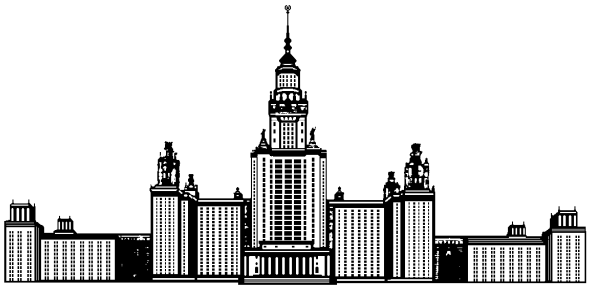

Московский государственный университет имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математических методов прогнозирования

Мазаев Павел Антонович

Состязательные методы для обучения представлений слов

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:
Е. А. Соколов

Москва, 2018

Содержание

Аннотация

В данной работе исследуется возможность модификации недавно предложенного метода для состязательного обучения представлений слов. Рассматриваются предобученные представления слов, с помощью состязательного обучения проводится отображение этих представлений в одно пространство. Исследуется перспективность использования вспомогательной задачи классификации для улучшения качества работы или ускорения процесса обучения, а также влияние различных факторов на динамику обучения генеративной состязательной модели.

Введение

Задачи автоматической обработки текста всегда занимали значительное место среди задач машинного обучения, а в течение нескольких последних лет стали ещё популярнее в связи с появлением значительных по объёму корпусов данных.

Одним из способов улучшить качество работы многих из алгоритмов для задач машинного обучения является применение transfer learning. Под этим термином подразумевается подход, когда для модели машинного обучения используется каким-либо образом информация, извлечённая из других задач. За счёт этого вносится дополнительная информация, и качество работы алгоритма растёт.

Одним из способов для осуществления такого переноса информации для задач обработки текстов является использование предобученных представлений слов (word embedding). Под ними подразумеваются в первую очередь вещественнозначные вектора.

Существующие методы предобучения таких векторов позволяют учесть статистическую связь между разными словами и позволяют оценивать вероятности словосочетаний, что необходимо для создания языковой модели [1].

Существуют различные подходы к обучению представлений слов, которые будут обсуждены ниже.

В случае применения алгоритмов обработки текстов часто встречается необходимость использовать схожие алгоритмы для текстов на разных языках. Поскольку нет никаких причин представлениям слов на разных языках быть равными или близкими, это означает, что необходимо обучать новую модель.

На практике некоторые методы обучения представлений слов для разных языков дают представления, обладающие некоторой степенью изоморфизма. То есть существует возможность с помощью линейного преобразования отобразить одно множество представлений в другое, с сохранением линейных операций, что важно для некоторых видов представления.

Отображение одного пространства представлений слов в другое может быть полезно для применения алгоритмов моделей, обученных на языках с большим количеством данных, для языков, где таких данных не очень много. Подобная задача рассматривается в [2] в контексте разметки частей речи в предложениях.

С другой стороны, состязательный генеративный подход (GAN) [3] также показывает значительные успехи в разных задачах машинного обучения, такик как обработка картинок и текстов.

Идеи использовать состязательные сети для нахождения отображения из одного пространства слов в другое исследуются последние несколько лет, и в данной работе исследуется возможность улучшения одного из существующих методов.

Постановка задачи

В данной работе ставится цель обучить с помощью состязательных методов новые представления слов на основе существующих представлений слов. Мы обучаем новые представления слов одного языка таким образом, чтобы они были близки к представлениям слов другого языка. Метрики близости также будут обсуждены.

Предлагаемый метод заключается в добавлении третьего компонента в схему генератор-дискриминатор для состязательного обучения, а именно классификатор текстов. Гипотеза состоит в том, что задача классификации текста способна положительно повлиять на качество обучаемых представлений слов или на динамику работы алгоритма.

Это предположение основывается на существующей практике проводить дообучение векторов представлений слов в процессе решения основных задач или для адаптации к более узким по тематике текстам [4].

Обзор литературы

Методы обучения представлений слов

Идея обучения представлений слов в виде веществennозначных векторов основываются на теории дистрибутивной семантики. Дистрибутивная семантика – раздел лингвистики, который изучает семантические отношения между лингвистическими объектами с помощью исследования распределений этих объектов [5]. Существует так называемая дистрибутивная гипотеза, согласно которой объекты, встречающиеся в схожих контекстах, имеют схожий смысл. Под контекстами могут подразумеваться, например, соседние слова или документы. В таком случае компоненты можно интерпретировать как некторые параметры распределения контекстов.

LSA

К ранним моделям, предназначенным для обучения таких представлений, относится латентный семантический анализ (LSA) [6]. Этот метод основан на факторизации терм-документной матрицы, такой как, например, матрица TF-IDF [7]. Эта матрица содержит статистику о частоте вхождения данного слова в данный документ. В такой матрице A столбцы соответствуют документам, а строки – столбцам, и в $A_{t,d}$ содержится инофрмация о вхождении слова t в документ d.

После того, как эта матрица построена, производится её низкораноговое приближение, например с помощью SVD: $A \approx U\Sigma V^*$. Строки матрицы U можно брать в качестве представлений слов.

LDA

Латентное размещение Дирихле – генеративная вероятностная тематическая модель, моделирующая взаимосвязь между документами и словами, вводя дополнительно скрытые переменные, интерпретируемые как темы. Согласно этой модели, каждому документу соотвествует распределение по темам, а для каждой темы – распределение по словам. После обучения парметров модели возможно извлечь представления слов из распределения слов в зависимости от темы.

Skipgram

Этот метод был предложен в [8]. Данный метод использует для обучения представлений слов нейронную сеть, обучаемую без учителя на корпусе текстов. Пусть рассматривается i-е слово. Обучаемый вектор представления слова w_i используется для того, чтобы предсказать t слов до позиции i и t – после.

Для этого вычисляется вероятность $p(w_{t+k}|w_t) = \frac{exp(v_{w_t}^T \tilde{v}_{t+j})}{\sum_{w_i \in V} exp(v_{w_t}^T \tilde{v}_i)}$, где v_{w_t} – обучаемые представления слов, $\tilde{v}_i$ – вектора параметров, которые можно тоже интерпретировать как другой набор представлений слов или как компоненты проекционной матрицы. Данные вектора также обучаются при оптимизации модели.

