

Санкт–Петербургский государственный университет

Девришев Надир Эльнурович

Выпускная квалификационная работа

***Применение нейросетевой архитектуры Transformer в
задаче машинного перевода***

Уровень образования: бакалавриат

Направление 01.03.02 «Прикладная математика и информатика»

Основная образовательная программа СВ.5005.2015 «Прикладная
математика, фундаментальная информатика и программирование»

Профиль «Исследование и проектирование систем управления
и обработки сигналов»

Научный руководитель:

доцент, кафедра теории систем управления элек-
трофизической аппаратурой, к.ф.-м.н. Козынчен-
ко Владимир Александрович

Рецензент:

профессор, кафедра теории управления д.ф.-м.н
Котина Елена Дмитриевна

Санкт-Петербург

2023 г.

Содержание

Введение	4
Постановка задачи	6
Обзор литературы	7
Цели и задачи работы	9
Глава 1. Подходы к решению задачи	10
1.1. Использование языка-посредника	10
1.2. Zero-shot translation	11
Глава 2. Используемая архитектура	14
2.1. Архитектура Transformer	14
2.1.1 Кодировщик	15
2.1.2 Механизм внимания	16
2.1.3 Multi-Head Attention	18
2.1.4 Декодер	19
Глава 3. Эксперименты и реализация	20
3.1. Набор данных	20
3.2. Метрики	21
3.2.1 BLEU	21
3.2.2 NIST	22
3.3. Эксперименты	22
3.3.1 Предобучение с помощью генерации пропущенного то- кена	22
3.3.2 Аугментация с помощью фиктивных данных	23
3.3.3 Лексическое сходство	24
3.4. Реализация	25
3.5. Результаты	27
3.6. Реальное оценивание	28
Выводы	29
Заключение	31
Список литературы	32

Приложения	34
Код	34

Введение

Машинный перевод – одна из наиболее быстро развивающихся сфер обработки естественного языка (Natural Language Processing, NLP). Однако с каждым днём не только появляются новые методы машинного перевода и модификации уже существующих, но и пополняется список подзадач. Так, например, в 2017 году команда Google Translate описала принцип работы своей кросс-языковой модели, использовавшей в процессе обучения большое количество корпусов парных предложений на различных языках, которая помимо осуществления универсального перевода между любыми двумя языками, встречающимися на этапе обучения, неплохо осуществляла новый для своего времени метод zero-shot translation.

Zero-shot translation – это подход к решению задачи машинного перевода с одного языка на другой, отличительным условием которой является отсутствие в явном виде обучающих данных между ними.

Очевидным решением будет построение нескольких моделей перевода и последующий двукратный перевод с исходного языка на целевой с использованием некоторого языка-посредника. Однако данный подход слишком затратен, и из самого алгоритма вытекает накопление ошибок при двукратном переводе.

Zero-shot подход предполагает построение большой кросс-языковой модели, которая была бы способна осуществлять перевод между всеми языками, встреченными на этапе обучения в соответствующих корпусах параллельных данных.

В качестве кросс-языковой модели в работе рассматривается state-of-the-art архитектура своего времени - Transformer.

Параллельные корпуса парных предложений были взяты из пакета данных для обучения моделей машинного перевода «OPUS». Построение кросс-языковой модели требует большого числа обучающих данных и вычислительных ресурсов для обучения крупной модели. Тем не менее задача zero-shot translation обычно ставится в условиях суровых ограничений, таких как отсутствие данных между парой языков.

В работе исследуются дешёвые с точки зрения затрат ресурсов мето-

ды повышения качества zero-shot перевода моделью, а также формируются выводы об их эффективности.

Наиболее успешным оказалось использование лексического сходства между парами языков. Показано, что при дополнительном обучении кросс-языковой модели переводу с исходного на лексически близкий к целевому язык, качество перевода улучшается. Тем самым появляется возможность получить более точную модель, все еще не привлекая напрямую обучающие данные между исходным и целевым языками.

Рассматриваемая задача крайне актуальна, поскольку большинство исследований на тему машинного перевода в настоящее время центрированы относительно английского языка из-за наличия большого количества корпусов обучающих данных, содержащих английский в качестве исходного или целевого языков. Zero-shot подход позволяет осуществлять перевод вне зависимости от наличия корпуса.

Актуальность исследования также не вызывает никаких сомнений, поскольку самым значимым недостатком подхода zero-shot является низкое качество перевода. Таким образом, любое улучшение, не связанное с увеличением корпуса данных или числа параметров модели, поможет продвинуться исследователям этой области.

Постановка задачи

Zero-shot translation (ZST) – это подход к решению задачи машинного перевода, отличительным условием которой является отсутствие в явном виде обучающих данных между ними [1].

Обычно задача ZST ставится при наличии большого количества параллельных данных (т. е. содержащих соответствие между предложениями, написанными на исходном и целевом языках) для широкого множества языковых пар. Так, например, если на этапе обучения доступны параллельные наборы предложений для языкового перевода **En** \leftrightarrow **Ru** и **En** \leftrightarrow **De**, то интерес представляет модель, осуществляющая перевод между русским (**Ru**) и немецким (**De**) языками вопреки отсутствию блока параллельных данных между ними.

Вход: <2ru> The cat was sitting in the couch.

Выход: Кот сидел на диване.

Рис. 1: Входные данные и вывод.

На вход модели подается предложение на исходном языке (рис. 1). В качестве выхода необходимо получить перевод данного предложения на целевой язык как можно более точно.

Для указания языка перевода каждое предложение, поступающее на вход дополняется меткой вида <2tgt>, где tgt - обозначение целевого языка перевода (например, <2ru>, <2uk>, <2en>).

Стоит также различать zero-shot translation и zero-resource [2, 3] translation. Zero-resource translation пытается подстроиться под отсутствие параллельных данных для исходного и целевого языков и создает их фиктивно. К примеру, имеется обученная кросс-языковая модель, и поставлена задача перевода **Ru** \rightarrow **De** при полном отсутствии набора пар предложений на русском и немецком языках. Тогда есть смысл перевести небольшое (относительно исходного датасета) количество предложений на английском в наборе данных **Ru** \leftrightarrow **En** на немецкий язык и применить тонкую настройку модели с помощью получившегося набора фиктивных данных.

Обзор литературы

Вопросу zero-shot translation посвящено множество исследований и научных статей.

Фундаментальной статьей в данной области считается исследование от команды «Google Brain» [1]. В труде Мэлвина Джонсона, Майка Шустера и др. был описан алгоритм построения большой кросс-языковой модели, осуществляющей перевод не только между парой языков, а между любыми двумя языками из тех, что встречались модели на этапе обучения. Исследователи также определили новую для того времени задачу zero-shot translation. Оказалось, что сконструированная модель неплохо справляется и с ней. Вопросам построения кросс-языковых моделей машинного перевода посвящена также статья [5].

