На правах рукописи

Малых Валентин Андреевич

Методы сравнения и построения устойчивых к шуму программных систем в задачах обработки текстов

Специальность 05.13.11 —
«Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей»

Автореферат
диссертации на соискание учёной степени
кандидата технических наук

Москва — 2019

Работа выполнена в Институте системного анализа Федерального исследовательского центра “Информатика и управление” Российской академии наук.

Научный руководитель: доктор технических наук, профессор, член-корреспондент Российской академии наук
Арлазаров Владимир Львович

Официальные оппоненты: **Тулупьев Александр Львович,**
доктор физико-математических наук, доцент,
Лаборатория теоретических и междисциплинарных проблем информатики, Федеральное государственное бюджетное учреждение науки Санкт-Петербургский институт информатики и автоматизации Российской академии наук,
заведующий лабораторией

Иванов Владимир Владимирович,
кандидат физико-математических наук,
Автономная некоммерческая организация высшего образования “Университет Иннополис”,
старший научный сотрудник

Ведущая организация: Федеральное государственное бюджетное образовательное учреждение высшего образования “Московский государственный университет им. М.В. Ломоносова”

Защита состоится 23 мая 2019 г. в 16 часов на заседании диссертационного совета Д 002.087.01 при Институт системного программирования им. В.П. Иванникова Российской академии наук по адресу: 109004, г. Москва,ул. А. Солженицына, дом 25.

С диссертацией можно ознакомиться в библиотеке и на сайте Федерального государственного бюджетного учреждения науки “Институт системного программирования им. В.П. Иванникова Российской академии наук”.

Автореферат разослан _____.

Ученый секретарь
диссертационного совета
Д 002.087.01,
к-т физ.-мат. наук Зеленов Сергей Вадимович

Общая характеристика работы

Актуальность темы. В последнее время в мире наблюдается быстрый рост накопления знаний, так называемый “информационный взрыв” . Поток генерируемой информации при этом имеет существенно другой характер, нежели наблюдаемый ранее, а именно, большая часть этого потока содержит шумы разного рода. Например, в случае новостных документов, новостные документы от информационных агентств проходят корректуру и содержат в себе минимальное количество опечаток. Но в настоящее время большая часть новостных документов поступает не от информационных агентств, а от обычных людей, что стало возможно с появлением сети интернет. Тексты из интернета часто содержат опечатки, уровень шума в них составляет 10-15%. Другой пример может быть приведен в распознавании документов, идентифицирующих личность. В случае использования специализированного оборудования точность распознавания может стремиться к идеальной, но существует большое количество документов, изображения которых были получены на бытовые фотокамеры в условиях плохого освещения и несоблюдения условий съемки.

В существующих системах для решения прикладных задач проблема устойчивости к шуму упускается из виду в виду того, что а) существует проблема открытого словаря - слов с опечатками на много порядков больше, чем словарных слов; б) предполагается, что системы проверки орфографии убирают опечатки из текста. Существующие системы для решения прикладных задач зачастую опираются на векторные представления слов. Существующие распространенные системы векторного представления слов обладают либо полностью ограниченным (закрытым) словарем, как система Word2Vec, либо частично открытым, как система FastText. В работах автора представлены методы построения систем для решения прикладных задач, обладающие открытым словарем и устойчивостью к шуму.

Еще одной проблемой является то, что ранее не было предложено методов сравнения качества систем по их устойчивости к шуму в рассматриваемых в этой диссертационной работе прикладных задачах. Существующие аналоги применяются для систем проверки орфографии и не рассчитаны на варьирование уровня шума, что не позволяет оценить устойчивость систем в разных условиях. В работах автора предложен метод сравнения систем в условиях разного уровня шума применительно к различным прикладным задачам.

Для решения проблемы шумности текстов существуют системы проверки орфографии, широко использующиеся в настоящее время. Но современные системы коррекции орфографии все еще могут ошибаться во многих случаях. Например, для русского языка точность современных систем проверки орфографии в терминах F-меры составляет ниже 85%.

Ошибки, допущенные в словах, приводят к ухудшению качества в различных задачах обработки естественного языка. Например, в работе [0] показано, что даже применение промышленных систем проверки орфографии для компенсации шума не дает преимущества перед системой, которая изначально устойчива к шуму. Так как не все опечатки могут быть исправлены или исправлены корректно (как показано, например, в работе Кучержана и Брилла существует некоторое количество исправлений, порядка 1%, некорректных относительно пользовательского намерения, но грамматически правильных), автором разработан альтернативный подход заложения в систему, выполняющую какую-либо задачу устойчивость к шуму, то есть создать систему, не полагающуюся на качество систем коррекции шума.

В настоящей диссертационной работе рассмотрены задачи сравнения качества систем векторных представлений слов, классификации текстов, распознавания именованных сущностей и извлечения аспектов, а также методы построения устойчивых систем для означенных задач.

При разработке программных систем обработки текстов в частности решается задача построения систем *векторных представлений слов*. Системы векторных представлений слов в частности решают задачу моделирования языка. Моделирование языка - это создание модели, которая может предсказать следующее слово, на основании окружающих. Задача *классификации текстов* является классической задачей классификации, где объектом выступает текст, а признаками - входящие в него слова. Задача *распознавания именованных сущностей* - это извлечение или разметка во входящем тексте последовательностей токенов, которые именуют сущности, например, людей или организации.

