МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ имени М.В.ЛОМОНОСОВА

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ и КИБЕРНЕТИКИ

КАФЕДРА СИСТЕМНОГО ПРОГРАММИРОВАНИЯ

Курсовая работа

Построение риторических деревьев текста на основе машинного обучения в рамках задачи автоматического реферирования.

Исполнитель:

Варламов Максим Игоревич

группа 427

Научный руководитель:

Майоров Владимир Дмитриевич

Москва, 2013

Содержание

[Введение 4](#_Toc363579012)

[1 Постановка задачи 7](#_Toc363579013)

[2 Обзор существующих решений 8](#_Toc363579014)

[3 Исследование и построение решения задачи 19](#_Toc363579015)

[4 Практическая часть 31](#_Toc363579016)

[Заключение 40](#_Toc363579017)

[Литература 41](#_Toc363579018)

Аннотация

Данная курсовая работа исследует основанные на машинном обучении подходы к построению риторических деревьев текста в рамках Теории Риторической Структуры (Rhetorical Structure Theory). В работе описывается реализация такого подхода в контексте модификации существующей системы автоматического реферирования, использующей риторическую структуру текста для построения его реферата. Алгоритм построения RST-деревьев, используемый в этой системе, является сугубо эвристическим и не использует машинное обучение. Работа оценивает эффективность реализованного подхода по сравнению с исходной эвристикой как на уровне точности риторического анализа текстов, так и на уровне качества рефератов, полученных на его основе.

Введение

Задачей реферирования является составление краткого изложения материала одного или нескольких информационных источников. В условиях постоянного роста количества свободно доступной информации человеку все труднее ориентироваться в ней, поэтому реферативные техники, позволяющие автоматически выделять главное и отбрасывать несущественное, приобретают особое значение. Новостные сводки, аннотации научных статей, сниппеты веб-страниц, дайджесты журналов, обзоры книг, трейлеры фильмов, репортажи спортивных событий – примеры рефератов, с которыми мы встречаемся каждый день.

Данная работа посвящена автоматическому реферированию текстовых документов. В этом аспекте задачу реферирования можно переформулировать как составление текста-реферата по одному или нескольким исходным документам, вкратце отражающему их основные положения. В зависимости от того, для каких целей составляется реферат, на какого читателя рассчитан, какую часть исходных документов должен покрывать, а также от числа текстов-первоисточников, выделяется несколько смежных подзадач автоматического реферирования текстов. Конкретно в данной работе мы рассмотрим составление рефератов наиболее общего назначения по единственному документу-источнику.

Большинство методов автоматического реферирования направлены на выделение в тексте важных фрагментов и перенос их в реферат в практически не измененном виде. Для этого обычно строится некоторая модель исходного текста. Первые и простейшие техники реферирования, такие как [1, 2], использовали частоты употребления отдельных слов и авторскую разметку текста. Другие методы ([3, 4, 5]) используют более сложные модели, например, представление текста в виде графа, где две вершины-предложения связываются дугой, если предложения содержат общие слова. Однако ни те, ни другие модели не претендуют на представление целостной структуры текста в том виде, в котором ее представляет человек. В рамках задачи автоматического реферирования построение таких моделей позволило бы не просто более точно определять наиболее важные фрагменты текста, но и составлять более связные, качественные рефераты. Стремление приблизиться к этому рубежу обосновало появление различных теорий организации структуры дискурса и их использование для моделирования текста. Было предложено несколько различных теорий ([6, 7, 8, 9]), однако наибольшей популярностью пользуется Rhetorical Structure Theory (RST, [10, 11, 12]).

RST определяет набор риторических отношений для описания связей и зависимостей между фрагментами текста. Термин “риторический” здесь близок по смыслу к термину “функциональный”: использование определенных отношений определяется целью, с которой автор пишет текст. Большинство отношений бинарные и асимметричные – определяют связь между двумя фрагментами текста, один из которых (спутник) является зависимым от другого (ядра). Многоядерные, или симметричные, риторические отношения задают связи между двумя и более равнозначными элементами. Поскольку аргументы отношений – непересекающиеся, обычно смежные фрагменты текста – сами могут иметь определенную риторическую структуру, по сути аргументом одного риторического отношения может являться другое риторическое отношение. Данный факт позволяет описывать строение текста в виде иерархической структуры – RST-дерева. Листьям такого дерева соответствуют элементарные, риторически неделимые единицы текста, в роли которых обычно выступают клаузы.

В рамках исследовательского проекта ИСП РАН была разработана система автоматического реферирования текстовых документов, использующая для выделения значимых фрагментов текста его риторическую структуру. Данная система строит RST-дерево подаваемого ей на вход текста с помощью набора эвристик, основанных на синтаксическом анализе предложений. Структура дерева используется для назначения весов его листьям-клаузам, и из наиболее весомых клауз составляется реферат.

Тестирование данной системы выявило относительно низкое качество получаемых с ее помощью рефератов. В качестве тестовых данных использовался набор текстов на английском языке с вручную сделанными аннотациями на основе материалов конференции DUC [17] 2001-го года. Для относительной оценки качества были реализованы два простейших алгоритма автоматического реферирования, выделяющие в качестве реферата соответственно первый абзац исходного текста и набор из первых предложений каждого абзаца. Сгенерированные системой и baseline-алгоритмами рефераты сравнивались с экспертными аннотациями с помощью метрик ROUGE [18]. Результаты приведены в Таблице 1.

Тем не менее, работы [13, 14, 15, 16] показывают, что риторическая структура документов может быть эффективно использована для получения качественных рефератов. Применимость используемого в рассматриваемой системе алгоритма взвешивания обоснована в [19]. Логично предположить, что причиной неудовлетворительных результатов системы ИСП РАН является низкая корреляция RST-представлений текстов, выделяемых системой, с их реальной риторической структурой. Улучшение механизма построения RST-деревьев может быть достигнуто за счет значительного расширения и усложнения набора используемых эвристик. Однако более интересную альтернативу предлагают работы на основе машинного обучения ([20, 26, 27, 28, 29]), показывающие, что определение риторических отношений в тексте на основе автоматически выведенных правил может быть эффективнее вручную реализованных эвристик.

Таблица 1. Результаты предварительного тестирования системы автоматического реферирования ИСП РАН на данных DUC-2001 в сравнении с baseline-алгоритмами

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Система ИСП РАН | Реферат - первый абзац текста | Реферат состоит из первых предложений каждого абзаца |
| ROUGE-1 | Average recall | 0.38252 | 0.20768 | **0.39090** |
| Average precision | 0.43041 | **0.58912** | 0.48388 |
| Average F-measure | 0.40461 | 0.29748 | **0.42928** |
| ROUGE-2 | Average recall | 0.13678 | 0.09795 | **0.17627** |
| Average precision | 0.15380 | **0.29344** | 0.21822 |
| Average F-measure | 0.14463 | 0.14292 | **0.19359** |
| ROUGE-L | Average recall | 0.33035 | 0.19771 | **0.36961** |
| Average precision | 0.37180 | **0.56068** | 0.45733 |
| Average F-measure | 0.34946 | 0.28317 | **0.40585** |

1 Постановка задачи

Целью данной работы является разработка и реализация алгоритма построения риторического дерева текста на основе машинного обучения.

Программная реализация разрабатываемого метода должна быть встроена в рассмотренную выше систему ИСП РАН вместо используемого на текущий момент эвристического алгоритма либо в комбинации с ним.

Разрабатываемый алгоритм принимает на вход исходный текст после синтаксического анализа составляющих его предложений. Алгоритм должен использовать данные, полученные в результате синтаксического анализа, для обучения и собственно построения RST-деревьев. Результатом работы алгоритма является древовидное представление риторической структуры текста в формате, определяемом исходной системой.

Разрабатываемый алгоритм должен обеспечивать лучшее качество выводимых с его помощью риторических структур по сравнению с используемой на текущий момент эвристикой. Качество RST-деревьев следует оценивать следующим образом:

* Через сравнение их структуры с риторическими деревьями, построенными вручную (внутренняя оценка). Для этого можно использовать метрики PARSEVAL [30].
* Через сравнение получаемых с их помощью рефератов с модельными аннотациями, построенными вручную, посредством уже упоминавшихся метрик ROUGE (внешняя оценка).

Улучшение качественных характеристик построения системой RST-деревьев после ее модификации с помощью разрабатываемого алгоритма не должно сопровождаться существенным увеличением времени ее работы. Текущая версия системы строит RST-дерево текста за линейное время от его размера в словах. Разрабатываемый алгоритм должен обеспечить сохранение линейного характера этой зависимости.

# 2 Обзор существующих решений

В процессе построения RST-дерева текста можно выделить два последовательных этапа. Первый из них – сегментация – представляет собой процесс определения границ элементарных сегментов (также их называют элементарными дискурсивными единицами, ЭДЕ), которые будут служить листовыми узлами риторического дерева. Второй этап заключается в построении иерархической структуры над набором элементарных сегментов.

В роли элементарных сегментов зачастую выступают клаузы [32], так что первый этап построения RST-дерева может быть переформулирован как процесс выделения в тексте границ клаузы. Использование средств синтаксического анализа предложений позволяет добиться высокой точности в данной задаче, как показывают исследования [20, 23]. Простая эвристика, используемая в модифицируемой нами системе, основана на достаточно компетентном синтаксическом анализе предложений (используется парсер ABBYY Compreno) и в модификации не нуждается. Ввиду вышесказанного, в дальнейшем мы сосредоточимся на втором этапе процедуры построения RST-деревьев, предполагая анализируемый текст уже корректно разбитым на элементарные сегменты. Соответственно, в обзоре методов построения RST-деревьев сегментации также будет уделено существенно меньшее внимание.

В оставшейся части главы сначала будут кратко рассмотрены известные нам эвристические методы построения RST-деревьев, не использующие машинное обучение, а затем будет дан более детальный обзор техник, так или иначе его использующих. В конце главы будут сделаны выводы о применимости рассмотренных методов к использованию в модифицируемой системе.

**2.1 Эвристические подходы**

Первые методы автоматического построения риторических деревьев, не накладывающие ограничений на вид подаваемых им на вход текстов, основывались на использовании вручную составленных наборов правил.

**Marcu, 1997**

Данная работа [21] может считаться одной из первых в области. В ее основе лежит определение допустимой риторической структуры текста, позволяющее для заданного набора ограничений построить все допустимые RST-деревья текста и выбрать из них наиболее правдоподобную.