Дополнительно были изучены иные методы улучшения качества перевода путем привлечения сгенерированных обучающих данных для интересующей нас пары языков [2, 3]. Формально, условия задачи zero-shot translation не выполняются, однако данный подход значительно улучшает качество перевода привлекая лишь небольшое количество данных.

Самый очевидный подход к решению задачи ZST описан в статье [4]. Использование языка-посредника долгое время считалось классическим способом осуществления перевода в отсутствие необходимых обучающих данных.

В статье [6] рассматривается метод переноса обучения, в котором в процессе тонкой настройки изменяются только параметры механизма перекрестного внимания.

Улучшению zero-shot translation подхода было посвящено большое количество работ. К примеру, в статье [7] описываются различные способы добиться этого.

ZST имеет основным недостатком качество осуществляемого перевода. Объяснить это явление и предложить собственные решения попытались авторы статей [8, 9].

Архитектура Transformer, используемая в данной работе, подробно описана в статье, изменившей в 2017 подход к решению всех задач NLP [10].

Важным аспектом в построении моделей машинного обучения является поиск правильного качественного набора обучающих данных. В задаче машинного обучения создание данных осложняется еще и привлечением большого количества специалистов-лингвистов. Однако иной способ добыть параллельные корпуса данных - собирать вместе уже готовые переведенные тексты по всем интернет ресурсам. Данный подход успешно реализовали авторы [11]. Логическим продолжением является формирование набора данных из субтитров к уже переведенным на разные языки фильмам [12].

Для сравнения моделей и подходов между собой в машинном обучении используются различные метрики, специфичные для каждой конкретной задачи. Популярной метрикой в задаче машинного обучения является BLEU [13]. Своей популярностью она обязана высокой степени схожести с человеческой экспертной оценкой перевода. Ещё одной распространённой метрикой является NIST [14].

Для решения задач NLP требуются крайне большие модели, способные учитывать в себе сложности естественного языка. Обычный исследователь, зачастую, не может позволить себе обучение хорошей модели с нуля, поэтому специалисты из «Google Brain» озаботились созданием универсальной модели BERT для решения задач NLP прямо «из коробки» в своей работе [15].

Цели и задачи работы

Самым простым способом повышения качества машинного перевода является увеличение корпуса обучающих данных или числа параметров модели.

С увеличением модели время на обучение может расти непропорционально быстро. Для обучения некоторых больших моделей и вовсе необходимо обладать большими ресурсами в виде специально оборудованных серверов. Еще более сложной задачей кажется получение дополнительных данных для обучения. Чтобы собрать набор данных для обучения, зачастую, требуется большое количество специалистов по data mining и лингвистов.

Ситуация усугубляется в случае zero-shot translation, поскольку подход предполагает обучение большой кросс-языковой модели для осуществления перевода между несколькими языками. Из этого следует необходимость использовать увеличенные модели с еще большим количеством данных, чем при классической постановке задачи машинного перевода.

Данная работа ставит своей целью исследование эффективных и дешевых с точки зрения ресурсоемкости методов улучшения качества перевода zero-shot translation.

Для достижения поставленной цели необходимо решить ряд задач:

1. Поиск подходящих данных для обучения
2. Подбор архитектуры
3. Обучение модели
4. Оценка перевода
5. Проведение экспериментов с улучшением перевода
6. Сравнение с базой исследования
7. Формулирование вывода

Глава 1. Подходы к решению задачи

Имеем несколько корпусов данных, состоящих из связных пар предложений на разных языках. Необходимо осуществлять перевод между любыми двумя языками, встреченными на этапе обучения.

Перевод между языками, пары которых встречались в наборе данных на этапе обучения, не вызывает вопросов и в данном случае постановка задачи полностью совпадает с задачей обычного машинного перевода. Интерес представляет осуществление перевода между парами языков, с которыми модель не сталкивалась при обучении.

1.1 Использование языка-посредника

Очевидным подходом для решения задачи является построение нескольких моделей, осуществляющих перевод с некоторого языка-посредника и на него [4].

При необходимости осуществить перевод с одного языка на другой сначала исходное предложение транслируется на язык-посредник моделью $\langle \text{src} \rangle \rightarrow \langle \text{agent} \rangle$, которая обучалась на корпусе предложений $\langle \text{src} \rangle \leftrightarrow \langle \text{agent} \rangle$, где $\langle \text{src} \rangle$ - исходный язык, а $\langle \text{agent} \rangle$ - язык посредник. Далее с языка-посредника $\langle \text{agent} \rangle$ осуществляется перевод на целевой язык с помощью модели $\langle \text{agent} \rangle \rightarrow \langle \text{tgt} \rangle$, обученной на корпусе $\langle \text{agent} \rangle \leftrightarrow \langle \text{tgt} \rangle$, где $\langle \text{tgt} \rangle$ - целевой язык перевода (рис. 2).

В качестве языка-посредника $\langle \text{agent} \rangle$ очень часто выступает английский язык, поскольку на сегодняшний день доступно большое количество обучающих корпусов, центрированных относительно английского языка, т.е. имеющих английский в качестве исходного или целевого языков.

Преимуществом подхода с использованием языка-посредника является независимость качества перевода от числа пар языков - модель для каждой пары языков обучается независимо от остальных моделей. По этой же причине при увеличении корпуса обучающих данных для какой-либо из пар или при улучшении качества данных нет необходимости в повторном обучении всех моделей - достаточно обучить заново лишь одну.

К недостаткам подхода можно отнести сложность. Число необходимых

Pivot vs. Zero-Shot Translation

Pivot Translation

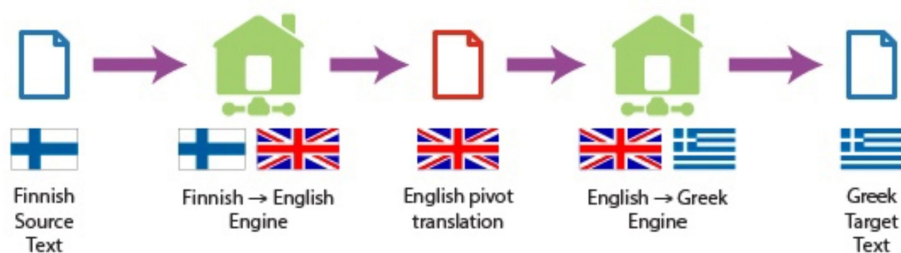


Рис. 2: Перевод с использованием языка-посредника.

к построению моделей может квадратично зависеть от числа рассматриваемых на этапе обучения языков.

