КОМПЬЮТЕРНАЯ ЛИНГВИСТИКА И ОБРАБОТКА ЕСТЕСТВЕННОГО ЯЗЫКА

УДК 004.896 ISSN 1995-5499
DOI: https://doi.org/
Поступила в редакцию 10.03.2022
Подписана в печать 22.04.2022

МНОГОЯЗЫЧНЫЙ МАШИННЫЙ ПЕРЕВОД С ПОМОЩЬЮ ИЕРАРХИЧЕСКОГО ТРАНСФОРМЕРА

© 2022 А. М. Хусаинова✉, В. А. Романов, А. М. Хан

Университет Иннополис
ул. Университетская, 1, 420500 Иннополис, Российская Федерация

Аннотация. Выбор стратегии распределения параметров между языками в моделях многоязычного машинного перевода определяет то, насколько оптимально используется пространство параметров. Следовательно, выбранная стратегия напрямую влияет на конечное качество перевода. Данная работа исследует новый подход к организации параметров в многоязычном машинном переводе на основе лингвистических деревьев, которые показывают степень родства между различными языками. Основная идея заключается в том, чтобы использовать эти экспертные языковые иерархии в качестве основы для архитектуры модели: чем ближе два языка, тем больше у них должно быть общих параметров.

Мы испытываем эту идею для архитектуры Трансформер и показываем, что, несмотря на успех в предыдущих работах, существуют проблемы, присущие обучению таких иерархических моделей. Мы демонстрируем, что при специально подобранной стратегии обучения иерархическая архитектура может превзойти как простые двуязычные модели, так и многоязычные модели перевода с общим пространством параметров.

Ключевые слова: нейронный машинный перевод, многоязычный перевод, организация параметров, языковые деревья, иерархическая архитектура, низкоресурсный перевод, родственные языки.

ВВЕДЕНИЕ

В настоящее время качество машинного перевода (МП) постепенно приближается к человеческому, однако это справедливо только при наличии массивных параллельных данных. Что же касается машинного перевода для низкоресурсных языковых пар, основной способ улучшить его качество заключается в использовании дополнительных данных. Это могут быть как одноязычные тексты (на исходном либо целевом языке), так и релевантные параллельные тексты, например, для родственных языков. При наличии таких параллельных текстов можно построить многоязычную модель перевода, в которой различные языки будут совместно использовать некоторые параметры. Учитывая, что языки, в особенности родственные, имеют много общего, правильно организованная стратегия совместного использования параметров может компенсировать недостаток обучающих примеров в низкоресурсных парах.

Исследования [1] показывают, что использование родственности языков может существенно повысить качество перевода. Нас интересует то, какая архитектура модели МП позволяет извлечь наибольшую выгоду из межъязыковой схожести. К данной задаче существуют различные подходы, такие как полное совместное использование параметров [2] или совместное использование энкодера с раздельными декодерами для каждого выход-

✉ Хусаинова Альбина Маратовна
e-mail: a.khusainova@innopolis.ru

ного языка [3]. Однако, мы считаем недавний подход [4] более перспективным, поскольку он систематически учитывает степень родства между языками в многоязычной модели.

Лингвистические деревья упорядочивают языки в иерархии по степени родства, и этот же подход может быть применен к многоязычным моделям машинного перевода. Идея заключается в иерархической организации энкодера и декодера, отражающей степень родства между языками таким образом, что наиболее родственные языки имеют наибольшее количество общих параметров. На рис. 1 показана схема такой модели (более подробно она будет описана в разделе 1.1).

В отличие от работы [4], в которой использовалась рекуррентная нейронная сеть Long Short Term Memory (LSTM) в качестве базовой архитектуры, мы реализовали эту идею с помощью современной архитектуры Трансформер [5]. Наши эксперименты продемонстрировали некоторые проблемы обучения иерархической модели, а именно, модель склонна к раннему переобучению в низкоресурсных направлениях, что приводит к низкому качеству перевода. Учитывая, что одной из основных целей внедрения многоязычных моделей в целом и иерархической модели в частности является повышение точности перевода именно для низкоресурсных направлений, эта проблема является критичной и мы предлагаем ее решение.

Основной вклад данной работы заключается в следующем:

1. Мы протестировали иерархическую многоязычную модель машинного перевода на основе архитектуры Трансформер;

2. Мы выявили и проанализировали проблемы, связанные с иерархической природой модели, а именно, переобучение в низкоресурсных направлениях;

3. Мы предложили и протестировали несколько способов решения проблемы переобучения, которые можно обобщить как различные формы регуляризации.

Анализ предшествующих работ

В этом разделе мы рассмотрим существующие подходы к организации параметров в многоязычных моделях МП и сравним их с иерархическим подходом.

Вероятно, самая простая форма распределения параметров была представлена в работе [2], где все параметры являются общими, а для различения выходных языков используются идентификаторы. Ввиду отсутствия каких-либо архитектурных подсказок, такая модель может потребовать больше ресурсов для определения отношений между языками.

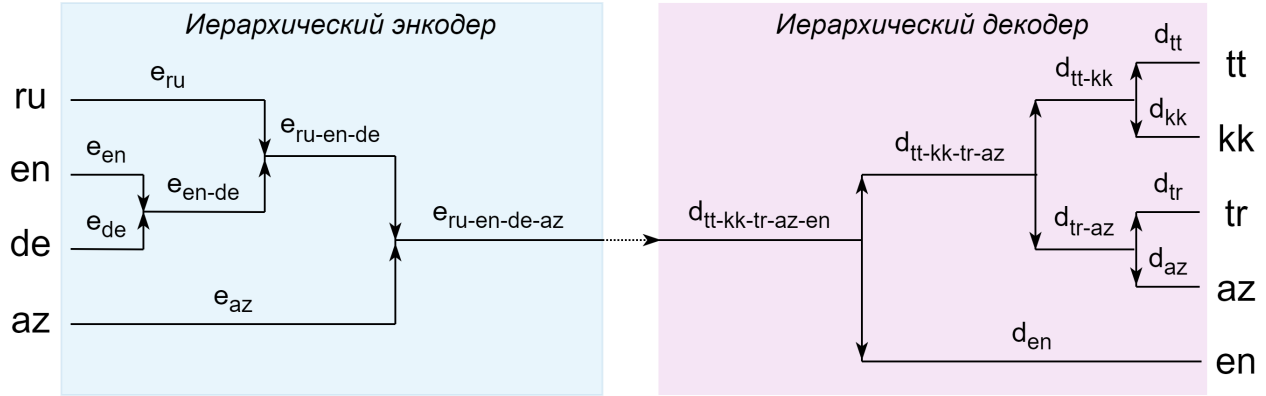

Рис. 1. Высокоуровневый вид многоязычной иерархической модели (пример). Горизонтальные линии обозначают блоки энкодера / декодера, вертикальные линии и стрелки показывают точки слияния / разделения, а подстрочные индексы указывают на то, какие именно языки совместно используют конкретный энкодер / декодер. Здесь и ниже приведены коды языков BCP-47 [18]

