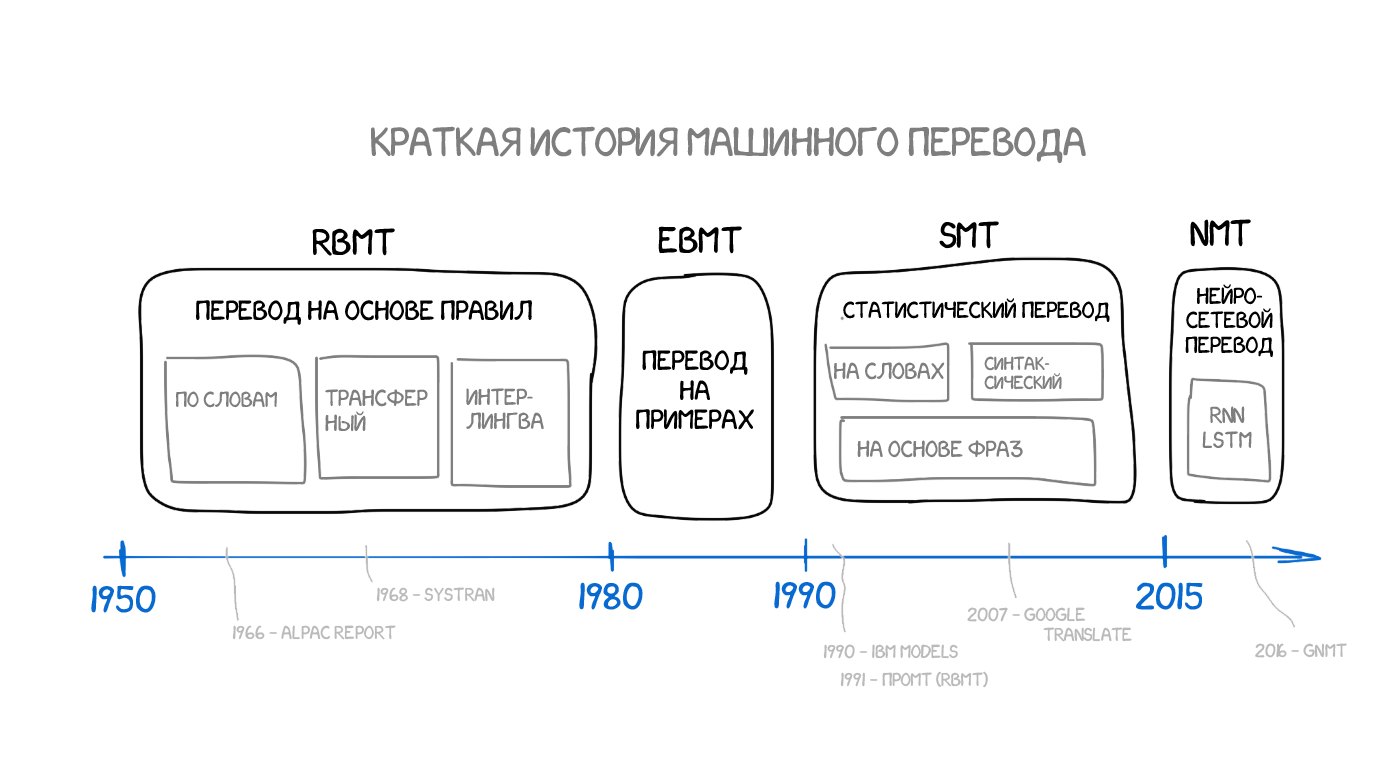


Рисунок 1. Краткая история машинного перевода

Идеи машинный перевода на основе правил (Rule-based Machine Translation) начали появляться в 1970 годах. Promt и Systran — одни из самыъ известных RBMT-систем.