Извлечение аспектов - это извлечение из входного текста описаний свойств некоторой сущности. Например в предложении “У этого телефона громкий динамик.” сущностью является “этот телефон”, а аспектом - “динамик”. То есть модель извлечения аспектов должна представить на выходе заключение, что в этом предложении содержится аспект “динамик”, так как, как правило, в задаче извлечения аспектов предполагается, что сущность фиксирована.

Актуальность данной работы состоит в том, что несмотря на то, что методы построения устойчивых программных систем предлагались и ранее, но не было разработано методов сравнения качества программных систем для данных задач, который позволяет выбрать лучший метод построения устойчивых к шуму систем.

Степень разработанности темы. Задачи по обнаружению и компенсации влияния шума решали такие зарубежные исследователи, как Ф. Дамерау (F.J. Damerau), П. Норвиг (P. Norvig), С. Кучержан (S. Cucerzan), Э. Карлсон (A. Carlson), Г. Кондрак (G. Kondrak), А. Винчьярелли (A.

Vinciarelli), а также отечественные исследователи, такие как В. Д. Соловьев, А. Байтин, И. Галинская, С. Татевосян, Н. Брызгалова, А. Сорокин.

Задачи устойчивости к шуму были освещены в работах отечественных ученых А. С. Титовой, В.В. Окатьева, и зарубежных исследователей - Т. Кёна (T. Cohn), Т. Болдуина (T. Baldwin), Ц. Ниу (J. Niu), Й. Блинка (Y. Blink).

Объектом данного исследования являются программные системы векторного представления слов, классификаторов текстов, извлечения именованных сущностей и извлечения аспектов, а **предметом** данного исследования является устойчивость к шуму вышеперечисленных программных систем.

Целью данной работы является разработка методов сравнения программных систем по их устойчивости к шуму в разных задачах, а именно в задачах получения векторных представлений слов, классификации текстов, распознавания именованных сущностей и выделения аспектов, а также разработка методов построения программных систем, устойчивых к шуму.

Для достижения поставленных целей необходимо было решить следующие **задачи**:

1. Исследовать устойчивость к шуму существующих программных систем векторных представлений слов, классификации текстов, распознавания именованных сущностей и извлечения аспектов.
2. Разработать методы сравнения программных систем векторных представлений слов, классификации текстов, распознавания именованных сущностей и извлечения аспектов по их устойчивости к шуму.
3. Разработать методы построения программных систем векторных представлений слов, классификации текстов и извлечения аспектов, более устойчивых к шуму, чем существующие аналоги,
4. Реализовать разработанные методы в комплексах программ и получить сравнение устойчивости программных систем к шуму.

Научная новизна:

1. Разработаны новые методы сравнения качества программных систем относительно их устойчивости к шуму для задач векторных представлений слов, классификации текстов, распознавания именованных сущностей и извлечения аспектов. Существующие аналоги разработанных методов применяются для оценки качества систем проверки орфографии и не предназначены для других задач. Также существенным отличием является наличие возможности регулирования уровня шума в разработанных методах.
2. Разработаны новые методы построения программных систем устойчивых к шуму векторных представлений слов, классификации текстов и извлечения аспектов. Разработанные методы

применены в описанных задачах и показали во многих экспериментах лучшие результаты.

3. Создан, апробирован и внедрен программный комплекс, реализующий разработанные методы.

Практическая значимость работы заключается в разработанных программных комплексах, реализующих:

– сравнение качества программных систем по устойчивости к шуму;
– построение устойчивых к шуму векторных представлений слов;
– построение устойчивых к шуму методов классификации текстов, распознавания именованных сущностей, извлечения аспектов.

Методы сравнения систем по их устойчивости к шуму могут быть легко адаптированы для широкого круга задач обработки текстов, помимо рассмотренных задач классификации, распознавания именованных сущностей и извлечения аспектов, это могут быть задачи распознавания текстов, распознавания речи, машинного перевода и другие. Разработанный программный комплекс векторного представления слов также может быть применен в широком круге задач обработки текстов.

Методология и методы исследования. Сущность методологии настоящего исследования состоит в формулировании гипотезы о недостаточной устойчивости существующих программных систем для рассматриваемых задач, а также в описании существующих программных и их особенностей, важных для исследуемого аспекта устойчивости к шуму. Описанные особенности программных систем используются в дальнейшем для постановки серий численных экспериментов, что характерно для научного поиска в области информатики в целом. В работе использованы методы теории алгоритмов, теории вероятностей и теории машинного обучения, а именно разделов связанных с теорией нейронных сетей и тематического моделирования.