Важно заметить, что данная вероятность зависит от скалярных произведений представлений слов, которые совпадают с косинусной мерой близости, в случае если эти вектора нормированы. Поскольку косинусная мера близости и скалярное произведение связаны монотонным, между оптимизацией этих двух мер близости есть связь.

Этот факт обосновывает возможность применения косинусной меры в качестве меры близости для представлений слов, обученных данным методом, а использование косинусной меры распространено в обработке естественных языков [1].

Также важным эмпирическим свойством данного метода обучения представлений слов является то, что можно придать смысл операциям над данными векторами [8].

Генеративные состязательные сети

Генеративные состязательные (конкурирующие) сети (Generative Adversarial Networks) [3] – модель машинного обучения, использующая нейронные сети, позволяющая моделировать сложные распределения.

Данный подход заключается в следующем. Пусть существует источник подлинных данных и источник «шума». Одновременно обучаются две нейросети: генеративная сеть G и дискриминативная сеть D. G учится преобразовывать то, что ей приходит на вход, в объекты, похожие на объекты из распределения подлинных

данных. D представляет из себя классификатор, выдающий вероятность того, что объект, пришедший к нему на вход, пришёл из подлинных данных, а не является результатом работы G. Цель обучения состоит в том, чтобы генератор создавал объекты, повышающие вероятность ошибки дискриминатора. Дискриминатор же обучается определять подлинность объектов. С формальной точки зрения, происходящее представляет из себя следующую игру с нулевой суммой:

$$\min_G \max_D \left[\mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_{noise}(z)}[\log(1 - D(G(z)))]\right] \quad (1)$$

В данной формуле первое слагаемое отвечает за качество работы дискриминатора на объекте из подлинного распределения, а второе – за качество работы на преобразованных объектах из шумовых данных. $p_{data}(x)$ – распределение на подлинных объектах, $p_{noise}(z)$ – распределение на шумовых данных. Название «шум» условное, имеется в виду, что это некоторое другое распределение.

В общем случае, G и D могут быть произвольными, но на практике используются лишь глубокие нейронные сети.

На практике обучение данной модели производится с помощью чередования обучения генератора и дискриминатора, и процесс обучения очень нестабилен. Теория полагает, что оптимальное значение достигается в случае, когда $\mathbb{E}_{x\sim p_{data}(x)}D(x) = \mathbb{E}_{x\sim p_{data}(x)}(1 - D(G(z)))$, но на практике это не так.

Нам важен тот факт, что данная модель учится отображать объекты из одного распределения (например, из стандартного нормального) в объекты другого распределения, например в изображения. Мы будем пользоваться этим свойством для обучения отображения из пространства представлений слов, обученных на языке A в пространство представлений слов языка B.

Методы отображения представлений слов

Существующие методы отображения делятся на две значительные подгруппы: требующие параллельной выборки для обучения и не требующие таковой.

Обучение с параллельным корпусом

К первому типу методов относятся методы, предложенные в [9]. Постановка задачи следующая: пусть даны N пар слов вида $\{a_i, b_i\}_{i=1}^N$, где слово b_i является векторным представлением перевода слова, соответствующему представлению слову a_i из языка A на язык B. Вычисляется матрица $W = argmin_W \sum_{i=1}^N ||Wa_i - b_i||^2$. Тогда для нового слова $\tilde{a}$ в качестве перевода берётся слово, соответствующее ближайшему по косинусной мере близости вектору $\tilde{b}$ после преобразования.

Таким образом, по известным парам слов строится матрица преобразования, после чего преобразуется весь словарь на соответствующем языке.

Также авторы той статьи рассматривают модификации меры близости, что приводит к улучшению качества пословного перевода.

Интересен подход, предлагаемый [10]. В той статье предлагается способ итеративно получать аналитическое локальное решение, последовательно улучшая качество пословного перевода. Важно, что их подход позволяет использовать очень малое количество слов в словаре (около 25), при этом используя векторы-представления для чисел в цифровой записи.

В [11] обсуждается согласованность оптимизирумой меры близости (евклидова расстояния или скалярного произведения) и меры близости, используемой при тестировании (косинусной меры близости). Также в той статье рассматривается необходимость накладывания ортогонального ограничения на матрицу преобразования, а также нормирования длины векторов представлений слов.

Эти вопросы также рассматриваются в [12].

Состязательный подход

Подобные методы могут использовать параллельные данные, но существуют работающие модели, не использующие словарей.

При таком подходе используют стандартную модель генеративной состязательной сети, фиксируя представления слов на одном языке в качестве выборки X, а представления слов на другом языке – в качестве источника «шума» Z в обозначениях, введённых выше. В результате работы генератор выучивает отображение одного пространства в другое.

В работе [13] для целей выравнивания пространств представлений слов рассматривается генеративная состязательная сеть, использующая метрику Вассерштейна в качестве меры близости распределений.

В [14] ставятся эксперименты со стандартной схемой генеративной состязательной модели и исследуется влияние размерности представления слова на качество пословного перевода. В работе [15] используется такой же подход, но предлагается усовершенствование метода пословного перевода, основанного на методе ближайших соседей, и валидационный критерий, не использующий словари.

Также существуют работы [16] посвящённые состязательным автокодировщикам [17] для такой же задачи.

Предлагаемый метод

Предлагаемый метод основан на работе [15]. Модель состоит из трёх основных частей: дискриминатора D, трансформатора (или трансформаторов) и классификато-

ра C. В данном разделе мы будем говорить, что используются две трансформации T_A и T_B, одна из которых может быть тождественной.

В качестве дискриминатора используется двуслойная нейросеть с нелинейностями вида ReLU. На вход её подаётся случайное подмножество векторов-представлений для языка A и случайное подмножество векторов представлений для языка B. Важно заметить, что эти вектора подаются преобразованными должным образом. Для i-го вектора из входного набора дискриминатор выдаёт вероятность того, что этот вектор до преобразования был для языка B.

В качестве трансформатора использовалось линейное преобразование: матрица $N \times N$ и вектор сдвига. Несмотря на то, что вектор сдвига не использовался в оригинальной работе, эксперименты проводились с ним. На матрицу накладывается ограничение приближенной ортогональности, как советутся в [12].

В качестве классификатора использовалась рекуррентная нейронная сеть (RNN) [18], основанная на двунаправленной LSTM-ячейке [19]. После кодирования последовательности слов в скрытое состояние, это состояние подавалось ещё двум полносвязным слоям. В качестве нелинейностей использовались так называемые LeakyReLU. На вход этой сети подавались предложения, преобразованные в ряды представлений слов. Эти представления слов также преобразовывались перед подачей в классификатор. Результатом работы этой сети являлась вероятность принадлежности к одному из классов текстов. Важно отметить, что с самого начала используется один классификатор для обоих языков, даже на тех итерациях, когда качество трансформации низкое.

