УДК 001.103:004:056

Д. В. Ландэ[1], Б. А. Березин[1], О. Ю. Павленко[2]

[1]Институт проблем регистрации информации НАН Украины,

ул. Н. Шпака, 2, 03113 Киев, Украина

[2]Открытый международный университет развития человека “Украина”,

ул. Львовская, 23, 03115 Киев, Украина

Алгоритм сегментации слов на основе поиска кратчайшего пути в графе

Рассмотрены особенности алгоритмов сегментации слов из текстов, не содержащих разделителей. Представлен новый алгоритм сегментации слов на основе поиска кратчайшего пути. Приведены оценки качества сегментации. Показаны возможности использования приведенного алгоритма в задачах поиска информации в национальных доменах сети Интернет. Реализация алгоритма сегментации слов используются для создания обобщенной модели предметной области на базе мониторинга ресурсов китайского сегмента Интернет.

Ключевые слова: *сегментация слов, сегментатор, качество сегментации слов, мониторинг, поиск кратчайшего пути, волновой алгоритм.*

Актуальность и анализ публикаций

Растущее количество информационных ресурсов, представленных в Интернете, ведет к необходимости развития поисковых систем для доступа к ним. В то же время, растет доля и важность мировых веб-ресурсов, представленных на языках, в которых в явном виде отсутствуют границы слов с помощью разделителей. К таким языкам относятся китайский, японский, тайский и др. Если люди-носители языка не имеют проблем с пониманием текстов на этих языках, то для использования традиционных поисковых систем с индексированными словами (а к таким относятся Google, Bing и др.), необходимо выделение каждого слова – их сегментация (word segmentation).

В работах [1,2] отмечаются такие особенности китайского сегмента сети Интернет как высокие темпы роста количесства веб-ресурсов; наличие собственных социальных сетей и глобальной поисковой системы Baidu, ориентированной на китайский язык, покрывающей значительную часть веб-ресурсов этого сегмента Интернет. В этих работах также показаны подходы к построению систем мониторинга национальных Интернет-ресурсов, обосновывается актуальность задачи сегментация слов при формировании индекса поисковых систем. В более ранней работе [3], рассматриваются возможности применения традиционных систем поиска и извлечения информации для информационных ресурсов

ресурсов, представленных на китайском и японском языках. Отмечается важность индексирования и проблемы автоматической сегментации китайских слов.

В работе [4] отмечается, что для решения проблемы сегментации китайского текста используются три основных способа: словарный (обычно с применением алгоритма максимального соответствия), статистический и комбинированный, сочетающий в себе оба предыдущих. При использовании этих методов удаётся сегментировать текст с точностью, превышающей 90%. В работе [5] рассматривается сегментация слов другого языка – урду (который также не использует разделителей между словами) как важную проблему приложений обработки естественных языков (NLP). Рассматриваются технология совпадения с наиболее длинными словами из словаря (longest matching); технология максимального совпадения (maximum matching) и методы статистической сегментации.

Особенности моделей алгоритма сегментации слов

Данная работа посвящена разработке алгоритма сегментации слов (АСС) для формирования индекса поисковой системы, быстродействие которого повышено благодаря использованию методов оптимизации, а также оценке качества сегментации с помощью этого алгоритма.

В работах, посвященных сегментации слов, выделяются две основные модели — статистические и использующие предопределенный (списки слов и правила) [6]. При этом отмечается вариант алгоритма с максимальным совпадением (maximal matching), для которого есть модификации – Forward Maximal Matching (FMM) и Backward Maximal Matching (BMM), в зависимости от направления обработки текста. Второй вариант для алгоритма со словарем - это алгоритм, который находит сегментацию с минимальным количеством слов shortest path (SP).