Допустимая риторическая структура текста по Marcu должна удовлетворять следующим свойствам:

* ЭДЕ являются непересекающимися фрагментами текста.
* Риторические отношения имеют место между смежными сегментами различных размеров.
* Отношения могут быть разделены на два класса: симметричные и асимметричные; симметричные отношения связывает одинаково значимые для авторских целей сегменты, асимметричные имеют место между сегментом-ядром, имеющим значение с авторской позиции, и сегментом-спутником, используемым для улучшения понимания читателем содержимого сегмента-ядра и самостоятельно не имеющим функциональной ценности.
* В большинстве случаев риторическая структура текста может быть представлена в виде двоичного дерева.
* Если некоторое риторическое отношение имеет место между двумя сегментами в составе RST-дерева текста, то это отношение имеет место между наиболее значимыми ЭДЕ в составе рассматриваемых сегментов. Набор наиболее значимых ЭДЕ для заданного сегмента называется promotion set и определяется рекурсивно как объединение наиболее значимых ЭДЕ его непосредственных потомков-ядер в RST-дереве; promotion set элементарного сегмента состоит из него самого. Это правило Marcu называет критерием строгой композиции (strong compositionality criterion).

Последнее свойство является здесь ключевым: оно позволяет генерировать все допустимые риторические деревья заданного текста по набору ограничений на риторические отношения между составляющими его элементарными сегментами. Соответствующий алгоритм описан в [33]. С каждым допустимым RST-деревом текста связывался некоторый эвристический весовой коэффициент, после чего выбиралось наиболее весомое дерево.

Для формулирования гипотез о возможных риторических отношениях между элементарными сегментами текста Marcu использовал сигнальные/ключевые слова и фразы (cue words) – специальные конструкции в тексте, обычно союзы и вводные слова, используемые для связывания аргументов риторических отношений и тем самым являющиеся сильными индикаторами их наличия. С помощью набора правил на основе регулярных выражений, полуавтоматически составленного в результате анализа большого корпуса примеров использования сигнальных фраз, Marcu определял границы связываемых таким образом сегментов и предполагаемые типы риторических отношений между ними.

**Corston-Oliver, 1998**

В целом данный подход [22] аналогичен использованному в предыдущей работе. Corston-Oliver при этом использовал более сложные наборы правил для каждого отношения, использующие как сигнальные фразы, так и другие синтаксические критерии. Часть правил, связанных с конкретным отношением, определяла набор необходимых условий для проведения этого отношения. Оставшиеся эвристики взвешивались для обеспечения возможности выбора между несколькими риторическими отношениями, для которых выполнены необходимые условия. Весовые коэффициенты первоначально определялись интуитивно и корректировались затем методом проб и ошибок. Автор также использовал механизм бэктрекинга [34] для сокращения пространства поиска оптимального RST-дерева.

**LeThanh, 2004**

LeThanh, Abeysinghe и Huyck в работе [23] представили систему построения качественных RST-деревьев на уровне всего текста без использования машинного обучения, задавшую стандарт качества среди полнотекстовых риторических анализаторов вплоть до появления работы duVerle (ниже).

Построение RST-дерева текста происходит здесь в два этапа. На первом этапе выявляется риторическая структура отдельных предложений, на втором происходит поиск наилучшего способа объединения полученных поддеревьев для получения полной RST-иерархии текста. Выделение элементарных дискурсивных единиц в составе предложения осуществляется с помощью набора правил, основанных на синтаксической структуре предложения и присутствии сигнальных фраз. Также, помимо местоположения границ ЭДЕ, эти правила позволяют выявить, какие именно группы ЭДЕ разделяет каждая граница, и инициировать риторическое отношение между ними. Тип и направления отношения определяются на основе различных факторов, включающих синтаксис, сигнальные фразы (в том числе специальные существительные и глаголы), а также средства семантического согласования (например, синонимия). Далее простая эвристика используется для объединения полученного набора отношений в RST-дерево предложения.

Получив риторические представления для каждого предложения, система начинает поиск оптимальной структуры для всего текста. На каждом шаге система оперирует некоторым набором уже построенных поддеревьев, соответствующих непересекающимся сегментам текста. Система далее определяет все возможные способы их однократного соединения, руководствуюсь следующими двумя принципами: связываемые сегменты должны быть соседними в тексте и находиться в рамках одной и той же минимальной объемлющей конструкции авторской организации текста (абзаца, параграфа и т.д.). На основе информации о возможных соединениях определяется набор риторических отношений, которые можно построить на данном шаге. Каждое возможное отношение получает оценку, равную суммарному весу голосующих за него эвристик. В качестве последних используются те же признаки, что и для построении RST-деревьев предложений, с эмпирически определенными весами. К оценке отношения также добавляется коэффициент, который тем больше, чем ниже уровень организации текста, в рамках которого происходит соединение сегментов. Используя оценки для возможных отношений, система выбирает одно из них для реализации и соответствующим образом обновляет набор риторических деревьев. Новый набор получает вес, равный сумме веса предыдущего набора деревьев и оценки использованного риторического отношения. Задача алгоритма – начиная с набора RST-деревьев предложений, имеющего нулевой вес, вывести требуемое количество покрывающих текст риторических иерархий (не обязательно одну) с наибольшим весом. Для оптимизации процесса перебора используется лучевой поиск и промежуточное сохранение выводимых отношений.

Авторы осуществили многоуровневую оценку качества работы системы на 20 документах из RST Discourse Treebank [35]. Использовались наборы из 14 и 22 риторических отношений. Авторы заявляют о F-мере 53% для отношений в рамках предложений (60% для определения направлений отношений, т.е. распознавания ядер и сателлитов) и 39,9% (47,1%) в рамках текста. Конкретный метод вычисления F-мер из текста статьи не ясен. Авторы сравнивали качество своей системы на уровне предложений с анализатором SPADE (ниже), на уровне всего текста – с системой Marcu (1997). В обоих случаях рассматриваемая система показала лучшие результаты.

**2.2 Подходы с использованием машинного обучения**

**Marcu, 1999**

Фактически, первая попытка использовать машинное обучение в риторическом анализе.

В работе [24] Marcu предложил использовать для построения риторических деревьев shift-reduce алгоритм, использующий для выбора следующего действия механизм деревьев решений. Набор действий состоял из операции SHIFT, помещающей следующий элементарный сегмент на вершину стека, и операций REDUCE, объединяющих два верхних RST-поддерева в стеке в одно. Для каждого типа риторического отношения и для каждого возможного распределения ролей (ядро-спутник) вводилась своя операция REDUCE. Итого, на каждом шаге алгоритм выбирал из 103 действий (1 SHIFT и 102 REDUCE).

Marcu использовал алгоритм C4.5 для построения решающего дерева, классифицирующего shift-reduce действия, на основе следующих классов признаков:

1. Структурные признаки
   1. Признаки, отражающие число RST-деревьев в стеке и число оставшихся элементарных сегментов на входе.
   2. Признаки, описывающие структуру первых трех деревьев в стеке в терминах типа охватываемых ими текстовых единиц (предложений, абзацев, заголовков), а также их риторическую структуру на верхнем уровне (число непосредственных потомков корневого узла, классы соответствующих отношений).
2. Лексические и синтаксические признаки (для первых трех деревьев в стеке и первого элементарного сегмента на входе)
   1. Собственно слова и POS-теги первых и последних двух лексем охватываемых сегментов.
   2. Признаки, отражающие наличие в охватываемых сегментах потенциальных сигнальных (ключевых) фраз и их расположение (в начале, в середине, в конце).
3. История парсера: 5 последних действий
4. Признаки на основе мер семантической близости: для первых трех деревьев в стеке и первого элементарного сегмента на входе семантическая близость охватываемых сегментов (как bag-of-words, составленных из слов элементарных сегментов, входящих в promotion set дерева) на основе косинусной меры, а также метрик Wordnet, оценивающих степень синонимии и т.д.

В бинарном представлении всего 2789 признаков на обучающий/тестовый пример.

Для обучения и тестирования Marcu использовал вручную размеченный корпус из 90 текстов (нет в свободном доступе). Впервые для оценки качества парсера применялись замеченные точность и полнота - метрики, оценивающее сходство RST-деревьев, выводимых парсером, с деревьями, построенными экспертами. Размеченная точность (полнота) – отношение числа корректно размеченных риторическими отношениями составляющих в RST-дереве, построенном парсером, к числу всех составляющих в этом (экспертном) дереве. Парсер достигал 70% точности и 80% полноты для направлений отношений и 60% точности и полноты для типов отношений. Экспертные верхние границы составляли здесь 77 и 63 % соответственно.

**Marcu, Echihabi, 2002**

Marcu и Echihabi в работе [25] предложили использовать наивный байесовский классификатор для выявления таких групп риторических отношений, как CONTRAST, CAUSE-EXPLANATION-EVIDENCE, CONDITION и ELABORATION, между предложениями и клаузами.

Данные для обучения и тестирования генерировались автоматически посредством извлечения из большого текстового корпуса – порядка 41,000,000 предложений - пар смежных предложений или клауз, содержащих характерные для одной из групп отношений союзы или вводные слова. К примеру, если предложение начиналось со слова *but,* оно извлекалось вместе с непосредственно предшествующим предложением, и пара помечалась отношением CONTRAST. Таким путем авторы получили в среднем по 2,000,000 примеров отношений каждой группы. В качестве отрицательных примеров авторы случайным образом извлекали из текстов корпуса пары не являющихся соседними предложений (NO-RELATION-SAME-TEXT), а также пары предложений из разных текстов (NO-RELATION-DIFFERENT-TEXT), по 1,000,000 на каждый класс.

Для каждой пары риторических отношений авторы обучили на извлеченных данных наивный байесовский классификатор. Признаками для каждой пары предложений служили всевозможные пары слов, взятых по одному из каждого предложения. Авторы предположили, что некоторые такие пары слов могут быть хорошими индикаторами для определенных риторических отношений. При этом ключевые слова и фразы, использовавшиеся для извлечения риторических отношений из текстового корпуса, перед обучением классификаторов были удалены из обучающей выборки. Все парные классификаторы тестировались на наборах данных с равным – по 5,000 – количеством примеров каждого класса и показали точность выбора между двумя отношениями от 65 до 93%. Авторы также обучили общий байесовский классификатор для всех шести рассматриваемых классов отношений, который при аналогичном тестировании показал точность 49,7%.

Далее авторы предложили использовать в качестве признаков для обучения и классификации пары наиболее информативных слов предложений – глаголов и существительных, а также характерных для рассматриваемых отношений союзов и вводных слов. Эксперименты на основе синтаксически размеченного текстового корпуса BLIPP (1,800,000 предложений) позволили добиться примерно той же точности парных классификаторов при значительно меньшем объеме обучающей выборке - 100,000 примеров на каждый класс.

**SPADE**

Soricut и Marcu в работе [20] представили вероятностный подход к построению риторических деревьев в рамках отдельных предложений с использованием их синтаксической и лексической структуры. Система впоследствии получила название SPADE –Sentence-level PArsing for DiscoursE. Вероятностная модель сначала используется для определения границ элементарных дискурсивных единиц в составе предложения. Затем SPADE использует динамическое программирование, чтобы сопоставить предложению его наиболее вероятную риторическую структуру. Вероятности потенциальных RST-деревьев для заданного предложения определяются с помощью признаков, выводимых из представления синтаксической структуры текста в виде лексикализованной иерархии составляющих.

