УДК 004.89: 004.4

Повышение эффективности интеллектуального поиска в полнотекстовых базах данных на основе автоматического аннотирования документов

Вороной С.М., к.т.н., доц., Егошина А.А., Донецкий государственный институт искусственного интеллекта, г. Донецк
postmaster@iai.donetsk.ua

В работе предложен вариант организации полнотекстовой информационно-поисковой системы с естественно-языковым интерфейсом и модулем контекстно-зависимого аннотирования документов. Проведен анализ методов контекстного аннотирования документов и предложены модифицированные алгоритмы работы модуля, позволяющие повысить качество аннотаций и сократить время их формирования.

Введение

Открытые электронные источники информации позволяют использовать огромное количество публикаций и тем самым ставят проблему эффективного поиска нужных сведений в больших объёмах документов. Для решения этой проблемы активно разрабатываются и совершенствуются системы информационного поиска. Современные поисковые системы являются незаменимым средством для работы с большими объемами текстовой информации. Главным критерием работы полнотекстовой поисковой системы является ее эффективность — скорость нахождения пользователем нужной ему информации. В результате поиска по запросу поисковая система обычно выдает огромное количество ссылок, большинство из которых слабо соответствуют запросу. Для анализа полученной информации и выбора нужного документа требуются большие затраты времени. Для сокращения затрат времени на просмотр найденных документов современные поисковые системы формируют для каждого документа в списке результатов поиска контекстно-зависимые аннотации, помогающие пользователю быстрее принять решение о полезности документа. Недостатком существующих средств автоматического

аннотирования, применяемых в поисковых системах, является использование достаточно простых алгоритмов контекстного реферирования. В большинстве поисковых машин в качестве аннотаций используется первый фрагмент текста, в котором встречаются слова запроса. Для повышения качества формирования аннотаций в модуле автореферирования информационно-поисковой системы [1], разрабатываемой авторами, проведен анализ существующих алгоритмов построения аннотаций. На основе оценок качества работы и быстродействия алгоритмов, предложены модифицированные алгоритмы, позволяющие повысить соответствие аннотаций смыслу документа и тем самым сократить время поиска

Постановка задачи. Целью работы является создание алгоритмического обеспечения компонентов системы автоматического аннотирования текстов в информационно-поисковых системах с интеллектуальным интерфейсом. Интеллектуальные интерфейсы интенсивно разрабатываются в настоящее время [2] и по своему названию предполагают использование методов и технологий человеко-машинного взаимодействия и искусственного интеллекта. Интеллектуальный интерфейс дает возможность вести с пользователем диалог на естественном языке. “Естественность” языка общения состоит не в том, чтобы он позволял использовать весь словарь и весь арсенал синтаксиса естественного языка, а в том, чтобы он позволял вести взаимодействие с системой без какой бы то ни было предварительной подготовки пользователя. Язык общения должен обеспечивать пользователя простыми средствами однозначной формулировки задачи поиска необходимых документов [1].

В работе проведен анализ существующих методов контекстно-зависимого аннотирования, существующих оценок качества и требований к составлению аннотаций. Проведено сравнение алгоритмов составления контекстно-зависимой аннотации по эффективности и временным затратам. С целью повышения качества выдаваемой аннотации и уменьшения временных затрат предложена модификация существующих алгоритмов автоматического аннотирования.

Алгоритмы построения аннотаций

Основные алгоритмы [3] аннотирования текста по запросу определяют вид и количество входящих в аннотацию фрагментов, а также их размер.

Алгоритм выбора наилучшего фрагмента базируется на предположении, что найденный по запросу единый фрагмент связного текста, отражает именно ту часть документа, которая необходима.

Основным составляющим элементом для построения аннотаций служит некоторый фрагмент текста, удовлетворяющий следующим требованиям:

- содержит наибольшее число термов запроса;
- между двумя предложениями из фрагмента, которые содержат термы запроса, может находиться не более одного предложения, не содержащего ни одного терма запроса.

Алгоритм построения аннотации состоит из следующих этапов.

Идентификация фрагментов текста, удовлетворяющих вышеупомянутому условию.

Выбор фрагментов текста, содержащих наибольшее количество различных термов запроса.

В случае существования фрагментов, содержащих равное число уникальных термов запросов происходила оценка фрагмента на основе частоты вхождения термов фрагмента в исходный документ. Вес фрагмента при этом вычислялся по следующей формуле:

$$W_F = \sum_{t \in T} \ln(F_t) \qquad (1)$$

где T – множество уникальных термов исходного документа, а F_t – частота терма t в документе.