Рассмотрим выражения для функций потерь дискриминатора и классификатора, выраженные непосредственно через входы соответствующих сетей, игнорируя распределение, откуда были поданы данные. В качестве функции потерь в обоих случаях берётся отрицательный логарифм функции правдоподобия, что для бинарного случая есть кросс-энтропия.

Для дискриминатора:

$$L_{discr}(X) = -\frac{1}{|X|}\sum_{i=1}^{|X|}\left[y_i \log D(x_i) + (1-y_i)\log(1-D(x_i))\right] \tag{2}$$

Для классификатора:

$$L_{class}(X) = -\frac{1}{|X|}\sum_{i=1}^{|X|}\sum_{k=1}^{N_{classes}}[y_i = k]\log C(x_i) \tag{3}$$

Процесс обучения разбит на так называемые глобальные итерации, которых производится N_{global}. На каждой из них производится по N_{discr} итераций обучения дискриминатора, N_{trans} итераций обучения трансформатора N_{class} итераций обучения классификатора. Число итераций – гиперпараметры алгоритма.

Рассмотрим теперь каждую итерацию отдельно.

На итерации обучения дискриминатора случайно без возвращения выбираются $E_{discr,A}$ и $E_{discr,B}$ представлений слов из соответствующего языка. К ним применяются соответствующие транформаторы T_A и T_B, из конкатенации этих данных получается вектор признаков X, имеющий размерность $(E_{discr,A} + E_{discr,B}) \times D_{embedding}$. После чего составляется вектор целевой переменной Y, где значение 0 соответствует тому, что вектор был из языка A, а 1 – языку B.

Существует методика, называющаяся сглаживанием меток, заключающаяся в том, что Y заполняется не 0 и 1, а $coeff$ и $1 - coeff$. На ранних стадиях экспериментов был сделан вывод, что эта методика не вносит улучшений, и дальше эксперименты с ней не проводились. Подробнее этот подход описан в статье [20].

На основе X и Y вычисляется функция потерь дискриминатора и делается оптимизационный шаг на параметры дискриминатора.

На итерации трансформатора случайно без возвращения выбираются $E_{trans,A}$ и $E_{trans,B}$ представлений слов из соответствующего языка, назовём их X_A и X_B. Применяются соответстсвующие трансформаторы, аналогично тому, как это было сделано на шаге дискриминатора и составляется X. В качестве вектора целевой переменной берётся вектор из бернулиевского распределения с параметром $p = 0.5$. Обычно при обучении генератора для генеративных состязательных сетей метки классов меняются местами. Эксперименты показывают, что это несколько уменьшает стабильность качества «обманывания» дискриминатора трансформатором, поскольку последний выучиывает, как менять метки. Подобный подход был предложен в статье [21]. В оригинальной статье используется традиционный подход.

Также в статье [15] предлагается не накладывать ограничение ортогональности на матрицу преобразования, а делать шаг для обеспечения близости матрицы к ортогональной. В предлагаемой модели делается такой же шаг.

После этого X (состоящий из преобразованных представлений слов) подаётся на вход дискриминатору. Вычисляется функция потерь трансформатора, совпадающая с функцией потерь дискриминатора, после чего делается оптимизационный шаг по параметрам трансформаторов.

Рассмотрим далее классификационный шаг. На нём выбирается S_A и S_B текстов из набора предложений для каждого из языка. Слова в этих предложениях кодируются в представления слов с помощью функции-словаря. Слишком длинные тексты обрезаются до длины L_{max}, которая также является параметром алгоритма. В данных подвыборках находятся максимальные по обрезанной длине предложения l_A и l_B, после чего слишком короткие предложения добиваются нулевым вектором-представлением. К каждому слову в каждом предложении применяется трансформатор для соответствующего языка. В итоге получается тензор размерности $(S_A + S_B) \times \max(l_a, l_b) \times D_{embedding}$.

Вектор целевой переменной получается конкатенацией векторов целевой переменной для предложений.

Вычисляется функция потерь для классификатора и делается оптимизационный шаг по параметрам классификатора и трансформаторов.

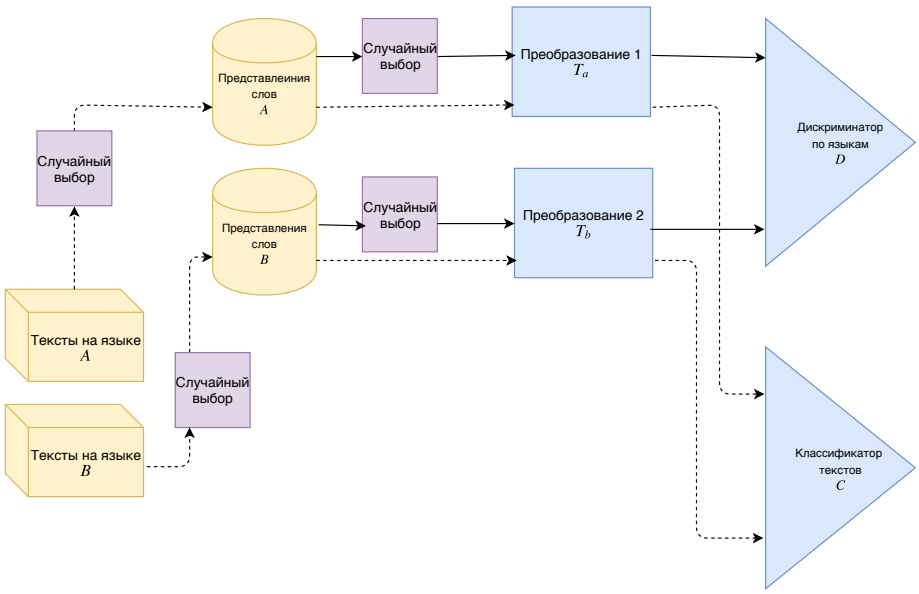

Рис. 1: Схема предлагаемой модели. Сплошные линии обозначают преобразования над представлениями слов при шаге дискриминатора и шаге трансформатора. Штрихованные линии обозначают преобразования текстов при шаге классификатора.