Для моделей на основе словаря предполагается наличие списка слов, каждое из которых связано с оценкой вероятности того, что оно является истинным словом. Рассматривается список – множество пар W:

$$W = \left\{ \left\{ w_i, g\left(w_i\right) \right\}_{i=1,\ldots,1} \right\},$$

где w_i является кандидатом на слово, а $g\left(w_i\right)$ – соответствующая ему функция качества. Алгоритм прямого максимального соответствия FMM обрабатывает текст T для путем нахождения на каждом шагу алгоритма текущего слова $w*$, при максимизации его качества:

$$\left\{w*, t*\right\} = \arg\max_{\{w,t\}\in W} g(w).$$

Алгоритм сегментации на основе кратчайшего пути [6,7] использует предположение о том, что правильная сегментация должна максимизировать длины всех слов или

минимизировать общее количество слов, т. е. для предложения S из m символов $\{c_1, c_2, ..., c_m\}$ – лучшее сегментированное предложение $S*$ из $n*$ слов $\{w_1^*, w_2^*, ..., w_n^*\}$:

$$S* = \arg\min_{w_1 w_2 ... w_n = T} n.$$

Данная задача оптимизации преобразуется в задачу нахождения кратчайшего пути для направленного нециклического графа.

При статистическом подходе [6] должен быть доступен для машинного обучения корпус текстов с уже сегментированными словами. В настоящее время становится особенно популярным метод CRF (Conditional Random Fields –условные случайные поля), применение которого рассмотрено в [8]. В [6] сравниваются результаты использования CRF- и FMM-сегментаторов. Для оценки выполнения сегментации слов используются два основных параметра: полнота и точность, а также некоторые их комбинации. Точность (*Precision*) при сегментации определяется как отношение количества совпадений позиций разделителей после применения алгоритма сегментации и истинными позициями разделителей (выполненных экспертами), к количеству разделителей в сегментированном тексте:

Precision = (число совпадений) / (число разделителей в полученном тексте) . (1)

Полнота (*Recall*) определяется как число совпадений позиций разделителей после применения алгоритма сегментации и истинными позициями разделителей, к количеству разделителей в тексте, сегментированном экспертами:

Recall = (число совпадений) / (число разделителей в стандартном тексте). (2)

F-мера определяется как комбинация полноты и точности по формуле:

$$F = 2 \times Precision \times \text{Recall} / (Precision + Recall) \quad (3)$$

В работе [6] сделаны выводы о том, что для улучшения качества машинной сегментации ключевым является не выбор стратегии сегментации, а лингвистические ресурсы.

В работе [9] алгоритм сегментации слов рассматривается с точки зрения модели марковских цепей. В рамках этой модели в зависимости от того, возможна ли сегментация при присоединении текущего символа к предыдущим, вводятся две фазы. Если текущий символ может быть присоединен к предыдущими, модель переходит в фазу s_0, в противном случае, модель преходит переходит к фазе s_1. Предполагается, что общая средняя вероятность переходов из s_0 в s_0, из s_0 в s_1, от s_1 к s_0 и от s_1 к s_1, являются q, $1 - q$, p и $1–p$ соответственно. В работе обосновывается, что для получения правильной и быстрой сегментации требуется, чтобы значение $|q - p|$ было достаточно малым, в то время, как p и q должны быть достаточно большими. В целом, рассматриваемый алгоритм сегментации состоит из двух основных частей: выбора из словаря слов, образованных на основе заданных

символов и вычисления функции оценки для полученных разбиений. Для оценки разбиений в [9] используется так называемая функция языковой ситуации.

Особенности технологии сегментации слов в китайских системах полнотекстового поиска исследуются в [10]. Методы сегментации слов, использующие словарь, статистические и комбинированные методы могут применяться для формирования индекса поисковой системы. Но опыт показывает, что эти методы имеют ограничения как в преодолении неоднозначностей, так и по затратам времени. Для преодоления этих ограничений в работе рассматривается древовидная, иерархическая организация словарной базы.