В качестве таковых признаков авторы вводят доминантные множества (dominance sets). Для заданного предложения доминантное множество определяется как набор узлов его иерархии составляющих, являющихся корневыми узлами соответствующих иерархий для содержащихся в предложении элементарных дискурсивных единиц, взятых вместе с их непосредственными родительскими узлами (за исключением корневой составляющей для всего предложения). Другими словами, доминантное множество предложения состоит из всех «точек присоединения» иерархий составляющих отдельных ЭДЕ друг к другу.

Авторы использовали RST Discourse Treebank для обучения и тестирования моделей, используя как полный (110 отношений), так и сокращенный (18) наборы риторических отношений. Для оценки производительности использовалась F-мера точности и полноты по метрике PARSEVAL. Система продемонстрировала F-меры 49% и 45,6% для сокращенного и полного наборов риторических отношений соответственно, что в сравнении с парсером Marcu (2000) – state-of-art на тот момент – дало выигрыш порядка 10%. Использование экспертного синтаксического анализа и выделения ЭДЕ позволило поднять F-меры для сокращенного и полного наборов отношений до 75,5% и 70,3% соответственно. При этом в терминах неразмеченной риторической структуры (без учета типов отношений) система достигла близкого к экспертному результата (96,2%).

**Reitter, 2003**

В данной работе [26] впервые было предложено использование машин опорных векторов для (SVM) для классификации риторических отношений. Reitter обосновывал выбор SVM следующими причинами:

• высокое качество классификации, показываемое SVM во многих приложениях;

• возможность мультиклассовой классификации (один против всех);

• большая размерность (>5000) множества признаков и их взаимосвязанность;

• относительно малый размер доступной тренировочной выборки (<1000).

Использовались следующие классы признаков:

• Присутствие в сегменте ключевых слов и их относительные позиции в сегменте. Набор ключевых слов автоматически извлекался из тренировочного корпуса посредством фильтрации по частям речи.

• Характеристики именных фраз.

• Вхождения в сегмент определенных знаков пунктуации.

• Части речи слов на границах сегментов.

• Лексическая схожесть сегментов.

• Отношение длин сегментов в словах.

Для обучения и тестирования классификатора использовались RST Discourse Treebank, а также вручную составлений корпус из новостных статей на немецком языке. Рассматривался сокращенный набор отношений RST-DT. Reitter заявляет о 61,8%-ной точности мультиклассовой классификации на тестовом подмножестве RST-DT и 39,1%-ной для своего корпуса.

**duVerle, 2009**

duVerle и Prendinger в работе [27] решили опустить этап сегментации и сконцентрировался на втором этапе парсинга дискурса, мотивируя это решение тем, что текущие state-of-the-art результаты в автоматическом сегментировании (Marcu и LeThanh) значительно ближе к экспертным по сравнению с качеством собственно RST-деревьев.

Развивая идеи Reitter, авторы использовали механизм SVM как для оценки вероятности наличия некоторого риторического отношения между заданными сегментами, так и для предположений о конкретном типе и направленности этого отношения, что вкупе с наивным агломеративным алгоритмом построения RST-иерархии позволило получать корректные риторические структуры за линейное время.

Авторы обучили два SVM-классификатора – бинарный для наличия/отсутствия и мультиклассовый для типов и направлений отношений – на данных RST-DT. Для второго классификатора использовался набор из 18 отношений, что вместе со всеми возможными комбинациями расстановки ролей ядро-спутник для двух сегментов давало 41 класс. Начиная с набора атомарных риторических деревьев, соответствующих элементарным сегментам, алгоритм итеративно выбирал наилучшую пару соседних сегментов для соединения (на основе оценок, полученных с помощью первого классификатора), определял тип отношения и роли объединяемых сегментов с помощью второго классификатора, добавлял в набор новое RST-поддерево и обновлял оценки первого классификатора. С учетом того, что для объединения рассматривались только смежные поддеревья, на каждой итерации требовалось пересчитать первым классификатором не более двух оценок, что давало существенный выигрыш в плане временной сложности. С последним оставшимся в наборе RST-деревом алгоритм заканчивал работу.

Как и в предыдущих работах (Soricut and Marcu, Reitter, LeThanh), duVerle работал исключительно с бинарными RST-деревьями. Не бинарные многоядерные отношения при обучении трансформировались в наборы вложенных одно в другое бинарных.

Оба классификатора использовали широкий набор признаков. Авторы исходили из предположения устойчивости SVM к влиянию неинформативных признаков при больших (~105) размерностях признакового пространства. Ниже представлены классы использованных признаков:

1. Авторская организация текста: признаки, отражающие принадлежность рассматриваемых сегментов к одному предложению или абзацу, а также длины сегментов в словах, ЭДЕ и т.д.

2. Лексические и пунктуационные признаки: вместо того, чтобы кодировать присутствие определенных сигнальных фраз, авторы автоматически собрали с помощью тренировочного корпуса словарь часто используемых n-грамм (n = 1, 2, 3), и кодировали их присутствие среди первых и последних n токенов каждого сегмента. В качестве токенов учитывались не только слова, но и знаки пунктуации, а также маркеры границ предложений и абзацев (в этом основное отличие метода от использования явного списка сигнальных фраз, как, например, у Marcu). Существенно лучшие результаты метода по сравнению с поиском сигнальных фраз в bag-of-words-представлении сегментов подтвердили гипотезу о том, что сильные риторические сигналы обычно расположены на границах сегментов.

3. Синтаксические признаки: в дополнение к предыдущему пункту, а также для достижения большей независимости от конкретного лексического контента, авторы кодировали части речи первых и последних трех слов каждого сегмента. Использование более длинных префиксов/суффиксов не оправдало себя.

4. Доминантные множества: ряд признаков, эмулирующих использование идеи доминантных множеств Soricut и Marcu. Данные признаки актуальны только для сегментов размером меньше одного предложения.

5. Признаки, эксплуатирующие критерий строгой композиции: дублирование признаков классов 1-4 для наиболее значимых ЭДЕ рассматриваемых сегментов.

6. Внутренняя риторическая структура: признаки, отражающие строение RST-поддеревьев рассматриваемых сегментов.

Впоследствии система duVerle и Prendinger обзавелась собственным механизмом сегментации (тоже, кстати, основанном на SVM) и получила название HILDA [29]. На данный момент эта система является state-of-art среди полнотекстовых риторических парсеров.

**2.3 Выводы**

В рамках рассматриваемой задачи внедрение любой из вышеописанных техник требует модификаций.

Основным недостатком подходов на основе правил является трудоемкость составления этих самых правил. Так, внедрение подхода Marcu требует составления регулярных выражений для каждой потенциальной сигнальной фразы. Учитывая тот факт, что модифицируемая система должна адекватно работать с несколькими языками, необходимо иметь свои наборы правил (в частности, регулярных выражений) для каждого языка. То же самое применимо к схемам Corston-Oliver (причем здесь эвристики значительно сложнее) и LeThanh (здесь конкретный вид используемых правил неизвестен). Подходы на машинном обучении в этом плане проявляют большую гибкость, так как в общем случае можно переобучить модель для нужного языка. Также стоит отметить, что все три рассмотренных нами подхода так или иначе используют перебор допустимых RST-деревьев, что может критически увеличить временную сложность модифицируемой системы – текущая ее версия строит RST-дерево заданного текста за один проход.

Среди рассмотренных нами работ, так или иначе использующих машинное обучение, предлагается лишь два полнотекстовых парсера – shift-reduce алгоритм Marcu и HILDA. Работа Reitter описывает лишь классификатор риторических отношений, метод же построения риторических деревьев на его основе не был реализован. Тем не менее, похожий механизм классификации отношений используется системой HILDA. Работы Marcu, Echihabi и Soricut, Marcu направлены на построение RST-деревьев исключительно в рамках предложений, так что их внедрение требует дополнительного механизма построения риторической структуры всего текста (например, переборной схемы из рассмотренных абзацем выше). Также следует отметить, что в обеих работах используются вероятностные подходы, для качественной работы которых необходимо значительное количество обучающих данных. Marcu и Echihabi предлагают решение данной проблемы, однако обобщающая способность моделей, обученных на таких данных, сомнительна. Еще одна проблема – использование формализма доминантных множеств (Soricut, Marcu) требует дополнительных накладных расходов по преобразованию грамматик зависимостей, используемый синтаксическим анализатором ABBYY Compreno, к лексикализованным грамматикам составляющих.

Shift-reduce алгоритм Marcu и алгоритм duVerle наиболее близки к решению поставленной нами задачи. Поскольку оба алгоритма строят RST-дерево заданного текста в один проход, внедрение любого из них позволит сохранить линейность времени работы системы. Обе работы, однако, описывают не конкретные наборы используемых признаков, а их классы (с примерами). Соответственно, их реализация требует более детальной спецификации признакового пространства.

Следует отметить, что все рассмотренные выше работы не делают предположений о целях, для достижения которых строятся RST-деревья. Такой подход обеспечивает универсальность, но может быть недостаточно эффективен в рамках поставленной задачи. Алгоритм взвешивания, использующийся в модифицируемой нами системе, учитывает только направленность отношений в RST-дереве текста и игнорирует их конкретный класс. Иначе говоря, разумно ограничиться ровно двумя классами риторических отношений (симметричные/асимметричные). Для работ, использующих машинное обучение, это позволит более эффективно использовать имеющиеся обучающие данные, однако неизбежно приведет к появлению неинформативных признаков. Для их элиминации либо корректировки требуется дополнительный анализ.

# 3 Исследование и построение решения задачи

**3.1 Обзор модифицируемой системы**

Рисунок 1 иллюстрирует общую схему работы модифицируемой системы.

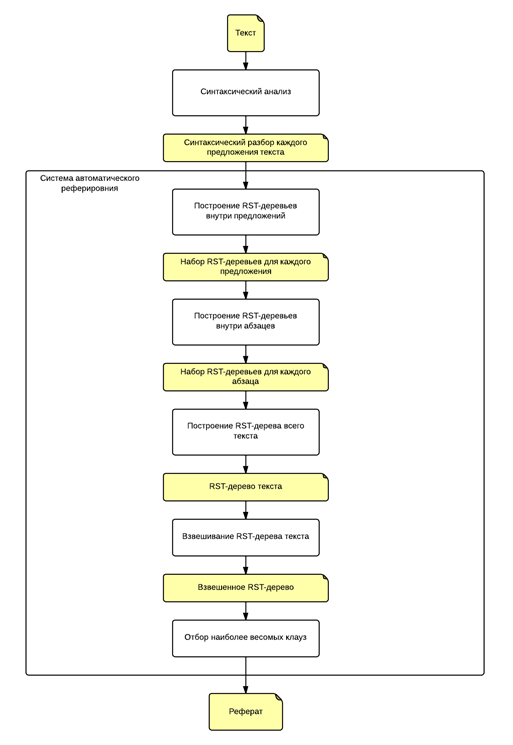
****

Рисунок 1. Общая схема работы системы автоматического реферирования ИСП РАН

В процессе построения риторического дерева текста система сначала осуществляет объединение элементарных сегментов в рамках предложений. Затем из RST-деревьев предложений строятся риторические деревья абзацев текста. Наконец, над набором RST-деревьев абзацев формируется единая риторическая структура всего текста.