Типы RBMT систем:

1. Системы дословного перевода – как можно понять, самый простой способ перевода, берём слово, переводим его и вставляем в предложение;
2. Трансферные системы – в этих системах мы сначала разбиваем предложение на подлежащее, сказуемые и так далее, и рассматриваем конструкции, а не каждое слово по отдельности. Проблемы данной системы является количество сочетаний слов, которых намного больше;
3. Интерлингвистические системы – это система, основанная на идеи самого Декарта, а именно промежуточное представление, которая едина для всех языков, названная интерлингва. Вручную создать интерлингву, которая подходила бы для всех языков до сих не являются осуществимым, хотя данную тему развивали многие учёные.

Машинный перевод на примерах (Example-based Machine Translation) – это перевод при уже существующих примерах, и простой замены в примерах, например, при переводе фразы “Я хочу спать” (“I want to sleep”), если у нас есть пример в нашем словаре, так как “Я хочу есть” (“I want to eat”), мы просто подменим слово eat на sleep и получим перевод. Такой перевод требует большого словаря примеров, и чем больше примеров, тем лучше выйдет перевод.

Перейдём наконец к SMT или же статистический машинный перевод (Statistical Machine Translation). Впервые SMT появился в IBM и именно там он получил наверно наибольше развитие. При переводе машина ничего не знала о лингвистике и правилах. Идея данного перевода была проста. Берём одно и то же предложение на двух языках, после этого разбиваем на слова и сопоставляем, повторяем эти действие огромное количество раз, а в это время программа считает сколько раз, например, слово яблоко переводилось как apple. В данном случае, мы не задавил не словари не правила, машина сама разобралась, руководствуясь статистикой. Такой перевод был очень эффективным и самое главное не требовал лингвистов, проблема такого перевода требовалась невероятное количество учебного материала, нужно огромное кол-во предложений на двух языках, что бы программа поняла, что яблоко в предложении это apple, а не tree, например.

Первые системы SMT начали переводить по словам, потому что всегда нужно начинать с чего-то простого, а уже постепенно усложнять. Первую систему IBM назвали IBM Model 1, и она напрямую разделяла предложения на слова, считала статистику и переводила слова. Во второй версии же появился учёт порядка слов в предложении и Model 2 запоминала наиболее популярный порядок слов в предложении. Иногда бывало, что при переводе не хватало слов, и для этого появилась 3-ая модель, которая проставляла нужные слова в пустые места перед этим вставив в промежуток значение NULL. Последующие модели уже дорабатывались и не привносили какие-то новые важные изменения. Последней моделью стала IBM Model 5. Сейчас уже такие системы не в обиходе, так как им на смену пришла более продвинутая система, Статистический перевод по фразам (Phrase-based SMT). В данной системы за основу взяли все принципы SMT систем, только для обучения использовали не слова, а фразы, хотя вернее сказать наборы пересекающихся слов подряд, так же известные как N-граммы. Программа уже училась переводить не слова, а их сочетания, это сильно помогло в точности перевода.

Нейронный машинный перевод (Neural Machine Translation) – прорыв в машинном переводе. В данном переводе предполагалось использовать нейросети глубокого обучения к машинному переводу. В 2016 году блоге Google появилась статистика качества перевода между Phase-based SMT, нейронным машинным переводом и обычным человеческим.

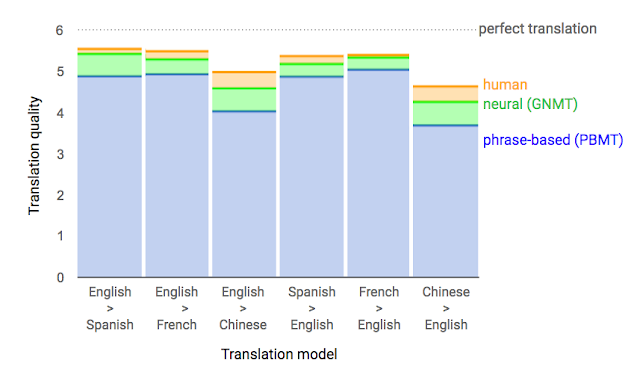


Рисунок 2. Статистика качества перевода

## Глава 2. Нейронный машинный перевод

Нейронный машинный перевод получил большую популярность в настоящее время, и уже на полную скорость внедряется в Google Translate, для выдачи перевода. Для перевода сначала использовались рекуррентные нейронные сети (RNN), но так как она запоминала всё, очень быстро это дало огромные нагрузки на вычислительную систему и позже стали использовать улучшенную версию RNN - LSTM (Long short-term memory).

