

**Последовательное обучение с помощью нейронных сетей**



**Илья Суцкевер Иоловые виньялы Куок против Ле**

Гугл Гугл Гугл

ilyasu@google.com vinyals@google.com qvl@google.com

# Абстрактный

Глубокие нейронные сети (DNN) — это мощные модели, которые достигли отличной производительности при выполнении сложных задач обучения. Несмотря на то, что DNN хорошо работают при наличии больших размеченных обучающих наборов, их нельзя использовать для сопоставления последовательностей с последовательностями. В этой статье мы представляем общий сквозной подход к обучению последовательностей, который делает минимальные предположения о структуре последовательности. Наш метод использует многоуровневую длительную кратковременную память (LSTM) для отображения входной последовательности на вектор фиксированной размерности, а затем еще одну глубокую LSTM для декодирования целевой последовательности из вектора. Наш главный результат заключается в том, что в задании на перевод с английского на французский из набора данных WMT'14 переводы, выполненные LSTM, получили оценку BLEU 34,8 по всему тестовому набору, где оценка BLEU LSTM была снижена за слова, выходящие за рамки словарного запаса. Кроме того, LSTM не испытывал трудностей при длительных предложениях. Для сравнения, система SMT на основе фраз получает оценку BLEU 33,3 на том же наборе данных. Когда мы использовали LSTM для переранжирования 1000 гипотез, выданных вышеупомянутой системой SMT, ее оценка BLEU увеличилась до 36,5, что близко к предыдущему лучшему результату в этой задаче. LSTM также изучил разумные представления фраз и предложений, которые чувствительны к порядку слов и относительно инвариантны к активному и пассивному залогу. Наконец, мы обнаружили, что изменение порядка слов во всех исходных предложениях (но не в целевых предложениях) заметно улучшило производительность LSTM, поскольку это привело к появлению множества краткосрочных зависимостей между исходным и целевым предложениями, что облегчило задачу оптимизации.

[4](http://arxiv.org/abs/1409.3215v3)

[arXiv:1409.3215v3 [cs. CL] 14 дек 201](http://arxiv.org/abs/1409.3215v3)

# Знакомство

Глубокие нейронные сети (DNN) — это чрезвычайно мощные модели машинного обучения, которые достигают отличной производительности при решении сложных задач, таких как распознавание речи [13, 7] и распознавание визуальных объектов [19, 6, 21, 20]. DNN являются мощными, потому что они могут выполнять произвольные параллельные вычисления за небольшое количество шагов. Удивительным примером мощи DNN является их способность сортировать *N* N-битных чисел, используя всего 2 скрытых слоя квадратичного размера [27]. Таким образом, хотя нейронные сети связаны с обычными статистическими моделями, они обучаются сложным вычислениям. Более того, большие DNN могут быть обучены с помощью контролируемого обратного распространения, когда помеченный обучающий набор содержит достаточно информации для указания параметров сети. Таким образом, если существует настройка параметров большой DNN, которая дает хорошие результаты (например, потому что человек может решить задачу очень быстро), контролируемое обратное распространение найдет эти параметры и решит проблему.

Несмотря на свою гибкость и мощность, DNN могут быть применены только к задачам, входные данные и цели которых могут быть разумно закодированы векторами фиксированной размерности. Это существенное ограничение, так как многие важные проблемы лучше всего выражаются с помощью последовательностей, длины которых априори неизвестны. Например, распознавание речи и машинный перевод — это последовательные задачи. Точно так же ответ на вопрос можно рассматривать как сопоставление последовательности слов, представляющих вопрос, с последовательностью слов, представляющих ответ. Таким образом, очевидно, что был бы полезен не зависящий от предметной области метод, который учится отображать последовательности на последовательности.

Последовательности представляют собой проблему для DNN, поскольку они требуют, чтобы размерность входов и выходов была известна и фиксирована. В этой статье мы показываем, что простое применение архитектуры Long Short-Term Memory (LSTM) [16] может решить общие задачи последовательности. Идея состоит в том, чтобы использовать один LSTM для чтения входной последовательности, шаг за шагом, чтобы получить большое представление вектора с фиксированной размерностью, а затем использовать другой LSTM для извлечения выходной последовательности из этого вектора (рис. 1). Второй LSTM по сути является рекуррентной нейронной сетевой языковой моделью [28, 23, 30], за исключением того, что он обусловлен входной последовательностью. Способность LSTM успешно обучаться на данных с большими временными зависимостями делает его естественным выбором для этого приложения из-за значительного временного лага между входами и их соответствующими выходами (рис. 1).

Был предпринят ряд связанных с этим попыток решить общую проблему обучения последовательности с помощью нейронных сетей. Наш подход тесно связан с Калхбреннером и Блансомом [18], которые первыми отобразили все входное предложение на вектор, и связан с Cho et al. [5], хотя последний использовался только для проверки гипотез, полученных с помощью системы, основанной на фразах. Грейвс [10] представил новый механизм дифференцируемого внимания, который позволяет нейронным сетям фокусироваться на различных частях входных данных, и элегантный вариант этой идеи был успешно применен к машинному переводу Бахданау и др. [2]. Коннекционистская классификация последовательностей является еще одним популярным методом отображения последовательностей на последовательности с нейронными сетями, но она предполагает монотонное выравнивание между входами и выходами [11].

