Московский Государственный Университет им. М.В. Ломоносова
Факультет Вычислительной математики и кибернетики
Кафедра Системного программирования

Дипломная работа

Исследование и реализация методов многоязыкового автоматического реферирования текстов

Выполнил:
Студент 527 группы
Павлович Борис

Научные руководители:
ак. РАН, проф. Иванников Виктор Петрович
н.с. к.ф.-м.н. Турдаков Денис Юрьевич

Москва — 2012

Содержание

Аннотация

Данная работа ставит целью исследование и разработку системы многоязыкового автоматического реферирования текстов. Для эффективной реализации требуется анализ существующих решений и подходов во многих областях обработки естественных языков:

- разделение текста на предложения и слова,
- определение языка,
- выделение ключевых фраз и слов,
- определение важных предложений,
- обработка предложений.

Основными проблемами при работе с естественным языком являются устранение лексической, синтаксической, семантической и морфологической многозначности.

Введение

Выделение главных мыслей в виде реферата или конспекта интересовало человечество со времен появления письменности. С появлением Интернета тема приобрела новую актуальность, так как нынешние объемы информации огромны и постоянно растут. Существует множество областей, в которых применение такого сокращенного представления давало бы существенный выигрыш. Например, система может быть полезна для человека, который должен по большому количеству статьей на определенную тему быстро получить представление о данной области. Автоматическое реферирование также можно использовать в поисковых системах для того, чтобы уменьшать область поиска.

Рефераты бывают нескольких типов: информативные, индикативные и критические. Индикативные рефераты должны предоставлять достаточно информации для принятия решения, стоит ли обращаться к оригиналу. Информативные рефераты должны сжимать исходный текст. Критические рефераты не только сокращают, но и дают оценку тексту. В данной системе будут рассматриваться только рефераты информативного типа.

Существует два основных подхода к автоматическому реферированию. Первый подход ориентирован на извлечение важных фрагментов, обычно предложений, так называемый sentenceextraction. Второй подход использует сложные методы семантического и лингвистического анализа, обычно это генерация рефератов (summarygeneration) на основе семантического представления текста. Ввиду сложности второго подхода, как в плане реализации, так и в плане вычислений, а также потому, что он

накладывает существенные ограничения на тексты, было принято основывать свою систему на первом подходе.

Разработанная система организована в виде веб-сервиса, позволяющего:

- определять язык,
- получить ключевые слова из текста,
- выделить в тексте основные предложения,
- создавать реферат состоящий из основных предложении.
- Указывать алгоритм реферирования, а также длину получаемого реферата.

Все алгоритмы должны корректно работать с несколькими языками.

Постановка задачи

Целью данной дипломной работы является разработка веб-сервис автоматического многоязыкого реферирования, и оценить качество работы данной системы. Для достижения данной цели необходимо решить следующие задачи:

- провести исследование целесообразности использования тех или иных технологий,
- исследование алгоритмов реферирования текстов,
- разработка архитектуры системы,
- реализация алгоритмов реферирования,
- исследование инструментов тестирования систем автоматического реферирование,
- оценка качества работы системы.

Разработка архитектуры осложняется тем, что система должна эффективно работать сразу с несколькими языками, при этом компоненты системы (определение языка, разделение текста на предложение, на токены и другие) должны быть слабосвязанными и легко заменяемыми, даже на сторонние библиотеки.

Обзор существующих решений

Задачу реализации системы многоязыкового автоматического реферирования можно разделить на ряд отдельных задач:

1) определение языка,
2) разбор текста – разделение текста по предложениям, по словам и по абзацам,
3) алгоритмы выделения и сглаживания n-grams,
4) выделение ключевых слов и фраз,
5) реферирование - выделение главных предложений и их упорядочивание и связывание.

Для решения подобных задач существует два основных подхода:

1) основанные на правилах – конечные автоматы,
2) основанные на машинном обучении – то есть использовании статистических данных, полученных из обучающей выборки.

Подходы, основанные на правилах, в настоящее время используются редко, ввиду проблем, возникающих при создании правил:

- необходимость создания ряда нетривиальных правил,
- потребность использования узкоспециальных лингвистических знаний,
- сложность, а порой и невозможность, обобщения результатов на другие языки.

Более перспективными являются методы машинного обучения: требующие лишь подготовленных для конкретной задачи, специально размеченных наборов документов.

Все методы машинного обучения можно свести к одной из трех общих задач:

- **Задача классификации** – дана обучающая выборка – набор объектов, разделенных на классы. Также дан набор объектов, для которых класс неизвестен – тестовая выборка. Требуется построить алгоритм, который определяет классы для тестовой выборки;
- **Задача кластеризации** – разделить исходное множество объектов на подмножества (кластеры), содержащие схожие объекты. При этом объекты разных кластеров должны сильно отличатся. Количество кластеров заранее неизвестно;
- **Задача идентификации** – все объекты обучающей выборки принадлежат классу А, и не существует возможности сделать репрезентативную выборку для «всех остальных». Требуется построить алгоритм, определяющий, принадлежит ли объект к классу А.

Реферирование

Реферирование текстов [1] – это процесс выделения наиболее важной информации из текста для создания новой сокращенной версии документа, исходя из конкретной цели. Основными типами резюме являются:

- основные положения любого документа,
- аннотации научных трудов,
- краткое содержание новостных рубрик,
- сниппеты – небольшие фрагменты исходного текста, содержащие слова запроса пользователя и используемые поисковыми системами для описания ссылок,

- краткое содержание email переписки,
- сжатие предложений для упрощения и сокращения размера текста,
- генерация ответов на сложные вопросы при помощи краткого содержания нескольких документов.

В данной работе основными являются два подхода к реферированию: упрощение предложений и выделение основных наиболее информативных предложений.