Другим изъяном подхода является накопление ошибок перевода. При наличии ошибки в переводе $\langle \text{src} \rangle \rightarrow \langle \text{agent} \rangle$ какого-либо токена с исходного языка на целевой, на вход следующей модели $\langle \text{agent} \rangle \rightarrow \langle \text{tgt} \rangle$ отправится уже неправильный запрос к переводу. В лучшем случае модель допустит ошибку в переводе лишь этого токена, в худшем - ошибка вызовет цепную реакцию и повлияет на перевод всего предложения.

1.2 Zero-shot translation

В отличие от подхода с использованием языка-посредника метод zero-shot translation основан на обучении крупной кросс-языковой модели для перевода между большим количеством языков [5]. Основа архитектуры такой модели не содержит никаких концептуальных отличий от моделей, используемых в переводе одной пары языков. Существенная разница заключается лишь в размере модели. Из-за использования универсальных для всех языков энкодера и декодера она должна быть гораздо больше, чтобы улавливать зависимости для всех пар языков (рис. 3).

В фундаментальной статье «Google’s multilingual neural machine translation system» [1] представлена первая версия рассматриваемого подхода. Исследователями была построена крупная кросс-языковая модель, осуществляющая перевод между большим количеством языков. Модель состояла из кодиров-

Zero-Shot Translation



Рис. 3: Zero-shot translation.

щика (8xLSTM) и декодировщика (8xLSTM). Также был составлен смешанный словарь, содержащий разбиения из всех языков, на основе которого определялись общие погружения.

Предполагалось, что модель сможет с помощью универсального кодировщика (8xLSTM) отображать предложения на любом языке в общее пространство признаков, после чего общий декодировщик (8xLSTM) так же успешно будет из данного пространства извлекать представления на нужном языке [6].

Чтобы модель самостоятельно различала нужный язык перевода, исходные предложения снабжаются метками **<2tgt>** в начале (например, **<2es>**, **<2en>** и т.д.). В среднем представленная модель справляется на 0.005-0.01 балл BLEU хуже, чем модель с такой же архитектурой, но обученная на однородном корпусе пар предложений WMT (таблица 1).

Явным достоинством подхода является обучение всего одной модели для осуществления перевода между любыми двумя языками. Это крайне упрощает работу с переводом, поскольку для перевода на конкретный язык необходимо лишь указать специальный тэг в начале предложения.

Другим преимуществом подхода является сохранение общих знаний об устройстве языков. Так как языки общих языковых групп лексически и грамматически схожи, у модели будет больше данных для обучения структуре языков.

Таблица 1: Результаты кросс-языковой модели.

Модель	Single model	Cross-lang model
WMT English → German	0.2467	0.2449
WMT English → French	0.3895	0.3623
WMT German → English	0.3043	0.2984
WMT French → English	0.3550	0.3489

К сожалению, и данный подход имеет свои недостатки. Одной из наиболее ярко выраженных слабых сторон zero-shot translation является низкий уровень качества. В некоторых случаях метод уступает в показателях даже обычному подходу с использованием языка-посредника [7]. Этому есть несколько объяснений.

Например, установление ложных корреляций [8]. При попытке найти объяснение этому феномену было установлено, что частой проблемой неправильного перевода является неверно выбранный целевой язык. В идеале, кросс-языковая модель должна использовать для определения целевого языка только метку **<2tgt>**, но в процессе обучения устанавливается связь между целевым языком и смысловой нагрузкой предложения. Например, если предложения о котах на разных языках встречались лишь с меткой **<2en>** (т. е. с требованием перевести их на английский язык), то при дальнейших попытках перевести схожие по смыслу предложения о котах с меткой **<2ru>** они будут ошибочно переведены на английский. В качестве решения было выдвинуто предложение предобучать модель отдельно на метках **<2tgt>**.

Иной проблемой является разреженность данных в общем пространстве. Объяснение выдвинули исследователи Naveen Arivazhagan, Orhan Firat и др. [9], предположив, что причиной является неправильное отображение энкодером исходных предложений в пространство признаков. Идея zero-shot translation предполагает, что схожие по смыслу предложения на разных языках будут расположены близко друг к другу в числовом пространстве. В связи с этим было принято решение «штрафовать» энкодер за слишком разрозненные представления.

Глава 2. Используемая архитектура

Для реализации подхода zero-shot translation необходимо обучить универсальную кросс-языковую модель, которая бы осуществляла перевод между всеми парами языков.

В статье “Google’s multilingual neural machine translation system: enabling zero-shot translation” 2017 года [1] исследователи использовали state-of-the-art подход того времени, основанный на механизме внимания, реализованном с использованием энкодеров и декодеров на основе BiLSTM.

Однако с того времени многое изменилось и был создан более эффективный метод для решения прикладных NLP (Natural Language Preprocessing) задач, а именно - архитектура Transformer.

2.1 Архитектура Transformer

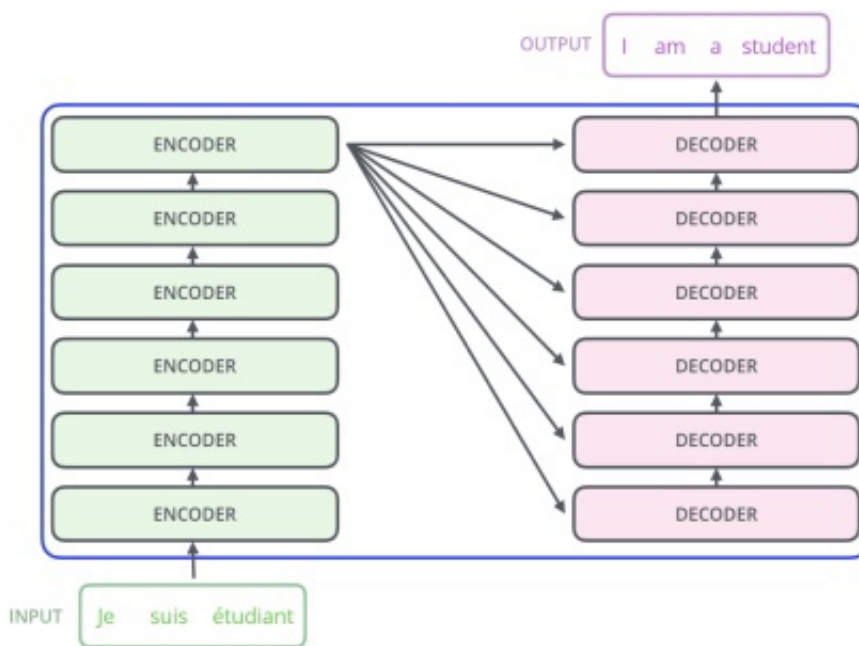


Рис. 4: Архитектура Transformer.

На сегодняшний день применительно к задаче ZST, как и к большинству других задач NLP, state-of-the-art-подходом является архитектура глубоких нейронных сетей Transformer, представленная в 2017 году исследователями из «Google Brain» [10]. Своим появлением модель оказала влияние на все

сферы, связанные с NLP, повысив планку результатов в большинстве задач и открыв новые возможности для исследований.