Первый этап проводится непосредственно при определении границ клауз, выступающих в роли элементарных сегментов, и основывается на синтаксическом подчинении: отношение между ядром и спутником соответствует отношению между синтаксически главной и зависимой клаузами. Каждый этап, кроме первого, представляет собой последовательное применение к набору риторических сегментов эвристик-коннекторов. Такие эвристики строят набор объемлющих риторических структур над входным набором RST-деревьев путем объединения некоторых из них. В архитектуре системы коннекторы представлены классами, реализующими интерфейс IRSTreesConnector:

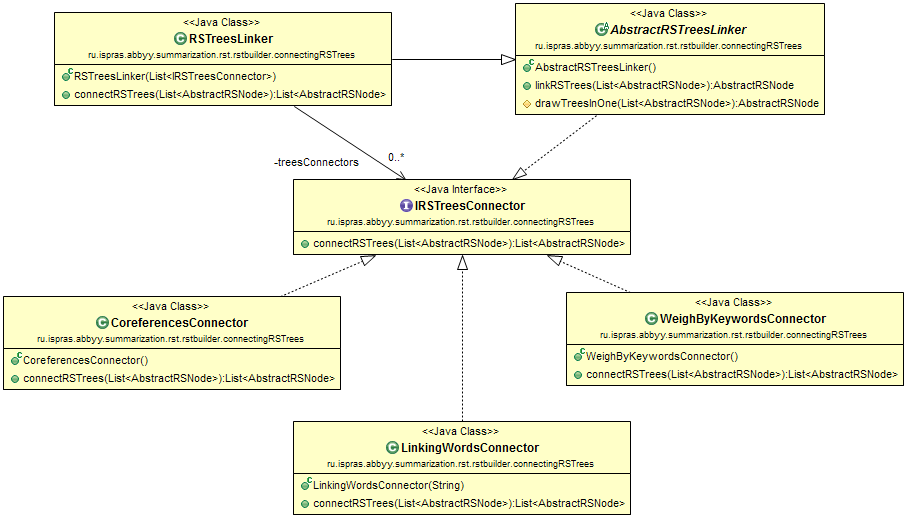


Рисунок 2. Архитектура подсистемы, ответственной за объединение риторических деревьев

Как видно из диаграммы на Рисунок 2, IRSTreesConnector оперирует списками экземпляров класса AbstractRSNode. Данный класс – представление узла риторического дерева, будь то элементарный сегмент (клауза, Clause) или же риторическое отношение (RSRelation): симметричное (ParatacticRelation) или асимметричное (HypotacticRelation). Рисунок 3 иллюстрирует описанную иерархию.

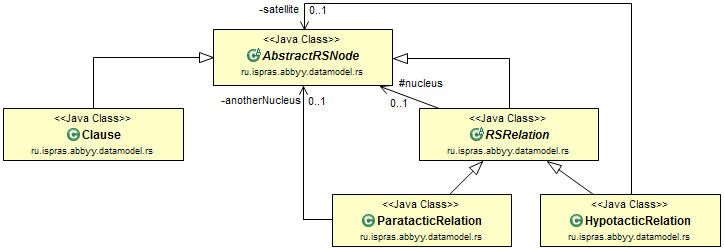


Рисунок 3. Модель данных модифицируемой системы

Система использует три эвристики для объединения RST-деревьев (см. Рисунок 2): CoreferencesConnector, LinkingWordsConnector и WeighByKeywordsConnector. Первая основана на поиске кореферентных – ссылающихся на один и тот же объект или явление – слов в объединяемых сегментах. LinkingWordsConnector с помощью эмпирического словаря ищет вхождения фраз и выражений, сигнализирующих наличие определенных риторических отношений. Наконец, WeighByKeywordsConnector использует информацию о содержании в сегментах ключевых слов, т.е. слов, в совокупности наиболее полно отражающих суть текста. Последняя эвристика используется для вывода отношений между абзацами, первые две – между предложениями в рамках абзацев.

Для того, чтобы с помощью заданного набора эвристик-коннекторов вывести ровно одно RST-дерево над входным набором, используется метод linkRSTrees класса RSTreesLinker. При этом сначала вызывается операция connectRSTrees, реализация которой в классе RSTreesLinker последовательно применяет ко входному набору RST-деревьев все коннекторы, ассоциированные с данным экземпляром класса. Если в результате остается два и более RST-деревьев, они объединяются с помощью паратактической связи (метод drawTreesInOne класса AbstractRSTreesLinker).

Исходя из особенностей архитектуры системы, для решения поставленной задачи разумно построить механизм объединения RST-деревьев на основе машинного обучения в виде класса, реализующего интерфейс IRSTreesConnector.

**3.2 Выбор схемы построения RST-дерева**

В качестве схемы построения RST-дерева было решено использовать жадный иерархический алгоритм аналогично системе duVerle и Predinger. Данный алгоритм прост в реализации и при этом имеет ряд преимуществ по сравнению с shift-reduce подходом Marcu и схемами поиска наиболее вероятного разбора, представленными в работах Marcu, Corston-Oliver и LeThanh.

К недостаткам подхода Marcu, основанного на обкатанной для синтаксических анализаторов схеме сдвиг-свертка, можно отнести его локальность: для принятия решения о следующем действии алгоритм просматривает три дерева в стеке и один сегмент на входе. Проводя аналогию с синтаксическим анализом, можно вспомнить, что применимость разбора с предпросмотром в один символ сужает класс используемых языков (грамматик). Перенос соответствующих требований на риторическую структуру не очевиден. В частности, при просмотре вперед только одного сегмента можно потерять информацию о содержащем его абзаце. Более корректный подход требовал бы предпросмотра всех оставшихся сегментов, что для данного алгоритма накладно. В свою очередь, иерархический подход duVerle, несмотря на свою простоту, на каждом шаге использует информацию о возможностях проведения отношений для каждого текущего сегмента.

Конечно, RST-дерево, построенное жадным алгоритмом, может не являться наиболее вероятной риторической структурой текста. Использование одного из алгоритмов поиска наиболее вероятной риторической структуры могло бы решить эту проблему, однако, судя по результатам обзора, временные потери при этом значительно перекрывают выигрыш в качестве.

Как и duVerle и Predinger, мы будем использовать два классификатора: один для возможности какого-либо отношения и один для его конкретизации. При этом, учитывая специфику поставленной задачи, второй классификатор ограничится исключительно распределением ядер и спутников, т.е. для данной пары сегментов будет выбирать из трех возможных отношений: ядро-спутник (nucleus-satellite, NS), спутник-ядро (SN) или ядро-ядро (NN).

При этом, в отличие от подхода duVerle и Predinger, мы будем более строго учитывать авторскую организацию текста, последовательно применяя описанный выше алгоритм сначала в рамках отдельных предложений, затем в рамках абзацев и, наконец, на уровне всего текста. Похожий подход использовал, хоть и эвристически, LeThanh. DuVerle и Predinger использовали для учета текстовой организации дополнительный набор признаков, утверждая, что предложения и особенно абзацы часто могут быть разнесены по различным RST-поддеревьям в риторическом представлении текста. Данное утверждение представляется нам весьма спорным, тем более при учете того, что наиболее весомым признаком в структурном классификаторе duVerle оказалась именно принадлежность сегментов к одному предложению (см. [29]).

**3.3 Выбор механизмов классификации**

Оба классификатора было решено реализовывать на основе машин опорных векторов. Исследования Reitter и duVerle показали хорошую применимость SVM к задаче классификации риторических отношений. Возможность использования SVM для определения вероятностей принадлежности объекта к классу согласуется с предполагаемым алгоритмом построения RST-дерева. В соответствии с ограничениями по времени построения RST-деревьев мы будем использовать линейные SVM-классификаторы. В работе [29] показано, что точность линейных классификаторов в данной задаче незначительно уступает классификаторам с полиномиальным ядром, обеспечивая при этом существенный выигрыш по времени.

**3.4 Признаки**

**3.4.1 Длины сегментов.**

Как отмечалось у Reitter и duVerle, можно предполагать корреляцию между длинами сегментов и типами риторических отношений между ними. Например, спутник асимметричного отношения CONTRAST обычно короче ядра. Ввиду вышесказанного, логично выделить несколько признаков, характеризующих размеры сегментов в различных единицах, в частности, в словах или элементарных сегментах.

**3.4.2 Сигнальные фразы**

Ключевые или сигнальные слова и фразы так или иначе использует каждая из рассмотренных работ, потому как являются наиболее явными показателями наличия риторических отношений. Наша работа не станет исключением. У нас имеются списки, содержащие по несколько десятков сигнальных фраз для русского и английского языков. Для каждой такой сигнальной фразы введем в качестве булевых признаков ее наличие/отсутствие среди первых/последних 3 слов каждого сегмента.

**3.4.3 Синтаксические признаки**

Используемый нами синтаксический анализатор позволяет для каждого предложения получить синтаксическое дерево зависимостей, выделяя для каждого слова ее часть речи, специфичные для части речи характеристики формы слова, а также синтаксическую функцию (subject, object, etc.). Будем кодировать эти свойства как бинарные признаки вида <свойство>\_is\_<значение>, например, POS\_is\_Noun, NounType\_is\_Common, SyntacticFunction\_is\_Subject.

Использование SVM требует конечности вектора признакового пространства, потому мы не можем выделять такие признаки для каждого слова в сегменте. Аналогично Marcu и duVerle, мы будем рассматривать префиксы и суффиксы сегментов конечной длины (3 слова). При наложении префикса и суффикса сегмента друг на друга признаки для общих слов будут дублироваться. Также вместо формализма доминантных множеств мы будем кодировать тот же набор признаков для 5 верхних слов соответствующего синтаксического дерева зависимостей (при его обходе в ширину). Если кодирование префиксов и суффиксов преследует примерно те же цели, что и кодирование сигнальных фраз, т.е. поиск и анализ связующих сегменты конструкций (как уже отмечалось многими исследователями, значимые риторические сигналы находятся на концах сегментов), то кодирование верхушки синтаксического дерева позволяет учесть наиболее общую синтаксическую структуру сегмента.

Для составных сегментов, покрывающих несколько ЭДЕ, может иметься несколько синтаксических деревьев. В таких случаях синтаксические признаки будут извлекаться для первого и последнего ЭДЕ сегмента. Соответственно, для сегментов-ЭДЕ все признаки будут дублироваться.

