**Аннотация**

Системы BERT (Devlin et al., 2018) и RoBERTa (Liuet al., 2019) установили новую современную производительность в задачах регрессии пар предложений таких как семантическое сходство текстов (STS). Однако для этого требуется, чтобы оба предложения подавались в сеть, что приводит к большим вычислительным затратам: Поиск наиболее похожей пары в коллекции из 10 000 предложений требует около 50 миллионов вычислений (~65 часов) при использовании BERT. Конструкция BERT делает его непригодным для поиска семантического сходства, а также для задач без контроля таких как кластеризация. В этой публикации мы представляем Sentence-BERT (SBERT), модификацию предварительно обученной

BERT, которая использует сиамские и триплетные сетевые структуры для получения семантически значимых вкраплений предложений, которые можно сравнивать с помощью косинусного сходства. Это уменьшает усилия по поиску наиболее похожей пары с 65 часов при использовании BERT / RoBERTa до примерно 5 секунд при использовании SBERT, сохраняя при этом точность BERT. Мы оцениваем SBERT и SRoBERTa на обычных задачах STS и задачах трансферного обучения, где они превосходят другие современные решения. методы встраивания предложений.1

**1 Введение**

В этой публикации мы представляем Sentence-BERT (SBERT), модификацию сети BERT с использованием сиамских и триплетных сетей, которая способна получать семантически значимые вкрапления предложений2. Это позволяет использовать BERT для решения некоторых новых задач, которые до сих пор были неприменимы для BERT. К таким задачам относятся крупномасштабные семан сравнение сходств, кластеризация и поиск информации через семантический поиск. BERT показал новые современные результаты в классификации различных предложений и пар предложений регрессии. BERT использует перекрестный кодер: Два предложения передаются в сеть трансформатора и предсказывается целевое значение. Однако такая схема не подходит для различных задач парной регрессии из-за слишком большого количества возможных комбинаций. Поиск в коллекции из n = 10 000 предложений пары с наибольшим сходством требует с помощью BERT n-(n-1)/2 = 49 995 000 вычислений. На современном графическом процессоре V100 это занимает около 65 часов. Аналогично, поиск того, какой из более чем 40 миллионов существующих вопросов Quora является наиболее похожим на новый вопрос, может быть смоделировано как парное сравнение с помощью BERT, однако для ответа на один запрос потребуется более 50 часов. Общим методом решения задач кластеризации и семантического поиска является отображение каждого предложения в векторное пространство таким образом, чтобы семантически схожие предложения были близки. Исследователи начали вводить в BERT отдельные предложения и получать вкрапления предложений фиксированного размера. Наиболее часто используемый подход заключается в усреднении выходного слоя BERT (известные как BERT embeddings) или с помощью вывода первой лексемы (лексема [CLS]). Как мы мы покажем, эта распространенная практика дает довольно плохие вкрапления предложений, часто хуже, чем усреднение GloVe embeddings (Pennington et al., 2014).

Для решения этой проблемы мы разработали систему SBERT. Архитектура сиамской сети позволяет можно получить векторы фиксированного размера для входных предложений. Используя такие меры сходства, как косинусоидальность или евклидово расстояние, можно найти семантически схожие предложения. Эти меры сходства могут быть выполнены чрезвычайно эффективно на современном оборудовании, что позволяет использовать SBERT использоваться как для поиска семантического сходства, так и а также для кластеризации. Сложность поиска поиск наиболее похожей пары предложений в коллекции из 10 000 сокращается с 65 часов при использовании BERT до вычисления вкраплений 10 000 предложений (~5 секунд с SBERT) и вычислению косинусоидального сходства (~0,01 секунды). Благодаря использованию оптимизированных индексных структур, поиск наиболее похожего вопроса на Quora можно сократить с 50 часов до нескольких миллисекунд (Johnson et al., 2017).

Мы точно настраиваем SBERT на данных NLI, что позволяет создавать вкрапления предложений, которые значительно превосходят другие современные методы вкрапления предложений.такие методы, как InferSent (Conneau et al., 2017) и Universal Sentence Encoder (Cer et al., 2018). На сайте семи задачах семантического текстового сходства (STS), SBERT достигает улучшения на 11,7 балла по сравнению с InferSent и на 5,5 балла по сравнению с Universal Sentence Encoder. В SentEval (Conneau and Kiela, 2018), наборе инструментов для оценки вкраплений предложений, мы достигли улучшения на 2,1 и 2,6 балла соответственно.