Размер отобранного фрагмента сравнивается с максимально допустимым размером аннотации. В случае, если размер фрагмента меньше размера аннотации, аннотация дополняется первым заголовком, предшествующим выбранному фрагменту текста, который помещается в начало аннотации. Если размер полученной аннотации все еще меньше максимально допустимого, то в аннотацию включаются предложения, следующие в исходном документе после выбранного фрагмента, но не выходящие за границы параграфа, которому принадлежит выделенный фрагмент.

В [4] рассмотрены три алгоритма формирования контекстно-зависимых аннотаций.

1. Базовый алгоритм, который анализирует в документе только слова запроса и отбирает фрагмент с их наибольшей плотностью.

2. Алгоритм Freq, учитывающий не только слова из запроса, но и другие, наиболее часто встречаемые вокруг фрагмента.

3. Алгоритм LRU-K, использующий при выборе фрагмента как слова запроса, так и слова, найденные по алгоритму LRU-K, которые подсчитывает повторяемость слов в тексте

Базовый алгоритм. При данном алгоритме производилось сканирование текста документа, при этом отбирался самый «тяжелый» фрагмент фиксированной длины не пересекающий границу абзаца. Требуемая длина выбиралась из того требования, что аннотация легко могла быть просмотрена и оценена пользователем. Была выбрана длина около 25 слов, но не длиннее 300 символов.

Вес фрагмента документа определялся следующим образом:

$$W_b = Sum(W_i)+n/L, \quad (2)$$

где $Sum(W_i)$ – сумма весов слов запроса, вошедших в фрагмент. Каждое слово учитывалось только один раз. Вес каждого слова, рассчитываемый по формуле (2), зависел от распределения слова в коллекции и был тем выше, чем более редкое это слово.

$$W_i = log2(N_i)/log2(N), \quad (3)$$

где N_i – число документов коллекции, где встретилось данное слово, N – общее число документов в коллекции,

n – число слов из запроса, которые встретились в фрагменте,

L – расстояние между первым и последним словом запроса, встретившимся во фрагменте.

Таким образом, указанная формула выше оценивала фрагменты, в которых слова запроса располагались более «кучно», встречалось больше слов запроса, и выше оценивались те слова запроса, которые реже встречались во всей коллекции.

В список результатов поиска для документа помещался фрагмент текста, который получил наибольший вес. Если несколько фрагментов получали одинаковые веса, то выдавался тот, который находился ближе к началу текста.

Отобранный фрагмент «выравнивался в тексте», то есть вырезалось некоторое количество слов до первого слова запроса, и несколько после, добавляя длину фрагмента до выбранной длины.

Алгоритм Freq. Описанный выше базовый алгоритм не может показать высокого качества формирования аннотаций, если запрос содержит всего одно слово или часто встречающееся словосочетание. В этом случае он просто возвращает первый фрагмент текста, в котором встретились слова запроса.

Проанализируем алгоритм, использующий кроме слов запроса другие слова документа, которые имеют высокую частоту встречаемости.

Для определения веса в тексте длиной в 1000 слов используется следующая формула:

$$Wfreq=Wb+Sum(log2(Fk)) \quad (4)$$

где *Wb* – вес, вычисленный по базовому алгоритму, Fk – количество повторений слова.

Отсюда следует, что максимальный вес имеют фрагменты, содержащие не только наибольшее число слов запроса, но и и большее количество слов часто встречающихся в документе.

Алгоритм LRU-K. При статистической обработке документа в качестве метрики значимости термина обычно используется частота его встречаемости, которая не несет информации о распределении слова в документе и к тому же требует относительно много ресурсов.

Проанализируем работу алгоритма LRU-K [5] при определении значимости термина в тексте. Алгоритм состоит из следующих шагов.

1. При инициализации создается 3 структуры данных: массивы слов (M) и 2 массива (M1 и M2) с указателями на слова длиной k.

2. Для каждого слова выполняется поиск в массиве слов M:

- если слово не найдено, то ссылка на него помещается в первую позицию массива M1, остальные слова в массиве сдвигаются, самое последнее слово удалялось из M1 и из массива слов M;

- если слово найдено и встречалось в M1, то оно удаляется из M1 и переносится на первую позицию в M2, при этом, если M2 полностью заполнен, то последнее слово из него так же удаляется, как и в первом случае;

- если слово найдено и уже было вM2, то оно просто перемещается на первую позицию.