Для каждого из этих типов реферирования существуют два важных параметра:

- количество документов, подаваемых на вход: один или несколько. В случае систем получающих на вход один документ, обычно не требуется упорядочивание информации, соответственно результатом работы является резюме содержащие предложения исходного документа с сохраненным порядком. Системы получающие на вход несколько документов связаны обычно с новостными рубриками и интернет ресурсами, где по большому количеству документов на одну тему требуется получить резюме, при этом приходится решать проблемы упорядочивания информации и разрешения конфликтов,
- тип реферирования: общее реферирование или исходя из запроса пользователя. В случае общего реферирования не делается предпочтение какой либо информации из предоставленного документа. В случае когда есть запрос для определения важности части информации документа используются также слова из запроса. Так можно например добавить к запросу пользователя синонимы всех слов, и

оставлять только предложения содержащие слова из этого набора.

Два основных подхода решения задачи реферирования

Существует два ключевых подхода к решению задач реферирования:

- аннотация – используется алгоритмы генерации текста, для создания нового текста, который передает суть исходного документа в сокращенной форме,
- извлечение – простейший способ реферирования с помощью комбинации фраз и предложений, полученных из документов.

Главными задачами современных исследований является разработка методов аннотационного реферирования, а также поиска признаков для методов машинного обучения.

Способ описания систем реферирования

Системы реферирования и системы генерации текстов, как правило, описываются используемыми подходами к решению трех задач:

- определение содержания – выделение важных единиц: фраз, предложений, параграфов из текста,
- упорядочивание информации – сортировка и структуризация извлеченных единиц,
- обработка предложений – какими методами происходит очистка и упрощение выделенных единиц.

Реферирование одного документа

Задача состоит в том, чтобы по исходному документу получить краткое содержание, то есть формально:

- Определение содержания – выделяем важные (информативные) предложения из текста
- Упорядочивание информации – определение порядка предложении. Данная часть сводится обычно к сохранению такого же порядка следовании предложений как в исходном документе
- Обработка предложении – Упрощение и согласование предложений, например удаление несущественных частей предложений, или разрешение проблем несогласованности.

Методы реферирования «без учителя»

Данная задача относится к задаче классификации, то есть разделяет все предложения на два класса:

- важные, попадающие в резюме,
- не информативные, не попадающие в резюме.

Под понятием «без учителя» - имеется в виду то, что данные методы не требуют специально составленных коллекций документов, в которых все важные предложения помечены меткой «ВАЖНО».

Самые ранние методы реферирования [2] были основаны на интуитивном предположении о том, что предложение является более важным, если оно содержит более важные и информативные слова. Важность слова определяется исходя из его частоты в документе, чем чаще оно употребляется, тем оно важнее.

Этот метод не используется напрямую ввиду того, что существуют слова с высокой частотой, но при этом не несущие полезной информации, например частицы и союзы. Поэтому используется один из нескольких подходов:

- tf-idf – увеличивает значимость слов, которые часто встречаются в документе, но редко в обучающем корпусе. $weight(w) = tf_{i,j} * idf_i$, где $tf_{i,j}$ – частота термина в текущем документе, idf_i - логарифм отношения количества всех документов к количеству документов содержащих данный термин;
- λ - отношение правдоподобия (log-likelihoodratio) является логарифмом отношения вероятности наблюдения слова с одинаковой вероятностью в корпусе исходных документов и корпусе советующих им резюме, к вероятности появления слова с разными вероятностями в этих корпусах. В статье [9]предложили использовать для расчета веса слова формулу: $\begin{cases} 1 \; if - 2 * \log(\lambda(w_i)) > 10 \\ 0 \; else \end{cases}$. Для оценки веса предложения $weight(s_i) = \sum_{w \in s_i} \frac{weight(w)}{|\{w | w \in s_i\}|}$.В резюме попадают предложения с более высоким весом;
- метод основанный на центрировании заключается в вычислении расстояний между предложениями и в выборе тех из них, которые в среднем ближе к другим. Для определения близости предложений обычно используются алгоритмы, основанные на наборах слов, содержащихся в предложении (bag-of-words). Пример определения самых близких в среднем предложений:

- вычислить близость между всеми парами предложений, то есть косинус угла между векторамиих слов, если вектора совпадают то результат равен 1, если вектора ортогональны то он равен 0:

$$cosine(i,j) = \frac{\sum_{w \in q,d} tf_{w_i} * tf_{w_j} * (idf_w)^2}{\sqrt{\sum_{w_i \in i}(tf_{w_i} * idf_{w_i}) * \sum_{w_j \in j}(tf_{w_j} * idf_{w_j})}}$$

- для каждого предложения посчитать среднююблизость к остальным предложениям,
- отсортировать и выбрать те у которых близость максимальна.

- симметричное реферирование [3] – подход, в котором вес предложения вычисляется как функция от количества связей с другими предложениями. Связь – это наличие одного и того же ключевого слова в двух предложениях. При этом учитываются словоформы ключевых слов.

Методы реферирования «с учителем»

Под этим определением понимается, что существует набор документов, в котором важные, попадающие в реферат, предложения помечены меткой. К данному набору документов применимы методы машинного обучения, состоящие из двух этапов:

- имеется простая выборка $X^l = (x_i, y_i)_i^l$ из неизвестно распределения $p(x,y) = P_y * p_y(x)$. Требуется построить эмпирические оценки априорных вероятностей $\check{P}_y$ и функции правдоподобия $p_{y(x)}$ для каждого из классов $y \in Y$,
- по известным плотностям распределении $p_{y(x)}$ и априорным вероятностям $\check{P}_y$ всех классов $y \in Y$ построить

алгоритм,минимизирующий вероятность ошибочной классификации.

В данной работе рассматриваются 3 разных подхода:

- дерево принятия решений,
- наивный Байесовский классификатор,
- метод опорных векторов.

Несмотря на разнообразие методов, основными характеристиками, влияющими на качество классификации, являются:

- качество обучающей выборки,
- выбранные признаки.