Посмотрим, как выглядят формулы для функций потерь, которые оптимизируются на каждом шаге, на этот раз учитывая разнородность данных, поступающих на вход сетям. Данные функционалы минимизируются в процессе обучения модели.

Для дискриминатора:

$$\mathcal{L}_{discr}(X_A, X_B) = -\frac{1}{E_{discr,A} + E_{discr,B}} \left[\sum_{i=1}^{E_{discr,A}} \log D(T_A(x_{a,i})) + \sum_{i=1}^{E_{discr,B}} \log(1 - D(T_B(x_{b,i}))) \right] \tag{4}$$

Для трансформаторов:

$$\mathcal{L}_{transform}(X_A, X_B) = -\frac{1}{E_{transform,A} + E_{transform,B}} \times$$
$$\times \left[\sum_{i=1}^{E_{transform,A}} y_{a,i} \log D(T_A(x_{a,i})) + \sum_{i=1}^{E_{transform,B}} (1 - y_{b,i}) \log D(T_B(x_{b,i})) \right] \quad (5)$$

где $y_{a,i}, y_{b,i} \sim Bernoulli(\frac{1}{2})$.

Наконец, для классификатора:

$$\mathcal{L}_{classifier}(X_A, X_B) = -\frac{1}{S_A + S_B} \times$$
$$\times [\sum_{i=1}^{S_A} \sum_{k=1}^{N_{classes}} [y_{a,i} = k] \log C(T_A(x_{a,i})) + \sum_{i=1}^{S_B} \sum_{k=1}^{N_{classes}} [y_{b,i} = k] \log C(T_B(x_{b,i}))] \quad (6)$$

Запишем это в виде алгоритма

Алгоритм 1 Обучение модели

for global_iteration = 1 ...N_{global} do
  for $k = 1 \dots N_{discr}$ do
    sample X_A^0 from A, $|X_A^0| = E_{discr,A}$
    sample X_B^0 from B, $|X_B^0| = E_{discr,B}$
    $X_A \leftarrow T_A(X_A^0)$, $X_B \leftarrow T_B(X_B^0)$
    $Y_A \leftarrow 0$, $Y_A \in \mathbb{R}^{E_{discr,A}}$
    $Y_B \leftarrow 1$, $Y_B \in \mathbb{R}^{E_{discr,B}}$
    $X \leftarrow [X_A; X_B]$, $Y \leftarrow [Y_A; Y_B]$
    $L \leftarrow L_{discr}(X, Y)$
    $\theta_D^k \leftarrow Optimizer(\theta_D^{k-1}, \nabla_\theta L_{|\theta_D^{k-1}})$
  for $k = 1 \dots N_{transform}$ do
    sample X_A^0 from A, $|X_A^0| = E_{transform,A}$
    sample X_B^0 from B, $|X_B^0| = E_{transform,B}$
    $X_A \leftarrow T_A(X_A^0)$, $X_B \leftarrow T_B(X_B^0)$
    $Y_A \leftarrow Bernoulli(0.5)$, $Y_A \in \mathbb{R}^{E_{discr,A}}$
    $Y_B \leftarrow Bernoulli(0.5)$, $Y_B \in \mathbb{R}^{E_{discr,B}}$
    $X \leftarrow [X_A; X_B]$, $Y \leftarrow [Y_A; Y_B]$
    $L \leftarrow L_{discr}(X, Y)$
    $\theta_{T_A}^k \leftarrow Optimizer(\theta_{T_A}^{k-1}, \nabla_{\theta_{T_A}} L_{|\theta_{T_A}^{k-1}})$
    $\theta_{T_B}^k \leftarrow Optimizer(\theta_{T_B}^{k-1}, \nabla_{\theta_{T_B}} L_{|\theta_{T_B}^{k-1}})$
  for $k = 1 \dots N_{classifier}$ do
    sample X_A^0, Y_A from $Dataset_A$, $|X_A^0| = S_A$
    sample X_B^0, Y_B from $Dataset_B$, $|X_B^0| = S_B$
    $X_A \leftarrow T_A(X_A^0)$, $X_B \leftarrow T_B(X_B^0)$
    $X \leftarrow [X_A; X_B]$, $Y \leftarrow [Y_A; Y_B]$
    $L \leftarrow L_{classifier}(X, Y)$
    $\theta_C^k \leftarrow Optimizer(\theta_C^{k-1}, \nabla_{\theta_C} L_{|\theta_C^{k-1}})$
    $\theta_{T_A}^k \leftarrow Optimizer(\theta_{T_A}^{k-1}, \nabla_{\theta_{T_A}} L_{|\theta_{T_A}^{k-1}})$
    $\theta_{T_B}^k \leftarrow Optimizer(\theta_{T_B}^{k-1}, \nabla_{\theta_{T_B}} L_{|\theta_{T_B}^{k-1}})$

Методы пословного перевода после отображения

Для того, чтобы перевести слово w с языка A в язык B необходимо

- Получить векторное представление этого слова

- Преобразовать его с помощью T_A

- Взять все имеющиеся представления для слов языка B и преобразовать их с помощью T_B.

- Поскольку мы предполагаем, что после преобразований пространства представлений слов становятся близкими, то в качестве перевода слова w необходимо брать ближайшего соседа.

Отношение «быть ближайшим соседом» несимметрично, то есть если w_1 – ближайший сосед w_2, то это не значит, что верно обратное. Есть и другая проблема, в [22] было показано, что при отображении пространств представлений слов существует точки, которые являются соседями для очень большого числа других точек, в то время как некоторые другие объекты не являются таковыми ни для какой другой точки. Таким образом, точки первого вида вносят в шум и не позволяют найти правильный перевод слова с помощью метода ближайшего соседа. Авторые статьи [15] предлагают усовершенствование поиска перевода с помощью метода ближайших соседей, который позволяет разрешить эту проблему лучше, чем методы, предложенные ранее. Этот метод был назван ими CSLS (cross-domain similarity local scaling).