Если бы слова в тексте имели равную вероятность появления, то после обработки фрагмента текста, содержащего слов намного больше k, содержимое массива M2 совпадало бы с k наиболее часто встретившихся слов. То есть данный алгоритм можно рассматривать как один из вариантов оценки локальной частоты терминов, при предположении равномерного распределения слов.

Однако, предлагаемый алгоритм, кроме этого, должен выделять слова, которые имеют не только высокую частоту, но и равномерно распределенные вблизи выбираемого фрагмента.

Вычисление веса фрагмента производилось также как и для алгоритма Freq, но вместо суммы по наиболее часто встречающимся словам, к весу, вычисленному по базовому алгоритму, прибавлялось количество слов из массива M2, встретившихся в анализируемом фрагменте.

Оценка качества работы и быстродействия алгоритмов

Эффективность описываемых алгоритмов (базового алгоритма, алгоритм Freq, алгоритм LRU-K) была проверена с помощью серии

экспериментов РОМИП [6] и получили наибольшие оценки качества в категориях And/And и Or/Or [7]. Экспериментальные оценки качества алгоритмов приведены в таблице 1.

Таблица 1 – **Результаты оценки качества работы базового алгоритма, алгоритм Freq и алгоритма LRU-K в конференции РОМИП-2005**

	Базовый алгоритм	алгоритм Freq	алгоритм LRU-K
	(алгоритм/лучший результат)	(алгоритм/лучший результат)	(алгоритм/лучший результат)
And/And	0,789832 /0,79	0,788395 /0,79	0,788395 /0,79
Or/Or	0.893758/0,89	0.890610 /0,89	0.890610 /0,89

Из приведенных выше оценок следует, что наилучшее быстродействие по сравнению с остальными показал базовый алгоритм. Временные оценки работы алгоритмов приведены в таблице 2. Оптимальное быстродействие базового алгоритма достигается за счет небольшого объема обрабатываемых данных. Как видно из таблицы 2 алгоритм Freq требует более чем в 5 раз больше времени, так как что вычисление локальной частоты для слов является достаточно трудоемкой операцией.

Таблица 2. **Результаты оценки временных затрат базового алгоритма, алгоритм Freq и алгоритма LRU-K в экспериментах Губина и Мерекулова**

Алгоритм	Время (сек)
Базовый	25
Freq	127
LRU-K	35

Учитывая вышесказанное, основой для дальнейшего исследования и возможной модификации будет служить базовый алгоритм.

Модификация базового алгоритма

На основе анализа существующих методов составления аннотаций для повышения эффективности базового алгоритма предлагаются следующие модификации.

1. Чаще всего число слов запроса к поисковым системам не превышает трех. Если запрос состоит более чем из одного слова, то наиболее адекватной будет аннотация, в которой встретились полностью все слова запроса, а не большое количество повторений отдельных слов запроса. То есть, наиболее существенной будет считаться аннотация, $Sum(W_i)$ которой максимальна. При совпадении некоторых $Sum(W_i)$ и $Sum(W_{i+k})$ сравнение будет производится по Wb (2):

$$W_1 > W_2 = \begin{cases} \textit{истина}, \text{если } [\mathrm{Sum}(W_{1i}) > \mathrm{Sum}(W_{2i})] \underline{\text{ИЛИ}} \\ \quad [\mathrm{Sum}(W_{1i}) > \mathrm{Sum}(W_{2i}) \underline{\text{И}}\ \mathrm{count}_1 > \mathrm{count}_2] \\ \quad \underline{\text{ИЛИ}} \\ \quad [\mathrm{Sum}(W_{1i}) \le \mathrm{Sum}(W_{2i}) \underline{\text{И}}\ W_{1b} > W_{2b}], \\ \textit{ложь}, \text{иначе}. \end{cases} \quad (5)$$

где *W1,W2* – вес аннотации 1 и 2 соответственно,

W1b, W2b – вес аннотации 1 и 2 соответственно, вычисленный по формуле базового алгоритма (1.1),

count1, count2 – количество слов запроса встретившихся в аннотации 1 и 2 соответственно, учитывается только 1 раз.

2. В любом тексте слова в информационном отношении неравнозначны, из чего следует, что вместо числа n слов запроса в (2) целесообразнее использовать сумму весов каждого встретившегося слова запроса.