**3.4.4 Лексические классы**

Для слов из предыдущего пункта будем кодировать в той же манере их лексические классы. Это позволит достичь большего уровня абстракции от конкретных слов и эффективнее использовать данные для обучения. Для выделения лексических классов будем использовать возможности синтаксического анализатора ABBYY Compreno.

**3.4.5 Риторическая структура**

В процессе построения риторических деревьев высших порядков необходимо учитывать риторическую структуру нижележащих. Для каждого сегмента будем выделять как признаки тип верхнего риторического отношения в его RST-дереве (три булева признака: \_is\_NS, \_is\_SN, \_is\_NN), а также размеры сегментов-потомков в ЭДЕ по отношению к размеру в ЭДЕ сегмента родительского.

Дополнительно будем дублировать признаки 1-4 для первой и последней ЭДЕ, принадлежащих promotion set рассматриваемого сегмента.

**3.5 Обучение и тестирование**

**3.5.1 Данные**

*3.5.1.1 RST*

RST Discourse Treebank [35], составленный в 2001 году усилиями Carlson, Marcu и Okurowski, является на настоящий момент наиболее объемным корпусом англоязычных текстов, аннотированных экспертами в соответствии с RST. Он содержит 385 статей из Wall Street Journal, сопровождаемых полнотекстовыми риторическими деревьями. Корпус распространяется через Linguistic Data Consortium [36] бесплатно для его членов, для остальных, однако, использование корпуса стоит хороших денег. Ввиду этого в настоящей курсовой работе было решено отказаться от использования RST-DT и обратить внимание на небольшой, но свободно распространяемый Discourse Relations Reference Corpus.

Discourse Relations Reference Corpus [31] Taboada и Renkema содержит 65 текстов на английском языке: 14 примеров разборов с сайта RST [11], 21 текст из RST-DT и 30 текстов из SFU Review Corpus [37]. Корпус распространяется бесплатно через сайт RST. Там же доступны файлы разборов текстов, полученные с помощью программы RSTTool [38] (утилита для автоматизации процесса RST-аннотирования). Разборы поставляются в форматах .rs2, .rs3 (форматы, основанные на XML) и .lisp (рекурсивное преставление дерева в синтаксисе языка Lisp) и имеют прозрачную структуру для анализа.

DRRC предполагается использовать для обучения англоязычных классификаторов и оценки качества выводимых системой RST-деревьев. Пары смежных сегментов из текстов корпуса, явно связанные некоторым риторическим отношением, будут служить обучающими примерами как для классификатора, оценивающего наличие или отсутствие риторической связи (такой пример соответствует «положительному» классу – риторическая связь присутствует), так и для классификатора, оценивающего конкретный вид риторического отношения (пример соответствует одному из классов NS, SN и NN).

Для обучения первого классификатора также необходимо некоторое количество примеров риторически не связанных сегментов. При извлечении таких примеров из обучающей выборки RST-деревьев мы будем руководствоваться следующим правилом: непересекающиеся сегменты текста X и Y могут считаться риторически не связанными, если в RST-дереве текста не имеется такого отношения R между сегментами P и Q, X ⊆ P и Y ⊆ Q, что в поддеревьях для P и Q сегменты, содержащие X и Y, всегда являются ядрами отношений. В частности, два элементарных сегмента X и Y риторически не связаны, если в RST-дереве текста не найдется риторического отношения, для аргументов которого X и Y являются наиболее значимыми ЭДЕ (т.е. входят в соответствующие promotion set’ы). Легко видеть, что данное определение отсутствия риторической связи согласуется с принципом строгой композиции Marcu. При подготовке обучающих данных на каждый пример наличия отношения мы будем извлекать одну пару риторически не связанных сегментов – это всегда возможно, поскольку «отрицательных» примеров в каждом тексте значительно больше, чем «положительных».

*3.5.1.1 Рефераты*

Для оценки качества системы реферирования мы располагаем корпусом текстов на основе материалов конференции DUC-2001. Корпус составлен из текстов новостных статей, взятых из таких газет как Wall Street Journal, San Jose Mercury News и др. К конференции было подготовлено в общей сложности 60 наборов текстов: 30 тренировочных и 30 тестовых, - содержащих каждый от 6 до 14 статей. Каждая статья содержит в среднем 800 слов (порядка 37 предложений) и сопровождается тремя экспертными аннотациями (abstracts) размером около 100 слов. Мы располагаем тестовым подмножеством оригинального корпуса (всего 311 статей), далее будем называть его DUC-Abstracts. Также имеется 147 статей из тренировочной части корпуса с аннотациями типа extract (наборы информативных предложений; средний размер – 160 слов), составленными Mary Ellen Okurowski и John M. Conroy на основе abstract-аннотаций. На этот набор мы будем ссылаться как на DUC-Extracts.

**3.5.2 Метрики тестирования**

*3.5.2.1 Оценка деревьев*

Для оценки корреляции RST-деревьев, построенных системой, с экспертными мы будем использовать метрики PARSEVAL (PARSing EVALuation). Данный набор метрик изначально использовался для оценки качества синтаксических парсеров и по аналогии был перенесен Marcu ([24]) на анализатор риторический.

Риторические деревья рассматриваются как наборы узлов (составляющих). Составляющая RST-дерева – элементарный сегмент или риторическое отношение. Мы будем называть составляющую в RST-дереве, полученном системой, правильной, если она присутствует в экспертном RST-дереве. В частности, правильность составляющей-отношения означает, что в экспертном дереве те же сегменты текста соединены тем же риторическим отношением.

Метрики PARSEVAL описывают три понятия: размеченные точность, полнота и F-мера. Размеченная точность – отношение числа правильных составляющих в RST-дереве, построенном системой, к общему числу составляющих в *этом* дереве. В свою очередь, размеченная полнота – отношение числа правильных составляющих в RST-дереве системы к общему числу составляющих в *экспертном* дереве. Наконец, размеченная F-мера – среднее гармоническое размеченных точности и полноты.

*3.5.2.2 Оценка рефератов*

Корреляцию получаемых системой рефератов с экспертными аннотациями мы будем оценивать с помощью метрик ROUGE (Recall-Oriented Understudy for Gisting Evaluation) – фактически, эти метрики являются на сегодняшний день стандартом оценки качества автоматически генерируемых рефератов. Метрики ROUGE основаны на подсчете количества перекрывающихся n-грамм в автоматическом и экспертном рефератах. Мы будем использовать ROUGE-N c N = 1,2, ROUGE-L и ROUGE-W-1.2.

ROUGE-N определяется как полнота набора n-грамм автоматически сгенерированного реферата по отношению к набору экспертных аннотаций данного текста и рассчитывается по формуле

|  |  |
| --- | --- |
|  | (1) |

где *ReferenceSummaries* – экспертные аннотации, n – длина n-грамм *gramn* и *Countmatch(gramn)* возвращает максимальное число n-грамм *gramn*, встречающихся как в автоматически сгенерированном, так и в экспертных рефератах.

Метрика Rouge-L вместо числа совпадающих n-грамм использует длину наибольшей общей подпоследовательности (с пропусками) автоматической и экспертной аннотаций. Rouge-W представляет собой взвешенный вариант Rouge-L, дающий преимущество непрерывным подпоследовательностям. Конкретные формулы для расчета этих метрик можно найти в [18]. Там же на основе данных конференций DUC 2001-2003 гг. показано, что выбранные метрики хорошо коррелируют с экспертными оценками рефератов.

**3.5.3 Оптимизация пространства признаков**

Хотя применимость выделенных нами признаков к классификации риторических отношений обоснована в работах Marcu и duVerle, заострение внимания на направленности отношений без учета их конкретных классов может привести к появлению среди признаков бесполезных и избыточных. Например, выше уже упоминалась о корреляции такого признака, как соотношение длин участвующих в отношении сегментов, с типом отношения между ними. При этом, однако, для некоторых отношений сателлит обычно короче ядра (CONCESSION), для других – длиннее (ELABORATION), а для третьих распространены оба случая (LISTING). Соответственно, данный признак слабо информативен, если дело касается исключительно направленности отношений (по крайней мере, в отрыве от остальных признаков). Для устранения таких признаков и обоснования релевантности остальных необходимо организовать отбор наиболее информативных признаков из рассмотренных выше. Сокращение размерности признакового пространства также позволит нам улучшить временные характеристики работы нашего алгоритма.

Чтобы получить данные о релевантности конкретных признаков и не привязываться к механизму классификации, решено было реализовать фильтрацию признаков по порогу некоторой статистики. В качестве последней использована простая в реализации хи-квадрат оценка. Для данных признака *f* и класса *c* хи-квадрат статистика рассчитывается как

|  |  |
| --- | --- |
|  | (2) |

где *A* – число примеров класса *c*, где *f* =1;

*B* – число примеров, не принадлежащих *c*, где *f* =1;

*C* – число примеров класса *c*, где *f* =0;

*D* – число примеров, не принадлежащих *c*, где *f* =0;

*m = A*+*B*+*C*+*D –* общее число примеров.

В качестве оценки признака мы брали максимальную по модулю оценку среди всех классов:

|  |  |
| --- | --- |
|  | (3) |

Заметим, что данные формулы оперируют исключительно бинарными признаками, поэтому для их использования все признаки, принимающие вещественные значения, были приведены к набору бинарных. Например, такой признак, как относительная длина сегмента, принимающий значения из полуинтервала (0, 1], был преобразован в набор булевых признаков следующего вида: относительная длина сегмента принадлежит полуинтервалу (0, 0.1], относительная длина сегмента принадлежит полуинтервалу (0.1, 0.2] и т.д. Размерность признакового пространства обоих классификаторов после бинаризации составила 19325 признаков.

Мы воспользовались корпусом DRRC для сбора обучающих данных и вычислили на его основе хи-квадрат оценки информативности всех используемых признаков для каждого из классификаторов.

Пять наиболее релевантных признаков для классификатора, отвечающего за оценку наличия или отсутствия риторической связи, составили:

|  |  |
| --- | --- |
| **Признак** |  |
| * Входные сегменты состоят из одинакового числа ЭДЕ | 57.805 |
| * Число ЭДЕ в первом сегменте больше, чем во втором | 22.600 |
| * Лексический класс слова, расположенного в вершине синтаксического дерева последней значимой клаузы первого сегмента, – SITUATIONAL AND ATTRIBUTIVE | 18.085 |
| * Лексический класс первого слова первой значимой клаузы первого сегмента - SITUATION | 17.362 |
| * Лексический класс последнего слова первой клаузы первого сегмента – VERBAL COMMUNICATION | 15.968 |

Пять наиболее релевантных признаков для классификатора, предсказывающего тип риторического отношения, составили:

|  |  |
| --- | --- |
| **Признак** |  |
| * Лексический класс предпоследнего слова первой клаузы первого сегмента – INTELLECTUAL ACTIVITY | 118.677 |
| * Лексический класс третьего слова первой клаузы первого сегмента –COMMUNICATIONS | 116.281 |
| * Часть речи второго слова последней клаузы первого сегмента – деепричастие | 116.280 |
| * Лексический класс первого слова последней клаузы второго сегмента – FEELING AS CONDITION | 92.828 |
| * Лексический класс слова, расположенного третьим при обходе в ширину синтаксического дерева последней клаузы второго сегмента, - ENTITY BY FUNCTION AND PROPERTY | 81.296 |

Посредством кросс-валидации мы установили оптимальные пороговые значения статистики равными 7.341 для первого классификатора и 71.206 для второго, что привело к сокращению размерности признаковых векторов соответственно до 465 и 72 признаков.