[Fig. 1. A high-level view of a sample multilingual hierarchical model. Horizontal lines denote encoder/decoder blocks, vertical lines and arrows show points of merging/splitting, and subscripts clarify which languages share a particular encoder/decoder. BCP-47 language codes are given here and below]

В [3] проблема перевода рассматривается как многозадачная, и предлагается архитектура с общим энкодером и отдельными декодерами для каждого целевого языка. Недостаток этого подхода в том, что здесь не учитывается возможность потенциального обмена информацией между целевыми языками.

В [6] и [7] вводят общие параметры между энкодером и декодером. Это интересная идея, которая может быть рассмотрена в будущем.

В работе [8] предлагают модель «один ко многим», где часть параметров совместно используется несколькими декодерами. По духу это похоже на иерархический подход, однако, в нашем случае вместо распределения параметров между отдельными декодерами идея заключается в построении иерархии декодеров.

Многие другие недавние работы, такие как [9] и [10] добавляют специфические для языка компоненты в декодер.

Мы наблюдаем, что в этих работах языки на стороне энкодера / декодера в многоязычных моделях рассматриваются одинаково, независимо от степени их родства. В одной из работ [10] языки группируются на основе пересечения их словарей, тем не менее, в их модели близкие языки могут оказаться в разных группах. В иерархической модели, однако, количество общих параметров напрямую зависит от степени родства между языками.

1. МЕТОДЫ И МАТЕРИАЛЫ

Моделирование генеалогических отношений между языками с помощью деревьев — это распространенный и давно существующий подход в лингвистике [11]. Эти экспертные знания о взаимоотношениях между языками могут быть использованы при построении многоязычной модели МП. Современная тенденция в обработке естественного языка состоит в разработке таких моделей, которые способны самостоятельно улавливать лингвистические правила и паттерны, без явного руководства. Однако в данном случае, когда экспертные знания легко доступны и для их получения не требуется дополнительных ресурсов, эти знания могут позволить моделям быстрее обучаться и эффективнее организовывать пространство параметров.

В [12] обнаружили, что разные уровни энкодера специализируются на разных аспектах языка: представления низшего уровня лучше улавливают информацию о частях речи, а представления высшего уровня лучше улавливают семантику.

Другое наблюдение [13] показало, что в многоязычных моделях МП с единым общим энкодером представления разных языков сначала группируются по языковым семьям, но по мере продвижения по энкодеру представления для разных исходных языков становятся все более схожими.

Эти факты в совокупности говорят о том, что энкодер пытается найти общее представление для различных языков, которые изначально группируются по языковым семьям. По мере продвижения вглубь энкодера модель находит промежуточные представления, которые сглаживают различия на разных уровнях — морфологическом, синтаксическом, семантическом.

Это очень напоминает структуру лингвистических деревьев, где связи на нижнем уровне означают наибольшее сходство между языками, а связи на самом высоком уровне иерархии говорят о том, что языки далеки друг от друга. Пример такого дерева приведен на рис. 2. Если организовать архитектуру многоязычной модели МП в соответствии с отношениями языков в лингвистическом дереве, это позволит языкам обмениваться параметрами на соответствующих уровнях.

Например, возьмем два очень близких языка, словари которых значительно пересекаются. Поскольку эти языки сильно похожи,

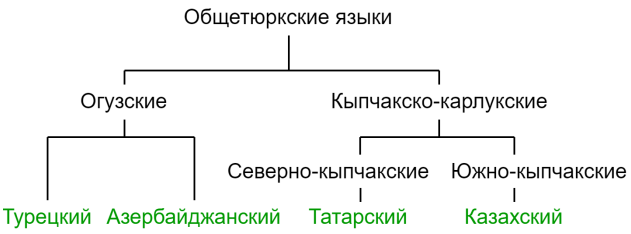

Рис. 2. Фрагмент лингвистического дерева для тюркских языков [14]
[Fig. 2. A fragment of the phylogenetic tree for Turkic languages]

мы предполагаем, что сведение их к одному представлению не займет много времени. Поэтому мы «объединяем» их на ранней стадии, т. е. вводим общие параметры для этих языков на первых уровнях энкодера. Третий язык, предположим, происходит из той же семьи, но многие базовые слова отличаются. Тем не менее, структура предложения остается такой же, как и в первых двух языках. Поэтому мы объединяем этот язык с первыми двумя на более поздних этапах (совместные параметры появляются глубже в слоях энкодера), и так далее. Такая стратегия позволяет экономично использовать пространство параметров и потенциально может привести к более качественному, т. е. независимому от исходного языка представлению на выходном слое энкодера.

1.1. Описание модели

Основная идея заключается в том, чтобы организовать параметры в энкодере и декодере в соответствии с лингвистическим сходством между языками. То есть, каждая сторона модели строится как иерархия, что соответствует тому как языки соединяются в лингвистических деревьях: чем ближе два языка, тем больше у них общих параметров.

Пример иерархической модели приведен на рис. 1. В данном примере мы видим четыре исходных и пять целевых языков, которые связаны цепью иерархически организованных энкодеров и декодеров. Рассмотрим кодирующую сторону (слева). Все исходные языки имеют свои собственные энкодеры e_{ru}, e_{en}, e_{de}, e_{az}, и параметры этих энкодеров не используются совместно с другими языками, поскольку они предназначены для изучения специфических особенностей конкретного языка. Глубже в модели представления на выходе некоторых энкодеров складываются и передаются общим энкодерам (связи обозначены стрелками). Эти энкодеры ($e_{en\text{-}de}$, $e_{ru\text{-}en\text{-}de}$) являются общими для двух или более языков и предназначены для обнаруживания признаков, общих для этих языков. Наконец, последний энкодер $e_{ru\text{-}en\text{-}de\text{-}az}$, который является общим для всех исходных языков, объединяет представления на выходе всех оставшихся энкодеров.