Что такое RNN? Это вид нейронных сетей, который эффективен для последовательных данных, например последовательный язык, или временные ряды. В нём мы учитываем не только отклик для элемента, но и так же “запоминаем” информацию о предыдущих ответах и используем для получения результата. На рисунке 3 представлена схема RNN.

Изображение выглядит как часы

Автоматически созданное описание

Рисунок 3. RNN

Почему LSTM? Причина этого, то, что такая сеть могла забывать ненужные вектора, и наоборот запоминать важные. На рисунке 3 предоставлен “cell” LSTM ячейки.

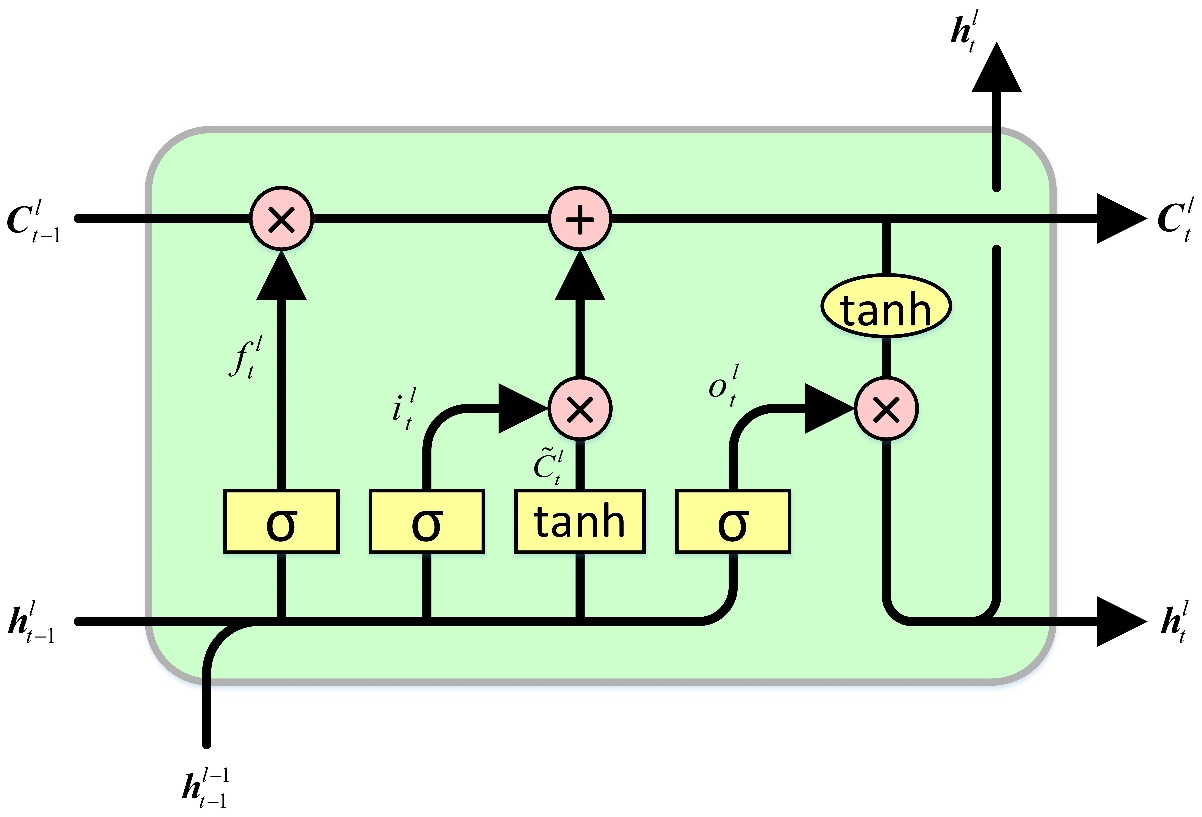


Рисунок 4. LSTM cell

Так же мы будем использовать 2 LSTM слоя, потому что будем использовать encode-decode перевод. На рисунке 5 представлена модель работы LSTM Encoder-Decoder