SBERT можно адаптировать под конкретную задачу. Он демонстрирует новые передовые результаты на сложном наборе данных о сходстве аргументов (Misra et al, 2016) и на наборе данных триплетов для различения предложений из разных разделов статьи Википедии (Dor et al., 2018). Статья построена следующим образом: В разделе 3 представлен SBERT, в разделе 4 проведена оценка SBERT на обычных задачах STS и на сложном корпусе Argument Facet Similarity (AFS) (Misra et al., 2016). В разделе 5 оценивается SBERT на SentEval. В разделе 6 мы проводим исследование абляции для проверки некоторых аспектов дизайна SBERT. В разделе 7 мы сравниваем вычислительную эффективность вкраплений предложений SBERT по сравнению с другими с другими современными методами встраивания предложений.

**2 Похожие работы**

Сначала мы представим BERT, а затем обсудим современные методы встраивания предложений. BERT (Devlin et al., 2018) - это предварительно обученная сеть-трансформатор (Vaswani et al., 2017), которая показывает новые современные результаты для различных задач НЛП, включая ответы на вопросы, классификацию предложений и регрессию пар предложений. Входные данные для BERT для регрессии пар предложений состоят из

двух предложений, разделенных специальной лексемой [SEP] токеном. Многоголовое внимание на 12 (базовая модель) или 24 слоев (большая модель), и выходные данные передаются в простую регрессионную функцию для получения окончательной метки. Используя эту схему, BERT установил новые передовые результаты в бенчмарке Semantic Textual Semilarity (STS) benchmark (Cer et al, 2017). RoBERTa (Liu et al., 2019) показала, что производительность BERT может быть дополнительно улучшена за счет небольшой адаптации процесса предварительного обучения. Мы также протестировали XLNet (Yang et al., 2019), но он показал в целом худшие результаты, чем BERT. в целом к худшим результатам, чем BERT. Существенным недостатком сетевой структуры BERT является то, что не вычисляются независимые вкрапления предложений, что затрудняет получение вкраплений предложений из BERT. Чтобы обойти это ограничение, исследователи пропускали отдельные предложения через BERT, а затем получали вектор фиксированного размера вектор фиксированного размера путем усреднения результатов (аналогично усреднения вкраплений слов), либо используя выход специальной лексемы CLS (например: May et al. (2019); Zhang et al. (2019); Qiao et al. (2019)). Эти два варианта также предоставляет популярное хранилище bert-as-a-service-repository3. Насколько нам известно, до сих пор не проводилась оценка того, являются ли эти методы приводят ли эти методы к полезным вкраплениям предложений.

Встраивание предложений - хорошо изученная область. с десятками предложенных методов. Пропуск мыслей (Kiros et al., 2015) обучает архитектуру кодера-декодера для предсказания окружающих предложений. InferSent (Conneau et al., 2017) использует маркированные данные Стэнфордского набора данных естественного языка (Bowman et al., 2015) и многожанрового набора данных NLI (Williams et al., 2018) для обучения сиамской сети BiLSTM с max-пулингом на выходе. Conneau et al. показали, что InferSent последовательно превосходит неконтролируемые методы, такие как SkipThought. Универсальный кодировщик предложений Encoder (Cer et al., 2018) обучает трансформаторную сеть и дополняет несамостоятельное обучение обучением на SNLI. Хилл и др. (2016) показали, что задача, на которой обучаются вкрапления предложений существенно влияет на их качество. Предыдущие работы (Conneau et al., 2017; Cer et al., 2018) показали, что наборы данных SNLI подходят для обучения вкраплений предложений. Янг и др. (2018) представили метод обучения на разговорах из Reddit используя сиамские сети DAN и сиамские сети-трансформеры, что дало хорошие результаты на STS эталонном наборе данных. .В работе Humeau et al. (2019) рассматривается вопрос времени выполнения накладные расходы на кросс-кодер из BERT и представляют метод (поликодеры) для вычисления оценки между m векторами контекста и пре вычисление вкраплений-кандидатов с помощью внимания. Эта идея работает для поиска предложения с наивысшей оценкой предложения в большой коллекции. Однако поликодеры имеют тот недостаток, что функция оценки не является симметричной, а вычислительные затраты слишком велики для таких случаев использования, как кластеризация, которая что потребует O(n2) вычислений баллов.