Основные положения, выносимые на защиту:

1. Разработаны новые методы сравнения качества программных систем относительно их устойчивости к шуму для задач обработки текстов. Шум в виде опечаток встречается во многих существующих текстах. Методы разработаны для задач сравнения программных систем векторного представления слов, классификации текстов, распознавания именованных сущностей и извлечения аспектов на различных языках. Существующие аналоги данного метода не применялись к рассматриваемым задачам.
2. Разработаны новые методы построения устойчивых к шуму программных систем, решающих следующие задачи: построение векторных представлений слов, классификации текстов и извлечения аспектов. Эти задачи часто решаются на текстах, содержащих естественный шум. Разработанный метод в задаче векторного представления слов позволяет построить системы

более устойчивые к шуму на большинстве исследованных приложений векторных представлений, а именно на задачах распознавания парафраз, распознавания логического следования и анализа тональности для русского, английского и турецкого языков. В задаче классификации текстов разработанный метод позволяет создавать программные системы, более устойчивые к шуму, чем существующие аналоги, для русского и английского языков. В задаче извлечения аспектов разработанный метод позволяет создавать системы, более устойчивые к шуму, чем существующие системы на основе нейросетевого и графического подходов к построению таких систем. Разработанные программные комплексы выложены в открытый доступ.

Достоверность Все полученные результаты подтверждаются экспериментами, проведенными в соответствии с общепринятыми стандартами.

Диссертационное исследование соответствует п. 10 "Оценка качества, стандартизация и сопровождение программных систем" паспорта специальности 05.13.11.

Апробация работы. Основные результаты работы докладывались на следующих конференциях:

- 13-я международная конференция о концептуальных решетках и их приложениях (CLA 2016) (18-22 июля 2016 г., г. Москва);
- 5-я международная конференция "Искусственный интеллект и естественный язык" (AINL FRUCT 2016) (10-12 ноября 2016 г., г. Санкт-Петербург);
- 6-я международная конференция по анализу изображений, социальных сетей и текстов (АИСТ 2017), (27-29 июля, г. Москва);
- 13-я международная конференция северо-американского отделения Ассоциации по компьютерной лингвистике (NAACL 2018, без публикации) (1-6 июня 2018 г., г. Новый Орлеан, США);
- 56-я международная конференция Ассоциации по компьютерной лингвистике (ACL 2018) (15-20 июля 2018 г., г. Мельбурн, Австралия).
- Конференция по эмпирическим методам в обработке естественного языка (EMNLP 2018) (31 октября - 4 ноября 2018 г., Брюссель, Бельгия);
- Международная конференция по искусственному интеллекту: приложения и инновации (IC-AIAI-2018) (31 октября - 2 ноября 2018 г., г. Никосия, Кипр);
- Открытая конференция ИСП РАН им. В.П. Иванникова (2018 Ivannikov ISPRAS Open Conference) (22-23 ноября 2018 г., г. Москва)

Публикации. Основные результаты по теме диссертации изложены в 11 печатных изданиях, 7 из которых издано в журналах, входящих в списки ВАК, 6 из которых опубликовано в изданиях, индексируемых Scopus, 4 — в трудах конференций.

Работа [5] опубликована в журнале, включённом в перечень рекомендованных изданий ВАК. Работы [1, 2, 3, 4, 7, 9] опубликованы в изданиях, индексируемых в Scopus, при этом работы [1, 2] опубликованы в журнале, включенном в перечень ВАК рецензируемых изданий, входящих в международные реферативные базы. Работа [8] опубликована в издании, индексируемом РИНЦ.

В работе [9] все результаты принадлежат автору. В остальных работах, также все результаты принадлежат автору, однако, в работе [7] Озерину А.В. принадлежат иллюстрации и частично постановка задачи; в работах [1, 3, 5, 8] Лялину В.А. принадлежат описания моделей и часть иллюстраций; в работе [2] Хахулину Т.А. принадлежат описания моделей, Логачевой В.К. вступление и часть иллюстраций.

Личный вклад автора. Все представленные в диссертации результаты получены лично автором.

Диссертационная работа была выполнена при поддержке Фонда поддержки проектов Национальной технологической инициативы и ПАО "Сбербанк". Идентификатор проекта 0000000007417F630002.

Объем и структура работы. Диссертация состоит из введения, шести глав, заключения и библиографии. Полный объем диссертации **145** страниц текста с **38** рисунками и **11** таблицами. Список литературы содержит **112** наименований.

Содержание работы

Во **введении** обосновывается актуальность исследований, проводимых в рамках данной диссертационной работы, приводится обзор научной литературы по изучаемой проблеме, формулируется цель, ставятся задачи работы, излагается научная новизна и практическая значимость представляемой работы.

Первая глава посвящена обзору работ по построению программных систем, устойчивых к шуму, для различных задач. Описаны существующие подходы к моделированию шума для построения программных систем для различных задач. Описаны общие методы, применяющиеся в обработке естественного языка с акцентом на применении в задачах, где требуется устойчивость к шуму. Существующие программные системы, как правило, не предполагают устойчивости к шуму, вместо этого используется проверка орфографии, что в некоторых случаях оказывается недостаточным.

Вторая глава работы посвящена описанию разработанного автором метода построения программных систем устойчивого к шуму в приложении к задаче векторного представления слов. Также в данной главе описывается разработанный автором метод сравнения существующих и перспективных программных комплексов на предмет устойчивости к шуму.

Метод построения систем, устойчивых к шуму, в задаче векторного представления слов

Система векторного представления слов состоит из модуля побуквенного представления слов, модуля обработки контекста и процедуры обучения. Предлагаемый метод состоит в том, что:

- модифицируется модуль побуквенного представления слов таким образом, чтобы он был устойчив в опечаткам,
- модифицируется модуль обработки контекста для учета контекста слова,
- задается процедура обучения модуля контекстного представления на обучающей выборке.