Впервые модель Трансформера была предложена в статье «Attention is All You Need» [10]. Данная архитектура состоит из кодирующего и декодирующего компонентов (рис. 4).

2.1.1 Кодировщик

Кодирующий компонент содержит набор последовательно соединённых кодировщиков (энкодеров). Он получает на вход векторизованную последовательность с позиционной информацией. Все энкодеры идентичны по структуре, хотя и имеют разные веса, которые обновляются независимо во время обучения. Каждый кодировщик можно разделить на два подслоя:

- Feed Forward Neural Network
- Механизм внимания Self-Attention

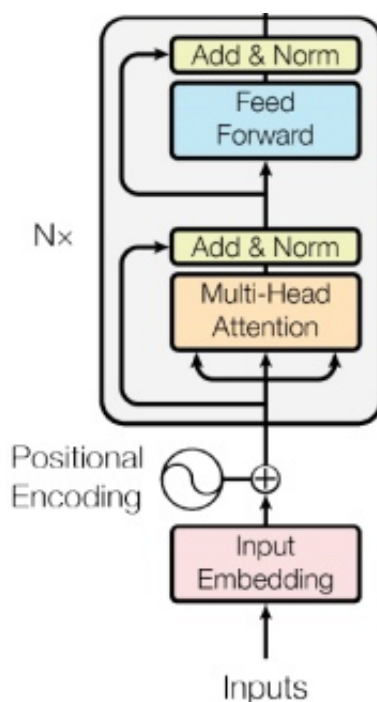


Рис. 5: Устройство энкодера.

Входная последовательность, поступающая в энкодер, сначала проходит через слой внутреннего внимания (self-attention), помогающий энкодеру

посмотреть на другие слова во входящем предложении во время кодирования конкретного слова (рис. 5). Выход слоя внутреннего внимания отправляется в нейронную сеть прямого распространения (feed forward neural network). Точно такая же сеть независимо применяется для каждого слова в предложении.

2.1.2 Механизм внимания

В качестве начального шага для подсчёта внутреннего внимания предполагается создание трёх векторов для каждого вектора, поданного на вход кодировщику (эмбединга каждого слова):

1. Query vector (запрос)
2. Key vector (ключ)
3. Value vector (значение)

Эти векторы создаются с помощью перемножения эмбединга на три матрицы, которые мы модифицируем во время процесса обучения.

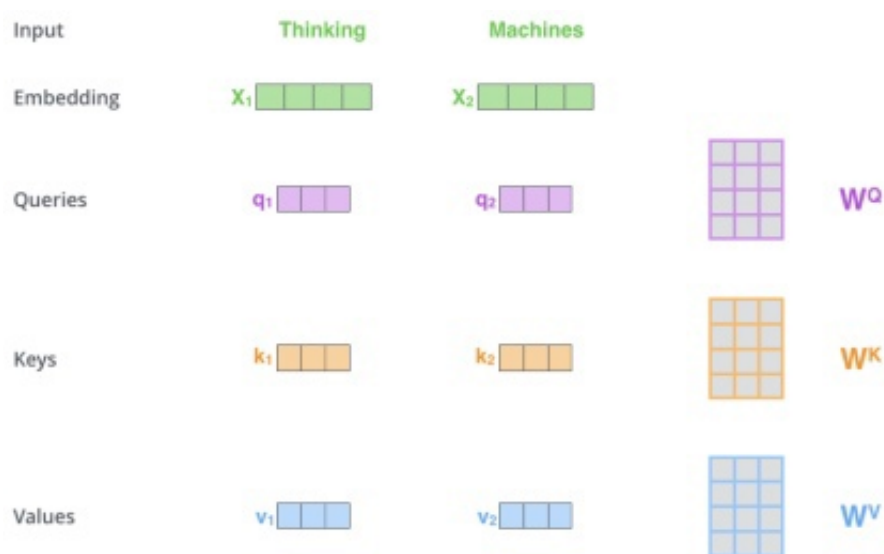


Рис. 6: Query, Key, Value векторы.

Полученные векторы, обычно, меньше в размере, чем входные векторы эмбедингов слов (рис. 6).

Следующий этап построения внутреннего внимания – вычисление коэффициента фокусирования (score). Оценивается коэффициент для каждого слова во входящем предложении по отношению к рассматриваемому слову. Коэффициент показывает, насколько стоит учитывать информацию о других словах входящего предложения во время кодирования слова в конкретной позиции.

Коэффициент высчитывается с помощью скалярного произведения вектора запроса (Query) и вектора ключа (Key) соответствующего слова. К примеру, при вычислении внутреннего внимания для слова в позиции 1, первый коэффициент будет скалярным произведением q_1 и k_1 , второй — скалярным произведением q_1 и k_2 (рис. 7).

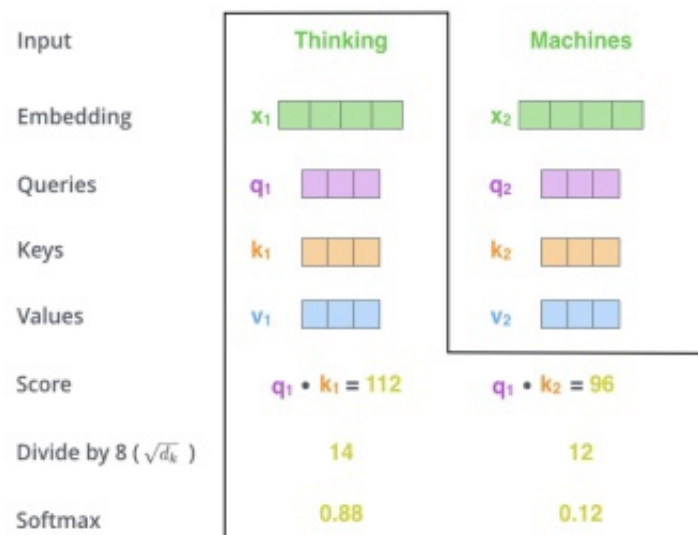


Рис. 7: Получение векторов key и query.

Третьим и четвертым этапами является нормировка коэффициентов. Они делятся на квадратный корень размерности векторов ключа. Выбирается именно такое значение, поскольку оно обеспечивает более стабильные градиенты. Однако возможны также и другие.

Затем результат пропускается через функцию softmax. Данная функция нормализует коэффициенты так, чтобы они были положительными и в сумме давали 1.

Полученный softmax-коэффициент (softmax score) решает, в какой мере каждое из слов предложения будет выражено в определенной позиции. Чаще

всего, слова в своей позиции получают наибольший softmax-коэффициент, но иногда дополнительно учитываются и другие слова, релевантные к рассматриваемому.