3. Следует заметить, что далеко не всегда часто встречаемое слово в предложении несет основной смысл. Например, в тексте говорится о различных видах птиц. Естественно, что чаще всего в тексте встретится слово «птица», а такие слова как «дятел», «воробей» не более одного или двух раз, хотя именно они несут основной смысл. Проанализируем влияние некоторой величины (W^*_i), обратной W_i:

$$W^*_j = 1/W_i. \quad (6)$$

4. Если ключевые слова и их вес рассчитывается для набора документов, объединенных одной темой, то важность слова для рубрики и для каждого конкретного документа различается. В связи с этим необходимо учитывать в весе слова запроса и частоту встречаемости в каждом документе:

$$W^{**}_j = W_i \cdot W_{1i}. \quad (7)$$

где *W1i = log2(N1i)/log2(N1), N1i* – число слов в документе, *N1* – общее кол-во слов в документе (все слова и ключевые считаются отдельно).

Экспериментальные исследования

Для оценки качества предложенных модификаций проведены экспериментальные исследования алгоритмов формирования аннотаций. Исследовались базовый алгоритм и алгоритмы автореферирования получаемые путем использования одной или нескольких модификаций. Обозначения алгоритмов приведено в таблице 3.

Таблица 3 – **Соответствие алгоритма использованным в нем модификациям**

Условное название алгоритма	Модификации (№)				
a	Базовый алгоритм				
b			3		
c	1		3		

Если в алгоритме используется модификация 1 с другими модификациями, то базовый вес слова вычисляется как и предполагает модификация, а сравнение аннотаций выполняется по формуле (5).

Эксперименты по автоматическому составлению аннотации к тексту проводились для тематики «Фотография», размер коллекции 16 документов, допустимые размеры аннотации от 50 до 300 символов.

В таблице 4 приведены оценки качества составления аннотаций.

Таблица 4 – **Результаты оценок качества составления аннотаций**

количество слов в запросе	Метрики	Алгоритмы		
		a	b	c
1	Аккуратность	0,67		
	Точность	0,67		
	Полнота	0,80		
	Ошибка	0,33		
2	Аккуратность	0,53	0,59	
	Точность	0,44	0,50	
	Полнота	0,57	0,71	
	Ошибка	0,47	0,41	
3	Аккуратность	0,64	0,67	0,67
	Точность	0,71	0,83	0,83
	Полнота	0,71	0,71	0,71
	Ошибка	0,36	0,33	0,33

Из полученных экспериментальных данных и подсчета количественных оценок видно, что влияние на качество составления аннотаций оказывает модификация 3.

Заключение

На основе предложенных вариантов модификации алгоритмов аннотирования текстов для полнотекстовой поисковой системы в настоящее время разрабатываются инструментальные средства для пополнения и обновления баз знаний интеллектуальных систем дистанционного обучения в Донецком государственном институте искусственного интеллекта.

Литература

1. *Егошина А.А.* Языковые и алгоритмические аспекты построения морфологических процессоров для интеллектуального поиска в полнотекстовых базах данных.// Сб. тр. VI международной конференции «Интеллектуальный анализ информации ИАИ-2006», с.102-111
2. *Поспелов Д.А.* Системы общения и экспертные системы /Искусственный интеллект -М.: Наука, 1990.
3. *Кондратьев М.А.* Аннотирование по запросу: связность или информативность? //Труды РОМИП'2005. *Под ред. И.С. Некрестьянова,* Санкт-Петербург: НИИ Химии СПбГУ, 214 с, сентябрь 2005
4. Г*убин М.В., Меркулов А.И.* Эффективный алгоритм формирования контекстно-зависимых аннотаций // Компьютерная лингвистика и интеллектуальные технологии. Труды международной конференции «Диалог'2005» (Звенигород, 1-6 июня 2005 г.). – М.: Наука, 2005. – С. 116–120.
5. *Luhn H.P.* The automatic creation of literature abstracts //IBM Journal of Research and Development. 1958. V. 2. №2
6. Г*убин М.В.,* Участие ИПС «Кодекс» в семинаре РОМИП 2005 //Труды РОМИП'2005. *Под ред. И.С. Некрестьянова,* Санкт-Петербург: НИИ Химии СПбГУ, 214 с, сентябрь 2005
7. *Агеев р.М., Кураленок И.* Приложение А. Официальные метрики РОМИП //Труды РОМИП'2005. *Под ред. И.С. Некрестьянова,* Санкт-Петербург: НИИ Химии СПбГУ, 214 с, сентябрь 2005