Модуль обработки контекста обеспечивает сохранение семантических свойств при использовании модуля буквенного представления.

Для описания модуля буквенного представления автором вводится специальное *побуквенное представление слова* BME.

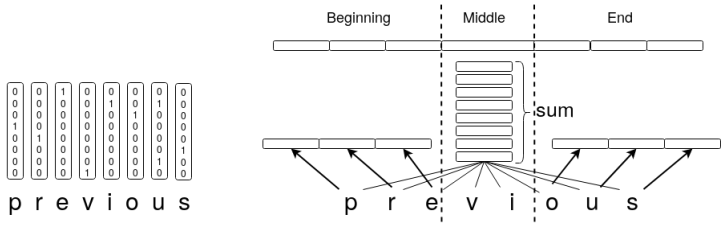

Рис. 1 — Генерация вектора для слова *previous*. Слева: генерация векторов one-hot кодирования букв, справа: генерация представления BME.

Каждый символ слова представляется, как one-hot вектор (вектор нулей размером с алфавит с одной 1 в позиции i, где i-индекс символа). Тогда имеются три вектора: начало (**B**), средний (**M**), и конечный (**E**) вектор, которые конкатенируются вместе. Вектор **M** - это сумма one-hot векторов всех символов слова. **B** - представляет собой объединение в one-hot вектора n_b первых букв слова. Аналогично, **E** компонент представляет собой объединение в один one-hot вектор n_e последних символов слова. n_b и n_e являются гиперпараметрами, которые могут различаться для разных наборов данных. Входное представление слова формируется путем объединения **B**, **M** и **E** векторов. Поэтому его длина равна $(n_b + n_e + 1) \times V$,

где V - алфавит языка. Этот входное представление далее обрабатывается нейронной сетью, описанной ниже. Формирование входного вектора показано на рис. 1.

Пусть слово w состоит из набора букв. Обозначим за $c_1..c_k$ one-hot представления этих букв. Тогда

$$B(w) = c_1\|..\|c_{n_b} \tag{1}$$

$$E(w) = c_{k-n_e}\|..\|c_k \tag{2}$$

$$M(w) = \sum_{1}^{k} c_i \tag{3}$$

$$BME(w) = B(w)\|M(w)\|E(w) \tag{4}$$

где $\|$ обозначает операцию конкатенации.

Выход из модуля побуквенного представления BME подается на модуль обработки контекста (см. рис. 2), который кодирует контекст в векторное представление с помощью нейронной сети.

$$RoVe(w) = enc(BME(w); C_{left}, C_{right})$$

где C_{left} и C_{right} обозначают левый и правый контексты для слова w соответственно.

Программная система называемая в дальнейшем RoVe создает контекстно-зависимые представления слов. Это означает, что она не генерирует фиксированный вектор для слова и должна производить его с нуля для каждого слова. Эта структура незначительно увеличивает время обработки текста, но дает более точные представления с учетом контекста.

Для того, чтобы создать представление слова в рамках предлагаемого метода, нужно кодировать его вместе с его контекстом. Для каждого слова контекста сначала создается его входное векторные представление. Это представление затем передается модулю обработки контекста (верхняя часть на рис. 2), который обрабатывает все слова контекста.

Модуль обработки контекста должен быть некоторой программой, например, нейронной сетью, которая может обрабатывать список слов и хранить информацию об их контекстах. После обработки всего контекста получается векторное представление для целевого слова. Данное векторное представление получается путем передачи скрытого состояние нейронной сети, соответствующее нужному слову, через полносвязный слой. В силу этого существует возможность генерировать векторные представления для всех слов в контексте одновременно.

Таким образом, вектора, генерируемые программной системой, всегда зависят от контекста. Контекст может быть полным предложением или окном вокруг слова. данная программная система не предназначена для получения векторных представлений слов изолированно. Однако, такая

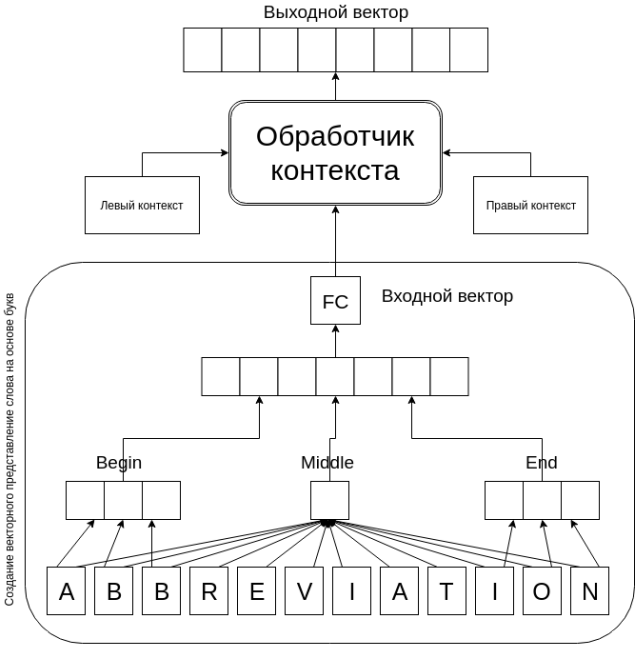

Рис. 2 — Программная система RoVe: генерация векторного представления слова *argument*.