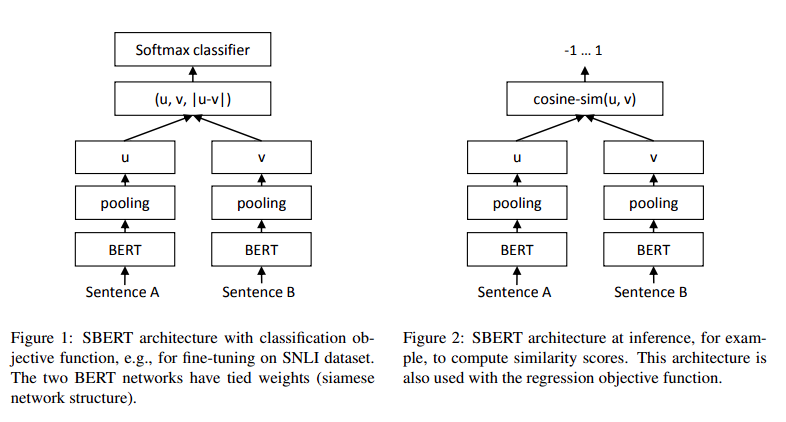


Рисунок 1. Архитектура SBERT с целевой функцией классификации, например, для тонкой настройки на наборе данных SNLI.Две BERT-сети имеют связанные веса (сиамские структура сети).

Рисунок 2. Архитектура SBERT при выводе, например, для вычисления оценок сходства. Эта архитектура также используется с объективной функцией регрессии.

Предыдущие нейронные методы встраивания предложений начинали обучение со случайной инициализации. В данной публикации мы используем предварительно обученную сеть BERT и RoBERTa и лишь тонко настраиваем ее, чтобы получить полезные вкрапления предложений. Это позволяет сократить время обучения: SBERT можно настраивается менее чем за 20 минут, а результат лучшие результаты по сравнению с аналогичными методами встраивания предложений

**3 Модель**

SBERT добавляет операцию объединения к выходу BERT / RoBERTa для получения вложения предложения фиксированного размера. Мы экспериментируем с тремя стратегиями объединения: Использование вывода CLS-токена, вычисление среднего значения всех выходных векторов (MEAN-стратегия) и вычисление максимального числа выходных векторов (MAX-стратегия). выходных векторов (MAX-стратегия). По умолчанию используется конфигурация MEAN. Для тонкой настройки BERT / RoBERTa мы создаем сиамские и триплетные сети (Schroff et al,2015) для обновления весов таким образом, чтобы полученные вкрапления предложений были семантически значимыми и могут быть сравнены с помощью косинусоидального сходства.

Структура сети зависит от имеющихся обучающие данные. Мы экспериментируем со следующими структурами и целевыми функциями. Целевая функция классификации. Мы объединяем вкрапления предложений u и v с разностью элементов |u-v| и умножаем ее с обучаемым весом Wt ∈ R 3n×k



где n - размерность вкраплений предложений, а k - количество меток. Мы оптимизируем

потери перекрестной энтропии. Эта структура изображена на Рисунок 1. Целевая функция регрессии. Вычисляется косинусоидальное сходство между двумя вкраплениями предложений

u и v (рис. 2). В качестве целевой функции мы используем среднее квадратичное отклонение. Триплетная целевая функция. Дано якорное предложение a, положительное предложение p и отрицательное предложение n, триплетная потеря настраивает сеть так, чтобы расстояние между a и p меньше, чем расстояние между a и p меньше, чем расстояние между a и n. Математически мы минимизируем следующую функцию потерь:



с sx - вложение предложения для a/n/p, || - || метрика расстояния и маржа. Маржа гарантирует. что sp по крайней мере ближе к sa, чем sn. В качестве метрики

мы используем евклидово расстояние и задаем = 1. экспериментах.

**3.1 Детали обучения**

Мы обучаем SBERT на комбинации SNLI (Bowman et al., 2015) и многожанрового NLI

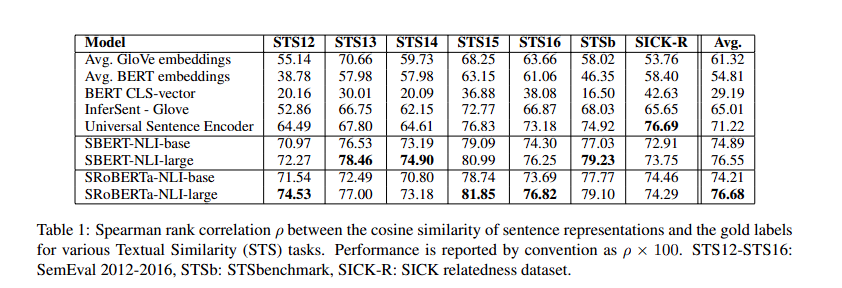


Таблица 1: Ранговая корреляция Спирмена ρ между косинусным сходством представлений предложений и золотыми метками для различных задач текстового сходства (STS). Производительность условно представлена как ρ × 100. STS12-STS16: SemEval 2012-2016, STSb: STSbenchmark, SICK-R: SICK relatedness dataset.