Наиболее употребляемыми признаки являются:

- расположение в тексте: предложения в начале или в конце текста являются более информативными;
- длина предложения: слишком короткие или слишком длинные предложения являются скорее всего мало информативными;
- наличие сигнальных фраз: наличие словосочетаний: «в данной работе», «в статье», «в заключение»;
- наличие слов из заголовка: если предложение содержит слова из заголовка, то значит, что оно относится к данной теме;
- наличие слов сравнения: «более», «менее», «лучше» и т.д;
- наличие ключевых слов: повышает интерес к предложению;
- другие статистические классификаторы, не требующие обучения.

Одной из основных проблем машинного обучения является то, что для данных алгоритмов требуется обучающая выборка, то есть набор документов в которых помечены предложения, которые попадают в резюме. То есть вручную сделанные рефераты, содержащие: объединения предложении, переформулированные фразы и добавление предложения не подходят. Если бы удалось убрать это ограничение, то можно было бы воспользоваться данными алгоритмами к широкому спектру областей, используя уже создание вручную рефераты.

Разделение текста на предложения

Разбор текста заключается в отображении исходного текста в удобном виде для последующей обработки, в виде набора абзацев, предложений, слов. Во время отображения исходного текста в нужном виде возникают неоднозначности, которые нужно разрешать. В качестве примера можно привести разделение следующего текста на предложения: “Мы выехали из города N. вчера вечером.” Основной вопрос, является ли точка после «N» концом предложения или нет?

Современные подходы

Для выделения предложений и слов используется алгоритм, основанный на отношении правдоподобия. Он подробно описан в [4]. Основная проблема при разделении текста на предложения заключается в том, что точка после предложения может иметь разный смысл: это может быть аббревиатура (или порядковый номер), либо конец предложения, либо и то и другое.

Алгоритм состоит из двух этапов:

1) Разметка аннотациями <A> - аббревиатур, <E> - многоточия, <S> - концов предложений.

2) Исправление ошибок:
    a. <A> -><A><S> обнаружение аббревиатур в конце предложения.
    b. <E> -><E><S> обнаружение многоточия в конце предложения.
    c. <S> -><A> (<A> -><A>) обнаружение точки внутри аббревиатуры.
    d. <S> -><A> обработка порядковых номеров и инициалов.

Оба этапа используют отношение правдоподобия – отношение вероятности получить положительный результат для положительного исхода к вероятности получить положительный результат для отрицательного исхода.

На первом этапе нужно выяснить, является ли слово аббревиатурой. Для этого используется эвристика о том, что после аббревиатур всегда стоит точка. Если встречаем слово, после которого часто стоит точка, то оно скорее всего является аббревиатурой. В статье [5] предлагают воспользоваться отношением правдоподобия с нулевой гипотезой:

$$H_0 : P(.|w) = \frac{C_w(.)}{N_w},$$

$C_w(.)$ – сколько раз после слова встречается ".",

N_w – количество слов w в корпусе,

и альтернативной гипотезой: $H_A : P(.|w) = 0.99$. Тогда отношение правдоподобия принимает вид:

$$\log(\lambda) = -2 * \log\left(\frac{H_0}{H_A}\right).$$

Для улучшения коэффициента правдоподобия используются 3 эвристики ($length(w)$ – длинна слова без учета внутренних точек):

- $F_{length}(w) = \frac{1}{e^{length(w)}}$ - чем длиннее слово, тем меньше вероятность, что оно окажется аббревиатурой,
- $F_{periods}(w) = количество\ внутренних\ точек + 1 - -$ если есть внутренние точки, то сильно возрастает вероятность того, что слово является аббревиатурой, при этом отсутствие точек не должно менять эту вероятность,
- $F_{penalty}(w) = \frac{1}{length(w)^{C(w,\neg.)}}$ - чем длиннее слово и чем чаще после него не стоит точка, тем меньше вероятность, что слово является аббревиатурой.

В [5] предлагается считать слово аббревиатурой если:

$$\log(\lambda(w)) * F_{length}(w) * F_{period}(w) * F_{penalty}(w) \geq 0.3$$

На втором этапе нужно выяснить, являются ли выделенные аббревиатуры и многоточия концом предложений.

Для аббревиатур длиннее 1 символа а также многоточий используются ортографические эвристики. Для определения, является ли точка концом предложения, используются два признака:

1. Отношение правдоподобия, в котором в виде нулевой гипотезы используется $H_0: P(w|S) = P(w|\neg S)$, в качетсве альтернативной гипотезы $H_A: P(w|S) = p_1 \neq p_2 = P(w|\neg S)$.
2. Пусть N – количество слов в корпусе, C(*) – количество элементов в корпусе. Тогда<w1,w2>является сочетанием слов, если $\frac{C(w1,w2)}{C(w2)} \geq \frac{C(w2)}{N}$.

Тогда точка является концом предложения, если $\log(\lambda) > const$ и выполняется то что $\frac{C(S,w2)}{C(S)} \geq \frac{C(w2)}{N}$.Данный метод дает точность порядка 98.3% точности, что является хорошим результатом.

Сравнение методов выделения предложений и слов

Алгоритм, основанный на регулярных выражениях, сканирует документ до обнаружения точек, а потом сравнивает окружающие токены с регулярными выражениями. Если ни одного регулярного выражения не обнаружено, то точка считается концом предложения. Точность данного алгоритма меньше, и при этом требует множества вручную составленных регулярных выражений (порядка 300 для одного языка), что также не подходит для решения данной задачи, ввиду требования многоязычности системы.

Дерево решения [6]. Имеет точность на корпусе (Brown) 99.8% что выше точности выбранного алгоритма, используются признаки:

- P[слово с «.» является концом предложения],
- P[слово после «.» является началом предложения],
- Длинна слова с «.»,
- Длинна слова после «.»,
- С чего начинается следующее слово (Большая(маленькая) буква, число),
- Является ли из класса аббревиатурой (названия месяца, мера или например адрес).

Существенным недостатком является требования к большому обучающему корпусу, для каждого языка, при этом такие корпуса отсутствуют для большинства языков.