Рассматривается двудольный граф, в котором вершины одной доли соответствуют словам из языка A, а из другой доли – словам из языка B. Пусть $\mathcal{N}_B(T_A(w_a))$ – множество представлений w_b для слов B, таких что $T_B(w_b)$ являющихся одним из K ближайших соседей для $T_A(w_a)$. Аналогично определим $\mathcal{N}_A(T_B(w_b))$. Рассмотрим среднюю меру близости $T_A(w_a)$ к представлениям слов из $\mathcal{N}_B(T_A(w_a))$:

$$r_A(T_A(w_a)) = \frac{1}{K} \sum_{w_b \in \mathcal{N}_B(T_A(w_a))} cos(T_A(w_a), T_B(w_b)) \tag{7}$$

Аналогично для $r_B(T_B(w_b))$. Тогда мера близости CSLS определяется следующим образом:

$$CSLS(T_A(w_a), T_B(w_b)) = 2cos(T_A(w_a), T_B(w_b)) - r_A(T_A(w_a)) - r_B(T_B(w_b)) \tag{8}$$

Метрики качества

В качестве метрики качества использовалась точность пословного перевода. Для этого использовались те же корпуса, что и в статье [15]. Эти параллельные корпуса

слов, то есть они состоят из пар вида «исходное слово – перевод». Параметры корпусов приведены в таблице ниже.

	валидационный	тестовый
английский – испанский	11977	2975
испанский – английский	8667	2416

Таблица 1: Количество пар слов в соответствующих корпусах.

Качество перевода измерялось при помощи того же кода, что и у авторов статьи [15].

В качестве кандидатов в перевод брались K ближайших соседей (в некотором смысле). Если эталонный перевод оказывается среди этих кандидатов, то считается, что перевод оказался правильным. K является параметром метрики.

Результаты были сравнены по мере сходства CSLS с $K = 1$.

Эксперименты

Данные

Данные для задачи классификации

Подробные данные о корпусах можно увидеть в таблице ниже.

Для обучения вспомогательной задачи классификации использовались корпуса на английском и испанском языках, в каждом из них около 18000 новостных текстов. Эти тексты имеют значительную длину (средняя длина больше 400 слов). В качестве разметки каждому сопоставлен один из 26 классов, соответствующих тематике текста. Классы несбалансированные.

В качестве предобработки к каждом тексту применялась нормализация: все слова приводились к нижнему регистру, удалялась пунктуация.

Слова, встречающиеся реже 5 раз во всём корпусе в дальнейшем игнорировались и заменялись на особое служебное слово.

Представления слов

В качестве представлений слов, которые позже преобразовывались, были взяты предобученные представления слов. Обучение осуществлялось на текстах из Википедии на соответствующем языке, с помощью метода skipgram [8].

Поскольку новостные тексты, которые использовались для классификации, содержат всего около 47000 уникальных слов, встречающихся чаще 5 раз, из числа

предобученных представлений были выбраны те, которые нужны для описания текстов. После этого были взяты самые частые не взятые ранее слова таким образом, что общее число слов в словаре составляло 50000.

Результаты экспериментов

Сравнение модифицированного метода с немодифицированным

В качестве метрики использовалась точность пословного перевода с помощью метода CSLS для поиска соответствий. Осуществлялось по 12 запусков каждого вида. Валидация и тестирование проводились на наборах, предоставляемых авторами facebook MUSE. Измерение метрики качества осуществлялось каждые 100 глобальных итераций. Поскольку данный вид эксперимента весьма продолжителен по времени, при каждой валидации осуществлялся подсчёт всех метрик, в том числе и на тесте. Анализ производился позже по логам. Это производилось следующим образом: по выбранному валидационному набору искались номера валидаций с наилучшим выбранным критерием качества, после чего выбиралось соответствующее значение тестовой метрики.

Для перевода в обратную сторону использовались те же номера оптимальных валидаций, что и для перевода в прямую сторону.

Все величины даны в процентах. Величины округлены до десятых.

Перевод английский-испанский

	Среднее	Медиана	σ	Средняя разность
С классификатором	74,5 %	74,7 %	1,9 %	1,6 %
Без классификатора	72,4 %	72,6 %	2,4 %	2,3 %

Таблица 2: Результаты эксперимента с измерением точности пословного перевода. σ – средневадратичное отклонение. Под средней разностью подразумевается среднее значения разности метрик на валидационной выборке и на тестовой.

Перевод испанский-английский

	Среднее	Медиана	σ	Средняя разность
С классификатором	70,1 %	71,0 %	2,6 %	6.0 %
Без классификатора	69,7 %	69,5 %	2,2 %	5,0 %

Таблица 3: Результаты эксперимента с измерением точности пословного перевода. σ – средневадратичное отклонение. Под средней разностью подразумевается среднее значения разности метрик на валидационной выборке и на тестовой.

Необходимо заметить, что при выборе оптимального состояния модели по точности перевода английский-испанский, статистически значимого прироста качества при переводе в обратную сторону не наблюдается.

Для проверки значимости наблюдаемого улучшения качества был рассмотрен непараметрический критерий Манна-Уитни. Этот критерий был применён к выборкам из значений качества пословного перевода с английского на испанский с классификатором и без классификатора. Значение критерие равно 36.0, что соответствует наличию статистической значимости.

Также был проделан симметричный эксперимент. Он во всё аналогичен предыдущему, только трансформация применяется к английскому языку.

Перевод испанский-английский

	Среднее	Медиана	σ	Средняя разность
С классификатором	71,6 %	71,5 %	1,4 %	1,2 %
Без классификатора	69,6 %	69.5 %	2,2 %	5,0 %

Таблица 4: Результаты эксперимента с измерением точности пословного перевода. σ – средневадратичное отклонение. Под средней разностью подразумевается среднее значения разности метрик на валидационной выборке и на тестовой.

Аналогичным образом был применён критерий Манна-Уитни, было получено значение статистики 8.0, что также соответствует статистической значимости. Количество запусков с классификатором 6, без классификатора – 8. Остальные параметры те же, что и в прошлом эксперименте.

Заметим также, что в экспериментах с классификатором, в которых качество пословного перевода измеряется в том же направлении, по которому осуществлялся выбор оптимального состояния системы, дисперсия ниже, чем в экспериментах без классификатора.