Эта архитектура разработана для агрегации знаний о различных языках на разных уровнях. В данном примере сначала объединяются параметры для английского и немецкого языков, поскольку они оба происходят из германской ветви индоевропейской языковой семьи. Далее к ним примыкает русский язык, поскольку он принадлежит к другой ветви той же языковой семьи. Позже к этим индоевропейским языкам присоединяется азербайджанский язык, который происходит из совершенно другой тюркской языковой семьи. Логика на стороне декодера аналогична — наиболее далекие языки разделяются первыми.

В нашей модели мы предлагаем задавать одинаковое количество параметров на любом пути от исходного языка к целевому. Например, если сравнивать направления перевода ru-tt и az-en на рис. 1, то, хотя az-en имеет значительно меньше общих параметров, суммарное количество параметров одинаково для обоих направлений. Именно поэтому некоторые блоки энкодера длиннее других, указывая на то, что, например, количество параметров в e_{az} должно быть таким же, как суммарно в e_{ru} и $e_{ru\text{-}en\text{-}de}$.

В целом, можно сказать, что предлагаемая модель сходна с [4], но есть важные отличия. Мы используем другую базовую модель — Трансформер вместо LSTM, мы не ограничиваем количество слоев и не урезаем языковые семьи. Есть также различия в процедуре обучения, которые будут описаны в разделе 2.2. И, что самое главное, мы выявили проблему, характерную для обучения иерархических моделей, и предложили улучшенные методы обучения для таких моделей, которые будут описаны в разделе 2.3.2.

2. РЕЗУЛЬТАТЫ И ИХ ОБСУЖДЕНИЕ

Для тестирования иерархического подхода к распределению параметров мы провели серию экспериментов с многоязычными моделями МП с разной глубиной иерархии. В отличие от [4], где рассматриваются только сложные модели с большим числом входных и выход-

ных языков, мы решили начать с простых моделей, в которых иерархия находится только на одной стороне (энкодер или декодер), а затем увеличить сложность до общего случая.

Мы проверили эффективность иерархической модели в трех различных случаях:

1. Простой случай с двумя родственными исходными языками и одним целевым языком (тип 1), см. рис. 3.

2. Простой случай с одним исходным языком и двумя родственными целевыми языками (тип 2).

3. Общий случай с несколькими языками разной степени родства как на стороне энкодера, так и на стороне декодера (тип 3), см. рис. 4.

Для каждого из этих случаев мы обучили по две иерархические модели, используя разные наборы языков. Далее, после выявления специфических проблем обучения иерархических моделей (описанных в разделе 2.3), мы предложили и применили две вариации к каждой иерархической модели.

Каждую иерархическую модель мы сравниваем с двумя базовыми моделями: ***двуязычной моделью МП и многоязычной моделью МП с общим пространством параметров***. Сравнение с двуязычными моделями (с одним входным и одним выходным языком), обученными на тех же параллельных данных, помогает понять, могут ли языки в иерархической модели «учиться друг у друга». В то время как сравнение иерархических моделей с моделями с полным совместным использованием параметров, с одним общим энкодером и декодером для всех языков, как в [2], отвечает на вопрос насколько целесообразно явно задавать иерархию в модели МП.

В каждой многоязычной модели, которую мы обучали, есть низкоресурсные и высокоресурсные направления, и мы исследовали, способны ли низкоресурсные обучаться у высокоресурсных, и какая архитектура модели лучше подходит для этой цели.

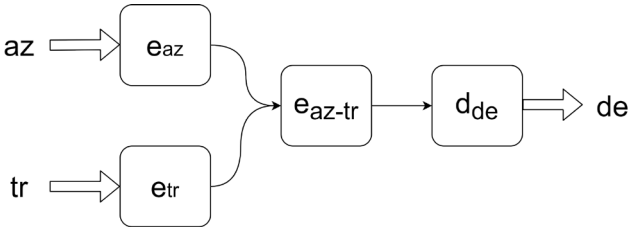

Рис. 3. Пример простой иерархической модели с двумя исходными и одним целевым языком. Закругленные прямоугольники обозначают блоки энкодера / декодера

[Fig. 3. The example of a simple hierarchical model with two source and one target language. Rounded rectangles denote encoder/decoder blocks]

2.1. Данные

Чтобы результаты обучения моделей были сравнимыми, мы использовали четыре параллельных корпуса из одного набора данных JW300 [15]:

1. Турецко-немецкий (tr-de), 500 тыс. предложений.

2. Азербайджанско-немецкий (az-de), 110 тыс.

3. Англо-польский (en-pl), 500 тыс.

4. Немецко-польский (de-pl), 110 тыс.

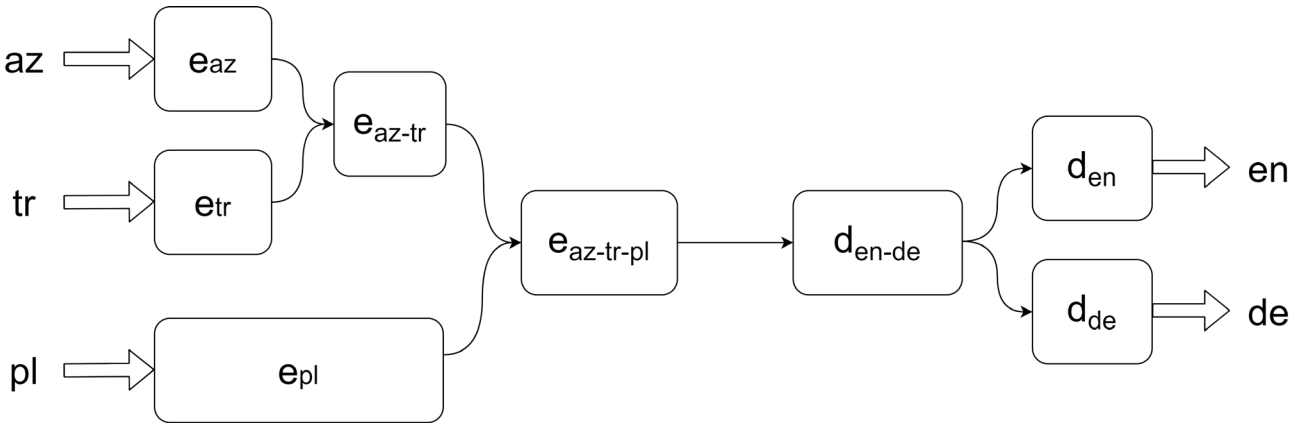

Рис. 4. Пример общей иерархической модели с несколькими исходными и целевыми языками

[Fig. 4. The example of a general multi-source multi-target hierarchical model]

Турецкий и азербайджанский языки являются родственными и относятся к огузской группе тюркской языковой семьи. Английский, немецкий и польский языки принадлежат к индоевропейской языковой семье. Английский и немецкий языки более родственны, так как они относятся к германской группе языков, в то время как польский относится к балто-славянской группе.