Предпоследний этап – умножение каждого вектора значения на softmax-коэффициент и сложение получившихся векторов. Идея состоит в том, что значения нерелевантных для конкретной позиции слов будут получать низкие softmax-коэффициенты и практически не будут оказывать влияния на результирующий вектор значения.

Наконец, последний шаг подразумевает простое сложение взвешенных векторов значения. Это и будет результатом слоя внутреннего внимания в данной позиции (для конкретного слова). Получаем вектор, который можем передавать дальше в нейронную сеть прямого распространения.

В настоящих реализациях, однако, эти вычисления делаются в матричной форме для более быстрой обработки. Каждый механизм внимания параметризован матрицами весов запросов W_Q , весов ключей W_K и весов значений W_V . Для вычисления внимания входного вектора X к вектору Y , вычисляются векторы $Q = W_Q X$, $K = W_K X$, $V = W_V Y$, которые используются для вычисления результата внимания:

$$Attention(Q, V, K) = softmax \left(\frac{QK^T}{\sqrt{d_k}} \right) V$$

2.1.3 Multi-Head Attention

Для улучшения результата и усложнения модели оригинальной статье был предложен механизм «Multi-Head attention», который заключается в использовании не одной, а нескольких (например, 4, 6 или 8) матриц W_Q , W_K , W_V . Это дает кодировщику возможность выделять из входной последовательности больше разнородной необходимой информации и улучшает производительность слоя внутреннего внимания за счет следующих аспектов:

1. Модель получает возможность фокусироваться на нескольких разных позициях.

2. Слой внимания снабжается множеством «подпространств представлений» (representation subspaces).

Результаты работы механизма множественного внимания конкатенируются, и полученный вектор умножается на матрицу обучаемых вместе со всей моделью весов, что понижает размерность вектора до исходной, сохраняя при этом информацию из всех параллельных механизмов внимания.

2.1.4 Декодер

Концептуально декодирующий элемент в архитектуре Transformer отличается от кодировщика лишь наличием дополнительного слоя, нацеленного на внедрение результатов механизма внимания с последнего слоя кодировщика (рис. 8).

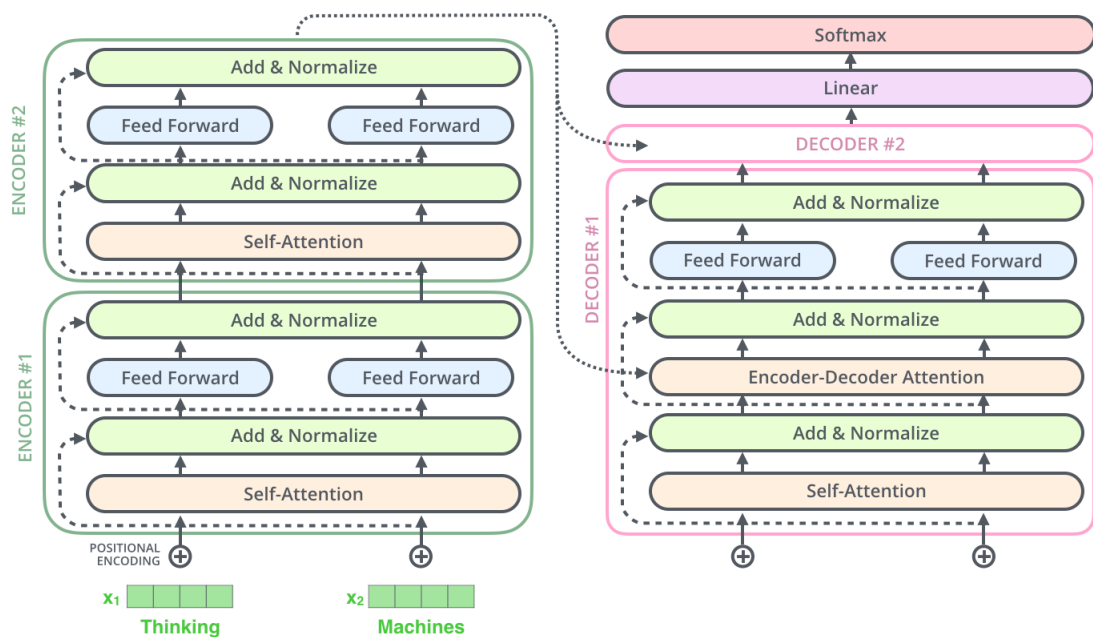


Рис. 8: Устройство декодера.

В отличие от кодировщика слой внутреннего внимания в декодере учитывает только предыдущие позиции в выходном предложении. Это реализуется с помощью маскирования всех позиций после текущей (устанавливая их в $-\infty$) перед этапом softmax в вычислении внутреннего внимания.

Глава 3. Эксперименты и реализация

Целью работы является поиск методов, которые бы улучшили подходов zero-shot translation. В процессе исследований было реализовано множество попыток усовершенствования.

3.1 Набор данных

«OPUS» - это растущая коллекция переведенных текстов из Интернета [11]. В проекте «OPUS» исследователи преобразовывают и выравнивают бесплатные открытые онлайн-данные, добавляют лингвистические аннотации и предоставляют сообществу общедоступные параллельные корпуса данных. «OPUS» основан на продуктах с открытым исходным кодом и поставляется в виде пакета открытого контента.

В «OPUS» включает в себя набор данных «OpenSubtitles» [12], который предоставляет выровненные между собой субтитры к фильмам и сериалам на разных языках. В общей сложности «OpenSubtitles» содержит корпуса данных для 62 языков.

Для проведения исследований были выделены корпуса:

- **De** → **En** (500 тыс. пар предложений)
- **En** → **De** (500 тыс. пар предложений)
- **Ru** → **En** (500 тыс. пар предложений)
- **En** → **Ru** (500 тыс. пар предложений)
- **En** → **Uk** (200 тыс. пар предложений)
- **De** → **Ru** (100 тыс. пар предложений)
- **De** → **Uk** (100 тыс. пар предложений)

Каждое предложение было дополнено токеном **<2tgt>**, где **tgt** - целевой язык перевода.

3.2 Метрики

В задаче машинного перевода при оценке качества перевода сложно достичь той же точности в оценке, что и оценка профессионального лингвиста. Однако такой подход требует много времени, ресурсов и специалистов. В попытках автоматизировать оценку качества перевода исследователи разработали множество метрик, стараясь максимально приблизиться к точности оценки лингвистов.

3.2.1 BLEU

BLEU - одна из популярнейших метрик в задачах NLP, связанных с генерацией некоторого текста и сравнением с эталоном [13].

Оценка BLEU принимает уже существующие идеально хорошие переводы как эталонный перевод и сравнивает выходные данные машинного перевода (кандидата) с этим эталоном. В конечном счете это сравнение выражается числом от 0 до 1. Чем выше цифра, тем лучше оценка.