Рисунок 1: Наша модель читает входное предложение "ABC" и выдает "WXYZ" в качестве выходного предложения. Модель перестает делать прогнозы после вывода маркера в конце предложения. Обратите внимание, что LSTM читает входное предложение в обратном порядке, потому что это приводит к появлению множества краткосрочных зависимостей в данных, которые значительно упрощают задачу оптимизации.

Основным результатом этой работы является следующее. В задаче на перевод с английского на французский язык в WMT'14 мы получили оценку BLEU **34,81** путем прямого извлечения переводов из ансамбля из 5 глубоких LSTM (с 384 млн параметров и 8000 размерным состоянием каждая) с помощью простого декодера поиска луча слева направо. Это, безусловно, лучший результат, достигнутый при прямом переводе с помощью больших нейронных сетей. Для сравнения, оценка BLEU базового уровня SMT на этом наборе данных составляет 33,30 [29]. Балл 34,81 BLEU был достигнут LSTM со словарным запасом 80 тыс. слов, поэтому балл был оштрафован всякий раз, когда эталонный перевод содержал слово, не охваченное этими 80 тыс. слов. Этот результат показывает, что относительно неоптимизированная архитектура нейронной сети с небольшим словарным запасом, которая имеет много возможностей для улучшения, превосходит систему SMT на основе фраз.

Наконец, мы использовали LSTM для повторной оценки общедоступных 1000 лучших списков базового уровня SMT по той же задаче [29]. Таким образом, мы получили оценку BLEU 36,5, что улучшает базовый уровень на 3,2 балла BLEU и близко к предыдущему лучшему опубликованному результату по этой задаче (который составляет 37,0 [9]).

Удивительно, но LSTM не страдал от очень длинных предложений, несмотря на недавний опыт других исследователей с родственными архитектурами [26]. Нам удалось преуспеть в длинных предложениях, потому что мы изменили порядок слов в исходном предложении, но не в целевых предложениях в обучающем и тестовом наборе. Таким образом, мы ввели множество краткосрочных зависимостей, которые значительно упростили задачу оптимизации (см. разделы 2 и 3.3). В результате SGD смог выучить LSTM, которые без проблем справлялись с длинными предложениями. Простой трюк с перестановкой слов в исходном предложении является одним из ключевых технических вкладов этой работы.

Полезным свойством LSTM является то, что он учится отображать входное предложение переменной длины в векторное представление с фиксированной размерностью. Учитывая, что переводы, как правило, являются пересказами исходных предложений, цель перевода побуждает LSTM находить представления предложений, которые отражают их значение, поскольку предложения со схожими значениями близки друг к другу, в то время как значения разных предложений будут далеки. Качественная оценка подтверждает это утверждение, показывая, что наша модель учитывает порядок слов и довольно инвариантна к активному и пассивному залогу.

# Модель

Рекуррентная нейронная сеть (РНС) [31, 28] является естественным обобщением нейронных сетей с прямой связью на последовательности. Имея последовательность входов (*x1,...,xT*), стандартная RNN вычисляет последовательность выходов (*y1,...,yT*) путем повторения следующего уравнения:

*ХТ*= sigm 

*yt* = *Wyhht*

RNN может легко сопоставлять последовательности с последовательностями в тех случаях, когда выравнивание между входами и выходами известно заранее. Однако неясно, как применять RNN к задачам, входные и выходные последовательности которых имеют разную длину со сложными и немонотонными отношениями.

Простейшая стратегия обучения общей последовательности состоит в том, чтобы отобразить входную последовательность на вектор фиксированного размера с помощью одного RNN, а затем отобразить вектор на целевую последовательность с помощью другой RNN (этот подход также был использован Cho et al. [5]). Хотя в принципе это может сработать, поскольку RNN предоставляется всей необходимой информацией, обучение RNN будет затруднено из-за возникающих в результате долгосрочных зависимостей (рис. 1) [14, 4, 16, 15]. Тем не менее, известно, что долгосрочная кратковременная память (LSTM) [16] обучается проблемам с долгосрочными зависимостями большого расстояния, поэтому LSTM может успешно работать в этой ситуации.