(Williams et al., 2018). SNLI - это коллекция из 570 000 пар предложений, аннотированных с

метками "противоречие", "недомолвка" и "нейтрально". MultiNLI содержит 430 000 пар предложений и охватывает целый ряд жанров устной и письменной речи текстов. Мы проводим тонкую настройку SBERT с помощью трехсторонней целевой функции softmax-классификатора в течение одной эпохи. Мы использовали размер партии 16, оптимизатор Адама с скоростью обучения 2e-5 и линейную скорость обучения прогрев на 10 % обучающих данных. Наша стратегия объединения по умолчанию - MEAN.

**4 Оценка - семантическое текстовое cходство**

Мы оцениваем производительность SBERT для распространенных задач семантического сходства текстов (STS). Современные методы часто обучают (сложную) регрессионной функции, которая сопоставляет вложения предложений с оценкой сходства. Однако эти регрессионные функции работают в паре и из-за комбинаторного взрыва часто не масштабируются, если коллекция предложений достигает определенного размера.

Вместо этого мы всегда используем косинус сходства для сравнения сходства между двумя вложениями предложений. Мы также провели эксперименты с отрицательными манхэттенскими и отрицательными евклидовыми расстояниями в качестве мер сходства, но результаты для всех подходов остались примерно одинаковыми.

**4.1 Неподконтрольные STS**

Мы оцениваем производительность SBERT для STS без использования каких-либо обучающих данных, специфичных для STS. Мы используем задачи STS 2012 - 2016 годов (Agirre et al., 2012, 2013, 2014, 2015, 2016), эталон STS (Ceret al., 2017), а также набор данных SICK-Relatedness

(Marelli et al., 2014). Эти наборы данных дают оценки от 0 до 5 о семантической связанности пар предложений. Мы показали в работе (Reimers et al,2016), что корреляция Пирсона плохо подходит для STS. Вместо этого мы вычисляем ранговую корреляцию Спирмена корреляцию между косинусным сходством вкраплений предложений и золотых меток. Сайт для других методов встраивания предложений эквивалентна, сходство вычисляется по косинусному сходству. Результаты представлены в таблице 1. Результаты показывают, что прямое использование выходных данных BERT приводит к довольно низким показателям. При усреднении вкраплений BERT достигается средняя корреляция всего 54,81, а при использовании результатов CLStoken - только средняя корреляция 29,19. Оба показателя хуже, чем при вычислении среднего значения вкраплений GloVe. Использование описанной структуры сиамской сети и механизма тонкой настройки существенно улучшает корреляцию, превосходя как InferSent, так и Universal Sentence Encoder. Сайт единственным набором данных, где SBERT работает хуже, чем Universal Sentence Encoder, является SICK-R. Универсальный Sentence Encoder был обучен на различных наборах данных, включая новости, страницы вопросов-ответов и дискуссионные форумы, что, по-видимому, больше подходит для данных SICK-R. В отличие от этого, SBERT был предварительно обучен только на Википедии (через BERT) и на данные NLI. В то время как RoBERTa смогла улучшить производительность для нескольких контролируемых задач, мы наблюдаем лишь незначительную разницу между SBERT и SBERT. наблюдаем лишь незначительную разницу между SBERT и SRoBERTa для генерации вкраплений предложений.

**4.2 Контролируемый STS**

Эталон STS (STSb) (Cer et al., 2017) является популярным набором данных для оценки контролируемых систем STS. STS-систем. Данные включают 8 628 пар предложений из трех категорий: подписи, новости и форумы. Они разделены на train (5 749), dev (1 500) и тестовые (1 379). BERT установил новый современный уровень на этом наборе данных, передав оба предложения в сеть и используя простую регрессию метод определения выходных данных

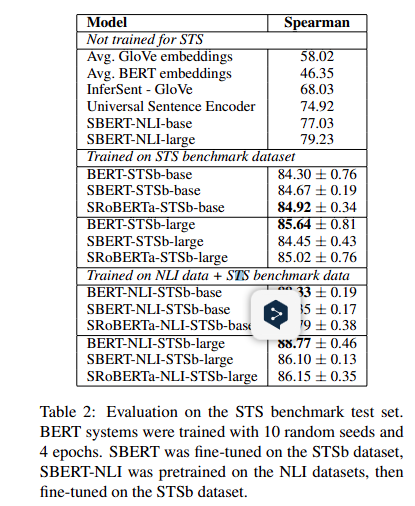


Таблица 2: Оценка на эталонном тестовом наборе STS. Системы BERT были обучены с использованием 10 случайных семян и 4 эпохи. Система SBERT прошла тонкую настройку на наборе данных STSb, SBERT-NLI был предварительно обучен на наборе данных NLI, затем тонкая настройка на наборе данных STSb.