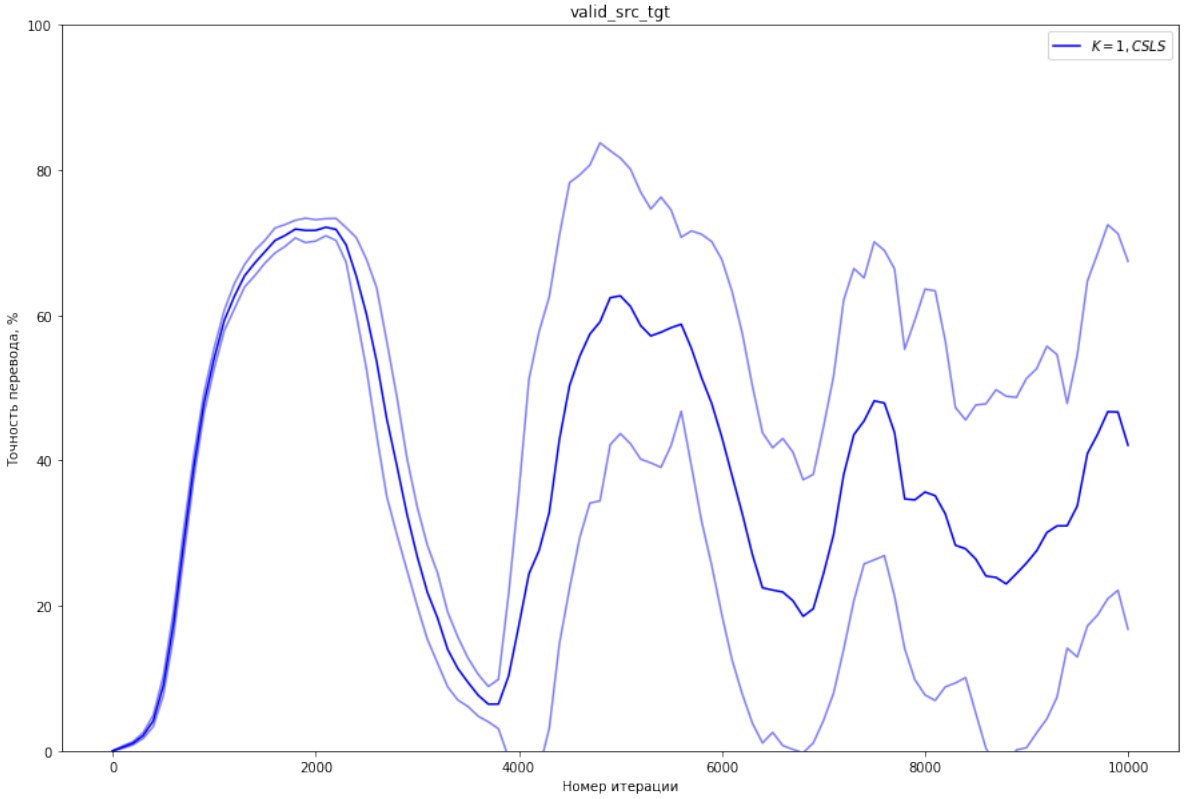

Рис. 2: Усреднённое по запускам значение метрики на валидации в зависимости от итерации с одним стандартным отклонением. Данная динамика обучения типична для данной задачи, схожие графики наблюдаются в частности, у [15]. Усреднение проводилось по экспериментам для перевода с английского на испанский, для тех экспериментов, по которым были получены результаты выше.

Параметры эксперимента

Параметр	Значение
$E_{discr,A}$	1024
$E_{discr,B}$	1024
$E_{trans,A}$	1024
$E_{trans,B}$	1024
S_A	256
S_B	256
L_{max}	100
N_{global}	10000
$N_{transform}$	25
N_{discr}	5
$N_{classifier}$	10
$D_{embedding}$	300

Таблица 5: В данной таблице приведены параметры для экспериментов с классификатором. Значения параметров объяснены в разделе «Предлагаемый метод». Для проведения экспериментов без классификатора значение $N_{classifier}$ выставлялось в 0.

Эксперимент для проверки значимости модификации

Для проверки значимости модификации был проведён следующий эксперимент. Практика показала, что при тех же параметрах, что и прошлом эксперименте, но с применением трансформаций к представлениям слов для обоих языков: качество перевода остаётся около нуля:

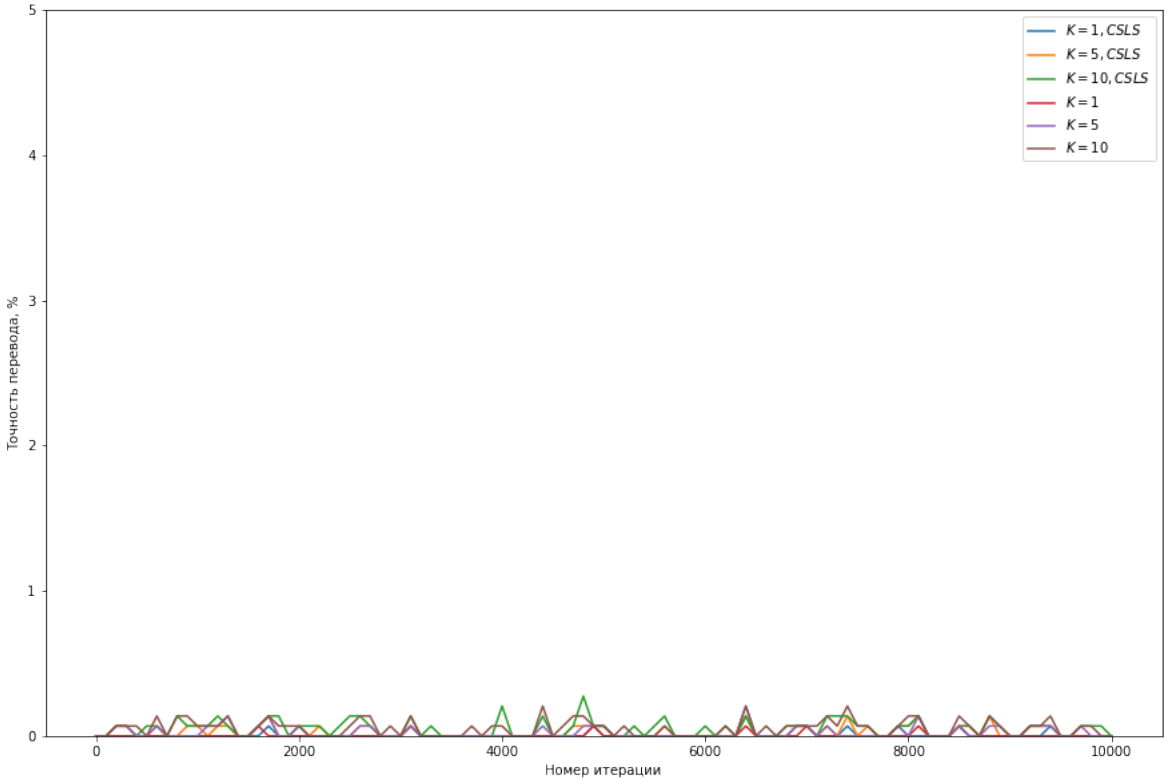

Рис. 3: Масштаб изменён до максимального значения 5% Можно видеть, что все метрики имеют пренебрежимо малое значение независимо от K соседей, рассматриваемых в качестве возможного перевода. Метрики качества при обучении отображения только при состязательном обучении, без классификатора