Целью LSTM является оценка условной вероятности *p*(*y1,...,yT*′|x1*,...,xT*), где (*x1,...,xT*) — входная последовательность, а *y1,...,yT*' — соответствующая ей выходная последовательность, длина которой *T*' может отличаться от *T*. LSTM вычисляет эту условную вероятность, сначала получая представление фиксированной размерности *v* входной последовательности (*x1,...,xT* ), заданной последним скрытым состоянием LSTM, а затем вычисляя вероятность *y1,...,yT*' с помощью стандартной формулировки LSTM-LM, начальное скрытое состояние которой установлено как представление *v* x1 *,...,xT* :

*Т*

*p*(*y1,...,yT*′|х1*,...,xT* ) = Yp(*yt|v,y1,...,yt−1*) (1)

*t=1*

В этом уравнении каждое  *распределение p*(*yt|v,y1,...,yt−1*) представлено с помощью softmax по всем словам в словаре. Мы используем формулировку LSTM от Грейвса [10]. Обратите внимание, что мы требуем, чтобы каждое предложение заканчивалось специальным символом в конце предложения "*<EOS*>", что позволяет модели определить распределение по последовательностям всех возможных длин. Общая схема представлена на рисунке 1, где показанный LSTM вычисляет представление "A", "B", "C", "<*EOS>*", а затем использует это представление для вычисления вероятности "W", "X", "Y", "Z", "*<EOS>*".

Наши фактические модели отличаются от приведенного выше описания по трем важным параметрам. Во-первых, мы использовали два разных LSTM: один для входной последовательности, а другой для выходной, потому что это увеличивает параметры числовой модели при незначительных вычислительных затратах и делает естественным одновременное обучение LSTM на нескольких языковых парах [18]. Во-вторых, мы обнаружили, что глубокие LSTM значительно превосходят поверхностные LSTM, поэтому мы выбрали LSTM с четырьмя слоями. В-третьих, мы сочли чрезвычайно ценным изменить порядок слов во входном предложении. Так, например, вместо того, чтобы сопоставлять предложение *a,b,c* с предложением *α,β,γ*, LSTM предлагается сопоставить *c,b,a* с *α,β,γ*, где *α,β,γ* является переводом *a,b,c*. Таким образом, *a* находится в непосредственной близости от *α*, *b* — довольно близко к *β* и т. д., что облегчает SGD «установление связи» между входом и выходом. Мы обнаружили, что это простое преобразование данных значительно повышает производительность LSTM.

# Эксперименты

Мы применили наш метод к задаче машинного перевода с английского на французский язык WMT'14 двумя способами. Мы использовали его для прямого перевода входного предложения без использования эталонной системы SMT, и мы использовали его для переоценки n-лучших списков базового уровня SMT. Мы сообщаем о точности этих методов перевода, представляем примеры переводов и визуализируем результирующее представление предложений.

## Сведения о наборе данных

Мы использовали набор данных WMT'14 с английского на французский. Мы обучили наши модели на подмножестве из 12 миллионов предложений, состоящем из 348 миллионов французских слов и 304 миллионов английских слов, что является чистым «выбранным» подмножеством из [29]. Мы выбрали эту задачу трансляции и это конкретное подмножество обучающего набора из-за общедоступности токенизированного обучающего и тестового набора вместе со списками 1000 лучших из базового SMT [29].

Поскольку типичные нейронные языковые модели полагаются на векторное представление для каждого слова, мы использовали фиксированный словарь для обоих языков. Мы использовали 160 000 наиболее часто встречающихся слов для исходного языка и 80 000 наиболее часто встречающихся слов для целевого языка. Каждое слово, выходящее за рамки словарного запаса, было заменено на специальный токен «UNK».

## Декодирование и восстановление

Суть наших экспериментов заключалась в обучении большого глубокого LSTM на многих парах предложений. Мы обучили его, максимизировав логарифмическую вероятность правильного перевода *T* при заданном исходном предложении *S*, поэтому целью обучения является

1*/*|С| X logp(*T|С*)

(*Т,С*)∈С

где S — обучающая выборка. После завершения обучения мы выполняем переводы, находя наиболее вероятный перевод в соответствии с LSTM:

*Tˆ = argmaxp*(*Т|S*) (2)

*T*

Мы ищем наиболее вероятный перевод с помощью простого декодера поиска луча слева направо, который поддерживает небольшое количество *B* частичных гипотез, где частичная гипотеза является префиксом некоторого перевода. На каждом временном шаге мы расширяем каждую частичную гипотезу в луче всеми возможными словами в словаре. Это значительно увеличивает количество гипотез, поэтому мы отбрасываем все, кроме *В,* наиболее вероятных гипотез в соответствии с логарифмической вероятностью модели. Как только *к* гипотезе добавляется символ «<EOS>», он удаляется из балки и добавляется в набор готовых гипотез. Хотя этот декодер является приблизительным, его легко реализовать. Интересно, что наша система хорошо работает даже при размере луча 1, а луч размера 2 обеспечивает большую часть преимуществ поиска луча (таблица 1).

Мы также использовали LSTM для повторной оценки списков 1000 лучших, полученных по базовой системе [29]. Чтобы переоценить список n-лучших, мы вычислили логарифмическую вероятность каждой гипотезы с помощью нашего LSTM и взяли четное среднее значение с их оценкой и оценкой LSTM.

## Перестановка исходных предложений

В то время как LSTM способен решать задачи с долгосрочными зависимостями, мы обнаружили, что LSTM обучается гораздо лучше, когда исходные предложения перевернуты (целевые предложения не перевернуты). Таким образом, тестовая сложность LSTM снизилась с 5,8 до 4,7, а оценка тестового BLEU его декодированных переводов увеличилась с 25,9 до 30,6.