Мы используем обучающий набор для точной настройки SBERT с помощью целевой функции регрессии. Во время предсказания мы вычисляем косинус сходства между вкраплениями предложений. Все системы обучены с 10 случайными семенами для борьбы с дисперсиями (Reimers and Gurevych, 2018). Результаты представлены в таблице 2. Мы экспериментировали с двумя установками: Только обучение на STSb, и сначала обучение на NLI, затем обучение на STSb. Мы заметили, что более поздняя стратегия приводит к незначительному улучшению на 1-2 балла. Этот двухэтапный подход оказал особенно большое влияние на кросс-кодера BERT, который улучшил производительность на 3-4 пункта. Мы не наблюдаем существенной разницы между BERT и RoBERTa.

**4.3 Сходство по фасеткам аргументов**

Мы оцениваем SBERT на корпусе Argument Facet Similarity (AFS) от Misra et al. (2016). Корпус

В корпусе AFS аннотировано 6 000 аргументов из диалогов в социальных сетях на три спорные темы: контроль над оружием, однополые браки и смертная казнь. Данные были аннотированы по шкале от 0 ("разные темы") до 5 ("полностью эквивалентные"). Понятие сходства в корпусе AFS довольно сильно отличается от понятия сходства в STS от SemEval. Данные STS обычно описательные, в то время как данные AFS представляют собой аргументированные отрывки из диалогов. Чтобы считаться схожими, аргументы должны не только содержать схожие утверждения, но и предоставлять аналогичную аргументацию. Кроме того, лексический разрыв между предложениями в AFS гораздо больше. Таким образом, как простые методы без контроля, так и как и современные системы STS, плохо работают на этом наборе данных (Reimers et al., 2019). Мы оцениваем SBERT на этом наборе данных в двух сценариях: 1) Как было предложено Мисрой и др. SBERT с помощью 10-кратной перекрестной валидации. Недостатком этой схемы оценки является то, что неясно насколько хорошо подходы обобщаются для различных тем. Следовательно, 2) мы оцениваем SBERT в кросс-тематическом настройке. Для обучения используются две темы, а подход оценивается на оставшейся теме. Мы повторяем Это повторяется для всех трех тем, и результаты усредняются. Тонкая настройка SBERT осуществляется с помощью регрессионной целевой функции. Оценка сходства вычисляется с помощью косинуса сходства на основе вкраплений предложений. Мы также приводим корреляцию Пирсона r, чтобы сделать результаты сопоставимыми с результатами Misra etal. Однако мы показали (Reimers et al., 2016). что корреляция Пирсона имеет ряд серьезных недостатков и ее следует избегать при сравнении систем STS . Результаты представлены в таблице 3.

Неподконтрольные методы, такие как tf-idf, среднее значение GloVe embeddings или InferSent, работают довольноплохо работают с этим набором данных, получая низкие оценки. Обучение SBERT при 10-кратной перекрестной проверке дает производительность почти на одном уровне с BERT. Однако при кросс-тематической оценке мы наблюдаем падение производительности SBERT примерно на 7 корреляция Спирмена. Чтобы считаться схожими, аргументы должны отвечать тем же требованиям и предоставлять одинаковое обоснование. BERT способен использовать внимание для прямого сравнения обоих предложений (например, сравнение слово за словом), в то время как SBERT должен сопоставить отдельные предложения из невидимой темы в векторное пространство так, чтобы аргументы с схожими утверждениями и причинами. Это гораздо более сложная задача, которая, по-видимому, требует не только двух тем для обучения. наравне с BERT.

**.4 Различение разделов Википедии**

Дор и др. (2018) используют Википедию для создания тематически тонкого набора обучающих, пробных и тестовых наборов для методов вкраплений предложений. Статьи Википедии разделены на отдельные разделы, сфокусированные на на определенных аспектах. Дор и др. предполагают, что сен-

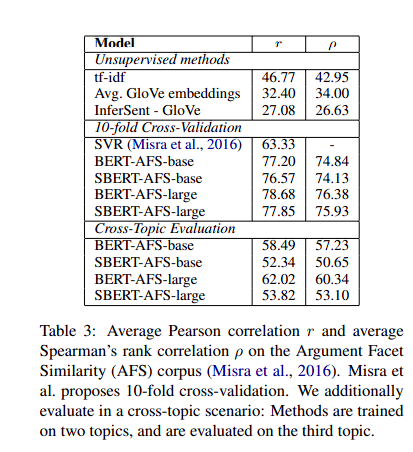


Таблица 3: Среднее значение корреляции Пирсона r и среднее значение ранговая корреляция Спирмена ρ на корпусе Argument Facet сходства (AFS) (Misra et al., 2016). Мисра и al. предлагают 10-кратную кросс-валидацию. Мы дополнительно оцениваем дополнительно в кросс-топическом сценарии: Методы обучаются на двух темах и оцениваются на третьей теме.

