# 1

Солодянкин А.А., БИВ174, «Исследование аффинных преобразований в семантическом пространстве BERT»

# 2 Актуальность работы

В последние годы технологии машинного обучения стали неотъемлемой частью нашей жизни. Они представлены голосовыми помощниками, рекомендательными системами, умными домами, умными автомобилями и другими системами. Важной частью этих систем являются модули, которые помогают сделать понятным для компьютера то, что от него требуется. Для систем по обработке текста это модули обработки естественного языка или Natural Language Processing (NLP).

Направление обработки естественного языка активно развивается, последний большой прорыв был сделан в 2018 году командой Google AI. Была представлена новая модель обработки естественного языка под названием BERT или Bidirectional Encoder Representations from Transformers [1]. BERT продемонстрировал лучшее качество на тесте SQuAD (Stanford Question Answering Dataset) [2] версии 1.1 для вопросно-ответных систем. На рисунке Рисунок 1 представлены первые строчки таблицы лидеров для теста SQuAD 1.1 на момент выхода модели BERT.

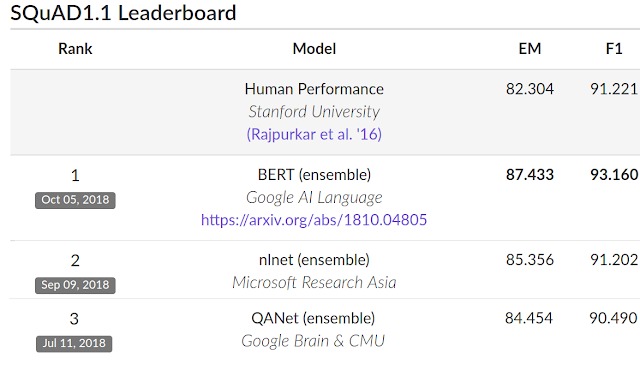


Рисунок 1 - Сравнение BERT с другими моделями на момент выхода модели

Сейчас появились модели с качеством лучше, чем у модели BERT. При этом многие модели с хорошим качеством на второй версии теста SQuAD [2] (SQuAD 2.0 [3]) используют те же принципы, что и модель BERT. На рисунке Рисунок 2 представлен топ 3 моделей на тесте SQuAD 2.0 [3] от 14.03.21.

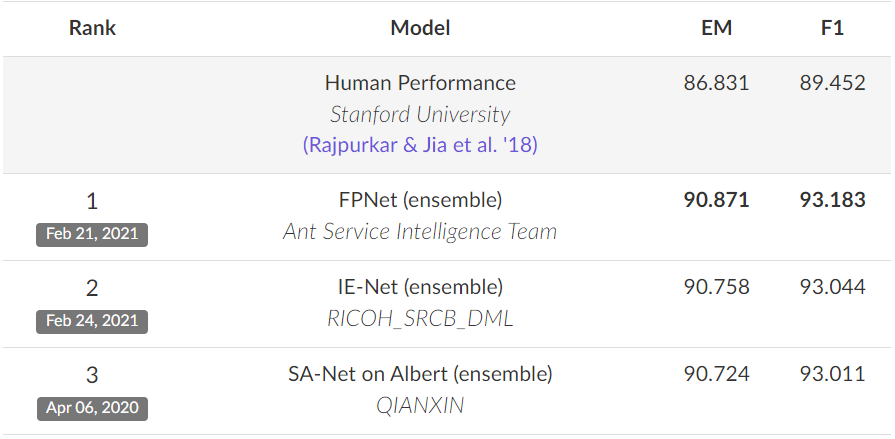


Рисунок 2 - Таблица лидеров теста SQuAD2.0 на 14.03.21

При всей популярности модели BERT остаются аспекты, которые плохо изучены или вообще не изучены, например, влияние вложений друг на друга в одном предложении. Большинство людей работают с моделью, не понимая, что происходит в модели и как модель генерирует результаты. Происходит это из-за того, что трудно интерпретировать промежуточные данные. Кроме того, в промышленной сфере есть потребность в интерпретации моделей машинного обучения.