Хотя у нас нет полного объяснения этому явлению, мы считаем, что он вызван введением множества краткосрочных зависимостей в набор данных. Обычно, когда мы объединяем исходное предложение с целевым предложением, каждое слово в исходном предложении находится далеко от соответствующего ему слова в целевом предложении. В результате задача имеет большой «минимальный временной лаг» [17]. При перестановке слов в исходном предложении среднее расстояние между соответствующими словами в исходном и целевом языках остается неизменным. Тем не менее, первые несколько слов на исходном языке теперь очень близки к первым нескольким словам на целевом языке, поэтому минимальная временная задержка задачи значительно сокращается. Таким образом, обратное распространение ошибки облегчает «установление связи» между исходным предложением и целевым предложением, что, в свою очередь, приводит к значительному улучшению общей производительности.

Первоначально мы полагали, что перестановка входных предложений приведет только к более уверенным прогнозам в первых частях целевого предложения и к менее уверенным прогнозам в более поздних частях. Тем не менее, LSTM, обученные на перевернутых исходных предложениях, гораздо лучше справлялись с длинными предложениями, чем LSTM, обученные на исходных предложениях (см. раздел 3.7), что говорит о том, что перестановка входных предложений приводит к лучшему использованию памяти LSTM.

## Детали тренировки

Мы обнаружили, что модели LSTM довольно легко поддаются обучению. Мы использовали глубокие LSTM с 4 слоями, с 1000 ячеек на каждом слое и 1000 размерными вложениями слов, с входным словарем 160 000 и выходным словарем 80 000. Таким образом, глубинный LSTM использует 8000 действительных чисел для представления предложения. Мы обнаружили, что глубокие LSTM значительно превосходят поверхностные LSTM, где каждый дополнительный слой снижает недоумение почти на 10%, возможно, из-за их гораздо большего скрытого состояния. Мы использовали наивный софтмакс более 80 000 слов на каждом выходе. Полученный LSTM имеет 384М параметров, из которых 64М являются чисто рекуррентными соединениями (32М для «энкодера» LSTM и 32М для «декодера» LSTM). Полная информация об обучении приведена ниже:

* Мы инициализировали все параметры LSTM с равномерным распределением от -0.08 до 0.08
* Мы использовали стохастический градиентный спуск без импульса, с фиксированной скоростью обучения 0,7. Через 5 эпох мы начали уменьшать скорость обучения вдвое каждые полэпохи. Мы обучали наши модели в общей сложности 7,5 эпох.
* Для градиента мы использовали пакеты из 128 последовательностей и разделили его на размер партии (а именно 128).
* Хотя LSTM, как правило, не страдают от проблемы исчезающего градиента, они могут иметь взрывные градиенты. Таким образом, мы наложили жесткое ограничение на норму градиента [10, 25], масштабируя ее, когда ее норма превышала пороговое значение. Для каждой обучающей партии мы вычисляем *s* = kgk2, где *g* — градиент, деленный на 128. Если *s >* 5, то ставим.
* Разные предложения имеют разную длину. Большинство предложений короткие (например, длина 20-30), но некоторые предложения длинные (например, длина *>* 100), поэтому мини-пакет из 128 случайно выбранных обучающих предложений будет состоять из множества коротких предложений и нескольких длинных предложений, и в результате большая часть вычислений в мини-пакете будет потрачена впустую. Чтобы решить эту проблему, мы убедились, что все предложения в мини-пакете имеют примерно одинаковую длину, что дает 2-кратное ускорение.

## Распараллеливание

Реализация глубокого LSTM на языке C++ с конфигурацией из предыдущего раздела на одном графическом процессоре обрабатывает скорость около 1 700 слов в секунду. Это было слишком медленно для наших целей, поэтому мы распараллелили нашу модель с помощью машины с 8 графическими процессорами. Каждый слой LSTM выполнялся на отдельном графическом процессоре и передавал свои активации следующему графическому процессору / слою сразу после их вычисления. Наши модели имеют 4 слоя LSTM, каждый из которых находится на отдельном графическом процессоре. Остальные 4 графических процессора использовались для распараллеливания софтмакса, поэтому каждый графический процессор отвечал за умножение на матрицу 1000× 20000 . В результате реализация достигла скорости 6300 слов (как на английском, так и на французском языках) в секунду при размере мини-пакета 128. При таком внедрении обучение заняло около десяти дней.

## Экспериментальные результаты

Для оценки качества наших переводов мы использовали оценку BLEU [24]. Мы рассчитали наши баллы BLEU, используя multi-bleu.pl[[1]](#footnote-1) на *основе токенизированных* прогнозов и наземных истин. Этот способ оценки оценки BELU согласуется с [5] и [2] и воспроизводит 33,3 балла [29]. Однако, если мы оценим лучшую систему WMT'14 [9] (прогнозы которой можно загрузить из statmt.org\matrix) таким образом, мы получим 37.0, что больше, чем 35.8, о которых сообщает statmt.org\matrix.