# 4 Практическая часть

**4.1 Инструментарий**

Решение задачи было выполнено на языке программирования Java. Выбор языка программирования был обусловлен языком реализации модифицируемой системы.

Для реализации классификаторов использовалась обертка для libsvm [39] – популярной библиотеки SVM классификации и регрессии, – входящая в состав библиотеки машинного обучения отдела Информационных систем ИСП РАН.

Для оценки качества рефератов с помощью метрик ROUGE использовалось программное средство ROUGE-1.5.5 [18], реализованное на языке программирования Perl.

**4.2 Архитектура решения**

Алгоритм построения RST-дерева на основе машинного обучения реализован в классе MLConnector:

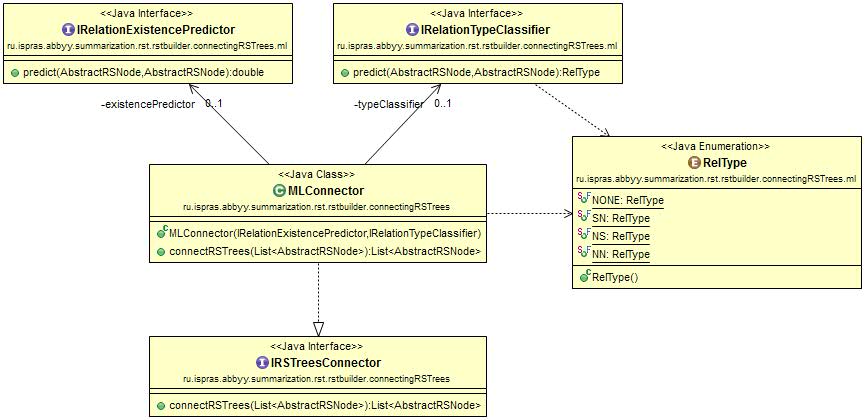


Рисунок 4. Общая архитектура решения

В перечислимом типе RelType определяется набор рассматриваемых классов риторических отношений:

* SN (Satellite-Nucleus) – асимметричное риторическое отношение, в котором спутник предшествует ядру
* NS (Nucleus-Satellite) – асимметричное риторическое отношение, в котором ядро предшествует спутнику
* NN (Nucleus-Nucleus) – симметричное риторическое отношение
* NONE – специальный тип для обозначения отсутствия риторической связи

Конструктор класса MLConnector принимает на вход два параметра – классификаторы для определения наличия риторической связи и ее направленности соответственно. Первый специфицируется интерфейсом IRelationExistencePredictor. Интерфейс определяет единственную операцию predict(AbstractRSNode, AbstractRSNode): double, которая принимает на вход два сегмента текста (в виде их RST-деревьев) и возвращает вероятность наличия какого-либо риторического отношения между ними. Классификатор для уточнения типа отношения реализует интерфейс IRelationTypeClassifier. Определяемая им операция predict возвращает для входных сегментов предполагаемый класс риторического отношения между ними в виде объекта типа RelType.

Описанные выше интерфейсы реализованы соответственно в классах RelExistencePredictor и RelTypeClassifier:

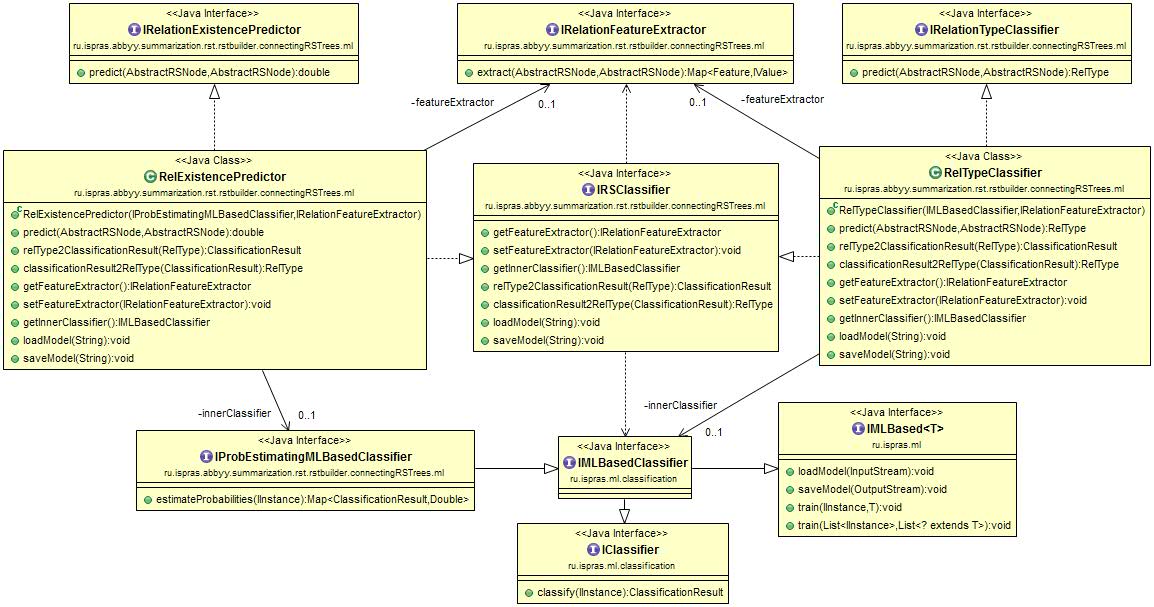


Рисунок 5. Классификаторы риторических отношений

Как видно из диаграммы на Рисунке 5, классы RelExistencePredictor и RelTypeClassifier имеют похожую структуру. Оба класса имеют атрибут featureExtractor – объект, реализующий интерфейс IRelationFeatureExtractor и отвечающий за извлечение признаков для классификации входной пары сегментов. Набор признаков затем конвертируется в объект класса Instance (этот класс – часть модели данных библиотеки машинного обучения ИСП РАН) и подается на вход внутреннему классификатору (innerClassifier). Отметим, что RelExistencePredictor использует для спецификации внутреннего классификатора интерфейс IProbEstimatingMLBasedClassifier, расширяющий интерфейс IMLBasedClassifier библиотеки машинного обучения ИСП РАН. IProbEstimatingMLBasedClassifier определяет единственную дополнительную операцию estimateProbabilities, обеспечивающую возможность оценки вероятностей классов. Поскольку рассматриваемый алгоритм построения RST-деревьев не требует вычисления вероятностей типов отношений, классификатор RelTypeClassifier использует интерфейс IMLBasedClassifier.

Классы RelExistencePredictor и RelTypeClassifier дополнительно реализуют интерфейс IRSClassifier, в который вынесены операции получения атрибутов, преобразования между RelType и ClassificationResult (представление результата работы классификатора в библиотеке машинного обучения ИСП РАН), а также операции загрузки модели из файла и ее сохранения в файл. Описанный интерфейс введен для удобства обучения классификаторов.

Внутренние SVM-классификаторы, как уже отмечалось, построены на основе обертки для libsvm, предоставленной библиотекой машинного обучения ИСП РАН (см. Рисунок 6). Обертка не поддерживает способность libsvm оценивать вероятности классов в процессе классификации. Нам удалось исправить этот недостаток с помощью классов ProbEstimatingCLibSvmClassifier и ProbEstimatingCLibSvmFacade.

Для обучения классификаторов используется класс RSTrainer (Рисунок 7). Его атрибутами являются classifier – обучаемый классификатор, который специфицируется описанным выше интерфейсом IRSClassifier, – а также precedentCollector – сборщик примеров для обучения, реализующий интерфейс IPrecedentCollector. На основе данных, получаемых от precedentCollector’а посредством операции collectPrecedents, RSTrainer предоставляет две операции: train, с помощью которой проводится обучение внутреннего классификатора, и optimizeFE. Последняя осуществляет выбор наиболее информативных признаков для классификации: признаки ранжируются с помощью хи-квадрат оценок (см. пункт 3.3.6), после чего кросс-валидацией на обучающей выборке находится оптимальное пороговое значение статистики; признаки, информативность которых ниже порога, элиминируются.