предложения в одном разделе тематически ближе чем предложения в разных разделах. Они используют это для создания большого набора данных слабо маркированных тройных предложений: якорь и положительный пример взяты из одного раздела, а отрицательный пример - из другого раздела той же статьи. Например, из статьи Alice Арнольд: Якорь: Арнольд присоединилась к BBC Radio Drama Company в 1988 г., позитив: Арнольд привлек внимание СМИ в мае 2012 г., отрицательные: Болдинг и Арнольд - заядлые гольфисты-любители. Мы используем набор данных из работы Dor et al. триплетную задачу, обучаем SBERT в течение одной эпохи на около 1,8 миллиона обучающих триплетов и оцениваем на 222 957 тестовых триплетах. Тестовые триплеты взяты из отдельного набора статей Википедии. В качестве оценки метрики мы используем точность: находится ли положительный пример ближе к якорю, чем отрицательный пример? Результаты представлены в таблице 4. Дор и др. настроили архитектуру BiLSTM с триплетными потерями для для получения вкраплений предложений для этого набора данных. Как видно из как видно из таблицы, SBERT явно превосходит подход BiLSTM от Dor et al.

**5 Оценка - SentEval**

SentEval (Conneau and Kiela, 2018) - это популярный

инструментарий для оценки качества вкраплений предложений. Вкрапления предложений используются в качестве признаков для классификатора логистической регрессии. Классификатор логистической регрессии обучается на различных задачах в режиме 10-кратной кросс-валидации, а предсказание точность предсказания рассчитывается для тестовой выборки.

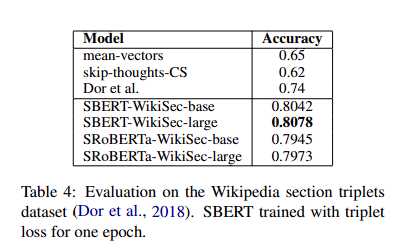


Таблица 4: Оценка на триплетах разделов Википедии (Dor et al., 2018). SBERT обучен с триплетом потери для одной эпохи.

Цель вкраплений предложений SBERT не должны использоваться для переноса обучения в другие задач. В данном случае мы считаем, что тонкая настройка BERT, как описано в Devlin et al. (2018), для новых задач является более подходящим методом, поскольку он обновляет все слои сети BERT. Однако SentEval все еще может дать представление о качестве наших вкраплений предложений вкраплений для различных задач. Мы сравниваем вкрапления предложений SBERT с другими методами вкрапления предложений на следующих семи задачах передачи SentEval:

- MR: предсказание настроения для рецензий на фильмы фрагментов по пятибалльной шкале (Pang and Lee,2005).

- CR: предсказание настроения отзывов покупателей о товарах (Hu and Liu, 2004).

- SUBJ: предсказание субъективности предложений из рецензий на фильмы и кратких описаний сюжета(Pang and Lee, 2004).

- MPQA: Классификация полярности мнений на уровне фраз из новостных лент (Wiebe et al., 2005).

- SST: Stanford Sentiment Treebank с бинарными метками (Socher et al., 2013).

- TREC: Мелкозернистая классификация типов вопросов из TREC (Li and Roth, 2002).

- MRPC: Корпус парафраз Microsoft Research из параллельных источников новостей (Dolan et al,

2004).

Результаты представлены в таблице 5. SBERT достигает наилучшей производительности в 5 из из 7 задач. Средняя производительность увеличивается примерно на 2 процентных пункта по сравнению с InferSent, а также с универсальным кодировщиком предложений. Несмотря на то, что обучение переносу не является целью SBERT, он превосходит другие современные методы вкрапления предложений в этой задаче.