Результаты представлены в таблицах 1 и 2. Наилучшие результаты мы получаем с ансамблем LSTM, которые различаются по своим случайным инициализациям и по случайному порядку минипакетов. Несмотря на то, что декодированные переводы ансамбля LSTM не превосходят лучшую систему WMT'14, это первый случай, когда чистая нейронная система трансляции превосходит базовую линию SMT на основе фраз в крупномасштабном машинном переводе

|  |  |
| --- | --- |
| **Метод** | **тест BLEU score (ntst14)** |
| Бахданау и др. [2] | 28.45 |
| Базовая система [29] | 33.30 |
| Одинарный передний LSTM, размер луча 12 | 26.17 |
| Одинарный реверсивный LSTM, размер луча 12 | 30.59 |
| Ансамбль из 5 перевернутых LSTM, размер луча 1 | 33.00 |
| Ансамбль из 2-х перевернутых LSTM, размер луча 12 | 33.27 |
| Ансамбль из 5 перевернутых LSTM, размер луча 2 | 34.50 |
| Ансамбль из 5 перевернутых LSTM, размер луча 12 | **34.81** |

Таблица 1: Производительность LSTM на тестовом наборе WMT'14 с английского на французский (ntst14). Обратите внимание, что ансамбль из 5 ЛСТМ с лучом размера 2 дешевле, чем из одного ЛСТМ с лучом размера 12.

|  |  |
| --- | --- |
| **Метод** | **тест BLEU score (ntst14)** |
| Базовая система [29] | 33.30 |
| Cho et al. [5] | 34.54 |
| Лучший результат WMT'14 [9] | **37.0** |
| Повторение базового уровня 1000 лучших с помощью одного прямого LSTM | 35.61 |
| Восстановление базового уровня 1000-best с помощью одного реверсивного LSTM | 35.85 |
| Повторение базового уровня 1000-best с ансамблем из 5 перевернутых LSTM | **36.5** |
| Анализ Oracle списков Baseline 1000 лучших | ∼45 |

Таблица 2: Методы, использующие нейронные сети вместе с системой SMT на тестовом наборе WMT'14 с английского на французский (ntst14).

со значительным отрывом, несмотря на его неспособность обрабатывать слова, выходящие за рамки словарного запаса. LSTM находится в пределах 0,5 балла BLEU от лучшего результата WMT'14, если он используется для повторного оценки списка 1000 лучших по базовой системе.

## Производительность при работе с длинными предложениями

Мы были удивлены, обнаружив, что LSTM хорошо справляется с длинными предложениями, что количественно показано на рисунке 3. В таблице 3 представлено несколько примеров длинных предложений и их переводов.

## Анализ модели

−6

−5

−4

−3

−2

−1

0

1

2

3

4

Джон уважает Марию

Мария уважает Джона

Джон восхищается Марией

Мария восхищается Джоном

Мэри влюблена в Джона

Джон влюблен в Марию

−20

−15

−10

−5

0

5

10

15

Я подарила ей карточку в саду

В саду я подарила ей карточку

Я подарил ей карточку в саду

Она подарила мне карточку в саду

В саду она дала мне карточку

В саду она подарила мне карточку

−8 −6 −4 −2 0 2 4 6 8 10 −15 −10 −5 0 5 10 15 20

Рисунок 2: На рисунке показана двухмерная проекция PCA скрытых состояний LSTM, которые получаются после обработки фраз на рисунках. Фразы сгруппированы по значению, которое в этих примерах в первую очередь является функцией порядка слов, который было бы трудно уловить с помощью модели мешка слов. Обратите внимание, что оба кластера имеют схожую внутреннюю структуру.

Одной из привлекательных особенностей нашей модели является ее способность превращать последовательность слов в вектор фиксированной размерности. На рисунке 2 визуализированы некоторые из изученных представлений. На рисунке ясно видно, что представления чувствительны к порядку слов, но при этом довольно нечувствительны к

|  |  |
| --- | --- |
| **Тип** | **Предложение** |
| **Наша модель** | Ульрих УНК, член совета директоров автопроизводителя Audi, говорит, что в течение многих лет было обычной практикой, когда мобильные телефоны можно собирать перед заседаниями совета директоров, чтобы они не использовались в качестве устройств дистанционного прослушивания. |
| **Истина** | Ульрих Хакенберг, член совета директоров автопроизводителя Audi, говорит, что сбор мобильных телефонов перед заседаниями совета директоров, чтобы их нельзя было использовать в качестве устройств дистанционного прослушивания, был обычной практикой в течение многих лет. |
| **Наша модель** | «Сотовые телефоны, которые действительно являются проблемой, не только потому, что они потенциально могут создавать помехи для навигационных устройств, но мы знаем, что, по данным FCC, они могут создавать помехи для вышек сотовой связи, когда они находятся в воздухе», — говорит UNK. |
| **Истина** | «Сотовые телефоны действительно представляют собой проблему, не только потому, что они потенциально могут создавать помехи для навигационных приборов, но и потому, что мы знаем, согласно FCC, что они могут создавать помехи для вышек сотовой связи, если их использовать на борту». — сказал Розенкер. |
| **Наша модель** | При кремации возникает «чувство насилия над телом любимого человека», которое будет «превращено в кучу пепла» за очень короткое время, вместо процесса разложения, «который будет сопровождать этапы траура». |
| **Истина** | В случае кремации происходит «насилие над любимым телом», которое будет «превращено в кучу пепла» в очень короткое время, а не после процесса разложения, который будет «сопровождать фазы траура». |

Таблица 3: Несколько примеров длинных переводов, выполненных LSTM наряду с наземными достоверными переводами. Читатель может убедиться в правильности перевода с помощью Google Translate.