Алгоритм, основанный на машинном обучении [7], рассматривающий вероятностное распределение для слов вокруг точек, а также разные эвристические данные.Имеет такую же точность как выбранный алгоритм, но требует обучающих корпусов.

N-граммы

N-gram – подпоследовательность длины N из элементов некоторой последовательности. Примеры N-грамм слов:

- Юниграммы - машина, холодильник, кровать...
- Биграммы - красивая машина, большой холодильник ...
- Триграммы - я хороший студент, долго и нудно...

В виду того, что язык не является случайным набором слов, N-граммы являются хорошей характеристикой текстов и языка. Многие алгоритмы обработки естественных языков используют в качестве базы N-граммы:

- определение языка: методы, основанные на использовании N-грамм букв дают большую точность (google langdetect);
- генерация текста: последовательность N-грамм, таких, что конец i-ой N-граммы является началом i+1 N-грамм, является синтаксически связанным текстом;
- поиск семантических ошибок: если слово в данном контексте не употребляется, то вероятность встретить N-грамму, содержащую данное слово в данном контексте будет маленькой (или 0), что позволяет делать вывод о семантической ошибке.

Для решения перечисленных задач создается языковая модель N-грамм. Языковая модель представляет из себя множество парвида

<N-грамма, вероятность ее появления>. При использовании созданной модели возникает проблема из-за разреженность N-грамм, то есть отсутствие многих N-грамм в обучающем корпусе. Например, если рассчитывать вероятность появления предложения как произведение N-грамм данного предложения, то часто будут возникать предложения с нулевой вероятность. Для разрешения этой проблемы используются специальные сглаживающие алгоритмы [8]:

- Add-one smoothing
- Witten-Bell discounting
- Good-Turing discounting
- Backoff smoothing

Методы сглаживания N-грамм моделей

Обозначения используемые в описании алгоритмов:

- V - количество уникальных слов (токенов),
- N - количество всех слов (токенов),
- C(w) - количество слов w.

Add-one smoothing

Ко всем N-граммам добавляется единица и пересчитывается вероятность:

$$p(w) = \frac{C(w) + 1}{N + V}$$

Метод вызывает сильную погрешность в вычислений, тесты показали что не сглаженная модель часто дает более точные результаты, чем сглаженная по этому методу. Поэтому данный подход не используется на практике.

Witten-Bell Discounting

Для неизвестных N-грамм используется контекст, в котором они находятся, то есть сколько N-грамм заканчивается на определенное слово.

В случае биграмм:

$$p^*(w_i|w_x) = \frac{T(w_x)}{Z(w_x)(N(w_x) + T(w_x))}, c(w_x w_i) = 0$$

,где $Z(w_x) = \sum_{i:c(w_x w_i)=0} 1$

$$p^*(w_i|w_x) = \frac{C(w_x w_i)}{C(w_x) + T(w_x)}, C(w_x w_i) > 0$$

Метод показывает хорошие результаты, но остается проблема с тем, что если вероятность

Good-Turing Discounting

Идея в том, что для N-грамм которые встретились ноль раз (с раз) оценка пропорциональна количеству N-грамм, встретившихся один раз (c+1 раз):

$$c^* = (c+1)\frac{N_{c+1}}{Nc}, Nc = \sum_{b:count(b)=c} 1$$

В случае если Nc = 0 то используется другая формула:

$$\log(Nc) = a + b * \log(c)$$

Katz'sbackoff

Основная идея метода в том, что можно оценивать вероятности N-грамм с помощью вероятностей (N-k)-грамм. Особенностью данного алгоритма является то, что его можно сочетать с другими алгоритмами.

Стемминг

Стемминг – процесс нахождения основы слова, то есть неизменяемой части слова, которая выражает его лексическое значение. Основа не обязательно совпадает с морфологическим корнем слова. Обычно стемминг используется в поисковых машинах для обобщения поискового запроса пользователя. Многие алгоритмы реферирования используют частоту слов как признак, этот признак дает более точные результаты, если учитывать все словоформы слова как одно слово.

Одним из способов стемминга является отсечение от слова суффиксов и окончаний, для того, чтобы оставшаяся часть слова была одинакова для всех грамматических форм слова. Стеммер такого вида может работать лишь с языками, реализующими изменение слов через аффиксы, то есть такие, как например русский, английский, немецкий и тд. При использовании такого стеммера возникает множество ошибок. Ошибки можно классифицировать:

- 1-ого рода – стем делает слишком большое обобщение, например, в случае с существительным «кошка» общим префиксом всех форм будет «кош», что при префиксном поиске будет также выдавать слово «кошмар»,
- 2-ого рода – стем оставляет слишком длинную общую часть, которая не совпадает со всеми формами слова,
- 3-ого рода – нельзя построить стем из-за изменения в корне, которое оставляет в стеме единственную букву или из-за использования приставок.

Системы реферирования

На рынке присутствуетне так много систем автоматического реферирования и аннотирования среди них стоят упоминания:

- AutoSummarize функция Microsoft Word,
- IBM Intelligent Text Miner,
- Oracle Context и Inxight Summarizer,
- Smart Search System (3S),
- Extractor,
- Prosum,
- МЛ Аннотатор.

Архитектура системы

Задачу автоматического резюмированния можно условно разделить на две основные подзадачи:

- обучение алгоритмов реферирования,
- оценка важности предложений текста.

При разработке архитектуры системы в первую очередь ставилась задача упрощения автоматического тестирования большого количества алгоритмов оценки важности предложений. По этому система является слабосвязанной состоящей из отдельных слабосвязанных компонентов:

- определения языка,
- выделение предложений и слов из текста
- стеммеров,
- оценки важности предложений,
- модуль для работы с материалами из DUC,
- модуль для работы с Rouge.

