Обучение машинного перевода без параллельных текстов.*

Саттаров Т. И.

sattarov.ti@phystech.edu

[1]Московский Физико-технический институт

Рассматривается метод обучения без учителя для задачи машинного перевода. В данном подходе используется нейросетевая модель с применением seq2seq. Каждое предложение на рассматриваемых языках отображается кодировщиком в вектор из скрытого пространства, а декодировщик восстанавливает предложение из вектора скрытого пространства. Для дальнейшего улучшения модели перевода предлагается применить модель T2T, позволяющую получить многоуровневые зависимости между словами в исходных предложениях. Качество работы проверяется на парах языков «английский-казахский» и «русский-украинский».

Ключевые слова: *Машинный перевод, seq2seq, T2T, обучение без учителя.*

1 Введение

Для решения задачи автоматического перевода с одного языка на другой существует несколько подходов: перевод по правилам, статистический перевод и др. Статистическая модель разделяет переводимое предложение на отдельные слова и фразы, перебирает все варианты перевода для каждого фрагмента и взвешивает вероятность каждого из них, исходя из того, какой вариант встречался чаще на обучающей выборке. Данный метод учитывает при переводе статистические закономерности для текстовых фрагментов небольшой длины, поэтому успешно справляется с небольшими предложениями. Альтернативным методом машинного перевода является машинный перевод, основанный на нейросетевых моделях. Несмотря на хорошую точность перевода, у этого метода есть большой недостаток: необходимость большого числа параллельных текстов — количество

*Работа выполнена при финансовой поддержке РФФИ, проект №00-00-00000. Научный руководитель: Стрижов В. В. Задачу поставил: Бахтеев О. Ю. Консультант: Бахтеев О. Ю.

предложений исчисляется миллионами. Построение корпусов параллельных текстов в таком объёме требует большого количества ресурсов, если таковых данных нет в открытом доступе.

Исследуется возможность оптимизации машинного перевода без учителя. При использовании данного метода не требуется наличие корпусов параллельных текстов, для оптимизации достаточно одноязычных наборов текстов. Основная идея состоит в том чтобы построить общее для обоих языков «скрытое» пространство. В отличие от работы [ссылка на работу про unsupervised translation], в данной работе в качестве модели "кодировщик-декодировщик"предлагается использовать T2T. Данная модель позволяет учесть зависимости между словами внутри предложений на нескольких уровнях, что улучшает итоговую модель перевода.

2 Постановка задачи

Пусть задана обучающая выборка состоящую из двух произвольных корпусов предложений для каждого из языков $\mathfrak{D}_{src}$ и $\mathfrak{D}_{tgt}$. Пусть также задана валидационная выборка параллельных корпусов $\mathfrak{D}_{valid}$ для проверки качества метода.
Пусть заданы модели декодировщика $\mathbf{g}$ и кодировщика $\mathbf{f}$. Кодировщик переводит предложения на каждом из языков в одно латентное пространство, декодировщик, в свою очередь, ставит в соответствие векторам из латентного пространства предложения на каждом из языков.

Нужно найти отображение в скрытое пространство и обратно, набор требований к этим отображениям может выглядеть следующим образом:
1. Переводчик быстро обучается выдавать одно и то же предложение. Эта проблема решается следующим образом. Пусть задано предложение $x \in \mathfrak{D}_{l_1}$ на $l_1 \in \{src, tgt\}$, по нему строится зашумленное предложение $C(x)$, которое переводится кодировщиком в латентное пространство, затем восстанавливается обратно. Рассматривается расстояние между изначальным предложением x и полученным $\hat{x} = \mathbf{g}(\mathbf{f}(C(x),\ l_1),\ l_1)$:

$$\mathscr{L}_{cd} = \mathsf{E}_{x \sim \mathfrak{D}_{l_1}, \hat{x} \sim \mathbf{g}(\mathbf{f}(C(x),\ l_1),\ l_1)}[\Delta(x, \hat{x})]$$

2. Для того, чтобы оптимизировать перевод, строится начальный перевод $M(x)$ [Ссылка], не использующий параллельные корпуса. Вводятся $l_1 \in \{src, tgt\},\ l_2 = l_1^c$. На вход кодировщику подаётся зашумленный перевод $C(M(x))$ предложения $x \in \mathfrak{D}_{l_1}$, из результата в латентном пространстве декодировщик получает предложение $\hat{x} = \mathbf{g}(\mathbf{f}(C(M(x)),\ l_2),\ l_1)$. Затем рассматривается расстояние между полученным предложением и изначальным:

$$\mathscr{L}_{cd} = \mathsf{E}_{x \sim \mathfrak{D}_{l_1}, \hat{x} \sim \mathbf{g}(\mathbf{f}(C(M(x)),\ l_2),\ l_1)}[\Delta(x, \hat{x})]$$

3. Также интуитивно хочется, чтобы по вектору из скрытого пространства не было понятно из какого языка оно кодировщик его отобразил. Для этих целей мы вводим нейронную сеть, которую назовём дискриминатор, он будет угадывать по вектору из скрытого пространства исходный язык. Исходная модель штрафуется, когда дискриминатор угадывает исходный язык.

$$\mathscr{L}_{adv} = -\mathscr{L}_{discr}$$

Итоговая функция потери:

$$\mathscr{L}_{res} = C_{auto} \cdot \mathscr{L}_{auto} + C_{cd} \cdot \mathscr{L}_{cd} + C_{adv} \cdot \mathscr{L}_{adv}$$

Литература