Подобный метод должен как-то компенсировать тот факт, что у каждого исходного сегмента может быть несколько совершенно хороших, но разных переводов. Оценка BLEU и допускает несколько эталонных переводов, каждый из которых считается одинаково хорошим. Но любое отклонение от эталона или эталонов получает более низкую оценку.

BLEU проверяет слова в переводе-кандидате, подсчитывает их, и всякий раз, когда слово из эталона отсутствует в кандидате, оценка снижается. Таким образом, алгоритм штрафует перевод ещё и за излишнюю лаконичность.

Тем не менее остается проблема порядка слов, ведь в данном случае слова эталонного предложения выставленные в произвольном порядке не будут иметь никакого смысла, но получат по метрике качество в 1 балл. BLEU решает эту проблему, проверяя не только наличие слов эталонного перевода в предложенном, но и наличие в нем групп последовательных слов. Подобный алгоритм даёт гарантию, что случайный порядок правильных слов не будет вознагражден, поскольку соответствием считается только присутствие слов в том же порядке.

Формула оценки BLEU:

$$BLUE = BP \times e^{\sum_{k=1}^n w_k \log(p_k)},$$

где

$$BP = e^{\min(1 - \frac{\text{len}(\text{reference})}{\text{len}(\text{prediction})}, 0)}$$

3.2.2 NIST

NIST (National Institute of Standards and Technology) является усовершенствованной версией метрики BLEU, полностью на ней основываясь [14].

Данная метрика так же как и BLEU сравнивает наличие групп последовательных слов в предложенном переводе, однако в неравной степени оценивает вклад n-грамм разных размеров в итоговую оценку. Чем больше последовательность слов, тем больше ее вклад в итоговую оценку качества перевода.

3.3 Эксперименты

3.3.1 Предобучение с помощью генерации пропущенного токена

В условиях дефицита обучающих данных необходимо повышать уровень понимания моделью структуры языка. Исследователи из «Google Research» внедрили новый подход к предобучению своей модели BERT (англ. Bidirectional Encoder Representations from Transformers) - предобучение на смежных прикладных задачах [15].

В качестве подобных задач были выбраны:

- предсказание следующего предложения (англ. next sentence prediction)
- генерации пропущенного токена (англ. masked language modeling)

Для задачи генерации пропущенных токенов на вход подается предложение, в котором некоторые токены заменены на служебный токен [MASK]. Модель должна по контексту понять, какие токены были в исходном предложении на месте [MASK].

Подобный подход к предобучению сильно улучшил результаты BERT в прикладных задач, и модель до сих пор остаётся state-of-the-art подходом.

В рамках проведения эксперимента предобучение генерацией пропущенных токенов было применено для предварительной подготовки модели к решению задачи zero-shot translation.

В качестве базы исследования были взяты архитектура Transformer и наборы **De** → **En** (500 тыс.) и **En** → **Ru** (500 тыс.).

Эксперимент проводился дважды - предобучение проводилось сначала маскированием предложений на русском языке, после чего было опробовано предобучение на корпусе немецкого языка.

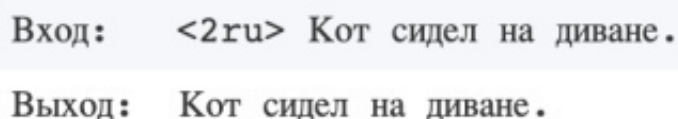
К сожалению, данный эксперимент не привёл к видимому улучшению перевода.

3.3.2 Аугментация с помощью фиктивных данных

Основной ошибкой при осуществлении перевода zero-shot является неправильное определение целевого языка перевода. Язык перевода должен быть определен лишь по тэгу **<2tgt>** в начале исходного предложения, где **tgt** - целевой язык перевода.

Объяснить неправильный выбор языка перевода можно склонностью модели обучаться ложным зависимостям [8]. Например, если на этапе обучения было встречено много предложений определённой тематики с запросом к переводу на английский (**<2en>**), тогда при осуществлении перевода схожих предложений на этапе тестирования на немецкий перевод будет осуществляться так же, как и при обучении модели - на английский.

Подобная проблема может быть также вызвана тем, что тэг **<2tgt>** на этапе обучения может быть ассоциирован моделью с языком исходного предложения, однако ввиду специфики задачи при тестировании **<2tgt>** будет использоваться вместе с предложениями на языках, с которыми он не комбинировался при обучении.



Вход: **<2ru>** Кот сидел на диване.
Выход: Кот сидел на диване.

Рис. 9: Пример фиктивных данных.

В качестве решения была выдвинута гипотеза о том, что дополнение исходного набора обучающих данных небольшим количеством фиктивных данных может помочь с правильным переводом.

Блок фиктивных данных состоит из пар предложений. Первое предложение является исходным и помечается тэгом со своим же языком (рис. 9). Второе предложение дублирует первое, но без тэга. Обучающие данные были дополнены фиктивными в размере 20% от исходных данных.

Предполагалось, что наличие подобных предложений повлияет на понимание моделью независимости тэга от языка исходного предложения. К сожалению, данное предположение оказалось неверным и не привело к улучшению качества перевода.

3.3.3 Лексическое сходство

По мере развития народы смешивались между собой, разделялись и соединялись культурно и территориально. При этом каждый из новообразованных народов видоизменял язык, адаптируя его под нужды людей для обеспечения максимальной взаимной понятности. Это и стало причиной столь огромного числа схожих между собой языков на нашей планете.

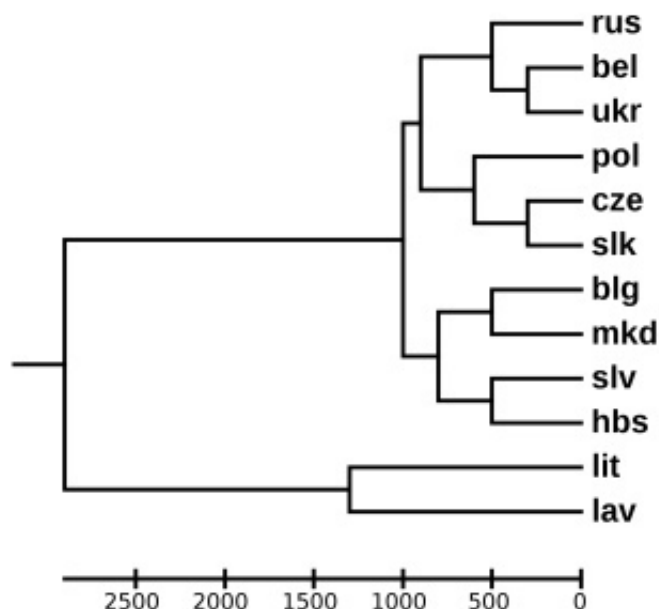


Рис. 10: Филогенетическое дерево групп славянских и балтийских языков.