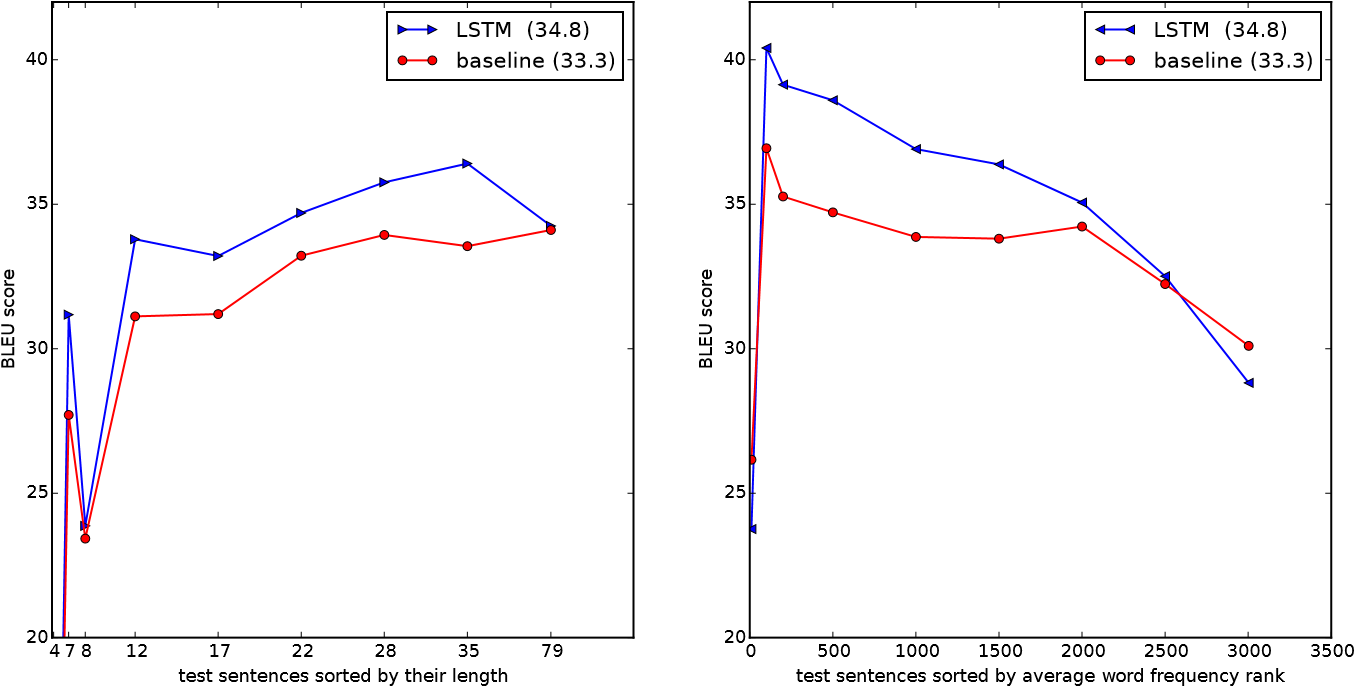


Рисунок 3: На левом графике показана производительность нашей системы в зависимости от длины предложения, где ось x соответствует тестовым предложениям, отсортированным по их длине, и отмечена фактической длиной последовательности. Нет ухудшения в предложениях, состоящих менее чем из 35 слов, есть лишь незначительное ухудшение в самых длинных предложениях. Правый график показывает производительность LSTM в предложениях со все более редкими словами, где ось x соответствует тестовым предложениям, отсортированным по их «среднему рангу частоты слов».

замена активного залога на пассивный. Двумерные проекции получены с помощью PCA.

# Связанная работа

Существует большой объем работ по применению нейронных сетей к машинному переводу. До сих пор самым простым и эффективным способом применения языковой модели RNN (RNNLM) [23] или языковой модели нейронной сети с прямой связью (NNLM) [3] к задаче машинного перевода является повторение списков nbest с сильным базовым уровнем машинного перевода [22], что надежно повышает качество перевода.

В последнее время исследователи начали искать способы включения информации об исходном языке в NNLM. Примеры этой работы включают Auli et al. [1], которые объединили NNLM с тематической моделью входного предложения, что улучшает производительность сканирования. Девлин и др. [8] придерживались аналогичного подхода, но они включили свой NNLM в декодер системы машинного перевода и использовали информацию о выравнивании декодера, чтобы предоставить NNLM наиболее полезные слова во входном предложении. Их подход оказался очень успешным, и он позволил добиться значительных улучшений по сравнению с их базовым уровнем.