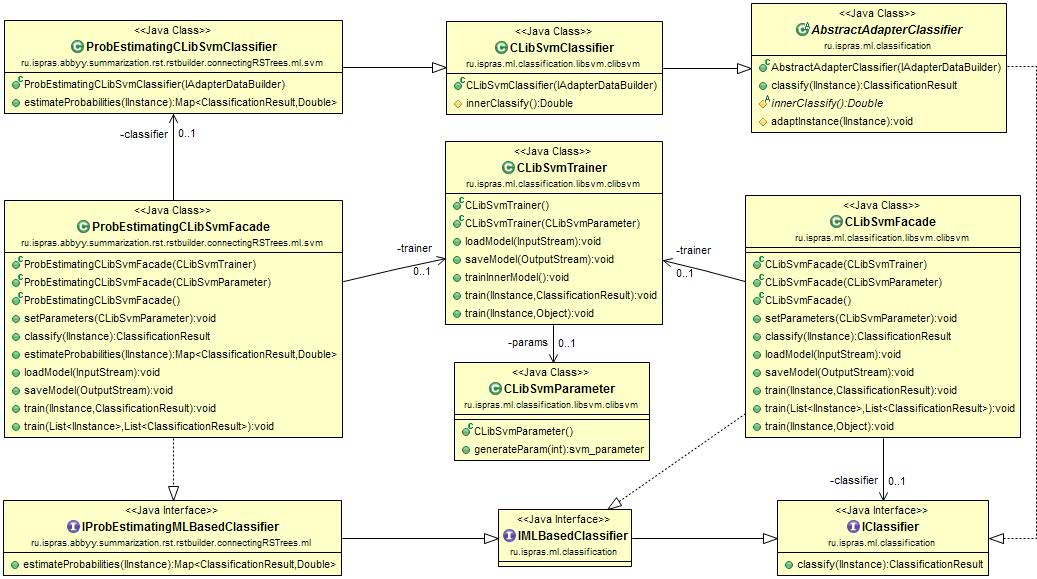


Рисунок 6. Обертка libsvm

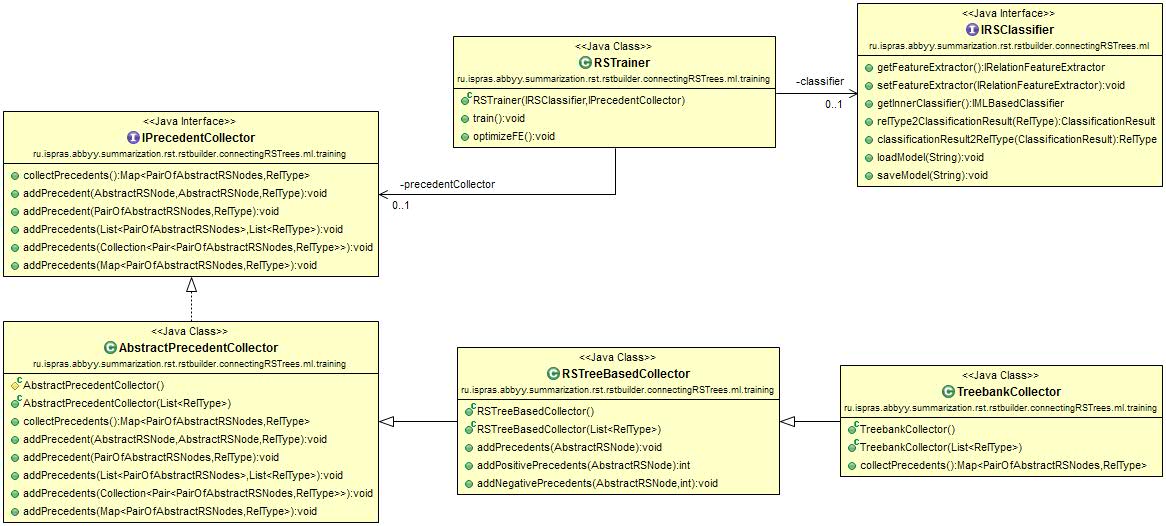


Рисунок 7. Подсистема обучения риторических классификаторов

Схема на Рисунке 7 отображает иерархию классов, предназначенных для подготовки обучающих данных. Каждый класс-сборщик имеет как конструктор по умолчанию, так и конструктор с аргументом – списком значений типа RelType – позволяющим ограничить набор типов извлекаемых примеров. Класс RSTreeBasedCollector предоставляет функциональность для извлечения риторических отношений и «не-отношений» прямо из RST-деревьев текстов. Класс TreebankCollector использует этот функционал для анализа корпуса DRRC.

Как уже отмечалось, за построение по двум сегментам набора признаков для классификации отвечают интерфейс IRelationFeatureExtractor и классы, его реализующие (см. Рисунок 8). Свой FeatureExtractor-класс организован для каждой группы признаков из рассмотренных в параграфе 3.3. Особняком стоит класс CompositeFE, позволяющий комбинировать несколько наборов признаков, последовательно применяя ассоциированные с ним FeatureExtractor’ы к паре риторических сегментов на входе.

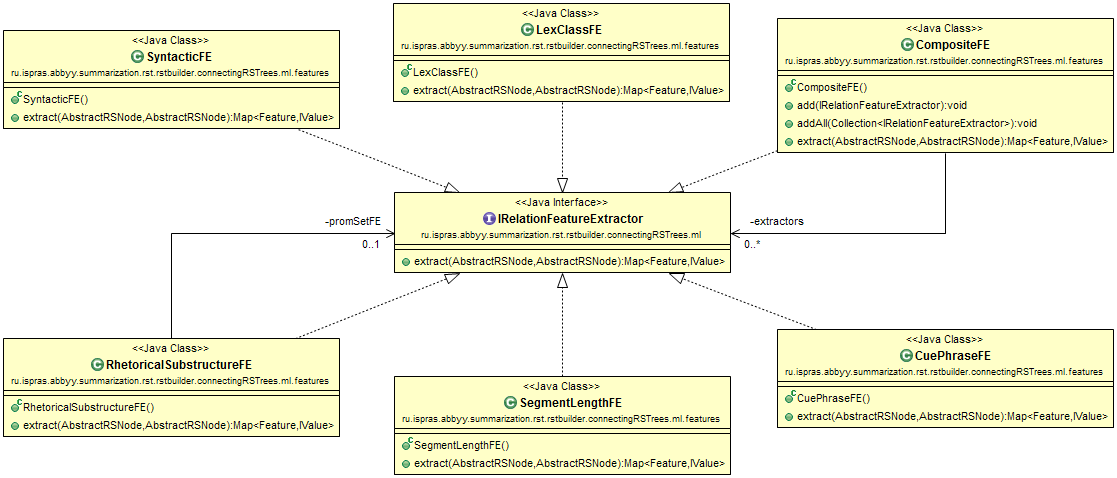


Рисунок 8. Подсистема извлечения признаков для классификации риторических отношений

**4.3 Характеристики функционирования**

Для тестирования использовался компьютер на базе процессора Intel Core i5-3210M (частота 2.50 ГГц) с 4 ГБ оперативной памяти.

**4.3.1 Качество RST-деревьев**

Для внутренней оценки качества алгоритма построения RST-деревьев мы вычислили средние значения точности, полноты и F-меры по метрикам PARSEVAL исходной и модифицированной версий системы на корпусе DRRC посредством кросс-валидации на 10 блоках. Результаты приведены в Таблице 2. Оценивалось как качество неразмеченной риторической структуры (без учета направлений и типов отношений; правильность составляющей определяется только через совпадение сегментов), так и правильность полноценных (размеченных) деревьев.

Таблица 2. Качественные характеристики риторических деревьев исходной (эвристический подход) и модифицированной (ML-подход) версий системы ИСП РАН на корпусе DRRC по метрикам PARSEVAL

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Неразмеченная структура | | | Размеченная структура | | |
| Точность | Полнота | F-мера | Точность | Полнота | F-мера |
| Эвристический подход | 0.5539 | 0.5539 | 0.5539 | 0.5307 | 0.5307 | 0.5307 |
| ML подход | **0.6538** | **0.6538** | **0.6538** | **0.5367** | **0.5367** | **0.5367** |

Результаты тестирования показывают, что использование машинного обучения позволяет существенно (на 10%) повысить качество неразмеченной структуры RST-деревьев, однако качество разметки осталось примерно на том же уровне.

**4.3.2 Качество рефератов**

Для внешней оценки качества алгоритма построения RST-деревьев мы построили с помощью обеих версий системы рефераты статей из корпусов DUC-Abstracts и DUC-Extracts и вычислили для каждого корпуса средние значения метрик ROUGE-N c N = 1,2, ROUGE-L и ROUGE-W-1.2. Мы также вычислили средние значения этих же метрик на каждом из корпусов для двух базовых эвристик, описанных во введении: реферат – первый абзац текста и реферат – набор первых предложений с каждого абзаца текста. Результаты тестирования приведены соответственно в Таблице 3 и Таблице 4.

Подход к построению риторических деревьев на основе машинного обучения показал более высокие результаты по сравнению с исходной эвристикой. Особенно существенна разница на корпусе DUC-Extracts, где выигрыш в F-мере модифицированной версии системы составляет от 8% по метрике Rouge-W до 17% по метрике Rouge-1. Вместе с этим, новый подход не смог достичь уверенного лидерства над базовыми алгоритмами. Модифицированная версия системы стабильно проигрывает 2-3% F-меры на DUC-Abstracts алгоритму, составляющему реферат из первых предложений каждого абзаца, однако выигрывает столько же на DUC-Extracts.

Таблица 3. Тестирование качества рефератов на корпусе DUC-Abstracts

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DUC-Abstracts** | | Система ИСП РАН, эвристический подход | Система ИСП РАН, ML подход | Реферат - первый абзац текста | Реферат состоит из первых предложений каждого абзаца |
| ROUGE-1 | Average recall | 0.38252 | **0.40559** | 0.20768 | 0.39090 |
| Average precision | 0.43041 | 0.45909 | **0.58912** | 0.48388 |
| Average F-measure | 0.40461 | **0.43013** | 0.29748 | 0.42928 |
| ROUGE-2 | Average recall | 0.13678 | 0.16609 | 0.09795 | **0.17627** |
| Average precision | 0.15380 | 0.18807 | **0.29344** | 0.21822 |
| Average F-measure | 0.14463 | 0.17616 | 0.14292 | **0.19359** |
| ROUGE-L | Average recall | 0.33035 | 0.35351 | 0.19771 | **0.36961** |
| Average precision | 0.37180 | 0.40017 | **0.56068** | 0.45733 |
| Average F-measure | 0.34946 | 0.37490 | 0.28317 | **0.40585** |
| ROUGE-W-1.2 | Average recall | 0.12704 | 0.13873 | 0.08428 | **0.14740** |
| Average precision | 0.23570 | 0.25890 | **0.40070** | 0.30121 |
| Average F-measure | 0.16491 | 0.18042 | 0.13591 | **0.19668** |

Таблица 4. Тестирование качества рефератов на корпусе DUC-Extracts

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DUC-Extracts** | | Система ИСП РАН, эвристический подход | Система ИСП РАН, ML подход | Реферат - первый абзац текста | Реферат состоит из первых предложений каждого абзаца |
| ROUGE-1 | Average recall | 0.44806 | **0.50243** | 0.21517 | 0.48709 |
| Average precision | 0.54773 | 0.64472 | **0.79703** | 0.63544 |
| Average F-measure | 0.47955 | **0.54889** | 0.31561 | 0.53073 |
| ROUGE-2 | Average recall | 0.25301 | **0.37211** | 0.17115 | 0.34992 |
| Average precision | 0.30908 | 0.47508 | **0.66696** | 0.45992 |
| Average F-measure | 0.26996 | **0.40496** | 0.25419 | 0.38088 |
| ROUGE-L | Average recall | 0.30446 | 0.40309 | 0.19795 | **0.40930** |
| Average precision | 0.37050 | 0.51357 | **0.75202** | 0.53723 |
| Average F-measure | 0.32464 | 0.43860 | 0.29266 | **0.44632** |
| ROUGE-W-1.2 | Average recall | 0.08369 | **0.13481** | 0.05809 | 0.11423 |
| Average precision | 0.28249 | 0.47552 | **0.62620** | 0.41104 |
| Average F-measure | 0.12553 | **0.20431** | 0.10242 | 0.17228 |