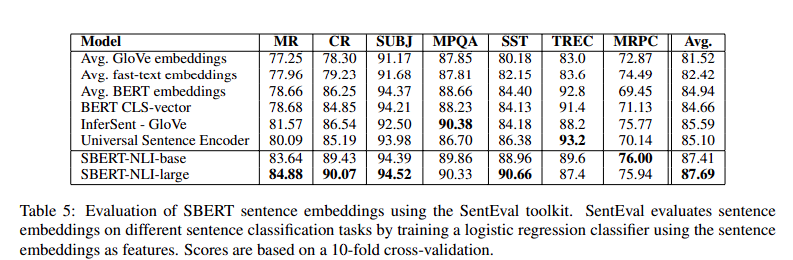


Таблица 5: Оценка вкраплений предложений SBERT с помощью инструментария SentEval. SentEval оценивает вкрапления предложений вкраплений на различных задачах классификации предложений путем обучения классификатора логистической регрессии с использованием вкраплений предложений в качестве признаков Оценки основаны на 10-кратной перекрестной валидации

Оказалось, что вкрапления предложений из SBERT хорошо отражают информацию о настроениях: Мы наблюдаем значительные улучшения для всех задач, связанных с чувством (MR, CR и SST) с помощью SentEval по сравнению с InferSent и Universal Sentence Encoder. Единственный набор данных, в котором SBERT значительно хуже, чем Universal Sentence Encoder, - это набор данных TREC. Универсальный кодировщик предложений был предварительно обучен на данных с вопросами-ответами, что, по-видимому, благоприятно сказывается на задаче классификации по типу вопросов в наборе данных TREC. Средние вкрапления BERT или использование CLStoken из сети BERT дали плохие результаты для различных задач STS (табл. 1), хуже, чем средние вкрапления GloVe. Однако для SentEval средние вкрапления BERT и выход BERT CLS-токен достигают достойных результатов (табл. 5), превосходя средние эмбеддинги GloVe. Причиной этого являются разные установки. Для задач STS мы использовали косинусоидальное сходство для оценки сходства между вкраплениями предложений. Косинус-подобие учитывает все измерения одинаково. В отличие от этого, SentEval подгоняет классификатор логистической регрессии к вкраплениям предложений. Этот допускает, что некоторые измерения могут иметь более высокие или меньшее влияние на результат классификации.

Мы пришли к выводу, что средние вкрапления BERT / CLS-токены, выведенные BERT, дают вкрапления предложений, которые невозможно использовать с косинусоидальным сходством или с расстоянием Манхэттена / евклидовым расстоянием. Для трансферного обучения они дают несколько худшие чем InferSent или Universal Sentence Encoder. Однако, используя описанную настройку с сиамской сетевой структурой на наборе данных NLI дает вкрапления предложений, которые достигают нового передового уровня для инструментария SentEval.

**6 Исследование абляции**

Мы продемонстрировали убедительные эмпирические результаты для качества вкраплений предложений SBERT. В этом разделе мы проведем исследование абляции различных аспектов SBERT, чтобы лучше понять понимания их относительной важности. Мы оценили различные стратегии объединения (MEAN, MAX и CLS). Для классификации объективной функции мы оценили различные методы конкатенации. Для каждой возможной конфигурации мы обучаем SBERT с 10 различными случайными семенами и усредняем результаты.

Целевая функция (классификация или регрессия) зависит от аннотированного набора данных. Для для функции классификации мы обучаем SBERTbase на наборах данных SNLI и Multi-NLI. Для регрессионной функции мы обучаемся на обучающем наборе эталонных данных STS. Производительность измеряется на обучающем наборе эталонного набора данных STS. Результаты представлены в Таблица 6.

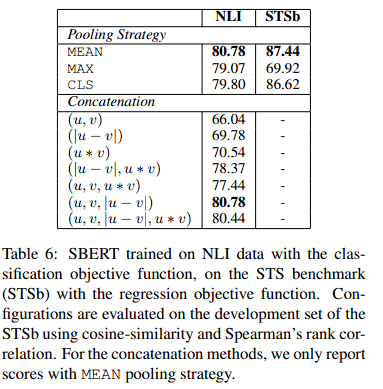


Таблица 6: SBERT, обученный на данных NLI с целевой функцией классификации, на эталоне STS (STSb) с целевой функцией регрессии. Конфигурации оцениваются на наборе разработки STSb с помощью косинусоидального сходства и ранговой корреляции Спирмена. Для методов конкатенации мы сообщаем только следующее оценки со стратегией объединения MEAN.