Для обучения простых моделей с иерархией на одной стороне (тип 1 и 2) мы использовали корпусы tr-de и az-de. Получившиеся модели можно обозначить как az-tr → de (рис. 3) и de → az-tr. Аналогично, на основе корпусов en-pl и de-pl получены модели en-de → pl и pl → en-de.

Для общих иерархических моделей (тип 3) были использованы все четыре корпуса вместе, на их основе были получены две модели: az-tr-pl → en-de (рис. 4) и en-de → az-tr-pl.

Размеры корпусов различны: tr-de и en-pl в нашем случае — высокоресурсные языковые пары, а az-de и de-pl — низкоресурсные пары.

Все корпусы были отфильтрованы по максимальной длине предложения в 40 BPE токенов [16].

2.2. Модель и ее обучение

В данной работе система многоязычного машинного перевода реализована на основе архитектуры Трансформер [5] с сокращенным количеством параметров:

- d_model = 128
- dff = 512
- num_heads = 8
- dropout_rate = 0.1

У всех иерархических и двуязычных моделей имеется 6 слоев Трансформера на пути от любого исходного языка к любому целевому языку. Таким образом, любая двуязычная модель состоит из 3 слоев энкодера и 3 слоев декодера. Модель первого типа (с иерархическим энкодером) состоит из 1 слоя в каждом индивидуальном энкодере, 2 слоев в общем энкодере и 3 слоев в декодере. Добавление большего количества языков в общую иерархическую модель третьего типа приводит к увеличению общего количества слоев, но между любой парой исходный-целевой язык все равно остается 6 слоев.

Чтобы модели с общим пространством параметров были сравнимыми с иерархическими моделями, мы задаем одинаковое общее количество слоев в энкодере и декодере. Например, модель с общим пространством параметров, обученная для сравнения с иерархической моделью первого типа, состоит из 4 слоев энкодера и 3 слоев декодера (соответствует 1+1+2 и 3). Однако, когда мы перешли к общему случаю с большим количеством слоев (тип 3), модель с общим пространством параметров не удалось ничему обучить. Так, для en-de → az-tr-pl проблему решило удвоение размера блоков выборки (batch size), а для az-tr-pl → en-de пришлось уменьшить количество слоев в энкодере с 6 до 4.

В [4] авторы не упомянули, обеспечивали ли они одинаковый размер иерархической и базовой модели, что затрудняет сравнительный анализ результатов.

Ввиду разного размера корпусов для тренировки, при обучении иерархических моделей мы использовали избыточную выборку (oversampling) из низкоресурсных корпусов. Однако для моделей с общим пространством параметров оказалось, что это снижает их производительность, поэтому мы не стали ее применять.

Что касается процедуры обучения, одним из способов обучения иерархической модели является чередование всех направлений перевода в системе, как в [4]. В этом случае возникает опасение, что параметры модели могут начать колебаться между этими направлениями. Поэтому мы решили одновременно подавать обучающие данные со всех исходных языков на соответствующие энкодеры, складывая представления в точках перехода специфических энкодеров в общие, и передавать их по цепочке декодеров вплоть до индивидуальных декодеров. Мы обучали все модели с размером блока выборки равным 128 до сходимости (максимум 50 эпох) и представляем лучшие результаты BLEU за все эпохи. При обучении иерархических моделей мы складываем полноразмерные (128) блоки

выборки различных языковых пар в энкодере и разделяем их в декодере.

В целях содействия обмену знаниями в многоязычных моделях мы используем общие словари для исходных и (отдельно) целевых языков.

2.3. Результаты и анализ

В данном разделе мы представляем и обсуждаем результаты проведенных экспериментов.

2.3.1. Оценка иерархической модели

Для оценки иерархического подхода мы использовали две базовые модели: двуязычную и модель с общим пространством параметров. Мы рассматривали несколько показателей. Во-первых, мы посчитали среднюю разницу в BLEU между многоязычными и двуязычными моделями. Для этого мы усреднили значения BLEU между всеми 16 направлениями перевода во всех обученных моделях с общим пространством параметров и, отдельно, во всех иерархических моделях. Во-вторых, мы разделили направления перевода на низкоресурсные (8) и высокоресурсные (8), и вычислили среднюю разницу в баллах BLEU только для высокоресурсных направлений, и, в-третьих, только для низкоресурсных направлений.

Эта информация визуализирована на рис. 5. Высота столбцов отражает величину разницы, а их направление (вниз / вверх) показывает, улучшаются или ухудшаются показатели по сравнению с базовой двуязычной моделью. На данный момент нас интересуют только модели с общим пространством параметров (*Full*) и обычные (описанные выше) иерархические модели (*Hie*).

Слева на рис. 5 приведена средняя разница в BLEU, и мы видим, что модели с общим пространством параметров в среднем работают почти так же, как двуязычные, а иерархические даже немного хуже (–0,29). Однако сравнение становится более информативным, если рассматривать отдельно высокоресурсные и низкоресурсные направления. Хотя в среднем может показаться, что модели с общим пространством параметров работают так же, как и двуязычные, теперь становится ясно, что на самом деле это происходит потому, что низкоресурсные направления улучшаются, обучаясь на родственных параллельных данных (+1,28), а высокоресурсные — ухудшаются (–1,26).

Подобная картина наблюдается и в иерархических моделях: низкоресурсные направления выигрывают от многоязычия (+1,31), а высокоресурсные страдают от него (–1,88) даже больше, чем модели с общим пространством параметров.

Таким образом, мы видим две проблемы: ухудшение показателей направлений с высокими ресурсами в многоязычных моделях в целом; и низкая производительность иерархических моделей по сравнению с моделями с общим пространством параметров. Первая проблема является общей, она наблюдалась в более ранних работах по многоязычным моделям МП [6] и поэтому в рамках данной статьи рассматриваться не будет. Однако вторая проблема является ставит под сомнение наше предположение о полезности иерархической организации многоязычной модели, и, следовательно, требует исследования.