Появляется потребность в интерпретации промежуточных данных. Промежуточными данными BERT являются эмбеддинги. Идея эмбеддингов заключается в том, что каждому слову ставится в соответствие вещественный многомерный вектор, что позволяет не просто представить текст в численном виде для работы компьютера с ним, но также применять векторные операции, выражая с их помощью различные семантические отношения между словами. В нашей работе проверяется гипотеза о возможности аффинных преобразований для эмбеддингов слов в семантическом пространстве BERT и проводится оценка качества проведенных аффинных преобразований. Это позволит приблизиться к пониманию эмбеддингов слов в модели BERT.

# 3 Обзор литературы

Для того, чтобы исследовать аффинные преобразования в семантическом пространстве BERT необходимо понимать историю развития NLP, как работают аффинные преобразования в моделях NLP, существующие методы оценки качества аффинных преобразований, как устроена модель BERT и какие исследования и эксперименты, уже проведены с моделью BERT.

## 3.1 История развития NLP

Обработка естественного языка берет свое начало в 1950-х годах. Уже в 1950 году Алан Тьюринг опубликовал статью [4] под названием «Вычислительные машины и интеллект», в которой предлагалось то, что сейчас называется тестом Тьюринга в качестве критерия интеллекта, задача, которая включает автоматическую интерпретацию и создание естественного языка, но в то время не сформулирована как проблема, отдельная от искусственного интеллекта.

C 1950 по начало 1990 NLP был символическим и опирался прежде всего на правила заранее прописанные в системе. Джорджтаунский эксперимент [5] в 1954 году включал полностью автоматический перевод более шестидесяти русских предложений на английский. Авторы утверждали, что в течение трех-пяти лет машинный перевод станет решенной проблемой. Однако реальный прогресс был намного медленнее, и после отчета ALPAC [6] в 1966 году, в котором было обнаружено, что десятилетние исследования не оправдали ожиданий, финансирование машинного перевода резко сократилось. Небольшие дальнейшие исследования в области машинного перевода проводились до конца 1980-х, когда были разработаны первые системы статистического машинного перевода.

Некоторые особенно успешные системы обработки естественного языка, разработанные в 1960-х годах, были SHRDLU [7], система естественного языка, работающая в ограниченных "блоках миров" с ограниченными словарями, и ELIZA [8], симуляция психотерапевта Роджера. Практически не используя информации о человеческих мыслях или эмоциях, ELIZA иногда обеспечивала поразительно человеческое взаимодействие. Когда «пациент» превышает очень маленькую базу знаний, ELIZA может дать общий ответ, например, ответив на «Моя голова болит» словами «Почему вы говорите, что у вас болит голова?».

Во время В 1970-х годах многие программисты начали писать «концептуальные онтологии», которые структурировали реальную информацию в понятные компьютеру данные. Примеры: MARGIE (Schank, 1975)[9], SAM (Cullingford, 1978)[10], PAM (Wilensky, 1978)[11].

1980-е и начало 1990-х годов знаменуют собой расцвет символических методов в NLP. время включало исследования синтаксического анализа на основе правил, морфологии, справочная информация и другие области понимания естественного языка. Были продолжены другие направления исследований, разработка чаттер-ботов с помощью Racter и Jabberwacky. Важным событием (которое в итоге привело к повороту в статистике в 1990-е годы) стало растущее значение количественной оценки в этот период.

До 1980-х годов большинство систем обработки естественного языка основывались на сложных наборах рукописных п правил. Однако, начиная с конца 1980-х годов, произошла революция в обработке естественного языка с появлением алгоритмов машинного обучения для языковой обработки. Это было связано как с постоянным увеличением вычислительной мощности, так и с постепенным уменьшением доминирования теорий лингвистики Хомского.