В работах, посвященных поисковым системам [11,12] рассматривается алгоритм сегментации слов с помощью средств Lucene (библиотека с открытым кодом для полнотекстового поиска, поддерживается Apache Software Foundation). В состав этих средств входит алгоритм сегментации китайских слов IKAnalyzer с открытым кодом. Используется алгоритм с максимальным совпадением на основе словаря библиотеки. Сравнение эффективности алгоритма сегментации китайских слов при использовании компонент Lucene - Standard Analyzer, Smart Chinese Analyzer, IKAnalyzer, а также особенности поисковых систем для образовательных ресурсов приводятся в [12].

Разработка алгоритма сегментации

Для реализации модели АСС предлагается использовать алгоритм поиском кратчайшего пути в графе [14,15]. Его реализация выполнена на языке Perl, а упрощенный псевдокод алгоритма сегментации слов приведен на Рис. 1. Для поиска кратчайшего пути в графе использовался волновой алгоритм, который был соответствующим образом доработан, модифицирован. Классический волновой алгоритм (алгоритм Ли) изначально создавался для формализации алгоритмов трассировки печатных плат, однако получил применение во многих других областях науки и техники [16]. Обычно рассматривается его работа на дискретном рабочем поле (ДРП), представляющем ограниченную замкнутой линией фигуру, разбитую на прямоугольные или квадратные ячейки. Множество всех ячеек ДРП разбивается на подмножества: “проходимые” (свободные), т. е при поиске пути их можно проходить, “непроходимые” (препятствия), пути через эти ячейки запрещены, стартовая ячейка и финишная. Основными этапами работы алгоритма является распространение волны (разметка) и восстановление пути. От стартовой ячейки порождается шаг в соседнюю ячейку, при этом проверяется, проходима ли она, и не принадлежит ли ранее меченой в пути ячейке. При выполнении условий проходимости и непринадлежности её к ранее помеченным в пути ячейкам, в атрибут ячейки записывается число, равное количеству шагов от стартовой ячейки (на первом шаге это будет 1). Каждая ячейка, меченая числом шагов от стартовой ячейки, становится стартовой и из неё

порождаются очередные шаги в соседние ячейки. Очевидно, что при таком переборе будет найден путь от начальной ячейки к конечной, либо очередной шаг из любой порождённой в пути ячейки будет невозможен.

Восстановление кратчайшего пути происходит в обратном направлении: при выборе ячейки от финишной ячейки к стартовой на каждом шаге выбирается ячейка, имеющая атрибут расстояния от стартовой на единицу меньше текущей ячейки. Очевидно, что таким образом находится кратчайший путь между парой заданных ячеек. На основе работ [17-19], вычислительная сложность классического волнового алгоритма определяется как $O(n^2)$, т.е., является квадратичной (где n – количество вершин в графе, или ячеек в ДРП).

Предложенный алгоритм сегментации слов состоит из трех частей:

– формирования таблицы Слов;

– формирования таблицы Шаг_Позиция;

– формирования массива сегментированных слов и вывода результатов в файл.

В первой части программы создается массив, каждая строка которого соответствует символу входной строки. При нахождении множества слов, на которые может быть разбита входная строка, для каждого входного символа анализируются возможные подстроки, начинающиеся с данного символа, длиной от 1 до n (n – максимально возможная длина слова, зависит от языка. Для китайского, в словаре можно найти слова до 5-6 иероглифов, для русского – до 18-20 букв и т.д.). Если для анализируемой подстроки находится соответствующее слово в словаре, то такое слово используется в разбиении.

Построенная таким образом таблица содержит множество слов, на которые может быть разбита входная строка для выбранного словаря. В Табл. 1 показано разбиение, полученное для входной строки на английском языке без разделителей: «IWORKATTHERESEARCHINSTITUTE» (в словаре найдены слова и часто используемые сокращения). Множество слов, на которые разбивается входная строка, может быть представлено направленным графом. Буквы входной строки являются вершинами такого графа, а слова разбиения, получаемые для каждой буквы, представляют ребра, исходящие из соответствующих вершин.