Было бы интересно сравнить наши результаты с [4], однако их способ представления результатов не позволяет нам это сделать. Они усредняют значения BLEU по исходным языкам, и не делают строгого разделения на высокоресурсные и низкоресурсные пары. Согласно их результатам, иерархическая модель работает лучше, чем все другие базовые модели, включая двуязычные и модели с общим пространством параметров. Однако в используемом ими наборе данных GlobalVoices [17] абсолютное большинство пар очень низкоресурсные. Таким образом, даже если значения BLEU для высокоресурсных пар в многоязычных моделях ухудшатся, это не будет заметно, если для большинства пар, которые являются низкоресурсными, значения BLEU улучшатся. Мы не можем утверждать, что это так, но вполне вероятно, что качество перевода высокоресурсных пар в их иерархических моделях также ухудшается. Этот вопрос мог

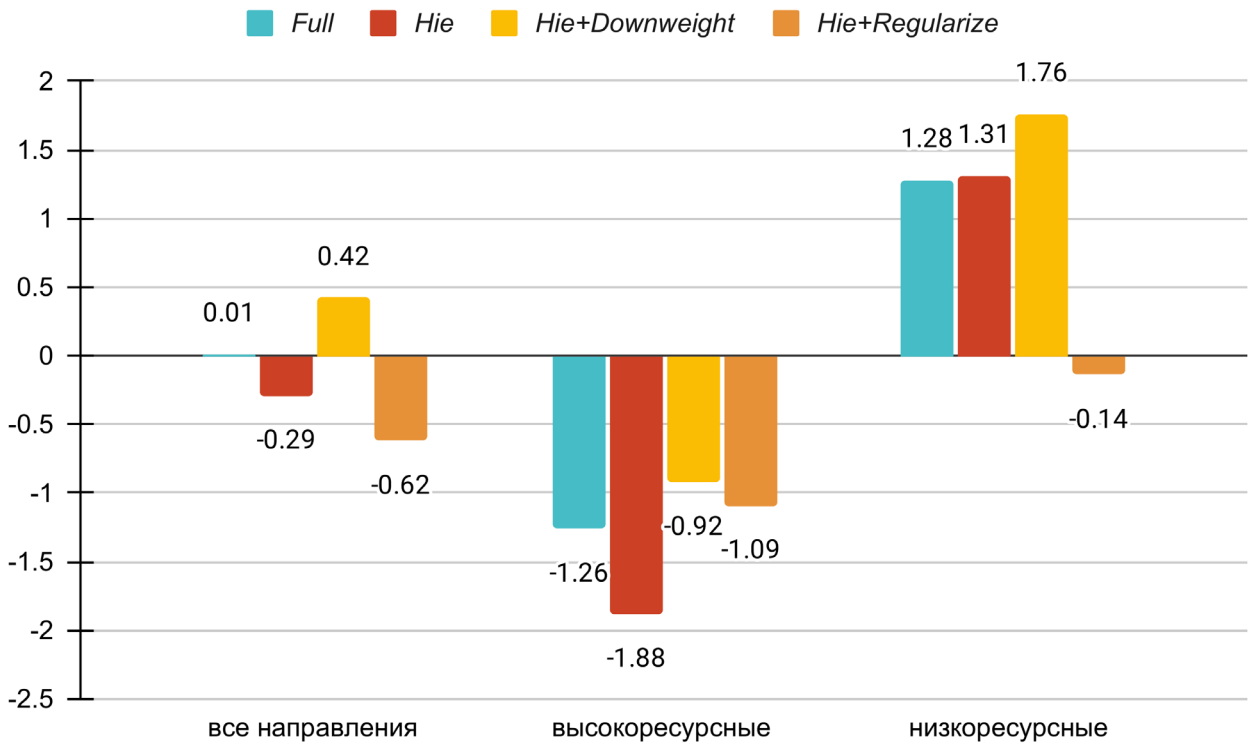

Рис. 5. Разница в баллах BLEU между многоязычными и двуязычными моделями. Нулевой уровень представляет собой средний балл двуязычных моделей. Положительные значения означают улучшение по сравнению с базовыми двуязычными моделями, отрицательные значения, соответственно, означают ухудшение. Full обозначает модели с общим пространством параметров, Hie обозначает обычные иерархические модели, Hie+Downweight обозначает иерархические модели, обученные путем понижения веса низкоресурсных тренировочных данных, и Hie+Regularize обозначает иерархические модели с регуляризацией с помощью высокоресурсных данных

[Fig. 5. Difference in BLEU scores between multilingual models and bilingual models. Zero level represents average bilingual models' score. Positive values mean improvement over bilingual baseline, negative values, respectively, signify decrease. Full denotes models with full parameter sharing, Hie stands for hierarchical models, Hie+Downweight is for hierarchical models trained by down-weighting low-resource samples, and Hie+Regularize is for hierarchical models regularized using high-resource samples]

бы быть прояснен на рис. 6 их статьи, где они группируют языковые пары по размеру корпуса, но, к сожалению, они решили опустить результаты для языковых пар с относительно большим объемом корпуса (более 100 тыс.).

Аналогично, может быть и так, что высокоресурсные пары в их моделях с общим пространством параметров (называемых там «one-to-one») имеют более высокие оценки, чем иерархические модели.

В целом, трудно сравнивать результаты в аспекте высокоресурсных / низкоресурсных пар из-за разницы в экспериментах. А именно, направления, которые в данной работе считаются низкоресурсными (100 тыс.), там могут считаться высокоресурсными. Кроме того, распределение направлений с высоким и низким ресурсом различно: в нашем случае половина направлений в многоязычных моделях являются высокоресурсными, в их же случае они составляют меньшинство.

Подведя итог, можно сказать, что выявленные проблемы могут иметь место и в [4], поэтому наши нижеописанные выводы могут внести весомый вклад в общую идею иерархической модели машинного перевода.

2.3.2. Улучшение иерархической модели

Проблема, которую мы выявили, заключается в том, что иерархическая модель не превосходит по качеству модель с общим

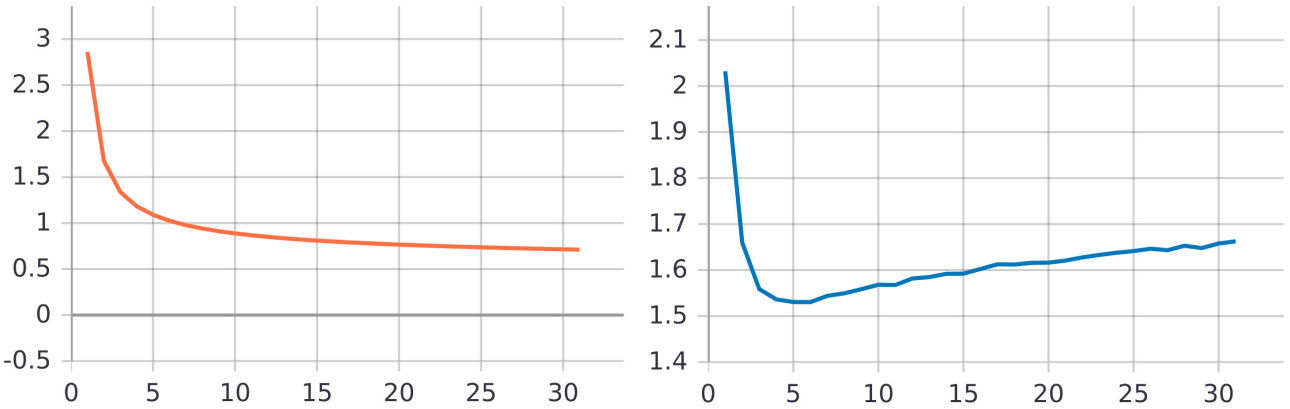

Рис. 6. Пример переобучения в иерархических моделях. Значения функции потерь на обучающей (слева) и валидационной (справа) выборках для низкоресурсной пары pl-de при обучении иерархической модели pl → en-de