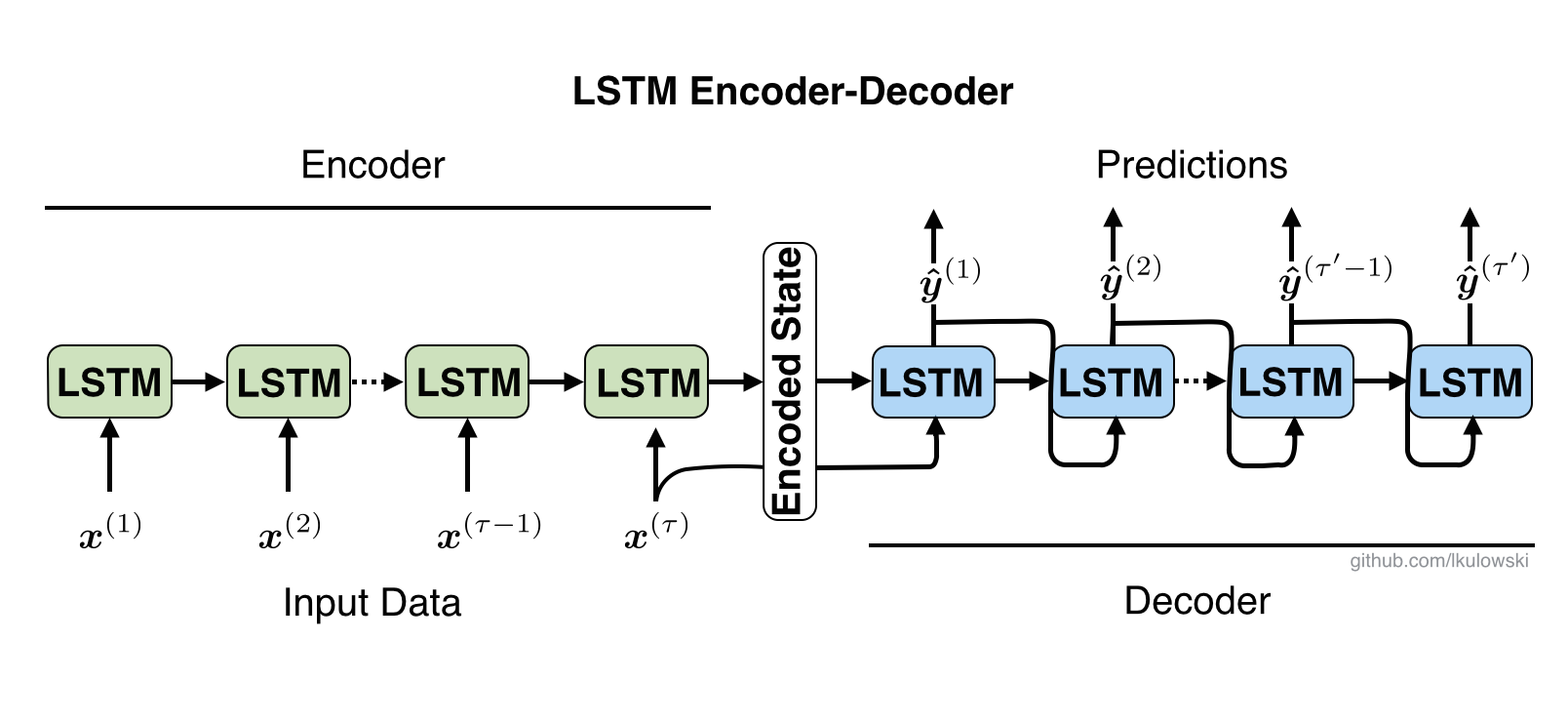


Рисунок 5. LSTM encoder-decoder

## Глава 3. Предварительная подготовка текстовых данных

В данной главе мы рассмотрим, какие шаги были принято для подготовки данных для обучения модели глубокого обучения.

Во-первых, нам необходимо было получить набор данных необходимый для обучения. В данном случае я взял пары предложения с разделителями табуляциями.

Следующим шагом мы загружаем данные, для этого была создана функция load\_doc (Рисунок 6)

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 6. load\_doc

После этого разбиваем данные по парам с помощью функции и удаляем ненужные данные (Рисунок 7).

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 7. to\_pairs

Теперь необходимо почистить данные (удалить знаки преминания, перевести в нижний регистр), это выполнит функция clean\_paris (Рисунок 8).

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 8. clean\_pairs

Теперь когда у нас есть почищенные данные, мы должны их сохранить, для этого я использую API pickle (Рисунок 9).

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 9. save\_clean\_data

Вызываю созданные мной функции и проверяю полученные данные на ограниченной выборке (Рисунок 10)

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 10. Использование созданных функций

В итоге мы получили очищенные данные (Рисунок 11)

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 11. Результат очистки данных

## Глава 4. Уменьшение кол-ва примеров и разделения на примеры для обучения и примеры для валидации и токенизация

Так как количество данных слишком много, сократим их до 40000 примеров и разделим их на 2 различные группы, для обучения и для проверки, и после сохраним в отдельные файлы. Листинг приведён ниже на рисунке 12.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 12. Уменьшения выборки данных

Для того что бы отобразить целые слова в какое-то численное значение мы будем использовать библиотеку Keras и создадим функцию токенизации слов. Под токенизацией понимается процесс, при котором идёт разбиение текста на части меньше и переводим в векторное представление (Рисунок 13)