Многие из первых заметных успехов статистических методов в NLP произошли в область машинного перевода, особенно благодаря работе в IBM Research. Эти системы могли использовать преимущества существующих многоязычных текстовых корпусов, которые были разработаны Парламентом Канады и Европейским Союзом в результате принятия законов, требующих перевод всех правительственных заседаний на все официальные языки соответствующих систем правления. Однако большинство других систем зависело от корпусов, специально разработанных для задач, реализуемых этими системами, что было основным ограничением успеха этих систем. В результате большое количество исследований было направлено на методы более эффективного обучения на ограниченных объемах данных.

С ростом Интернета стало доступно все большее количество необработанных языковых данных. С середины 1990-х гг. Таким образом, исследования все больше фокусируются на алгоритмах неконтролируемого и полу-контролируемого обучения. Такие алгоритмы могут учиться на данных, которые не были аннотированы вручную с желаемыми ответами, или с использованием комбинации аннотированных и неаннотированных данных. Как правило, эта задача намного сложнее, чем контролируемое обучение, и обычно дает менее точные результаты для заданного количества входных данных. Однако доступно огромное количество неаннотированных данных, которые часто могут компенсировать худшие результаты, если используемый алгоритм достаточно низкая временная сложность для практического применения.

В 2010-х годах репрезентативное обучение и глубокая нейронная сеть Методы машинного обучения в стиле стали широко распространены в обработке естественного языка, отчасти из-за большого количества результатов, показывающих, что такие методы могут достигать самых современных результатов во многих задачах естественного языка, например, в языковом моделировании, синтаксическом анализе и многие другие.

Одной из первых и самой знаковых нейросетевых моделей NLP является word2vec. Word2vec был создан и опубликовано в 2013 году группой исследователей под руководством Томаса Миколова в Google [12–14]. Встраивание векторов, созданных с помощью алгоритма Word2vec, имеет некоторые преимущества по сравнению с более ранними алгоритмами, такими как скрытый семантический анализ.

Word2vec — это метод обработки естественного языка. Алгоритм word2vec использует модель нейронной сети для изучения словесных ассоциаций из большого корпуса текста. После обучения такая модель может обнаруживать синонимы слов или предлагать дополнительные слова для частичного предложения. Как следует из названия, word2vec представляет каждое отдельное слово с определенным списком чисел, который называется вектором или эмбеддингом. Векторы выбираются тщательно так, чтобы простая математическая функция (косинусное сходство между векторами) указывала уровень семантического сходства между словами, представленными этими векторами.

В word2vec реализованы два основных алгоритма обучения: CBoW и Skip-gram. CBoW — архитектура, которая предсказывает текущее слово, исходя из окружающего его контекста. Архитектура типа Skip-gram действует наоборот: она использует текущее слово, чтобы предугадывать окружающие его слова. Построение модели word2vec возможно с помощью двух данных алгоритмов. Порядок слов контекста не оказывает влияния на результат ни в одном из этих алгоритмов.

Получаемые на выходе векторные представления слов позволяют вычислять «семантическое расстояние» между словами. Так, можно находить похожие по значению слова. Обычно приводят пример с королём и королевой: король относится к мужчине также, как королева к женщине. Word2vec выполняет прогнозирование на основании контекстной близости этих слов. Так как инструмент word2vec основан на обучении простой нейронной сети, чтобы добиться его наиболее эффективной работы, необходимо использовать большие корпусы для его обучения. Это позволяет повысить качество предсказаний.

Далее знаковым шагом было появление модели fastText. FastText — это библиотека для изучения вложений слов и классификации текстов, созданная лабораторией AI Research (FAIR) Facebook [15]. Модель позволяет создать алгоритм неконтролируемого обучения или контролируемого обучения для получения векторных представлений слов. Facebook делает доступными предварительно обученные модели для 294 языков. FastText использует нейронную сеть для встраивания слов. Алгоритм fastText основан на содержании этих двух работ [16,17].

2018 год стал переломной точкой для развития моделей машинного обучения, направленных на решение задач обработки. Быстро растет концептуальное понимание того, как представлять слова и предложения для наиболее точного извлечения их смысловых значений и отношений между ними. Более того, NLP-сообщество продвигает невероятно мощные инструменты, которые можно бесплатно скачать и использовать в своих моделях и пайплайнах. Эту переломную точку также называют NLP’s ImageNet moment, ссылаясь на тот момент несколько лет назад, когда схожие разработки значительно ускорили развитие машинного обучения в области задач компьютерного зрения.