задача возникает не часто, в силу того что на практике часто обрабатываются слова, имеющие определенный контекст.

Метод сравнения систем на предмет устойчивости к шуму в задаче векторного представления слов

Метод сравнения программных систем на предмет устойчивости к шуму состоит из следующих этапов:

– этап контролируемого внесения шума во входные данные,

– этапа проверки качества тестируемой системы по целевой метрике.

Для того, чтобы проверить качество некоторой системы при заданном уровне шума нужно внести во входные данные для тестирования системы шум заданного уровня и проверить качество вывода системы относительно известной разметки для тестовых данных. Данную процедуру нужно провести для каждой системы и каждого уровня шума. Каждый такого рода эксперимент нужно повторить некоторое количество раз для статистической достоверности получаемых результатов. В описываемой главе каждый эксперимент был повторен 10 раз, полученные результаты имеют достоверность $p < 0.05$.

В рамках предлагаемого метода для сравнения качества программных систем векторных представлений слов по их устойчивости к шуму были произведены эксперименты на 6 различных корпусах и 3 различных языках: для английского языка это были корпуса Microsoft Research Paraphrase Corpus (задача нахождения парафраз), Stanford Sentiment Treebank (задача анализа тональности) и Stanford Natural Language Inference (задача определения логической связи, textual entailment), для русского языка это корпуса Parapharaser.ru (задача нахождения парафраз) и Russian Twitter Sentiment (задача анализа тональности), для турецкого языка это корпус Turkish Paraphrase Corpus (задача нахождения парафраз). В качестве целевой метрики использовалась метрика ROC-AUC для классификатора, построенного на основе векторных представлений, получаемых для слов рассматриваемыми системами.

В качестве иллюстрации полученных результатов будут приведены результаты для части корпусов.

Stanford Natural Language Inference - набор данных, состоящий из 570.152 пар предложений, извлечённых из подписей к фотографиям на сервисе Flickr, размеченных 5 людьми каждая. Этот корпус также состоит из трех классов, один из которых был удален из данных, для упрощения методов сравнения. Оставшиеся классы: противоречие (когда предложение 2 противоречит по смыслу предложению 1) и логическое следование. В этом корпусе все классы примерно равны по мощности. Результаты на этом корпусе представлены в таблице 1.

noise (%)	0	10	20	30
Базовые системы				
Word2Vec	0.624	0.593	0.574	**0.557**
fastText	0.642	0.563	0.517	0.480
fastText+speller	0.642	0.498	0.481	0.482
RoVe				
biSRU	**0.651**	**0.621**	**0.598**	0.536

Таблица 1 — Результаты для задачи определения логической связи (для английского языка) на корпусе Stanford Natural Language Inference.

Корпус Paraphraser содержит новостные заголовки на русском языке от различных информационных агентств, которые предполагаются (с помощью автоматической системы оценивания) близкими по смысловому значению. Кроме того, все они протестированы, чтобы быть близкими во время создания. Корпус содержит около 6000 пар предложений, которые размечены как -1 - не парафраз, 0 - слабый парафраз и 1 - сильный парафраз. Для оценки были взяты только классы -1 и 1, т. е. не парафраз и сильный парафраз. Таких пар в корпусе 4470. Результаты на этом корпусе представлены в таблице 2.

noise (%)	0	10	20	30
Базовые системы				
Word2Vec	0.800	0.546	0.535	0.647
fastText	0.813	0.645	0.574	0.632
fastText + spell-checker	0.813	0.693	0.525	0.490
RoVe				
stackedLSTM	0.723	0.703	**0.674**	0.601
SRU	0.823	0.716	0.601	0.647
biSRU	**0.841**	**0.741**	0.641	**0.718**

Таблица 2 — Результаты для задачи определения парафраз на корпусе Paraphraser для русского языка.

Проведенные эксперименты показали, что разработанная программная система более устойчива к опечаткам, чем программные системы Word2Vec и fastText, обычно используемые для получения векторных представлений слов. Качество данных систем существенно падает при добавлении шума даже небольшого уровня. Кроме того было проведено исследование в результате которого было показано, что программная система RoVe более устойчива к опечаткам, чем программная система fastText вместе с системой для исправления ошибок в текстах (по всей видимости, из-за ошибок, которые допускает система проверки орфографии).

В **третьей главе** представлены метод сравнения программных систем и метод построения устойчивых программных систем к задаче классификации текстов на примере задачи анализа тональности (sentiment analysis). Задача анализа тональности в настоящий момент актуальна прежде всего для текстов, написанных не профессиональными писателями или журналистами, а обычными людьми в социальных сетях. В связи с этим проблема устойчивости к шуму в этой задаче весьма актуальна.

Метод построения систем, устойчивых к шуму, в задаче классификации текстов

Система классификации текстов может быть представлена, как совокупность модулей векторного представления слов и принятия решения о присвоении метки класса и процедуры обучения программной системы. *Метод построения устойчивых к шуму программных систем* для данной задачи состоит из следующих этапов:

1. модификации модуля векторного представления слов,
2. модификации процедуры обучения программной системы.