На рис. 2 приведено представление разбиения входной строки «IWORKATTHERESEARCHINSTITUTE» на слова в виде графа. При таком представлении, выбор слов правильного разбиения входной строки может рассматриваться как поиск кратчайшего пути в графе. Для решения этой задачи, наряду с другими методами нахождения кратчайшего пути в графе, могут использоваться различные варианты волнового алгоритма.

Таблица 1. Разбиение, полученное для входной строки.

Входная строка	Найденные в словаре слова разбиения				
i	i				
w	work				
o	or				
r					
k					
a	a	at			
t					
t	th	the	there	theres	
h	he	her	here		
e	ere				
r	re	res	research		
e	es				
s	se	sea	sear	search	
e	ear				
a	a	ar	arc	arch	
r					
c	ch	chi	chin	chins	
h	hi				
i	i	in	ins	inst	institute
n	ns				
s	st				
t	ti	tit			
i	i	it			
t	tu	tut			
u	ut				
t					
e	e				

Во второй части программы, в соответствии с волновым алгоритмом, с помощью таблицы Слов формируется таблица Шаг_Позиция, с помощью которой выполняется разметка вершин графа, определяется их расстояние (соответствующие количеству ребер в случае невзвешенного графа) от начальной вершины (первой буквы входной строки). Для этого, на каждом шаге (волне) алгоритма определяется множество соседних, достижимых в данный момент вершин. Таким образом, таблица Шаг_Позиция содержит множество вершин графа, которые достижимы при анализе текущего входного символа (текущей строки таблицы Слов). А также содержит позиции начала слов, на которые разбивается входная строка при сегментации. В целом, программа может рассматриваться как модифицированный волновой алгоритм нахождения кратчайшего пути в графе.

На третьем этапе алгоритма производится выбор ребер в графе, составляющих минимальный путь, начиная с последней вершины, на основе расстояний до предшествующих вершин, т.е. происходит выбор минимального количества слов, на которые может быть разбита, сегментирована входная строка. На этом этапе, при выборе слов, для улучшения качества сегментации, кроме расстояний могут также учитываться другие особенности языков, например частота использования слов в текстах на разных языках и т.п.

Таким образом, в этой части программы, на основе таблицы Шаг_Позиция и входной строки символов заполняется массив Сегментированных Слов. Затем слова из сформированного массива выводятся в выходной файл программы.

```
Чтение слов из файла словаря в хэш
Цикл по входным строкам {
Чтение входной строки из файла
#Формирование таблицы Слов
Цикл по символам входной строки {
    Цикл по n символам вх. строки, следующим за текущим{
    Если подстрока совпадает со словом словаря {
        Запись подстроки в табл. Слов }
        Иначе {
        слово не найдено}
    }Конец Цикл по символам
}Конец Цикла по символам входной строки
#Формирование таблицы Шаг_Позиция
Цикл по строкам табл. Слов {
    Если текущая строка табл. Шаг_Позиция свободна {
        Запись строки в таблицу}
    Если в строке табл. Слов есть слова {
        Цикл по словам в строке табл. Слов {
        Определение позиции нач. след. Слова =длина слова+текущая позиция
        #Формирование строки в  таблице Длин_Слов
        Если позиция нач. след. Слова табл. Шаг_Позиция свободна{
        Запись строки в таблицу}
        Иначе{
        Если номер новой волны на 1 больше текущей {
        Перезапись старой строки табл. Шаг_Позиция }#Уменьшение количества слов
        } Конец иначе
        } Конец Цикла по словам в строке табл. Слов
    }Конец Если в строке табл. Слов есть слова
} Конец Цикла по строкам табл. Слов
#Формирование массива сегментированных слов и вывод в файл
    Цикл по символам {
    Чтение подстрок входной строки в массив Сегментированных Слов, используя
    позиции из табл. Шаг_Позиция
    }Конец Цикл по символам
    Цикл по кол. сегм. слов в строке {
    Вывод сегментированных слов в файл
    }Конец Цикл по кол. сегм. слов
} Конец Цикл по входным строкам
```