Одна из последних основополагающих разработок в этой сфере – это релиз BERT, событие, которое положило начало новой эре в NLP. BERT – это модель, побившая несколько рекордов по успешности решения ряда NLP-задач. Вскоре после выхода статьи, описывающей модель, команда разработчиков также выложила в открытый доступ код модели и сделала возможным скачивание различных версий BERT, которые уже были предобучены на больших наборах данных. Этот знаменательный шаг позволил любому разработчику встраивать в свои модели машинного обучения для обработки естественного языка уже готовый мощный компонент, сохраняя свои время, энергию и ресурсы, необходимые для обучения модели обработки языка с нуля.

BERT построен на целом ряде недавних разработок, предложенных NLP-сообществом, включая, но не ограничиваясь: Semi-supervised Sequence learning [18], ELMo [19], ULMFiT [20], OpenAI Transformer [21] и Трансформер[22].

## 3.2 Аффинные преобразования в моделях NLP

Аффинные преобразования для моделей NLP стали возможны с появлением модели word2vec [12–14], из-за того, что в этой модели предлагалось работать со словами как c векторами.

Для моделей word2vec и fastText задача аффинных преобразований решалась проще, так как преобразование из токена в вектор не зависело от контекста слов и для одинаковых токенов в разных контекстах получались одинаковые вектора.

Для моделей, работающих на основе трансформеров, так не работает. Эмбеддинги одинаковых токенов в разных контекстах будут отличаться, так как при создании эмбеддинга трансформеры используют еще и контекст. Отсюда появляется сложность для проведения аффинных преобразований, непонятно какой из эмбеддингов брать.

## 3.3 Существующие методы оценки качества аффинных преобразований

Если модель векторного представления слов хорошо обучена, то для задачи, чтобы найти ответ на вопрос (мужчина:король)::(женщина:?), нужно посчитать вектор, ближайший к вектору 𝑣женщина + 𝑣король – 𝑣мужчина. При условии качественно обученной модели этим вектором будет 𝑣королева.

Данный метод решения аналогии называется 3CosAdd. Как было показано во многих работах [23–28], 3CosAdd имеет ограничения. Точность решения в разных категориях аналогий сильно варьируется. При помощи этого метода нельзя решать аналогии некоторых категорий, например аналогии с синонимами или антонимами [24,25]. С одной стороны, это можно объяснить несовершенством современных моделей или неправильным выбором корпуса текстов для их обучения.

Однако в работе [25] утверждается, что проблема может крыться и во взаимном расположении векторов, например, в высокой косинусной близости некоторых пар слов и, наоборот, в большом косинусном расстоянии среди других пар, что может влиять на точность решения. Более того, метод 3CosAdd предполагает, что верный ответ – единственный, в то время как для многих задач аналогии это неверно [27], например, для задач аналогии с типом «целое-часть» подразумевается наличие более одного ответа. Вследствие всех этих ограничений встает вопрос о состоятельности и обоснованности метода. Впоследствии были предложены другие методы решения задачи аналогии. Некоторые, как и метод 3CosAdd, основываются на одном конкретном примере аналогии, где ответ на вопрос аналогии выражается исключительно через 𝑎, 𝑎∗, 𝑏 и соответствующие им вектора. Также существуют методы, в которых рассматриваются все пары слов, объединенных одним и тем же отношением. Например, при известных парах (Москва:Россия), (Париж:Франция), (Лондон:Англия) нужно восстановить аналогию (Осло:?), где правильным ответом будет являться Норвегия. Рассмотрим, каким ещё способом можно решать вопрос аналогии.

В работе [29] предлагается метод 3CosAvg, ответ на вопрос аналогии при этой метрике выражается через средний вектор разности между.