Экспериментально было произведено сравнение качества следующих программных систем:

FastText. Текст представлен как последовательность векторов из программной системы FastText, выступающей в качестве модуля векторного представления слов. Последовательность векторов подается на вход программе представляющей собой нейронную сеть, содержащую слой GRU.

Далее применяется процедура дропаут к последнему скрытому состоянию слоя GRU, а полученный вектор проецируется в 2-мерное пространство. Данная программа рассматривается, как модуль приятия решения данной системы.

CharCNN-WordRNN. Данная программная система повторяет программную систему, описанную в работе Кима [0]. Каждое слово представляется в виде последовательности символов, а текст представляется как последовательность представлений слов, что рассматривается в качестве модуля векторного представления слов. Векторные представления поступают в программный модуль нейронной сети, содержащий слои GRU, слой, реализующий процедуру дропаут и слой проецирования - аналогично предыдущему разделу. Данный программный модуль рассматривается, как модуль приятия решения данной системы.

RoVe. Данная программная система аналогична системе FastText, за исключением того, что используются векторные преставления, получаемые из программной системы RoVe вместо векторов, получаемых из программной системы fastText.

Метод сравнения систем на предмет устойчивости к шуму в задаче классификации текстов

Метод сравнения программных систем на предмет устойчивости к шуму в задаче классификации текстов имеет следующий вид. Следует провести три вида экспериментов, с наложением и отсутствием наложения шума на входные данные тестируемых систем с разными уровнями шума.

В рамках предлагаемого метода сравнения качества программных систем классификации текстов были проведены три типа экспериментов на различных способах обработки обучающей и тестовой выборок.

Эксперимент 1. Для проведения эксперимента на исходных данных данные в тестовой и обучающей выборках берутся без изменения.

Эксперимент 2:

– тестовая и обучающая выборки обрабатываются системой проверки орфографии;
– к тестовой и обучающей выборкам добавляется искусственный шум.

Эксперимент 3:

– обучающая выборка обрабатывается системой проверки орфографии;
– к обучающей выборке добавляется искусственный шум;
– тестовая выборка остается без изменений.

Используемые для экспериментов наборы данных:

Набор данных SentiRuEval-2015 состоит двух доменов т.н. “ресторанного” и “автомобильного”. Автомобильный домен содержит 207 и 205

Система	SentiRuEval-2015	Airline Twitter Sentiment
CharCNN	0.40	0.77
FastTextGRU	0.45	0.76
CharCNN-WordRNN	0.39	0.81
RoVe	0.38	0.78

Таблица 3 — Результаты эксперимента 1. F_1 на тестовой выборке.

сообщений в обучающей и тестовой выборках соответственно. Ресторанный домен содержит 207 и 205 сообщений в обучающей и тестовой выборках соответственно.

Набор данных Airline Twitter Sentiment состоит из 14485 твитов, описывающих впечатления пользователей от пользования услугами определенных авиакомпаний. Набор данных является публично доступным.

В таблице 3 представлены результаты на исходных данных.

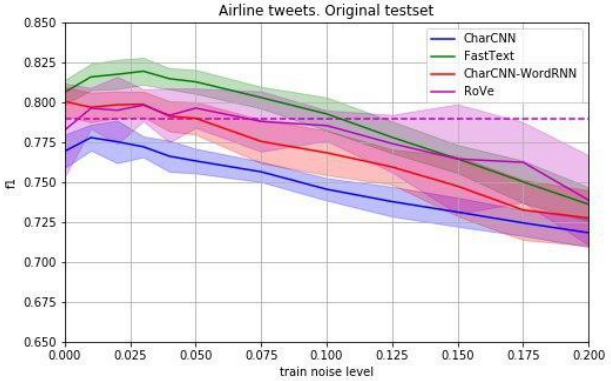

Рис. 3 — Набор данных Airline Twitter Sentiment. Результаты эксперимента 2.

В проведенных экспериментах программная система FastText показывает лучшее качество на низких уровнях шума. На высоких уровнях шума преимущество переходит к программной системе RoVe, которая теряет качество с ростом уровня шума медленнее.

Можно отметить, что программная система FastText показывает лучшее качество на низких уровнях шума, но теряет его достаточно быстро и в на высоких уровнях шума проигрывает программной системе RoVe в обоих экспериментах с изменением уровня шума. Также стоит отметить, что в эксперименте с неизмененной тестовой выборкой программная система RoVe показывает стабильное качество на уровнях шума $[0, 0.1]$, в то время как остальные модели оказываются существенно более уязвимыми для шума.

Четвертая глава настоящей работы посвящена описанию разработанного метода сравнения систем устойчивых к шуму в задаче распознавания именованных сущностей. Задача распознавание сущностей становится все более и более актуальна для текстов из социальных сетей, которым присущ некоторый уровень шума. Для этой задачи на трех языках была протестирована архитектура программных систем, показавшая самые высокие результаты на сегодняшний день для английского, русского и французского языков - BiLSTM-CRF. Для французского языка описываемая архитектура была применена автором впервые для создания программной системы извлечения именованных сущностей.