Лексическое сходство (в лингвистике) — мера того, до какой степени слова двух данных языков лексически сходны. Лексическое сходство, равное единице (или 100%) означает полное совпадение двух данных языков, тогда как равенство 0 означает полное отсутствие в них общих слов. Например, русский и украинский языки имеют 62% лексического сходства (рис. 10).

Существуют разные способы определения лексического сходства и результаты, полученные разными способами, соответственно, будут различаться. Например, метод, принятый в этнологии, состоит в том, чтобы сравнивать стандартизированный список слов в разных языках и находить сходные среди них одновременно как по написанию, так и по смыслу. Используя этот метод, было найдено, что английский язык имеет лексическое сходство с немецким 60% и с французским 27%.

Задача zero-shot translation ставит своей целью осуществление перевода между двумя языками в условиях полного отсутствия обучающих данных для данной пары. Однако, используя свойства лексической схожести двух языков, мы можем обучить модель переводу на схожий язык, тем самым облегчив ей задачу.

Эксперименты проводились замещением украинского языка русским. Для обучения кросс-языковой модели были использованы корпуса предложений **De** → **En** и **En** → **Uk**, а в качестве дополнительного - **De** → **Ru**. Оценивался лишь перевод **De** → **Uk**. Предполагалось, что отсутствие прямых обучающих данных **De** → **Uk** может быть компенсировано привлечением корпуса **De** → **Ru**. Из датасета **De** → **Ru** были отфильтрованы все предложения, содержащие символы Ёё, ъ, ы, и Ээ, так как в украинском алфавите отсутствуют данные буквы, и обучение модели лишним словам могло ухудшить качество перевода на украинский.

3.4 Реализация

Доступ к массиву данных, предобработка данных, построение и обучение модели осуществлялись с помощью открытой программной библиотеки для машинного обучения tensorflow, разработанной компанией «Google» для решения задач построения и обучения нейронных сетей.

Модель Transformer имела параметрами:

- Число слоёв кодировщика/декодировщика - 4;
- Размер входных данных (размер эмбедингов) - 128;
- Размерность выхода нейронной сети - 512;
- число heads в «MultiHeadAttention» - 8;
- Доля dropout-параметров - 0.1;

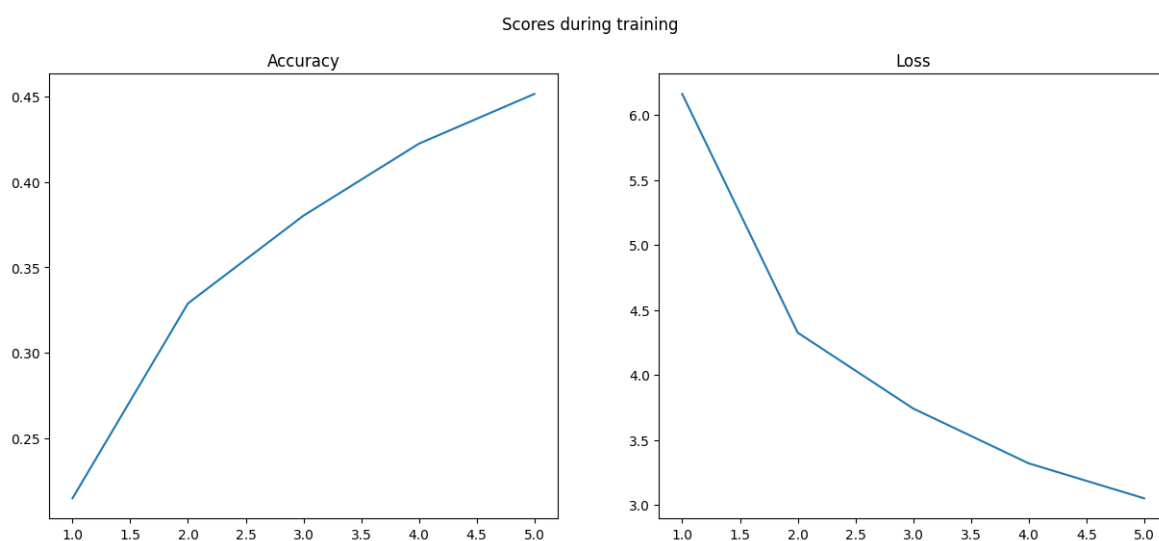


Рис. 11: Показатели во время обучения.

Размеры обучающих данных варьировались от 50 до 100 тыс. пар предложений. Число эпох для обучения было выбрано равным 5. В среднем на одну эпоху затрачивалось время равное одному часу (≈ 3325.42 секунд). Уменьшение ошибки (Loss) наблюдалось на протяжении всех этапов обучения (рис. 11).

Реализация метрик BLEU и NIST была взята из библиотеки для символической и статистической обработки естественного языка nltk (Natural Language Toolkit).

3.5 Результаты

Использование лексического сходства украинского и русского языков поспособствовало улучшению качества перевода с немецкого на украинский **De** → **Uk**.

Таблица 2: Результаты работы.

Набор данных	BLEU	NIST
De → En - 50 тыс., En → Uk - 50 тыс.	0.082	0.025
De → En - 100 тыс., En → Uk - 100 тыс.	0.111	0.036
De → En - 50 тыс., En → Uk - 50 тыс., De → Ru - 50 тыс.	0.117	0.037
De → En - 100 тыс., En → Uk - 100 тыс., De → Ru - 100 тыс.	0.132	0.046

Эксперименты проводились дважды на наборе данных «OpenSubtitles» от OPUS. Были взяты корпуса **De** → **En** и **En** → **Uk** размерами по 50 тыс. и по 100 тыс. пар предложений. В обоих экспериментах основные данные дополнялись корпусом **De** → **Ru** размером 20% от основных.

Для тестирования качества использовались метрики BLEU и NIST. Тестирование проводилось с использованием корпуса размером 20 тыс. **De** → **Uk** предложений.

Из Таблицы 2 можем сделать вывод, что для двух экспериментов дополнение корпусом **De** → **Ru** размерами 10 и 20 тыс. пар предложений обеспечивает улучшение в 35% и 13% для метрики BLEU соответственно и 44% и 24% — для NIST соответственно.

3.6 Реальное оценивание

Такие метрики как BLEU и NIST хороши лишь для поверхностной оценки перевода. Зачастую, они плохо отражают действительное качество перевода.

По-настоящему показательной оценкой машинного перевода всегда являлось оценивание людьми.

Недостатком подобного подхода является его ресурсоёмкость, поскольку для наиболее полного оценивания необходимо привлечь большое количество людей.