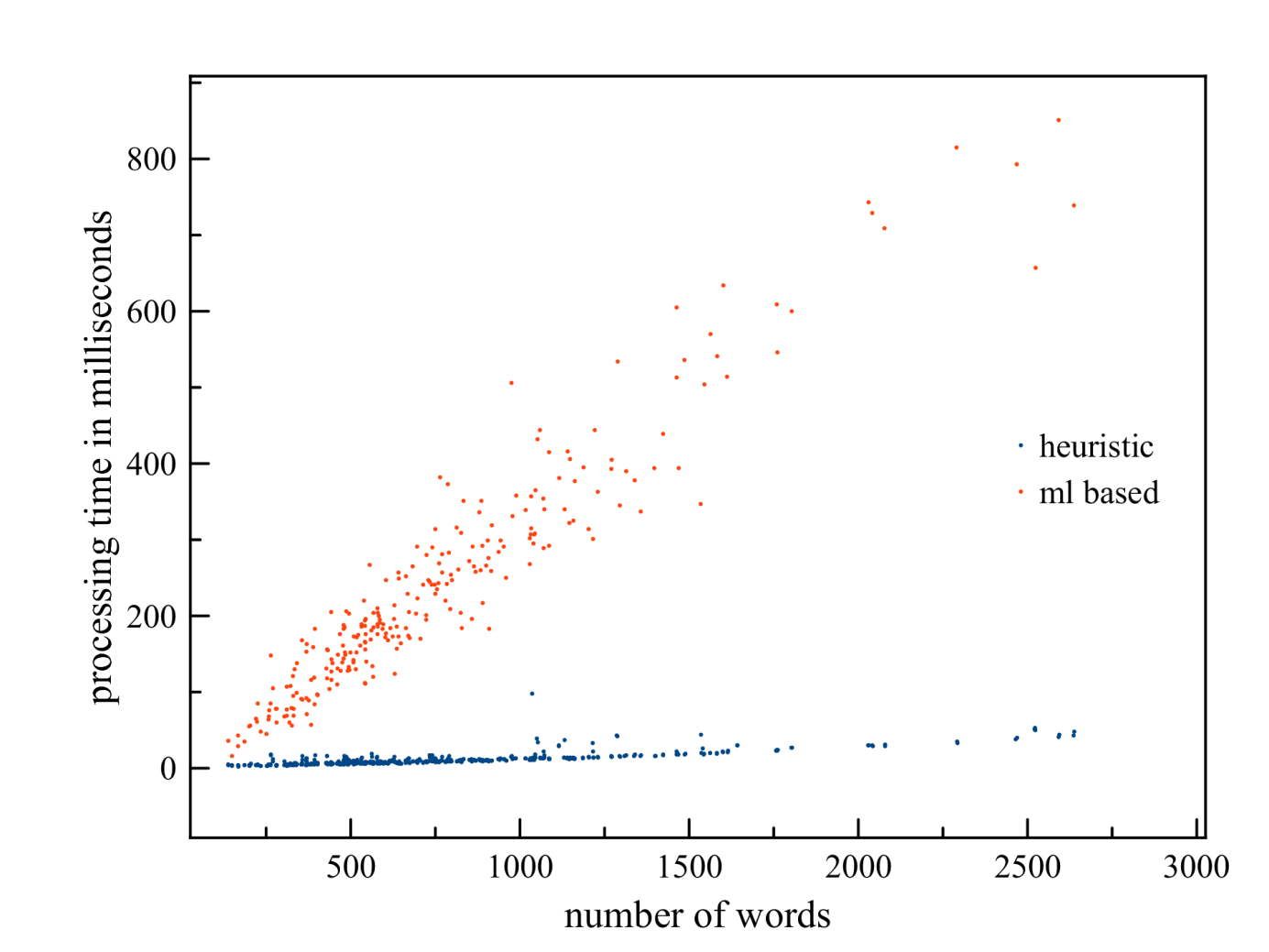


Рисунок 9. Зависимость времени построения RST-дерева исходной (heuristic) и модифицированной (ml based) версиями системы от размера текста в словах

**4.3.3 Время работы**

Для оценки временных характеристик работы системы мы для каждого документа, использованного для тестирования качества работы системы, рассчитали среднее время построения его RST-дерева каждой из версий системы на основе 10 запусков. Результаты отображены на Рисунке 9. Видно, что время работы версии системы на машинном обучении с увеличением размера текстов возрастает линейно, что удовлетворяет поставленным требованиям.

# Заключение

В рамках данной курсовой работы были исследованы существующие подходы к построению риторических деревьев на основе машинного обучения. Был реализован алгоритм построения RST-деревьев текста, использующий линейные SVM-классификаторы для оценки вероятности наличия либо отсутствия риторической связи между сегментами текста, а также для определения, какой из участвующих в отношении сегментов является ядром риторической связи, а какой – спутником. Данный алгоритм был встроен в систему автоматического реферирования ИСП РАН, что позволило повысить по сравнению с исходной эвристикой как качество проведения риторического анализа, так и качество получаемых с его помощью рефератов текстов. При этом реализованный алгоритм осуществляет обработку текста за линейное от его длины время.

Тем не менее, проведенные эксперименты выявили необходимости доработки реализованного алгоритма. В частности, внедрение данного подхода практически не улучшило размеченную структуру выводимых RST-деревьев, что свидетельствует о «холостой» работе классификатора, ответственного за определение типов риторических связей. Также следует отметить, что простейшая реферативная эвристика, составляющая реферат текста из первых предложений каждого абзаца, в половине случаев продемонстрировала лучшее качество рефератов по сравнению с модифицированной версией системы ИСП РАН, что фактически сводит на нет какое-либо расширение области ее применения.

# Литература

1. Luhn H. P. The automatic creation of literature abstracts. // IBM Journal of research and development, 1958, pp.159-165.
2. Edmundson H. P. New methods in automatic extracting. // Journal of the ACM (JACM), 1969, pp.264-285.
3. Barzilay R. et al. Using lexical chains for text summarization. // Proceedings of the ACL workshop on intelligent scalable text summarization, 1997, pp.10-17.
4. Erkan G., Radev D. R. LexRank: Graph-based lexical centrality as salience in text summarization. // J. Artif. Intell. Res. (JAIR), 2004, Vol. 22, pp.457-479.
5. Mihalcea R., Tarau P. TextRank: Bringing order into texts. // Proceedings of EMNLP, 2004, Vol. 4, №. 4.
6. Kamp H., Reyle U. From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation. // Studies in Linguistics and Philosophy, Kluwer, 1993.
7. Webber B., Joshi A. Anchoring a lexicalized tree-adjoining grammar for discourse. // Coling/ACL workshop on discourse relations and discourse markers, 1998, pp.86-92.
8. Polanyi L. et al. A rule based approach to discourse parsing. // Proceedings of SIGDIAL, 2004, Vol. 4.
9. Lascarides A., Asher N. Segmented discourse representation theory: Dynamic semantics with discourse structure. // Computing meaning, Springer Netherlands, 2007, pp.87-124.
10. Mann W. C., Thompson S. A. Rhetorical structure theory: Toward a functional theory of text organization. // Text, 1988, Vol. 8, №. 3, pp.243-281.
11. RST Web Site: <http://www.sfu.ca/rst/index.html>
12. Литвиненко А.О. Описание структуры дискурса в рамках теории Риторической структуры: применение на русском материале. // Труды Международного семинара Диалог '2001 по компьютерной лингвистике и ее приложениям, Аксаково, 2001, Т.1, стр.159-168.
13. Rösner D., Stede M. Customizing RST for the automatic production of technical manuals. // Springer Berlin Heidelberg, 1992, pp.199-214.
14. Ono K., Sumita K., Miike S. Abstract generation based on rhetorical structure extraction. // Proceedings of the 15th conference on Computational linguistics-Volume 1, Association for Computational Linguistics, 1994, pp.344-348.
15. Marcu D. Discourse trees are good indicators of importance in text. // Advances in automatic text summarization, 1999, pp.123-136.
16. Marcu D. The Theory and Practice of Discourse Parsing and Summarization. // The MIT press, 2000.
17. Document Understanding Conferences (DUC) Web Site: <http://duc.nist.gov/>
18. Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. // Text Summarization Branches Out: Proceedings of the ACL-04 Workshop, 2004, pp.74-81. (<http://acl.ldc.upenn.edu/acl2004/textsummarization/pdf/Lin.pdf>)
19. Bosma W. E. Query-based summarization using rhetorical structure theory. // 2005.
20. Soricut R., Marcu D. Sentence level discourse parsing using syntactic and lexical information. // Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, Association for Computational Linguistics, 2003, pp.149-156.
21. Marcu D. The rhetorical parsing of natural language texts. // Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 1997, pp. 96-103.
22. Corston-Oliver S. H., Corston-oliver S. H. Beyond string matching and cue phrases: Improving efficiency and coverage in discourse analysis. // The AAAI Spring Symposium on Intelligent Text Summarization, 1998, pp.9-15.
23. LeThanh H., Abeysinghe G., Huyck C. Generating discourse structures for written texts. // Proceedings of the 20th international conference on Computational Linguistics, Association for Computational Linguistics, 2004, p.329.
24. Marcu D. A decision-based approach to rhetorical parsing. // Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics, Association for Computational Linguistics, 1999, pp.365-372.
25. Marcu D., Echihabi A. An unsupervised approach to recognizing discourse relations. // Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, 2002, pp.368-375.
26. Reitter D. Simple Signals for Complex Rhetorics: On Rhetorical Analysis with Rich-Feature Support Vector Models. // LDV Forum, 2003. Vol. 18, №. 1/2, pp.38-52.
27. Duverle D. A., Prendinger H. A novel discourse parser based on support vector machine classification. // Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2, Association for Computational Linguistics, 2009, pp.665-673.
28. Sagae K. Analysis of discourse structure with syntactic dependencies and data-driven shift-reduce parsing. // Proceedings of the 11th International Conference on Parsing Technologies, Association for Computational Linguistics, 2009, pp.81-84.
29. Hernault H. et al. HILDA: a discourse parser using support vector machine classification. // Dialogue & Discourse, 2010, Vol.1, №. 3.
30. Black E. et al. A procedure for quantitatively comparing the syntactic coverage of English grammars. // Proceedings of Workshop on Speech and Natural Language, Association for Computational Linguistics, 1991, pp.306-311.
31. Taboada M., Renkema J. Discourse Relations Reference Corpus. // 2008 (<http://www.sfu.ca/rst/06tools/discourse_relations_corpus.html>)
32. <https://en.wikipedia.org/wiki/Clause>
33. Marcu D. Building up rhetorical structure trees. // Proceedings of the National Conference on Artificial Intelligence, 1996, pp.1069-1074.
34. <http://en.wikipedia.org/wiki/Backtracking>
35. Carlson L., Marcu D., Okurowski M. E. Building a discourse-tagged corpus in the framework of rhetorical structure theory, Springer Netherlands, 2003, pp.85-112.
36. LDC – Linguistic Data Consortium – web site: <http://www.ldc.upenn.edu/>
37. Taboada M. SFU Review Corpus // 2008 (<http://www.sfu.ca/~mtaboada/research/SFU_Review_Corpus.html>)
38. O'Donnell M. RSTTool 2.4: a markup tool for Rhetorical Structure Theory // Proceedings of the first international conference on Natural language generation-Volume 14, Association for Computational Linguistics, 2000, pp.253-256.
39. Chang C. C., Lin C. J. LIBSVM: a library for support vector machines. // ACM Transactions on Intelligent Systems and Technology (TIST), 2011, Vol. 2, №. 3, p.27.