Наша работа тесно связана с работой Кальхбреннера и Блансома [18], которые первыми отобразили входное предложение в вектор, а затем обратно в предложение, хотя они отображают предложения на векторы с помощью сверточных нейронных сетей, которые теряют порядок слов. Как и в этой работе, Cho et al. [5] использовали LSTM-подобную архитектуру RNN для отображения предложений в векторы и обратно, хотя их основное внимание было сосредоточено на интеграции нейронной сети в систему SMT. Bahdanau et al. [2] также попытались выполнить прямой перевод с помощью нейронной сети, которая использовала механизм внимания для преодоления низкой производительности при работе с длинными предложениями, с которой столкнулись Cho et al. [5], и достигли обнадеживающих результатов. Аналогичным образом, Pouget-Abadie et al. [26] попытались решить проблему памяти у Cho et al. [5], переводя фрагменты исходного предложения таким образом, чтобы получить плавный перевод, что похоже на подход, основанный на фразах. Мы подозреваем, что они могли бы достичь подобных улучшений, просто обучая свои сети на перевернутых исходных предложениях.

Сквозное обучение также находится в центре внимания Hermann et al. [12], чья модель представляет входы и выходы сетей прямой связи и отображает их на аналогичные точки в пространстве. Однако их подход не может генерировать переводы напрямую: чтобы получить перевод, им нужно выполнить поиск ближайшего вектора в предварительно вычисленной базе данных предложений или переоценить предложение.

# Заключение

В этой работе мы показали, что большой глубокий LSTM, который имеет ограниченный словарный запас и который почти не делает предположений о структуре проблемы, может превзойти стандартную систему на основе SMT, словарный запас которой неограничен в крупномасштабной задаче машинного перевода. Успех нашего простого подхода к машинному переводу на основе LSTM предполагает, что он должен хорошо справляться со многими другими задачами обучения последовательностей, при условии, что у них достаточно обучающих данных.

Мы были удивлены степенью улучшения, полученного при перестановке слов в исходных предложениях. Мы пришли к выводу, что важно найти задачу кодирования, которая имеет наибольшее количество краткосрочных зависимостей, так как они значительно упрощают задачу обучения. В частности, хотя мы не смогли обучить стандартную РНС на задаче необратного перевода (показано на рис. 1), мы считаем, что стандартная РНС должна быть легко обучаемой, когда исходные предложения перевернуты (хотя мы не проверяли это экспериментально).

Мы также были удивлены способностью LSTM правильно переводить очень длинные предложения. Первоначально мы были убеждены, что LSTM не справится с длинными предложениями из-за ограниченного объема памяти, а другие исследователи сообщали о низкой производительности при работе с длинными предложениями с моделью, аналогичной нашей [5, 2, 26]. И все же LSTM, обученные на обратном наборе данных, не испытывали особых трудностей при переводе длинных предложений.

Самое главное, мы продемонстрировали, что простой, понятный и относительно неоптимизированный подход может превзойти систему SMT, поэтому дальнейшая работа, скорее всего, приведет к еще большей точности перевода. Эти результаты позволяют предположить, что наш подход, скорее всего, будет хорошо работать и в других сложных задачах последовательности.

# Подтверждения

Мы благодарим Сэми Бенджио, Джеффа Дина, Матье Девина, Джеффри Хинтона, Нала Калчбреннера, Тханга Луонга, Вольфганга Мачери, Раджата Монгу, Винсента Ванхуке, Пэн Сюя, Войцеха Зарембу и команду Google Brain за полезные комментарии и обсуждения.