[Fig. 6. The example of overfitting in hierarchical models. Train (left) and validation (right) losses for low-resource pl-de pair when training pl → en-de hierarchical model]

пространством параметров, как это должно было быть согласно нашему предположению. Иерархическая идея не работает так, как ожидалось — она показывает почти такое же улучшение для низкоресурсных пар и большее ухудшение для высокоресурсных пар.

В попытке решить эту проблему мы исследовали динамику обучения (значения функции потерь на обучающей и валидационной выборках) для обоих типов моделей. Оказалось, что существует постоянная проблема с обучением иерархических моделей: для всех обученных моделей наблюдается явное переобучение для низкоресурсных пар, см. рис. 6, в то время как высокоресурсные пары обучаются нормально. Вместе с тем результаты для низкоресурсных пар улучшились по сравнению с базовыми двуязычными моделями. Вероятно, иерархическая структура позволяет языкам «учиться друг у друга», но этому мешает проблема переобучения, и, если ее исправить, значения BLEU должны вырасти еще больше.

В то же время проблема переобучения никогда не возникает в моделях с общим пространством параметров (даже когда мы делаем избыточную выборку для низкоресурсных пар). Этому может быть несколько объяснений. В иерархических моделях различные пути перевода имеют одинаковое количество параметров независимо от размера корпуса. Это означает, что низкоресурсных пары могут иметь слишком много параметров для имеющегося у них объема данных, что может привести к переобучению. Высокоресурсным же парам может потребоваться больше параметров, чем низкоресурсным. В этом смысле модели с общим пространством параметров могут быть более эффективными, поскольку пространство параметров распределяется между языками автоматически. Другое возможное объяснение заключается в том, что в моделях с общим пространством параметров тренировочные данные высокоресурсных пар действуют как неявный регуляризатор, предотвращая чрезмерную специализацию параметров модели на низкоресурсных парах. Это не совсем так в иерархических моделях — там имеются индивидуальные слои, отдельные для каждого языка, а также слои общие для нескольких близких низкоресурсных языков.

Еще одной возможной проблемой является избыточная выборка (oversampling) данных низкоресурсных пар. Мы применяли ее при обучении иерархических моделей, и это само по себе может быть причиной переобучения. Действительно, в наших первоначальных экспериментах без избыточной выборки мы не наблюдали переобучения, однако в этом случае результаты для низкоресурсных пар были значительно ниже. Поэтому мы решили подойти к проблеме переобучения используя другой подход, а не убирая избыточную выборку.

Исходя из этих рассуждений, мы предлагаем два способа решения проблемы переобучения:

1. Уменьшить вес низкоресурсных тренировочных данных, т. е. уменьшить их вес в функции потерь пропорционально дисбалансу данных. Под дисбалансом здесь понимается отношение объема низкоресурсного корпуса к объему ближайшего высокоресурсного корпуса. Например, если это отношение равно 1:5, то низкоресурсные тренировочные данные имеют в пять раз меньший вес в функции потерь. Идея в том, что таким образом будет осуществляться регуляризация низкоресурсных направлений и в то же время высокоресурсные данные будут иметь большее влияние в общих частях, что позволит улучшить и результаты для высокоресурсных направлений.

2. Регуляризация низкоресурсных направлений с помощью высокоресурсных данных. Эта идея основана на вышеизложенной гипотезе о неявной регуляризации, происходящей в моделях с общим пространством параметров. Здесь мы явно применяем эту регуляризацию, подавая ближайшие (в плане родства языков) высокоресурсные данные вместо низкоресурсных данных, предназначенных для данного направления. То есть, в течение эпохи в заданном направлении подаются все доступные соответствующие низкоресурсные данные (единожды, без избыточной выборки) плюс родственные высокоресурсные данные. Мы уменьшаем вес высокоресурсных данных, используемых для регуляризации, пропорционально дисбалансу данных, чтобы ограничить возможный негативный эффект.

Мы применили обе эти идеи и обе они решили проблему переобучения. Теперь снова обратимся к рис. 5. Как из него видно, уменьшение веса низкоресурсных данных (*Hie+Downweight*) принесло значительные улучшения. Во-первых, модель теперь работает в среднем лучше, чем обе предыдущие модели (+0,42). Во-вторых, результаты для высокоресурсных пар значительно улучшились — с –1,88 до –0,92. И, в-третьих, показатели для низкоресурсных пар также заметно выросли — с +1,31 до +1,76. Что касается второго подхода (*Hie+Regularize*), то он также улучшил результаты для высокоресурсных пар (–1,09), но сильно ударил по низкоресурсным парам. Возможно, причина в излишней регуляризации, препятствующей нормальному обучению.

Итак, подводя итог, можно сказать, что оба подхода решили проблему переобучения. И если регуляризация с помощью высокоресурсных данных оказалась слишком строгой, то уменьшение веса низкоресурсных данных позволило значительно улучшить результаты как для высокоресурсных, так и для низкоресурсных направлений.

Теперь, взяв улучшенную версию иерархической модели в качестве основной, попытаемся ответить на вопросы, поставленные нами в начале раздела 2. Нам было интересно узнать, имеет ли значение на какой стороне находится иерархия — в энкодере или декодере. На основании наших данных можно сказать, что существенной разницы нет: иерархическая модель работает на одном уровне, независимо от того, где находится иерархия. Однако, что интересно, простые иерархические модели работают значительно лучше сложных: высокоресурсные направления ухудшаются меньше (в среднем –0,52 против –1,33), а низкоресурсные улучшаются больше (+2,02 против +1,49). Это может происходить потому, что добавление большего количества языков в многоязычную модель неизбежно приводит к смешению более удаленных пар, и они могут мешать друг другу вместо того, чтобы помогать учиться.