Для оценки качества преобразований необходимы датасеты, существует Google analogy test set [13]. Он используется для сравнения всех существующих на данный момент моделей векторного представления слов. Содержит 19544 пар аналогий, разделенных на 14 категорий, в 9 из которых аналогия основана на семантических связях между словами, в 5 – на морфологических связях. Датасет переведён и адаптирован для русского языка, дополнен большим количеством примеров. Многие категории исключены из рассмотрения, так как не являются актуальными для русского языка.

## 3.4 Устройство модели BERT

Точно так же, как и в обычном Трансформере, BERT принимает на вход последовательность слов, которая затем продвигается вверх по стеку энкодеров (рисунокРисунок 3). Каждый слой энкодера применяет внутреннее внимание (self-attention) и передает результаты в сеть прямого распространения, после чего направляет его следующему энкодеру.

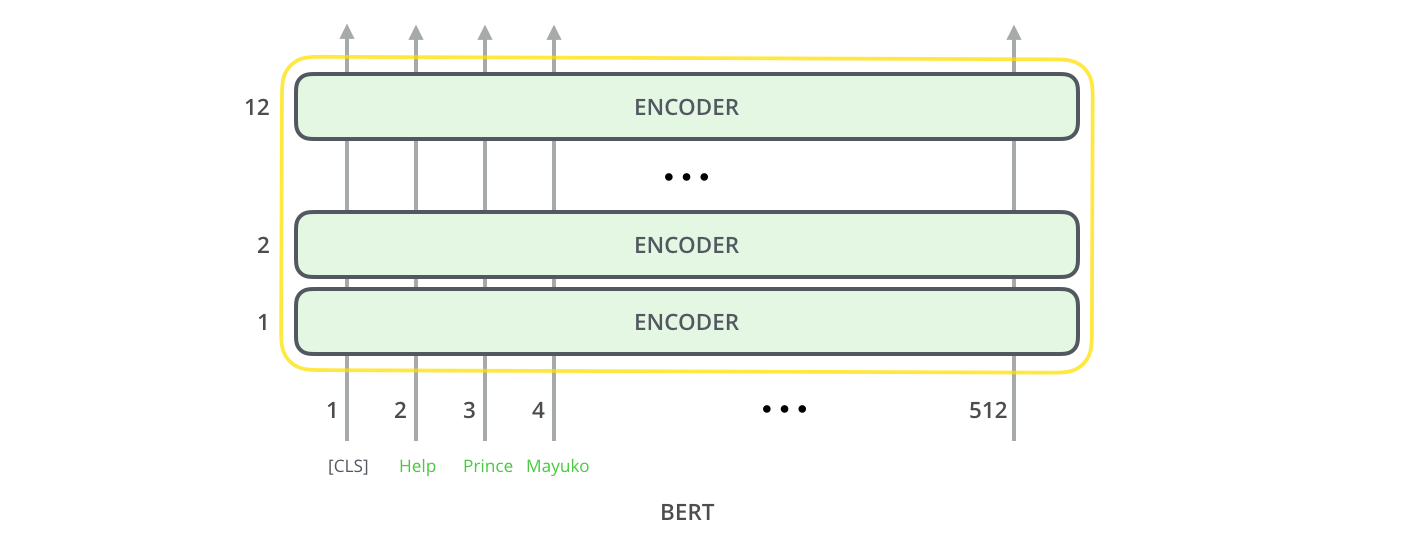


Рисунок 3 - Структура слоев модели BERT

С точки зрения архитектуры, данный процесс идентичен Трансформеру вплоть до следующего этапа (если не учитывать размер, который является настраиваемым параметром). Но относительно выходов эти две модели значительно отличаются.

## 3.5 Исследования и эксперименты, уже проведенные с моделью BERT

В работе [30], посвященной визуализации вложенных отношений, Gergely D. Németh (2019) предложил подход к визуализации полного предложения.

В центре внимания статьи-исследование вложений BERT в 2D-подпространстве. Автор исследует, как этот подход может идентифицировать различные времена и знаки препинания. Кроме того, автор анализирует способность модели BERT обрабатывать более сложные предложения и выявлять возможный ответ. Другой эксперимент был связан с распознаванием неправильных предложений.