Эти компоненты используют 2 высокоуровневых приложения:

- приложение для тестирования алгоритмов оценки важности предложений с помощью DUC и Rouge:
    - получить корпуса для обучения и тестирования из DUC
    - обучить тестируемые алгоритмы (на обучающем корпусе DUC)
    - получить рефераты
    - подготовить данные для утилиты Rouge
- веб сервис позволяющий создавать рефераты:
    - определение языка текста
    - разделение текста на предложения и слова

- применение одного из алгоритмов оценки важности предложений,
- выделения запрошенного количества предложений исходя из запроса пользователя и оценки,
- Возвращаем размеченный документ, содержащий либо только извлеченные предложения, либо исходный документ с выделенными важными предложениями.

В данной системе используются сторонние библиотеки:

- Google langdetect в качестве компонента для определения языка, преимуществами являются большое количество поддерживаемых языков, высокая точность и простой программный интерфейс,
- NLTK WordPunktTokenizer и PunktSentenceTokenizer - для выделения предложений и слов,
- NLTK PorterStemmer - для стемминга,
- scipy.linalg.svd - для сингулярного разложения матриц,
- NLTK NaiveBayesClassifier - реализация наивного Байесовского классификатора.

Оценка важности предложения является основной частью системы. Каждый алгоритм представляет собой класс унаследованный от базового абстрактного класса содержащего всего два метода:

- возвращающий текст из предложений с максимальной оценкой до N слов,
- абстрактный метод, возвращающий оценки для предложений.

Оба метода на вход получают список предложений текста, представляющий из себя кортежи вида (исходное предложение, список слов предложения). Данная организация удобна тем, что позволяет создавать линейные комбинации из алгоритмов оценки важности предложений.

В случае с алгоритмами требующих обучения с учителем, таким как наивный Байесовский классификатор, отедельно передается объект умеющий по предложению получить вектор признаков. Это добавляет гибкости решения, позволяя легко одни и те же признаки использовать в разных классификаторах.

Из особенностей архитекутры, стоит также упомянуть, что nltk стеммер не используется напрямую, ввиду того, что он работает медленно. На его основе создан новый класс, который реализует такой же интерфейс, только перед стеммингом проверяет наличие данного слова в собственно кеше, в котором находятся все ранее обработные слова. Кеш в данном случае многократно ускоряет работу стеммера, особенно при больших текстах.

Оценка качества систем реферирования

Задача тестирования систем является чрезвычайно сложной. Тестирование систем вручную требует много времени и усилий, тестирование одной системы может занимать порядка трех тысяч часов. Следовательно, данные подходы крайне неэффективны, и требуется разработка систем автоматического оценивания разнообразных алгоритмов.

Разработка системы автоматического тестирования осложняется тем, что не существует общего алгоритма оценки резюме, исходя из конечного набора признаков и правил. Если бы существовал подобный алгоритм, то не существовало бы задачи реферирования. Современные подходы к оцениванию основываются на сравнении полученного резюме с несколькими модельными, вручную созданными резюме. Задача делится на два этапа:

1) исследование и разработка методов сравнения резюме с модельными резюме;
2) создание набора модельных резюме (корпусов).

Система тестирования Rouge

Rouge – это система тестирования, которая является одной из лучших на сегодняшний день. Данная система содержит много метрик: “Rouge-N”, “Rouge-L”, “Rouge-W”, “Rouge-S”, “Rouge-SU”. Все метрики основаны на идее максимального покрытия тестируемыми резюме модельных и, наоборот, при этом во всех методах используются N-граммы. В качестве модельных резюме берутся в ручную написание аннотации.

Rouge-N

Rouge-N – основывается на вычислении совпадающих N-грамм из полученного резюме и модельных резюме. N – это длина используемых N-грамм. Оценка резюме вычисляется по следующей формуле:

$$Score_{Rouge-N}(C, Ref) = \frac{\sum_{S \in \{Ref\}} \sum_{gram_n \in S} count_{match}(gram_n)}{\sum_{S \in \{Ref\}} \sum_{gram_n \in S} count(gram_n)}$$

C – тестируемое резюме,

Ref – модельное резюме,

$count_{match}(gram_n)$, количество общих N-грамм.

В случае, когда существует несколько модельных резюме, используется другая формула:

$$ScoreMulti_{Rouge-N} = argmax_i Score_{Rouge-N}(C, R_i)$$

Данный подход не является идеальным, что видно из следующего примера:

1) C : «police killed the gunman.»,
2) R1 : «police kill the gunman.»,
3) R2 : «the gunman kill police.».

$$Score_{Rouge-2}(C, R1) = \frac{1}{2},\ Score_{Rouge-2}(C, R2) = \frac{1}{2}$$

Предложения R1, R2 имеют одинаковую оценку, хотя по смыслу очень сильно отличаются. Следующий метод Rouge-L, основанный на самой длинной цепочкеулучшает ситуацию.

Rouge-L

Последовательность $Z = [z1, z2, \dots, zn]$ является подпоследовательностью другой последовательности $X = [x_1, x_2, .., x_m]$, если $\exists [i_1, i_2, \dots, i_k]: j = \overline{1..k} \Rightarrow x_{i_j} = z_j$

В отличие от Rouge-N,в данном методе нас интересует не совпадение подряд идущих слов двух предложении, а самая длинная подпоследовательность этих предложении, то есть совпадение слов с возможными пропусками.

$$Score_{Rouge-L}(C, Ref) = \frac{\sum_{r \in \{Ref\}} LCS_{\cup}(r, C)}{M}$$

$LCS_{\cup}(r, C)$ длина объединенных самых длинных общих подпоследовательсное между модельным предложением и предложениями оцениваемого резюме. M – это количество слов во всех модельных резюме. Например, если $r_1=w_1w_2w_3w_4w_5$, $c_1=w_1w_2w_6w_7w_8$, $c_2=w_1w_3w_8w_9w_5$, то самая длинная общая подпоследовательность r_1и c_1 "w_1w_2", а r_1 и c_2 "$w_1w_3w_5$". Следовательно объединением общих подпоследовательностей является $w_1w_2w_3w_5$ $LCS_{\cup}(r_1, \{c_1, c_2\}) = {}^{4}/_{5}$.