Сравнение иерархических моделей с двуязычными подтвердило наше предположение о том, что иерархическая архитектура позволяет языкам «учиться друг у друга», однако это верно в основном для низкоресурсных пар. Если посмотреть на конкретные (не усредненные) баллы BLUE, то показатели высокоресурсных направлений не всегда ухудшаются: в 2 из 8 случаев они улучшаются, хотя и незначительно. Результаты для низкоресурсных направлений улучшаются во всех 8 случаях.

Что касается сравнения иерархической модели с моделью с общим пространством

параметров, то улучшенная иерархическая модель превосходит ее как для высокоресурсных (+0,34), так и для низкоресурсных пар (+0,48). В то же время, модель с общим пространством параметров эффективна в том смысле, что она обладает естественным механизмом регуляризации, и, возможно, этот аспект может быть улучшен также и в иерархических моделях.

ЗАКЛЮЧЕНИЕ

В данной работе был исследован новый иерархический подход в многоязычном машинном переводе. Мы реализовали иерархическую модель на основе архитектуры Трансформер и сравнили ее с двумя базовыми моделями: двуязычной и моделью с общим пространством параметров. Оказалось, что обычная стратегия обучения ограничивает возможности иерархической модели. Конкретно, мы выявили проблему переобучения, которая характерна для обучения иерархических моделей. Регуляризация низкоресурсных направлений решила эту проблему, существенно улучшив производительность модели.

Мы показали, что использование иерархической модели позволяет значительно улучшить качество перевода низкоресурсных пар, однако, для высокоресурсных направлений качество перевода чаще понижается. Также результаты сравнения с моделью с общим пространством параметров подтверждают наше предположение о полезности явного определения стратегии совместного использования параметров в многоязычных моделях.

В целом, иерархический подход выглядит перспективным и, следовательно, нуждается в дальнейшем изучении и развитии. В будущем мы бы хотели проанализировать, как именно происходит обучение в иерархических моделях, и действительно ли иерархическая архитектура способна отражать различные языковые аспекты (морфологические, синтаксические, семантические) на разных уровнях модели.

КОНФЛИКТ ИНТЕРЕСОВ

Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

СПИСОК ЛИТЕРАТУРЫ

1. *Tan X., Chen J., He D., Xia Y., Qin T., Liu T. Y.* Multilingual Neural Machine Translation with Language Clustering // In EMNLP/IJCNLP. – 2019.

2. *Johnson M., Schuster M., Le Q. V., Krikun M., Wu Y., Chen Z. [et al.]* Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation // Transactions of the Association for Computational Linguistics. – 2017. – 5. – P. 339–351.

3. *Dong D., Wu H., He W., Yu D., Wang H.* Multi-Task Learning for Multiple Language Translation // In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Vol. 1: Long Papers); 2015 Jul; Beijing: Association for Computational Linguistics. – P. 1723–1732.

4. *Azpiazu I. M., Pera M. S.* A Framework for Hierarchical Multilingual Machine Translation. – 2020.

5. *Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N. [et al.]* Attention is All you Need. In Guyon I., Luxburg U. V., Bengio S., Wallach H., Fergus R., Vishwanathan S. [et al.] editors. Advances in Neural Information Processing Systems 30.: Curran Associates, Inc. – 2017. – P. 5998–6008.

6. *Firat O., Cho K., Bengio Y.* Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism // In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2016 Jun; San: Association for Computational Linguistics. – P. 866–875.

7. *Wang Y., Zhou L., Zhang J., Zhai F., Xu J., Zong C.* A Compact and Language-Sensitive Multilingual Translation Method // In Proceedings of the 57th Annual Meeting of the Association for

Computational Linguistics; 2019 Jul; Florence: Association for Computational Linguistics. – P. 1213–1223.

8. *Sachan D., Neubig G.* Parameter Sharing Methods for Multilingual Self-Attentional Translation Models. In Proceedings of the Third Conference on Machine Translation: Research Papers; 2018 Oct; Brussels: Association for Computational Linguistics. p. 261–271.

9. *Bapna A., Arivazhagan N., Firat O.* Simple, Scalable Adaptation for Neural Machine Translation. In EMNLP/IJCNLP. – 2019.

10. *Fan A., Bhosale S., Schwenk H., Ma Z., El-Kishky A., Goyal S. [et al.]* Beyond English-Centric Multilingual Machine Translation. ArXiv. 2020; abs/2010.11125.

11. *Schleicher A., Schleicher S.* Die ersten Spaltungen des indogermanischen Urvolkes [The first splits of the Proto-Indo-European people]. – 1853.

12. *Belinkov Y., Màrquez L., Sajjad H., Durrani N., Dalvi F., Glass J.* Evaluating Layers of Representation in Neural Machine Translation on Part-of-Speech and Semantic Tagging Tasks // In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers); 2017 Nov; Taipei: Asian Federation of Natural Language Processing. – P. 1–10.

13. *Kudugunta S., Bapna A. Caswell I, Firat O.* Investigating Multilingual NMT Representations at Scale // In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); 2019 Nov; Hong: Association for Computational Linguistics. – P. 1565–1575.

14. *Savelyev A., Robbeets M.* Bayesian phylolinguistics infers the internal structure and the time-depth of the Turkic language family // Journal of Language Evolution. – 2020 Feb.

15. *Agić Ž., Vulić I.* JW300: A Wide-Coverage Parallel Corpus for Low-Resource Languages // In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics; 2019 Jul; Florence: Association for Computational Linguistics. – P. 3204–3210.

16. *Sennrich R., Haddow B., Birch A.* Neural Machine Translation of Rare Words with Subword Units // In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7–12, 2016, Berlin, Germany, Volume 1: Long Papers; 2016: The Association for Computer Linguistics.

17. *Tiedemann J.* Parallel Data, Tools and Interfaces in OPUS. In Chair) NC(, Choukri K, Declerck T, Dogan MU, Maegaard B, Mariani J, et al., editors. Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12); 2012 May; Istanbul: European Language Resources Association (ELRA).

18. *Phillips A., Davis M.* Tags for Identifying Languages. – 2009 Sep.