Но при добавлении классификатора ситуация изменяется:

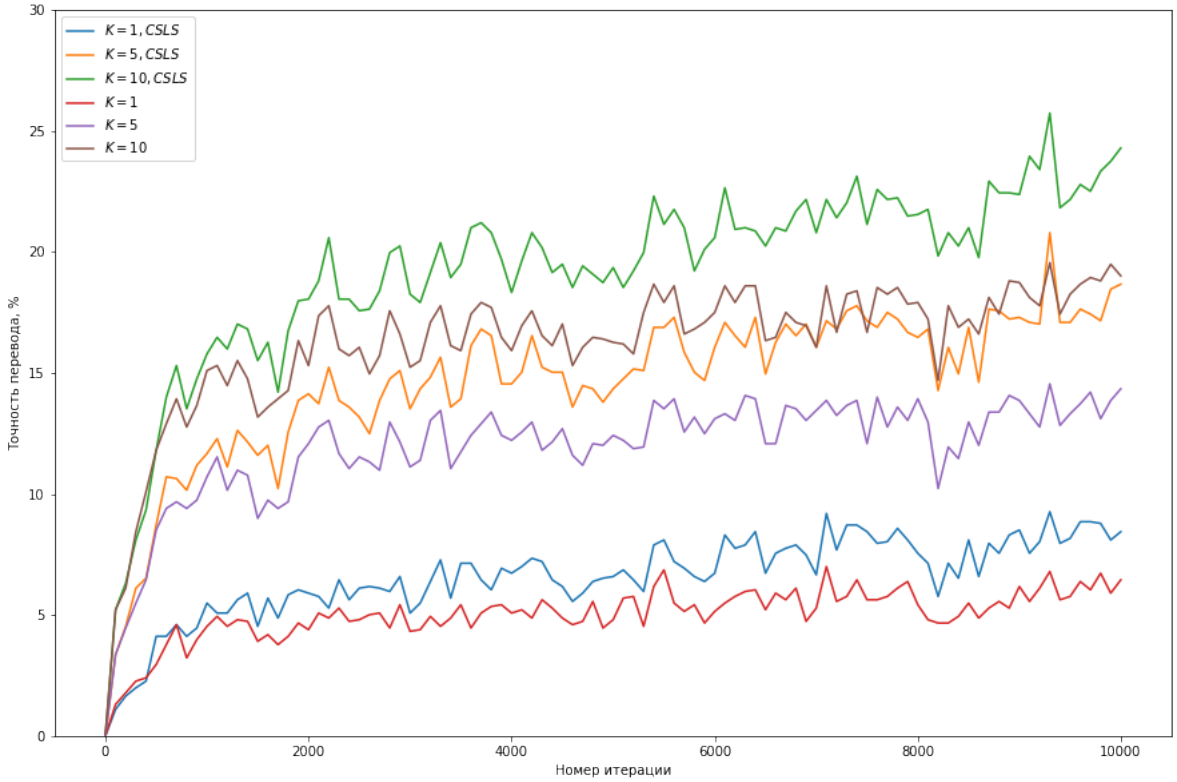

Рис. 4: Масштаб изменён до максимального значения 30%. Несмотря на тот факт, что качество остаётся значительно ниже качества модели с одной трансформацией, можно наблюдать значительное отличие метрик от нуля (до 8.5% при $K = 1$)

При этом был проведён эксперимент, где не производилось обучения дискриминатора или трансформаторов:

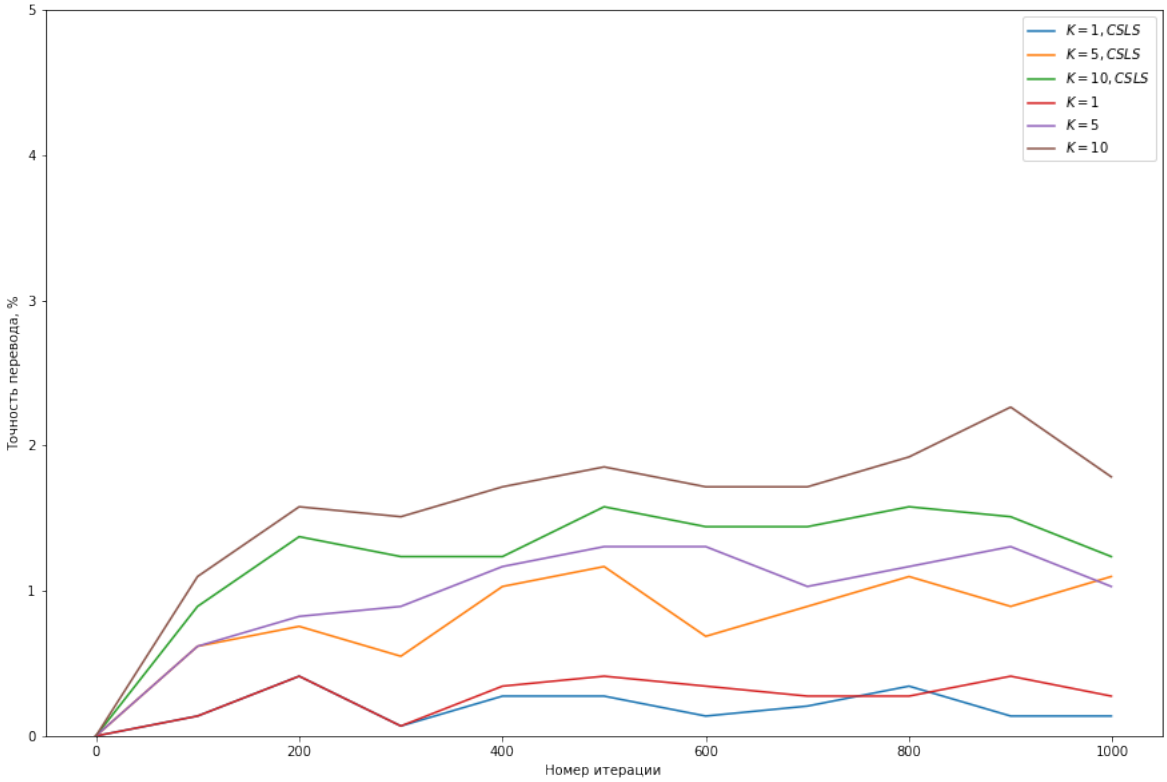

Рис. 5: Масштаб изменён до максимального значени 5%. Метрики качества при обучении отображения при помощи только классификатора

Из этого делается вывод, что изучение комбинаций состязательной модели и классификатора текстов, является перспективным.