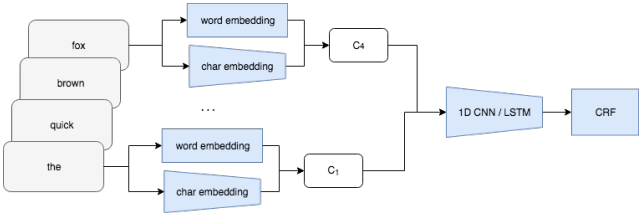

Рис. 4 — Базовая архитектура программных систем BiLSTM-CRF.

Были рассмотрены следующие варианты построения программных систем, комбинирующие векторные представления для слов и для символов:

- Векторные представления слов:
  - Word2Vec - инициализация векторами из программной системы Word2Vec для матрицы векторных представлений слов;
  - FastText - аналогично для программной системы fastText;
  - EmbedMatrix - матрица векторных представлений выучивается в процессе обучения;
  - RandomEmbed - матрица векторных представлений задается случайно и не изменяется в процессе обучения.
- Буквенные представления символов:
  - nochar - без добавления признаков от побуквенного представления слова;
  - CNN - сверточная сеть на уровне символов.

Метод сравнения систем на предмет устойчивости к шуму в задаче распознавания именованных сущностей

Метод сравнения программных систем в данной главе строится аналогично предыдущей главе.

В экспериментах были использованы наборы данных для русского, английского и французского языков.

Модель	CoNLL'03	Persons-1000	CAp'2017
EmbedMatrix+CNN	0.81	0.85	0.43
EmdebMatrix-nochar	0.80	0.81	0.44
RandomEmbed+CNN	0.69	0.77	0.31
RandomEmbed-nochar	0.61	0.48	0.22
FastText+CNN	0.86	0.69	0.41
FastText-nochar	0.86	0.69	0.41
Word2Vec+CNN	0.73	0.72	н/д
Word2Vec-nochar	0.72	0.72	н/д

Таблица 4 — Результаты экспериментов с исходными наборами данных. Метрика F_1 на тестовой выборке.

На рис. 5 представлены результаты для русскоязычного корпуса с исправленными опечатками.

Лучшая устойчивость к шуму в проведенных экспериментах была продемонстрирована программно системой EmbedMatric-CNN, чьими характерными особенностями являются выучивание матрицы векторных представлений слов, а также выучивание признаков для векторных представлений слов, основанных на сочетаниях символов.

Пятая глава описывает разработанные метод сравнения программных систем выделения аспектов на предмет их устойчивости к шуму и метод построения таких систем. Выделение аспектов также актуально для текстов из социальных сетей. В этой главе была исследована лучшая на сегодняшний день программная система извлечения аспектов ABAE и ее модификации.

Целиком программная система представляет собой в своей основе автокодировщик, то есть систему, которая восстанавливает на выходе то, что было подано ей на вход с помощью промежуточного (сжимающего) представления. Главной ее особенностью является функция потерь, так называемая функция реконструкции между векторным представлением текста, основанным на векторных представлениях слов (выхода компоненты векторного представления слов) и линейной комбинацией векторных представлений аспектов.

Метод построения систем, устойчивых к шуму, в задаче извлечения аспектов

Система извлечения аспектов может быть представлена как совокупность модулей векторного представления слов и сопоставления аспектов. *Метод создания программных систем* состоит в модификации модуля векторного представления слов таким образом, чтобы он был устойчив к опечаткам. Модуль векторного представления слов может состоять из двух компонент - компоненты векторного представления слов на основе символов и компоненты представления слов, как целого.

Метод сравнения систем на предмет устойчивости к шуму в задаче извлечения аспектов

Метод сравнения систем на предмет устойчивости к шуму в данной главе строится аналогично второй главе.

Для измерения сравнения качества программной системы ABAE и предлагаемых расширений используется корпус Citysearch отзывов на рестораны Нью-Йорка, в котором были размечены категории аспектов. Для проверки берется подкорпус тех отзывов, в которых выделена только одна категория. Далее решается задача классификации, но без учителя. Выделенные программной системой аспекты размечаются по категориям и проверяется их соответствие известным категориям. Для оценки качества используется метрика F_1.

Для программной системы ABAE были разработаны расширения в виде использования различных устойчивых к шуму векторных представлений слов. Результаты тестирования разработанных расширений представлены на рис. 6. Все разработанные расширения обладают большей устойчивостью к шуму, нежели оригинальная система. При этом модифицированная программная система с использование векторных представлений слов RoVe показала лучшие результаты и лучшую устойчивость к шуму.

В **шестой главе** выведены оценки алгоритмической сложности для рассматриваемых в главах 2, 3, 4, и 5 программных систем. Получены оценки для лучших по качеству программных систем. Для главы 2 - это система biSRU, алгоритмическая сложность данной программной системы составляет

$$O(6k \times d + 20k).$$

Для главы 3 - это программная система RoVe, сложность системы составляет

$$O(l \times (n_b + n_e + 1 + 17k + 9k^2)).$$

Для главы 4 - это программная система fasttext-CharCNN, сложность системы составляет

$$O(l \times (d^2 + 2d) + d + 2l + k + a \times k + 1).$$

Для главы 5 - это программная система RoVe, сложность системы составляет

$$O(\frac{L}{L_w} \times (n_b + n_e + 1 + 3k \times d + 10k)).$$

Также отмечена тенденция к тому, что программные системы, демонстрирующие лучшую устойчивость к шуму обладают большей алгоритмической сложностью.