## Ссылки

1. М. Аули, М. Галли, К. Квирк и Г. Цвейг. Совместное языковое и переводческое моделирование с рекуррентными нейронными сетями. В *EMNLP*, 2013.
2. Д. Бахданау, К. Чо и Ю. Бенжио. Нейронный машинный перевод путем совместного обучения выравниванию и переводу. *Препринт arXiv arXiv:1409.0473*, 2014.
3. И. Бенжио, Р. Дюшарм,. Винсент и К. Жовен. Нейронная вероятностная языковая модель. В *журнале Journal of Machine Learning Research*, стр. 1137–1155, 2003.
4. Ю. Бенджио,. Симард и. Фраскони. Изучение долгосрочных зависимостей с помощью градиентного спуска является сложной задачей. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.
5. К. Чо, Б. Мерриенбур, К. Гюльцере, Ф. Бугарес, Х. Швенк и Ю. Бенджио. Обучение представлениям фраз с помощью кодера-декодера RNN для статистического машинного перевода. В *препринте Arxiv arXiv:1406.1078*, 2014.
6. Д. Чиресан, У. Мейер и Й. Шмидхубер. Многоколоночные глубокие нейронные сети для классификации изображений. В *CVPR,* 2012.
7. Г. Э. Даль, Д. Ю, Л. Денг и А. Асеро. Контекстно-зависимые предварительно обученные глубокие нейронные сети для распознавания речи большого словарного запаса. *IEEE Transactions on Audio, Speech, and Language Processing - Специальный выпуск по глубокому обучению для обработки речи и языка*, 2012.
8. Д. Девлин, Р. Збиб, З. Хуанг, Т. Ламар, Р. Шварц и Д. Махул. Быстрые и надежные модели суставов нейронных сетей для статистического машинного перевода. В *ACL*, 2014.
9. Надир Дуррани, Барри Хэддоу, Филипп Коэн и Кеннет Хифилд. Эдинбургские системы машинного перевода на основе фраз для wmt-14. В *WMT*, 2014.
10. А. Грейвс. Генерация последовательностей с помощью рекуррентных нейронных сетей. В *препринте Arxiv arXiv:1308.0850*, 2013.
11. А. Грейвс, С. Фернандес, Ф. Гомес и Й. Шмидхубер. Коннекционистская временная классификация: маркировка несегментированных данных последовательностей с помощью рекуррентных нейронных сетей. В *ICML,* 2006.
12. К. М. Герман и. Блансом. Многоязычные распределенные представления без выравнивания слов. В *ICLR*, 2014.
13. Г. Хинтон, Л. Денг, Д. Ю, Г. Даль, А. Мохамед, Н. Джейтли, А. Сениор, В. Ванхуке,. Нгуен, Т. Сайнатх и Б. Кингсбери. Глубокие нейронные сети для акустического моделирования в распознавании речи. *Журнал IEEE Signal Processing*, 2012.
14. С. Хохрейтер. Исследования динамических нейронных сетей. *Магистерская диссертация, Институт компьютерных наук Технического университета, Мюнхен*, 1991 г.
15. С. Хохрейтер, Ю. Бенджио,. Фраскони и Й. Шмидхубер. Градиентный поток в рекуррентных сетях: сложность изучения долгосрочных зависимостей, 2001.
16. С. Хохрайтер и Й. Шмидхубер. Длительная кратковременная память. *Нейронные вычисления*, 1997.
17. С. Хохрайтер и Й. Шмидхубер. LSTM может решить сложные проблемы с длительными задержками. 1997.
18. Н. Кальхбреннер и. Блансом. Модели рекуррентного непрерывного перевода. В *EMNLP*, 2013.
19. А. Крижевский, И. Суцкевер, Г. Э. Хинтон. Классификация ImageNet с глубокими сверточными нейронными сетями. В *NIPS*, 2012.
20. К.В. Ле, М.А. Ранзато, Р. Монга, М. Девин, К. Чен, Г.С. Коррадо, Д. Дин и А.Й. Нг. Создание высокоуровневых функций с использованием крупномасштабного неконтролируемого обучения. В *ICML,* 2012.
21. Ю. Лекун, Л. Ботту, Ю. Бенжио и. Хаффнер. Обучение на основе градиента применительно к распознаванию документов. *Труды IEEE,* 1998.
22. Т. Миколов. *Статистические языковые модели на основе нейронных сетей*. Кандидатская диссертация, Технологический университет Брно, 2012 г.
23. Т. Миколов, М. Карафиат, Л. Бургет, Я. Черноцкий, С. Худанпур. Рекуррентная языковая модель на основе нейронных сетей. В *журнале INTERSPEECH,* страницы 1045–1048, 2010.
24. К. Папинени, С. Роукос, Т. Уорд и У. Дж. BLEU: метод автоматической оценки машинного перевода. В *ACL*, 2002.
25. Р. Паскану, Т. Миколов, Ю. Бенджио. О сложности обучения рекуррентных нейронных сетей. *Препринт arXiv arXiv:1211.5063*, 2012.
26. Ж. Пуже-Абади, Д. Бахданау, Б. ван Мерриенбур, К. Чо и Ю. Бенджио. Преодоление проклятия длины предложений для нейронного машинного перевода с помощью автоматической сегментации. *Препринт arXiv arXiv:1409.1257*, 2014.
27. А. Разборов. На цепях с малым порогом глубины. *3-й Скандинавский семинар по теории алгоритмов*, 1992.
28. Д. Румельхарт, Г. Э. Хинтон и Р. Дж. Обучение представлений путем обратного распространения ошибок. *Природа*, 323(6088):533–536, 1986.
29. H. Schwenk. University le Mans. http://www-lium.univ-lemans.fr/ ̃schwenk/cslm\_ joint\_paper/, 2014. [Электронный ресурс; дата обращения: 03 сентября 2014 г.].
30. М. Зундермейер, Р. Шлютер и Х. Ней. Нейронные сети LSTM для моделирования языка. В *журнале INTERSPEECH,* 2010.
31. . Вербос. Обратное распространение во времени: что это делает и как это сделать. *Труды IEEE,* 1990.

1. Существует несколько вариантов оценки BLEU, и каждый вариант определяется с помощью perl-скрипта. [↑](#footnote-ref-1)