В целом автор [30] добился хороших результатов в визуализации вложений BERT. Тем не менее, нет никакого объяснения, почему графики выглядят именно так. Более того, эта статья не содержит никаких математических формул, описаний применяемых методов или примеров кода. Трудно проверить или повторить действия автора статьи.

Исследователи из Кембриджа [31] пошли дальше в своих исследованиях и визуализации вложений BERT.

В своей работе [31], «Visualizing and measuring the geometry of bert» (2019), исследователи провели серию экспериментов, которые помогают узнать больше о внутренних представлениях лингвистической информации BERT. Они обнаружили доказательства синтаксического представления в матрицах внимания, которые состоят из определенных направлений в пространстве, представляющем определенные отношения зависимости. Исследователи показали конкретные синтаксические подпространства, представляющие семантическую информацию. Их эксперименты также дают ответ на вопрос о том, как все эти различные представления сочетаются друг с другом. Они сделали предположение, что внутренняя геометрия BERT может быть разделена на несколько линейных пространств, где отдельное подпространство содержит определенную синтаксическую или семантическую информацию.

Подводя итог, следует отметить, что исследователи из Кембриджа [31] достигли существенных результатов в исследовании семантического подпространства BERT и выдвинули интересные гипотезы о языковых подпространствах. Их предположения подкреплялись убедительными математическими методами. Кроме того, исследования геометрии представлений BERT могут помочь в понимании и совершенствовании моделей, основанных на трансформерах. Однако эта работа с недостатками, они не исследовали существование других значимых подпространств и типов лингвистической информации.

## 3.6 Выводы

Была рассмотрена история развития NLP от примитивных правил до сложных моделей генерации текстов.

Были рассмотрены аффинные преобразования в моделях NLP и существующие методы оценки переносов.

В приведенных статьях по изучению BERT все рассмотренные источники раскрывают важные особенности семантического подпространства BERT, но основное внимание в них уделяется анализу полных предложений. В своем исследовании я анализирую вложения отдельных слов и их значения, которые основаны на представлениях BERT из разных предложений. В этой статье я не буду фокусироваться на отношениях внутри предложения или между различными предложениями, моя цель – проанализировать значение слова, проверить возможность и оценить качество аффинных преобразований в отрыве от контекста.

# 4 Постановка задачи

Исходя из сделанных выводов об актуальности использования и популярности модели BERT, а также исходя из не исследованности аффинных преобразований для эмбеддингов BERT, поставлена следующая цель работы.

**Целью** данной работы является исследование и измерение качества аффинных преобразований для модели BERT.

Для достижения поставленной цели необходимо решить следующие задачи:

* Найти русскую версию модели BERT;
* Найти данные для обработки моделью;
* Токенизировать весь текст;
* Обработать полученные токены моделью и получить эмбеддинги для всех токенов;
* Усреднить эмбеддинги для каждого токена;
* Найти тесты для оценки качества аффинных преобразований модели;
* Провести аффинные преобразования;
* Посчитать качество аффинных преобразований.

Исследование будет проводиться на языке python в среде Jupyter Notebok при использовании Google Colab. Jupyter Notebok является наиболее удобной платформой для проведения исследований на python. Google Colab является бесплатной и мощной платформой для запуска кода. При этом дается 12Гб оперативной памяти, доступ к Google диску для доступа к данным, а также есть возможность запускать код с использованием GPU.

# 5 Список используемых источников

1. Devlin J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding // arXiv Prepr. arXiv1810.04805. 2018.

2. Rajpurkar P. et al. Squad: 100,000+ questions for machine comprehension of text // arXiv Prepr. arXiv1606.05250. 2016.

3. Rajpurkar P., Jia R., Liang P. Know what you don’t know: Unanswerable questions for SQuAD // arXiv Prepr. arXiv1806.03822. 2018.