Если рассмотреть предыдущий пример, то результаты Rouge-L для R1 и R2 будут соответственно ¾ и ½.

1) C : «police killed the gunman.»
2) R1 : «police kill the gunman.»
3) R2 : «the gunman kill police.»

Rouge-W

Rouge-W представляет из себя улучшению версию Rouge-L.Например, если X = ABCDEFG, Y1 = ABCDHIK, Y2 = AHBKCID, то оценка Rouge-L одинакова для (X,Y1) и (X, Y2). При этом более желательным является предложение Y1, так как Y1 непрерывно совпадает с X. Чтобы учесть этот факт, сперва вводится функция относительно веса $f: f(x + y) > f(x) + f(y), x, y > 0$. Вес это длина

самой длинной непрерывной общей подпоследовательности. Формула оценки принимает вид:

$Score_{Rouge-W}(f^{-1}(\frac{WLSC(X,Y)}{f(m)}))$, где WLSC(X,Y) получается из LSC(X,Y) + f(длина самой длинной непрерывной подпоследовательности).

Rouge-S

Данный метод основывается на оценке при помощи использования совпадений биграмм с пропусками.

Биграммы с пропусками – это любая пара элементов последовательности, у которых сохранен порядок следования. Например для предложения “Я еду сегодня на работу” биграммами будут: «Я еду», «Я сегодня», «Я на», «Я работу», «еду сегодня», «еду на», «еду работу», «сегодня на», «сегодня работу», «на работу». Формула оценки принимает вид:

$Score_{Rouge-S}(C, Ref) = \frac{SKIP2(X,Y)}{C(m,2)}$, где SKIP2(X,Y) – количество совпадающих биграмм с пропусками, а$C(m,n) = \frac{m!}{(m-n)!n!}$

Ввиду того, что возможны синтаксические неверные последовательности: «что что», «в от», вводится ограничение на максимальное расстояние между элементами.

Rouge-SU

Если рассмотреть два предложения: «ABCDEF» и «FEDCBA» то оценка Rouge-S будет 0. Получается, что предложения совсем разные, хотя и состоят из одних и тех же слов. Для того чтобы учитывать эту особенность в методе Rouge-SU к оценке Rouge-S добавляется оценка Rouge-1.

Корреляция между оценками экспертов и Rouge

Определить, насколько хорошо работает тот или иной метод Rouge, можно, сравнивая оценки, выдаваемые Rouge с оценками экспертов. В таблице приведена корреляция между оценками экспертов и методами Rouge.

	DUC 2001 100 WORDS SINGLE DOC						DUC 2002 100 WORDS SINGLE DOC					
	1 REF			3 REFS			1 REF			2 REFS		
Method	**CASE**	**STEM**	**STOP**	**CASE**	**STEM**	**STOP**	**CASE**	**STEM**	**STOP**	**CASE**	**STEM**	**STOP**
R-1	0.76	0.76	0.84	0.80	0.78	0.84	0.98	0.98	**0.99**	0.98	0.98	**0.99**
R-2	0.84	0.84	0.83	**0.87**	0.87	0.86	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
R-3	0.82	0.83	0.80	0.86	0.86	0.85	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
R-4	0.81	0.81	0.77	0.84	0.84	0.83	**0.99**	**0.99**	0.98	**0.99**	**0.99**	**0.99**
R-5	0.79	0.79	0.75	0.83	0.83	0.81	**0.99**	**0.99**	0.98	**0.99**	**0.99**	0.98
R-6	0.76	0.77	0.71	0.81	0.81	0.79	0.98	**0.99**	0.97	**0.99**	**0.99**	0.98
R-7	0.73	0.74	0.65	0.79	0.80	0.76	0.98	0.98	0.97	**0.99**	**0.99**	0.97
R-8	0.69	0.71	0.61	0.78	0.78	0.72	0.98	0.98	0.96	**0.99**	**0.99**	0.97
R-9	0.65	0.67	0.59	0.76	0.76	0.69	0.97	0.97	0.95	0.98	0.98	0.96
R-L	0.83	0.83	0.83	0.86	0.86	0.86	**0.99**	**0.99**	0.99	**0.99**	**0.99**	**0.99**
R-S*	0.74	0.74	0.80	0.78	0.77	0.82	0.98	0.98	0.98	0.98	0.97	0.98
R-S4	0.84	**0.85**	0.84	**0.87**	**0.88**	**0.87**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
R-S9	0.84	**0.85**	0.84	**0.87**	**0.88**	**0.87**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
R-SU*	0.74	0.74	0.81	0.78	0.77	0.83	0.98	0.98	0.98	0.98	0.98	0.98
R-SU4	0.84	0.84	**0.85**	**0.87**	0.87	**0.87**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
R-SU9	0.84	0.84	**0.85**	**0.87**	0.87	**0.87**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
R-W-1.2	**0.85**	**0.85**	**0.85**	**0.87**	0.87	**0.87**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**

Из приведенной таблицы следует, что алгоритмы Rouge-SU и Rouge-W позволяют достаточно точно оценивать качество резюме. Также стоит обратить внимание на Rouge-2, он имеет высокую точность при низкой вычислительной сложности, что делает его привлекательным при оценке большого числа резюме.

Материалы DUC

DUC (Document Understanding Conference) - это проект организации NIST, представляющий начиная с 2001года, материалы для обучения и тестирования систем автоматического реферирования.

Содержание DUC 2001

DUC 2001 это набор документов, содержащий в себе корпуса для обучения и тестирования алгоритмов реферирования и аннотирования.

Корпус для обучения состоит из примерно 30 кластеров, каждый кластер содержит:

- около 10 новостных сообщении на одну тему в формате xml,
- аннотации по кластеру длинной в 100, 200, 300 и 400 слов для многодокументного реферирования,
- 100 словные аннотации для каждой новости из кластера.