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 13. Токенизация

Далее закодируем предложения в числовые вектора, для этого создаём функции encode\_sequences и encode\_output, а также сразу закодируем входные данные (Рисунок 14)**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 14. Токенизация предложений

## Глава 5. Создание алгоритма обучения модели

Следующей задаче, которую мы выполним, будет наконец определение NMT и будем использовать RNN, а именно LSTM для решения данной задачи, на рисунке 12, Создание модели и её последующее обучение, а также её сохранения (Рисунок 15)

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 15. создание и обучения модели

Информация о модели, её слои и кол-во параметров на рисунке 16

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 16. Информация о модели

Обучение модели, и информация о работе loss функции на рисунке 17

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 17. Обучение модели

## Глава 6. Оценка модели c помощью BLEU

Для оценки модели, мы будем использовать BLUE (bilingual evaluation understudy) алгоритм, проведём тесты сразу на 4 граммах от униграмма до 4-грамма. На рисунке 18 предоставлена информация о результатах теста BLUE, для униграммы мы получили результат в 0.079154. Так как у нас небольшая выборка, и сложность русского языка так же влияет на качестве перевода, мы получили не очень хорошие результаты, для улучшения результата необходимо увеличить кол-во обучений и данных для обучения.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 18. BLEU

# ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы я подробно изучил способы машинного перевода слов, узнал об обучении нейронной сети и об её оценке. Так же изучил LSTM архитектуру RNN и обучил собственную сеть машинному переводу на примерах двупарных языковых предложений. Узнал о encode-decode методе перевода, и разобрался в библиотеке Keras для глубокого машинного обучения.