4. Machinery C. Computing machinery and intelligence-AM Turing // Mind. 1950. Vol. 59, № 236. P. 433.

5. Hutchins J. The first public demonstration of machine translation: the Georgetown-IBM system, 7th January 1954 // noviembre de. 2005.

6. Committee N.R.C. (US). A.L.P.A., (Organization) A. Language and Machines: Computers in Translation and Linguistics; a Report. National Academies, 1966. Vol. 1416.

7. Fang I.E. It isn’t ETAOIN SHRDLU; it’s ETAONI RSHDLC // Journal. Mass Commun. Q. Association for Education in Journalism, etc., 1966. Vol. 43, № 4. P. 761.

8. Weizenbaum J. ELIZA—a computer program for the study of natural language communication between man and machine // Commun. ACM. ACM New York, NY, USA, 1966. Vol. 9, № 1. P. 36–45.

9. Schank R.C., Abelson R.P. Scripts, plans, and knowledge // IJCAI. 1975. Vol. 75. P. 151–157.

10. Cullingford R.E. Script application: computer understanding of newspaper stories. 1978.

11. Wilensky R. Understanding goal-based stories. 1978.

12. Mikolov T., Yih W., Zweig G. Linguistic regularities in continuous space word representations // Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies. 2013. P. 746–751.

13. Mikolov T. et al. Efficient estimation of word representations in vector space // arXiv Prepr. arXiv1301.3781. 2013.

14. Mikolov T. et al. Distributed representations of words and phrases and their compositionality // arXiv Prepr. arXiv1310.4546. 2013.

15. Joulin A. et al. Fasttext. zip: Compressing text classification models // arXiv Prepr. arXiv1612.03651. 2016.

16. Joulin A. et al. Bag of tricks for efficient text classification // arXiv Prepr. arXiv1607.01759. 2016.

17. Bojanowski P. et al. Enriching word vectors with subword information // Trans. Assoc. Comput. Linguist. MIT Press, 2017. Vol. 5. P. 135–146.

18. Dai A.M., Le Q. V. Semi-supervised sequence learning // arXiv Prepr. arXiv1511.01432. 2015.

19. Peters M.E. et al. Deep contextualized word representations // arXiv Prepr. arXiv1802.05365. 2018.

20. Howard J., Ruder S. Universal language model fine-tuning for text classification // arXiv Prepr. arXiv1801.06146. 2018.

21. Radford A. et al. Improving language understanding by generative pre-training. 2018.

22. Vaswani A. et al. Attention is all you need // arXiv Prepr. arXiv1706.03762. 2017.

23. Levy O., Goldberg Y. Linguistic regularities in sparse and explicit word representations // Proceedings of the eighteenth conference on computational natural language learning. 2014. P. 171–180.

24. Köper M., Scheible C., im Walde S.S. Multilingual reliability and “semantic” structure of continuous word spaces // Proceedings of the 11th international conference on computational semantics. 2015. P. 40–45.

25. Vylomova E. et al. Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning // arXiv Prepr. arXiv1509.01692. 2015.

26. Rogers A., Drozd A., Li B. The (too many) problems of analogical reasoning with word vectors // Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (\* SEM 2017). 2017. P. 135–148.

27. Newman-Griffis D., Lai A.M., Fosler-Lussier E. Insights into analogy completion from the biomedical domain // arXiv Prepr. arXiv1706.02241. 2017.

28. Gladkova A., Drozd A., Matsuoka S. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn’t. // Proceedings of the NAACL Student Research Workshop. 2016. P. 8–15.

29. Drozd A., Gladkova A., Matsuoka S. Word embeddings, analogies, and machine learning: Beyond king-man+ woman= queen // Proceedings of coling 2016, the 26th international conference on computational linguistics: Technical papers. 2016. P. 3519–3530.

30. Gergely D. Németh. Visualisation of embedding relations (word2vec, BERT) [Electronic resource] // https://towardsdatascience.com/visualisation-of-embedding-relations-word2vec-bert-64d695b7f36. 2019.

31. Coenen A. et al. Visualizing and measuring the geometry of bert // arXiv Prepr. arXiv1906.02715. 2019.