Корпус для тестирования также содержит около 30 кластеров, только для каждого новостного сообщения соответствуют три вручную написанных аннотации.

Ввиду того, что аннотации не являются набором предложении исходных документов, то возникает сложность с применением алгоритмов машинного обучения.

Эксперименты

В данной части проводится сравнение алгоритмов реферирования по основным характеристикам:

- точность – насколько хорошо модельные резюме покрывают полученное резюме. В качестве оценки используется Rouge Precision,
- полнота – насколько хорошо полученное резюме покрывает модельные. Используется Rouge Recall,
- F-мера - среднее гармоническое от точности и полноты.
- скорость – время требуемое для получения резюме. Имеется ввиду сколько времени необходимо для создания всех резюме из тестового набора DUC 2001,

Интерес также представляет влияние использования стемминга и tf-idf в разных алгоритмах.

Все системы обучаются и тестируются на материалах DUC 2001, при этом тестирование происходит с помощью использования алгоритмовRouge-2 и RougeSU*.

Алгоритмы машинного обучения требуют специально подготовленной обучающей выборки, то есть специально размеченные документы в которых указаны какие предложения попадают в резюме. К сожалению в материалах DUC 2001 доступны только аннотации в ручную написание с использование других слов, переформулированных фраз, объединении предложении и прочих методов аннотирования. По этому первым этапом будет получение специально размеченных документов из DUC 2001. Для этого применяется алгоритм центрирования, при этом самые близкие предложения ищутся не относительно входного документа, а относительно готового модельного резюме.

Получившийся текст имеет оценку Rouge-SU* F=0.23, следовательно имеет смысл взять в качестве максимально верхней оценки именно это число.

Метод «Нижняя граница»

В качестве нижней границы обычно используется алгоритм возвращающий первые 100 слов. Данный алгоритм является весьма эффективным, так как требует наименьшего количества вычислительных ресурсов, не требует моделей. При этом его точность и полнота зачастую превышает сложные алгоритмы. РезультатыRouge: R=0.12964 / P=0.15944 / F=0.14132

Методы «без учителя»

«Позиция»

Данный алгоритм основывается на том, что вводится оценка предложения исходя из его расположения в тексте.

Формула имеет вид: $Si = \frac{CountOfSentence - i}{CountOfSentence}$. Данный алгоритм выдает такой же результат как и «Нижняя граница», но при этом данный алгоритм позволяет сочетать его с другими алгоритмами.

TF+IDF

Алгоритм основывается на учете частоты слов(TF).При этом для того, чтобы не учитывать слова которые встречаются во всех документах используется обратная частота документа IDF, которая равна логарифму отношения количества документов к количеству документов в которых они встретились. Частота слова рассматривается относительно конкретного документа. Алгоритм состоит из нескольких пунктов:

1) Рассчитываем idf обучающей выборки :
   a. Разбиваем все тексты на предложения

b. Каждое предложение разбиваем на слова
c. Удаляем все стоп-слова
d. Применяем к оставшимся стеммер
e. Подсчитываем количество предложений (N)
f. $idf_w = \log\left(\frac{N}{tf_w}\right)$

2) Подсчитываем для каждого слова из текста частоту tf_w.
3) $\text{Score}(S_i) = \sum_{\text{w in } S_i} \text{tf}_\text{w} * \text{idf}_\text{w}$
4) Предложения с максимальной оценкой остаются в резюме

TF+IDF + Позиция

Используются сразу два алгоритма Позиция и TF+IDF в виде линейной комбинации.

Центрирование

Алгоритм основывается на предположений о том, что самые важные предложения являются самыми близкими в среднем к другим в данном тексте. Под расстоянием между предложением предполагается, совпадение слов двух предложений. То есть если нету совпадающих слов то близость 0, если все слова совпадают то 1.

1) Аналогично рассчитывается td-idf
2) Каждое предложение представляем в виде вектора слов:
   a. Стоп-слова игнорируются
   b. Перед добавлением слов, применяется стемминг
3) $cosine(i,j) = \frac{\sum_{w \in q,d} tf_{w_i} * tf_{w_j} * (idf_w)^2}{\sqrt{\sum_{w_i \in i}(tf_{w_i} * idf_{w_i}) * \sum_{w_j \in j}(tf_{w_j} * idf_{w_j})}}$
4) $Score(S_i) = avg\left(\sum_{j=\overline{1..N}, j\ !=i} cosine\left(S_i, S_j\right)\right)$

Центрирование + Позиция

Используются сразу два алгоритма Центрирование и позиция в виде линейной комбинации.

Сингулярное разложение матриц

Алгоритм состоит в составлении матрицы строкам которой соответствуют слова, а столбцам предложения текста, значение $a_{ij} = \begin{cases} 1, word_i \in sentence_j \\ 0, else \end{cases}$. После составления таблицы производится сингулярное разложение этой матрицы $A = U\Sigma V^T$, где в строках U - информативность слов, в столбцах V^T - информативность предложений. Соответственно важность предложения рассчитывается из формулы: $Score(S_i) = \sqrt{\sum_k (\sigma_{kk} \upsilon_{ik})^2}$

Методы «С учителем»:

Под методами с учителем, имеются ввиду, методы обучающийся на заранее подготовленных наборах документов, для которых известно какие предложения попадают в резюме, а какие нет. Использование данных алгоритмов состоит из двух этапов:

- обучение, на вход подается список кортежей (Вектор Признаков, Метка). Вектор Признаков - это вектор i - значение которого соответствует тому обладает ли предложение признаком i. Например находится ли данное предложение в первых 5 предложениях данного текста. Метка - попадает ли предложение в реферат. Результатом данного этапа является, подготовленный классификатор, определяющий вероятность попадания предложения в реферат исходя из вектора признаков данного предложения,
- реферирование, на вход подается текст, который в последствий разбивается на набор предложений. Для каждого предложения рассчитывается вектор признаков. С помощью

классификатора и данных векторов признаков рассчитывается оценка каждого предложения, как вероятность попадание предложения в реферат. В качестве предложений попадающих в реферат, берутся предложения с максимальной оценкой.