Хусаинова Альбина Маратовна — аспирант 4-го года обучения, ассистент в лаборатории машинного обучения и представления данных Университета Иннополис.
E-mail: a.khusainova@innopolis.ru
ORCID iD: https://orcid.org/0000-0002-0636-3449

Романов Виталий Анатольевич — аспирант 4-го года обучения, ассистент в лаборатории промышленной разработки ПО Университета Иннополис.
E-mail: v.romanov@innopolis.ru
ORCID iD: https://orcid.org/0000-0003-3772-0039

Хан Адил Мехмуд — канд. физ.-мат. наук, профессор, начальник лаборатории машинного обучения и представления данных Университета Иннополис.
E-mail: a.khan@innopolis.ru
ORCID iD: https://orcid.org/0000-0003-2220-8518

DOI: https://doi.org/
Received 10.03.2022
Accepted 22.04.2022

ISSN 1995-5499

MULTILINGUAL MACHINE TRANSLATION USING HIERARCHICAL TRANSFORMER

© 2022 A. M. Khusainova✉, V. A. Romanov, A. M. Khan

Innopolis University
1, Universitetskaya Street, 420500 Innopolis, Russian Federation

Annotation. The way parameters are organized in multilingual machine translation models defines the effectiveness of parameter space usage. Therefore, it directly influences the translation quality. This work explores the idea of using language trees as the basis for the multilingual machine translation models architecture. Language trees show how different languages are related to each other and the primary idea is to organize multilingual models according to these expert hierarchies: the more related two languages are, the more parameters they share.

We test this approach for the Transformer architecture and demonstrate that despite the success in previous works there are persistent problems inherent to training hierarchical models. We investigate it and propose a solution to this problem and show that with the suggested training fix the hierarchical model can considerably outperform both bilingual and multilingual models with full parameter sharing.

Keywords: neural machine translation, multilingual translation, parameter organization, language trees, hierarchical architecture, low-resource translation, related languages.

CONFLICT OF INTEREST

The authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.

REFERENCES

1. *Tan X., Chen J., He D., Xia Y., Qin T. and Liu T. Y.* (2019) Multilingual Neural Machine Translation with Language Clustering. *In EMNLP/IJCNLP*.

2. *Johnson M., Schuster M., Le Q. V., Krikun M., Wu Y., Chen Z. [et al.]* (2017) Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguistics*. 5. P. 339–351.

3. *Dong D., Wu H., He W., Yu D. and Wang H.* (2015) Multi-Task Learning for Multiple Language Translation. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Vol. 1: Long Papers); 2015 Jul; Beijing: *Association for Computational Linguistics*. P. 1723–1732.

4. *Azpiazu I. M. and Pera M. S.* (2020) A Framework for Hierarchical Multilingual Machine Translation.

5. *Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N. [et al.]* (2017) Attention is All you Need. In Guyon I., Luxburg U. V., Bengio S., Wallach H., Fergus R., Vishwanathan S. [et al.], editors. *Advances in Neural Information Processing Systems 30*.: Curran Associates, Inc. P. 5998–6008.

6. *Firat O., Cho K. and Bengio Y.* (2016) Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism. *In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2016 Jun; San: Association for Computational Linguistics*. P. 866–875.

7. *Wang Y., Zhou L., Zhang J., Zhai F., Xu J. and Zong C.* (2019) A Compact and Language-Sensitive Multilingual Translation Method. *In Proceed-*

✉ Khusainova Albina M.
e-mail: a.khusainova@innopolis.ru

ings of the 57th Annual Meeting of the Association for Computational Linguistics; 2019 Jul; Florence: Association for Computational Linguistics. P. 1213–1223.

8. *Sachan D. and Neubig G.* (2018) Parameter Sharing Methods for Multilingual Self-Attentional Translation Models. In Proceedings of the Third Conference on Machine Translation: Research Papers; 2018 Oct; Brussels: Association for Computational Linguistics. P. 261–271.

9. *Bapna A., Arivazhagan N. and Firat O.* (2019) Simple, Scalable Adaptation for Neural Machine Translation. In EMNLP/IJCNLP.

10. *Fan A., Bhosale S., Schwenk H., Ma Z., El-Kishky A., Goyal S. [et al.]* (2020) Beyond English-Centric Multilingual Machine Translation. ArXiv. 2020; abs/2010.11125.

11. *Schleicher A. and Schleicher S.* Die ersten Spaltungen des indogermanischen Urvolkes [The first splits of the Proto-Indo-European people]. 1853.

12. *Belinkov Y., Màrquez L., Sajjad H., Durrani N., Dalvi F. and Glass J.* (2017) Evaluating Layers of Representation in Neural Machine Translation on Part-of-Speech and Semantic Tagging Tasks. *In Proceedings of the Eighth International Joint Conference on Natural Language Processing* (Vol. 1: Long Papers); 2017 Nov; Taipei: Asian Federation of Natural Language Processing. P. 1–10.

13. *Kudugunta S., Bapna A., Caswell I. and Firat O.* (2019) Investigating Multilingual NMT Representations at Scale. *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*; 2019 Nov; Hong: Association for Computational Linguistics. P. 1565–1575.

14. *Savelyev A. and Robbeets M.* (2020) Bayesian phylolinguistics infers the internal structure and the time-depth of the Turkic language family. Journal of Language Evolution. 2020 Feb.

15. *Agić Ž and Vulić I.* (2019) JW300: A Wide-Coverage Parallel Corpus for Low-Resource Languages. *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*; 2019 Jul; Florence: Association for Computational Linguistics. P. 3204–3210.

16. *Sennrich R., Haddow B. and Birch A.* (2016) Neural Machine Translation of Rare Words with Subword Units. *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*, August 7–12, 2016, Berlin, Germany, Volume 1: Long Papers; 2016: The Association for Computer Linguistics.

17. *Tiedemann J.* (2012) Parallel Data, Tools and Interfaces in OPUS. In Chair) NC(, Choukri K, Declerck T, Dogan MU, Maegaard B, Mariani J, et al., editors. *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*; 2012 May; Istanbul: European Language Resources Association (ELRA).

18. *Phillips A & Davis M.* (2009) Tags for Identifying Languages. 2009 Sep.

Khusainova Albina M. — 4th year post-graduate student, assistant in Machine Learning and Knowledge Representation Laboratory, Innopolis University.
E-mail: a.khusainova@innopolis.ru
ORCID iD: https://orcid.org/0000-0002-0636-3449

Romanov Vitaly A. — 4th year post-graduate student, assistant in Industrial Software Production Laboratory, Innopolis University.
E-mail: v.romanov@innopolis.ru
ORCID iD: https://orcid.org/0000-0003-3772-0039

Khan Adil M. — Candidate of Science in Physics and Mathematics, Professor, Head of the Machine Learning and Knowledge Representation Laboratory, Innopolis University.
E-mail: a.khan@innopolis.ru
ORCID iD: https://orcid.org/0000-0003-2220-8518