При обучении с целевой функцией классификации на данных NLI, стратегия объединения оказывает довольно незначительное влияние. Влияние режима конкатенации гораздо больше. InferSent (Conneau et al., 2017) и Universal Sentence Encoder (Ceret al., 2018) используют (u, v, |u - v|, u ∗ v) в качестве входных данных для классификатора softmax. Однако в нашей архитектуре добавление мудрых элементов u ∗ v снизило производительность. Наиболее важным компонентом является поэлементная разность |u - v|. Отметим, что режим конкатенации важен только для обучения классификатора softmax. При выводе, при предсказании сходства для эталонного набора данных STS, используются только вкрапления предложений u и v используются в сочетании с косинусным сходством. Поэлементная разница измеряет расстояние между измерениями двух вкраплений предложений, обеспечивая сходство пар ближе, а несходство - дальше. более далеки друг от друга. При обучении с использованием объективной функции регрессии мы видим, что стратегия объединения оказывает большое влияние. Здесь стратегия MAX работает значительно хуже, чем MEAN или стратегия CLS-токенов. Это противоречит данным (Conneau et al., 2017), которые обнаружили, что для BiLSTM-слоя выгоднее InferSent использовать MAX вместо MEAN-пулинга.

**7 Вычислительная эффективность**

Вкрапления предложений потенциально должны быть вычислены для миллионов предложений, поэтому требуется высокая скорость вычислений желательна. В этом разделе мы сравниваем SBERT со средними вкраплениями GloVe, InferSent (Conneau et al., 2017) и Universal Sentence Encoder (Cer et al., 2018). Для сравнения мы используем предложения из бенчмарка STS (Cer et al., 2017). Мы вычисляем средние вкрапления GloVe с помощью простого цикла for-loop с поиском по словарю в python и NumPy. InferSent4 основан на PyTorch. Для Universal Sentence Encoder мы используем версию TensorFlow Hub5. которая основана на TensorFlow. SBERT основан на PyTorch. Для улучшения вычисления вкраплений предложений, мы применили стратегию интеллектуального пакетирования: Предложения с одинаковой длины группируются вместе и и добавляются только к самому длинному элементу в мини-партии. Это значительно сокращает вычислительные затраты от добавления лексем. Производительность измерялась на сервере с процессор Intel i7-5820K @ 3,30 ГГц, Nvidia Tesla

Графический процессор V100, CUDA 9.2 и cuDNN. Результаты представлены в таблице 7.

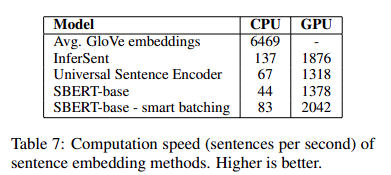


Таблица 7: Скорость вычислений (предложений в секунду) методов встраивания предложений. Выше – лучше

На центральном процессоре InferSent примерно на 65 % быстрее, чем SBERT. Это объясняется гораздо более простой архитектурой сети. В InferSent используется один слой BiLSTM, в то время как в BERT - 12 слоев стекированных трансформаторов. Однако преимуществом трансформаторных сетей является эффективность вычислений на графических процессорах. Так, SBERT с интеллектуальной сортировкой примерно на 9 % быстрее, чем InferSent, и примерно на 55 % быстрее, чем Universal Sentence Encoder. Умнаяпакетная обработка достигает ускорения 89 % на CPU и 48 % на GPU. Средние вкрапления GloVe, очевидно, с большим отрывом являются самым быстрым методом вычисления вкраплений предложений.

**8 Заключение**

Мы показали, что готовая программа BERT отображает предложения в векторное пространство, которое довольно плохо подходит для использования с обычными мерами сходства такими как косинус-сходство. Производительность для семи STS-задач оказалась ниже, чем у средних вкраплений GloVe. Чтобы преодолеть этот недостаток, мы представили Sentence-BERT (SBERT). SBERT точно настраивает BERT в архитектуре сиамской/триплетной сети. Мы оценили его качество на различных распространенных бенчмарках, где он смог добиться значительного улучшения по сравнению с современными методами встраивания предложений. Замена BERT на RoBERTa не дала значительного улучшения в наших экспериментах.

SBERT требует больших вычислительных затрат. На графическом процессоре он примерно на 9 % быстрее, чем InferSent, и примерно на 55 % чем Universal Sentence Encoder. SBERT может быть использован для задач, которые с вычислительной точки зрения нецелесообразно моделировать с помощью BERT. Например, кластеризация 10 000 предложений с помощью иерархической кластеризации требует от BERT около 65 часов, поскольку необходимо вычислить около 50 миллионов комбинаций предложений необходимо вычислить около 50 миллионов комбинаций предложений. С помощью SBERT нам удалось сократить это время примерно до 5 секунд.

**Благодарности**

Эта работа была поддержана Немецким Исследовательский фонд Германии через германо-израильский проектного сотрудничества (DIP, грант DA 1600/1-1 игрант GU 798/17-1). Она была профинансирована совместно с

Федеральное министерство образования и научных исследований Германии (BMBF) в рамках рекламных ссылок 03VP02540 (ArgumenText).