Классификатор

В качестве классификатора был выбран наивный Байесовский классификатор. Среди его преимуществ является, быстрая работа алгоритма, как обучения так и классификаций, при этом реализация является достаточно простой.

Признаки

- «Позиция предложения»:
  - На 1, 2, 3, 4, 5, ..., 15 месте
  - В первых 5, 10,15,20 предложениях
  - После 20, 25, ..., 50 предложений
- «Имена»
  - Нету имен в предложении
  - 1,2,3,4 или более 5 имен в предложений
- Слова
  - содержится ли слово(для всех слов из обучающей выборки) в предложений

Результаты

	R	P	F
«Нижняя граница»	**0,14433**	**0,15082**	**0,14604**
Позиция	0,14433	0,15082	0,14604
TF+IDF	0,13462	0,13021	0,13108
TF+IDF + (длина < 3)	0,13629	0,13251	0,1329
TF+IDF + (список стоп-слов)	0,13618	0,12911	0,13131
TF+IDF + (список стоп слов и длина <3)	**0,13835**	**0,13105**	**0,13332**
TF+IDF + стеммер	0,13061	0,12667	0,12735
TF+IDF + стеммер + (стоп-слова)	0,13671	0,13089	0,13251
Центрирование	0,10761	0,12292	0,11325
центрирование + TF+IDF	0,12345	0,12549	0,12316
центрирование TF+IDF + стоп-слова	0,12508	0,12595	0,12424
центрирование + стоп-слова	0,13326	0,13573	0,13309
центрирование + стеммер	0,11231	0,12857	0,11839
центрирование + стеммер + стоп-слова	**0,13723**	**0,13959**	**0,13711**
центрирование TF+IDF + стоп-слова + стеммер	0,12651	0,12802	0,12605
0.65*Позиция + 0.35*(центрирование + стеммер + стоп-слова)	**0,15656**	**0,16040**	**0,15703**
0.65 *Позиция + 0.35*(TF+IDF+стоп-слова)	**0,15427**	**0,15202**	**0,15196**
сингулярное разложение	0,10533	0,10945	0,10641
сингулярное разложение + стоп-слова	0,11722	0,11001	0,11269
сингулярное разложение + стоп-слова + стеммер	**0,11739**	**0,11068**	**0,11311**
наивный Байесовский кл. (слова)	0,12430	0,11584	0,11894
наивный Байесовский кл. (позиция)	**0,14375**	**0,15033**	**0,14550**
наивный Байесовский кл. (имена)	0,10299	0,11119	0,10598
наивный Байесовский кл. (позиция + имена)	0,14403	0,14992	0,14544

Из приведенной таблицы видно, что удаление стоп-слов всегда приводит к улучшению результатов работы алгоритмов.

Использование по отдельности tf-idf и стемминга также всегда приводят к улучшению ситуации, при этом совместное их использование приводит к ухудшению результатов.

Одни из ключевых признаков является позиция предложения относительно начала текста, чем ближе к началу находится предложение, тем информативность данного предложения выше. Данное утверждение подтверждается и наивным Байесовским классификатором, с признаками позиций, который работает почти идентично алгоритму нижней границы.

Заключение

В работе рассматриваются методы автоматического многоязыкового реферирования. Разработана декомпозиция задачи на этапы: определение языка, выделение слов и предложений, загрузка языкозависимых данных и оценка важности предложения. Исследованы способы автоматического тестирование получившихся рефератов и был определен алгоритм нижней границы.

Основную часть данной работы составляет исследование популярных алгоритмов оценки важности предложений, а также влияния TF+IDF и стемминга на данные алгоритмы. Исследовано 5 популярных алгоритмов, среди которых:

- метод основанный на позиций,
- метод основанный на частоте слов,
- метод основанный на близости предложений в среднем,
- метод основанный на сингулярном разложений матрицы слов и предложений текста,
- наивный Байесовский классификатор.

На основе анализа подходов, заложенных в них, предложены два новых алгоритма автоматического реферирование, являющиеся линейными комбинациями данных алгоритмов. Проведено исследование каждого алгоритма на предмет оптимальных коэффициентов линейной комбинации.

Только два предложенных алгоритма превзошли алгоритм нижней границы, то есть удовлетворяют предъявленным к ним требованиям.

Литература

1. Mani I., Maybury M. Advances in Automatic Text Summarization. MIT Press, 1999.
2. Luhn H.P. The automatic creation of literature abstracts // IBM J. of Research and Development. 1958. 2. N 2. P.159-165.
3. Яцко В.А. Симметричное реферирование: теоретические основы и методика // НТИ. 2002. 2. N 5. С. 18-28.
4. Kiss T., Strunk J. Unsupervised multilingual sentence boundary detection // Computational Linguistics. 2006. 32. N 4. P. 485–525.
5. Dunning T. Accurate methods for the statistics of surprise and coincidence // Computational Linguistics. 1993. 19. N 1. P. 61–74.
6. Riley M. D. Some applications of tree-based modeling to speech and language indexing // In Proceedings of the DARPA Speech and Natural Language Workshop. Cape Cod, Massachusetts. 1989. P. 339–352.
7. Müller H., Amerl V., Natalis G. Worterkennungs verfahren als Grundlageeiner Universal methode zur automatischen Segmentierung von Texten in Sätze. Ein Verfahren zur maschinellen Satz grenzen bestimmung im Englischen // Sprache und Datenverarbeitung. 1980. 4. N 1. P. 46–64.
8. Chenand S.F., Goodman J. T. An empirical study of smoothing techniques for language modeling // Technical Report TR-10-98. Harvard University. 1998.
9. Lin C.-Y., Hovy E.H. The Automated acquisition of topic signatures for text summarization // In Proceedings of COLING-00. Saarbrücken, Germany. 2000. P. 495-501.