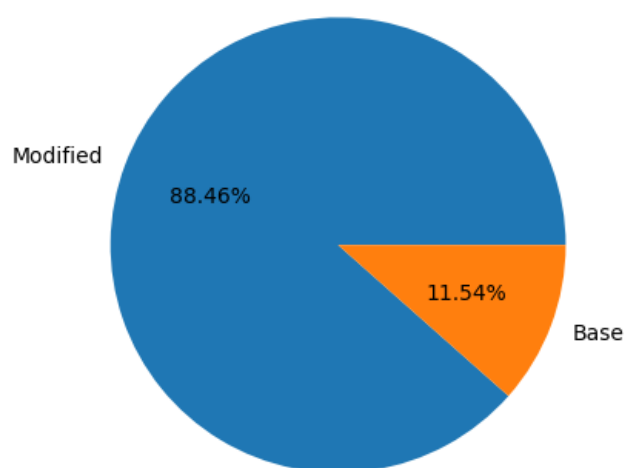


Рис. 12: Независимая оценка.

Тем не менее для подтверждения результатов был проведён независимый опрос, который показал, что большинство людей (около 88.5%) предпочитает перевод, сформированный дообученной моделью, переводу базовой модели (рис. 12).

Выводы

В рамках данной работы были исследованы различные модификации классического zero-shot подхода.

Целью исследований был поиск дешевого и эффективного способа улучшения качества zero-shot перевода. Были выдвинуты гипотезы по поводу влияния некоторых модификаций на качество перевода моделью относительно базы исследования. Наряду с удачным методом улучшения были осуществлены и другие попытки повысить качество перевода, не давшие значительного прироста в качестве.

Предложенными модификациями были:

- Предобучение модели с помощью задачи генерации пропущенного токена;
- Аугментация с помощью фиктивных данных;
- Использование лексического сходства языков.

Лексическое сходство языков можно использовать для улучшения качества машинного перевода zero-shot (ZST), компенсировав отсутствие корпуса параллельных данных другим корпусом, имеющим лексически похожий язык в качестве целевого.

Однако предобучение модели на смежной задаче и аугментация фиктивными данными, рассмотренные в работе, не оказали сильного влияния на конечный результат.

Вопреки расхожему мнению, повысить качество машинного перевода можно не только значительными дорогостоящими увеличениями модели, сборами новых, необходимых для обучения корпусов данных или привлечением большого числа лингвистов. Например, при использовании свойств самих языков, таких как лексическая и грамматическая схожесть некоторых из них, также можно добиться улучшения.

Исследованное влияние свойств лексической схожести на перевод zero-shot способно существенно помочь в реальных условиях, если необходимо осуществить перевод с одного языка на другой в отсутствие параллельного

корпуса данных. Например, необходимо обучить модель машинного перевода с русского языка **Ru** на один из языков индоарийской группы бходжпури **Bho**. Для этого достаточно будет корпусов **Ru** \rightarrow **En**, **En** \rightarrow **Bho** и корпуса параллельных данных переводов с русского (**Ru**) на язык, схожий с языком бходжпури (**Bho**), – Хинди (**Hi**).

Заключение

В ходе исследования были проведены эксперименты по улучшению качества zero-shot translation.

В качестве данных рассматривались корпуса параллельных данных от открытого пакета «OPUS». Были использованы наборы субтитров на английском, немецком, украинском и русском языках.

Основной моделью для проведения экспериментов была выбрана архитектура Transformer. Все эксперименты проводились, опираясь на обучение данной модели с идентичными параметрами.

В качестве базы для экспериментов были выбраны модели, обученные на 50 и 100 тысячах пар предложений. Направлениями перевода для исследований были **De** → **Ru** и **De** → **Uk**.

Перевод оценивался с помощью двух популярных для задач машинного перевода метрик - BLEU и NIST. Именно эти показатели были ключевыми при сравнении подходов.

В процессе исследований было реализовано несколько модификаций базы, основанных на выдвинутых гипотезах, в надежде получить улучшение в качестве машинного перевода.

Гипотеза о предобучении модели с помощью задачи генерации пропущенного токена не оправдала ожиданий и не привела к видимым улучшениям в качестве. Также при использовании аугментация фиктивными данными существенного улучшения достичь не удалось.

В то же время использование лексического сходства двух языков дало значительное улучшение в качестве в сравнении с базой исследования.

В результате был сформирован вывод о пользе полученных результатов и их актуальности в современных условиях.

Список литературы

- [1] Johnson M., Schuster M., Le V. Q. et al. Google's multilingual neural machine translation system: enabling zero-shot translation // Transactions of the Association for Computational Linguistics. 2017. Vol. 5. P. 339–351.
- [2] Currey A., Heafield K. Zero-resource neural machine translation with monolingual pivot data // Proceedings of the 3rd Workshop on Neural Generation and Translation. 2019. P. 99–107.
- [3] Firat O., Sankaran B., Alonaizan Y. et al. Zero-resource translation with multilingual neural machine translation // Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. 2016. P. 268–277.
- [4] Wu Hua, and Haifeng Wang. Pivot Language Approach for Phrase-Based Statistical Machine Translation. // Machine Translation 21. 2007. № 3. P. 165–181.
- [5] Freitag M., Firat O. Complete multilingual neural machine translation // Proceedings of the Fifth Conference on Machine Translation. 2020. P. 550–560.
- [6] Gheini M., Ren X., May J. Cross-attention is all you need: adapting pretrained transformers for machine translation [Электронный ресурс]: URL:<https://doi.org/10.48550/arXiv.2104.08771> (дата обращения: 15.02.23).
- [7] Zhang B., Williams P., Titov I. et al. Improving massively multilingual neural machine translation and zero-shot translation // Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020. P. 1628–1639.
- [8] Jiatao Gu, Yong Wang, Kyunghyun Cho, Victor O.K. Li. Improved zero-shot neural machine translation via ignoring spurious correlations. // In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019. P. 1258–1268.

- [9] Arivazhagan N., Bapna A., Firat O. et al. The missing ingredient in zero-shot neural machine translation [Электронный ресурс]: URL:<https://doi.org/10.48550/arXiv.1903.07091> (дата обращения: 15.02.23).
- [10] Vaswani A., Shazeer N. et al. Attention is all you need // In Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017. P. 6000–6010.
- [11] Tiedemann J., Parallel Data, Tools and Interfaces in OPUS. In Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'2012)
- [12] Lison P. and Tiedemann J., 2016, OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)
- [13] Kishore Papineni, Salim Roukos. et al. Bleu: a Method for Automatic Evaluation of Machine Translation // In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. 2002. P. 311–318.
- [14] Przybocki M., Peterson K. et al. The NIST 2008 Metrics for machine translation challenge—overview, methodology, metrics, and results // Machine Translation 23. 2009. P 71-103.
- [15] Jacob D., Ming-Wei C. et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL-HLT. № 1. 2019. P 4171–4186.

Приложения

Код

Реализация метода улучшения с использованием лексического сходства доступна в GitHub-репозитории по ссылке:

`https://github.com/RolfAdolf/graduate-project`