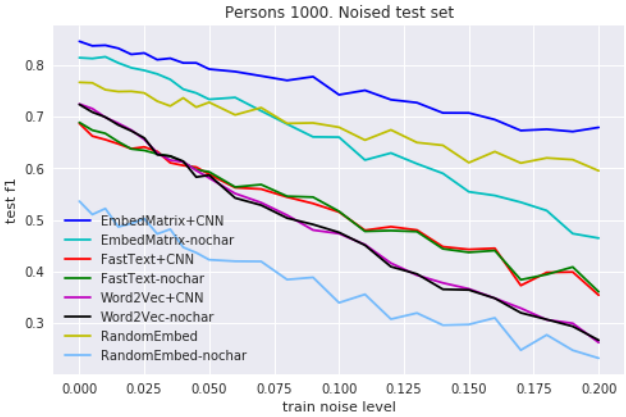

Рис. 5 — Набор данных Persons-1000. Результаты эксперимента 2.

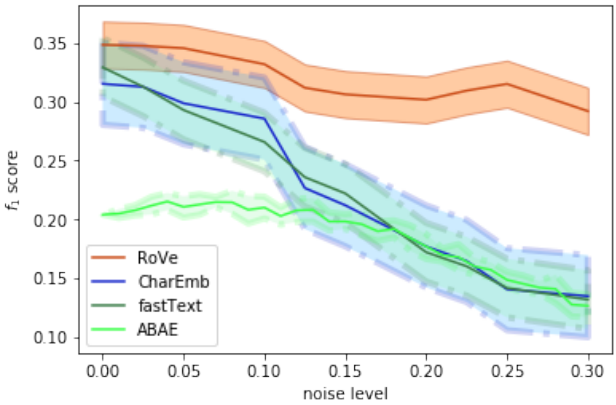

Рис. 6 — Качество по метрике F_1 для оригинальной программной системы ABAE и предлагаемых расширений.

В **заключении** в краткой форме излагаются итоги выполненного диссертационного исследования, представляются отличия диссертационной работы от ранее выполненных родственных работ других авторов, даются рекомендации по использованию полученных результатов и рассматриваются перспективы дальнейшего развития темы.

Публикации автора по теме диссертации

В изданиях из списков ВАК РФ

1. *Malykh, V.* Improving Classification Robustness for Noisy Texts with Robust Word Vectors / V. Malykh, V. Lyalin // Записки научных семинаров ПОМИ. Серия “искусственный интеллект”. — 2019.
2. *Malykh, V.* Robust to Noise Context-Aware Word Vectors / V. Malykh, V. Logacheva, T. Khakhulin // Записки научных семинаров ПОМИ. Серия “искусственный интеллект”. — 2019.
3. *Malykh, V.* Named Entity Recognition in Noisy Domains / V. Malykh, V. Lyalin // The Proceedings of the 2018 International Conference on Artificial Intelligence: Applications and Innovations. — 2018.
4. *Malykh, V.* Noise Robustness in Aspect Extraction Task / V. Malykh, T. Khakhulin // The Proceedings of the 2018 Ivannikov ISP RAS Open Conference. — 2018.
5. *Малых, В.* К вопросу о классификации шумных текстов / В. Малых, В. Лялин // Труды ИСА РАН. Специальный выпуск. — 2018.
6. *Malykh, V.* Generalizable Architecture for Robust Word Vectors Tested by Noisy Paraphrases / V. Malykh // Supplementary Proceedings of the Sixth International Conference on Analysis of Images, Social Networks and Texts (AIST 2017). — 2017. — С. 111—121.
7. *Malykh, V.* Reproducing Russian NER Baseline Quality without Additional Data. / V. Malykh, A. Ozerin // CDUD at CLA. — 2016. — С. 54—59.

В сборниках трудов конференций

8. *Malykh, V.* What Did You Say? On Classification of Noisy Texts / V. Malykh, V. Lyalin // XX Международная научно-техническая конференция “Нейроинформатика-2018’: Сборник научных трудов. В 2-х частях. Ч. 1. — М. : НИЯУ МИФИ, 2018.
9. *Malykh, V.* Robust word vectors for Russian language / V. Malykh // Proceedings of Artificial Intelligence and Natural Language AINL FRUCT 2016 Conference, Saint-Petersburg, Russia. — 2016. — С. 10—12.

10. DeepPavlov: Open-Source Library for Dialogue Systems / M. Burtsev [и др.] // Proceedings of ACL 2018, System Demonstrations. — 2018. — С. 122—127.

11. *Malykh, V.* Robust Word Vectors: Context-Informed Embeddings for Noisy Texts / V. Malykh, V. Logacheva, T. Khakhulin // Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text. — 2018. — С. 54—63.

Малых Валентин Андреевич

Методы сравнения и построения устойчивых к шуму программных систем в задачах обработки текстов

Автореф. дис. на соискание ученой степени канд. тех. наук

Подписано в печать ___.___._____. Заказ № _____
Формат 60×90/16. Усл. печ. л. 1. Тираж 100 экз.
Типография _____