Другие эксперименты

Было проведено сравнение с качеством работы эталонной реализации подхода из [15] от авторов статьи. Для пары английский-испанский был получен средний результат точности перевода $K = 1, CSLS$ равный 72.8% со стандартный отклонением 3.6. Проводилось 9 запусков на тех же представлениях слов, что и в экспериментах с предложенным изменением.

Выводы

Состязательные генеративные сети способны выучить отображение из одного пространства представлений слов одного языка в пространство представлений слов другого языка, при этом достигается значительная точность пословного перевода.

Добавление в модель текстового классификатора положительно влияет на качество пословного перевода. С помощью добавление классификатора удалось добиться статистически значимого улучшения точности пословного перевода на 2% и уменьшить в некоторых случаях дисперсию меры качества.

Гипотеза о благотворном влиянии классификатора на качество была дополнительно проверена в постановке с ослабленной исходной моделью.

Заключение

В данной работе была рассмотрена модификация подхода, исследованного в [15]. В той статье рассматривается состязательный подход к обучению представлений слов для пар языков, основанный на стандартной схеме, состоящей из дискриминатора и генератора. Наша модификация заключается в добавлении дополнительного элемента в эту схему, а именно классификатора, осуществляющего многоклассовую классификацю текстов на разных языках одновременно.

В данной работе было проведено сравнение предложенного подхода с немодифицированным подходом, а также были проведены дополнительные эксперименты, исследующие значимость предложенного подхода.

Список литературы

[1] Jurafsky, D. Speech and language processing / Dan Jurafsky, James H Martin. — Pearson London:, 2014. — Vol. 3.

[2] Ten pairs to tag-multilingual pos tagging via coarse mapping between embeddings / Yuan Zhang, David Gaddy, Regina Barzilay, Tommi Jaakkola / Association for Computational Linguistics. — 2016.

[3] Generative adversarial nets / Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza et al. // Advances in neural information processing systems. — 2014. — P. 2672–2680.

[4] Yang, X. Supervised fine tuning for word embedding with integrated knowledge / Xuefeng Yang, Kezhi Mao // CoRR. — 2015. — Vol. abs/1505.07931. — http://arxiv.org/abs/1505.07931.

[5] Sahlgren, M. The distributional hypothesis / Magnus Sahlgren // Italian Journal of Disability Studies. — 2008. — Vol. 20. — P. 33–53.

[6] Landauer, T. K. An introduction to latent semantic analysis / Thomas K Landauer, Peter W Foltz, Darrell Laham // Discourse processes. — 1998. — Vol. 25, no. 2-3. — P. 259–284.

[7] Introduction to information retrieval / Mark Sanderson, D Christopher, Hinrich Manning et al. // Natural Language Engineering. — 2010. — Vol. 16, no. 1. — P. 100.

[8] Efficient estimation of word representations in vector space / Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean // CoRR. — 2013. — Vol. abs/1301.3781. — http://arxiv.org/abs/1301.3781.

[9] Mikolov, T. Exploiting similarities among languages for machine translation / Tomas Mikolov, Quoc V. Le, Ilya Sutskever // CoRR. — 2013. — Vol. abs/1309.4168. — http://arxiv.org/abs/1309.4168.

[10] Artetxe, M. Learning bilingual word embeddings with (almost) no bilingual data / Mikel Artetxe, Gorka Labaka, Eneko Agirre // Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). — Vancouver, Canada: Association for Computational Linguistics, 2017. — July. — P. 451–462. — http://aclweb.org/anthology/P17-1042.

[11] Normalized word embedding and orthogonal transform for bilingual word translation / Chao Xing, Dong Wang, Chao Liu, Yiye Lin // Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. — 2015. — P. 1006–1011.

[12] Artetxe, M. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance / Mikel Artetxe, Gorka Labaka, Eneko Agirre // Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. — 2016. — P. 2289–2294.

[13] Earth mover's distance minimization for unsupervised bilingual lexicon induction / Meng Zhang, Yang Liu, Huanbo Luan, Maosong Sun // Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. — 2017. — P. 1934–1945.

[14] Adversarial training for unsupervised bilingual lexicon induction / Meng Zhang, Yang Liu, Huanbo Luan, Maosong Sun // Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). — Vol. 1. — 2017. — P. 1959–1970.

[15] Word translation without parallel data / Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato et al. // CoRR. — 2017. — Vol. abs/1710.04087. — http://arxiv.org/abs/1710.04087.

[16] Barone, A. V. M. Towards cross-lingual distributed representations without parallel text trained with adversarial autoencoders / Antonio Valerio Miceli Barone // CoRR. — 2016. — Vol. abs/1608.02996. — http://arxiv.org/abs/1608.02996.

[17] Adversarial autoencoders / Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian J. Goodfellow // CoRR. — 2015. — Vol. abs/1511.05644. — http://arxiv.org/abs/1511.05644.

[18] Liu, P. Recurrent neural network for text classification with multi-task learning / Pengfei Liu, Xipeng Qiu, Xuanjing Huang // arXiv preprint arXiv:1605.05101. — 2016.

[19] Hochreiter, S. Long short-term memory / Sepp Hochreiter, Jürgen Schmidhuber // Neural computation. — 1997. — Vol. 9, no. 8. — P. 1735–1780.

[20] Improved techniques for training gans / Tim Salimans, Ian Goodfellow, Wojciech Zaremba et al. // Advances in Neural Information Processing Systems. — 2016. — P. 2234–2242.

[21] Simultaneous deep transfer across domains and tasks / Eric Tzeng, Judy Hoffman, Trevor Darrell, Kate Saenko // CoRR. — 2015. — Vol. abs/1510.02192. — http://arxiv.org/abs/1510.02192.

[22] Dinu, G. Improving zero-shot learning by mitigating the hubness problem / Georgiana Dinu, Angeliki Lazaridou, Marco Baroni // arXiv preprint arXiv:1412.6568. — 2014.