Рис. 1 – Обобщенный псевдокод алгоритма сегментации слов с поиском минимального пути в графе

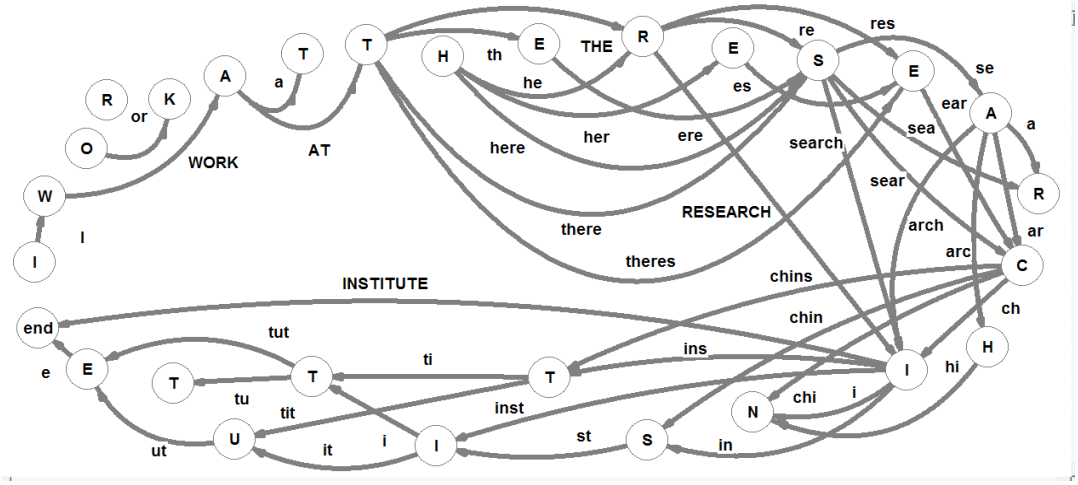

Рис. 2 – Представление разбиения входной строки на слова в виде графа

Оценка моделей сегментации

Для оценки качества сегментации и возможности использования реализованных вариантов АСС авторами проводилось тестирование алгоритма максимального соответствия FMM и алгоритма с поиском кратчайшего пути в графе, а также других, эксплуатируемых в сети Интернет сегментаторов. Аналогично работам [20,21], для оценки сегментаторов использовался инструментальный набор для оценки сегментации китайских слов, представленный в сети Интеренет по адресу https://web.archive.org/web/20100702191858fw_/http://projectile.sv.cmu.edu/research/public/tools/segmentation/eval/index.htm. В состав набора входят программные средства и тестовый корпус. Также при оценке сегментаторов использовался тестовый корпус на основе китайских новостных текстов (данные проекта the Chinese Treebank, с китайскими текстами из Xinhua news, Sinorama news magazine, and Hong Kong News [20]). Кроме того, для оценки алгоритмов был подготовлен собственный тестовый корпус объемом 10 тысяч слов на основе частотного китайского словаря.

Для тестирования сегментаторов авторами использовался метод EDWS (Edit Distance of the Word Separator), заключающийся в следующем. Пусть $C_1C_2...C_iC_{i+1}...C_n$ представляет собой предложение без сегментации. Сегментатор разделяет это предложение на последовательность слов путем вставки разделителя S между символами, в результате предложение может иметь, например, такой вид: $C_1C_2SC_3C_4C_5S...C_iC_{i+1}...SC_n$. Разные программы сегментации могут размещать разделители в разных позициях одного и того же предложения. Расстояние при редактировании разделения слов в EDWS определяет, сколько операций редактирования (вставки, удаления и совпадения) необходимы для модификации автоматически полученной сегментации текста до стандартной сегментации того же текста, полученной вручную, экспертами. Например, пусть стандартная сегментация имеет вид первой строки таблицы 2, а результат, выдаваемый сегментатором имеет вид второй строки таблицы.

Таблица 2. Пример операций редактирования при оценке сегментации.

C_1	C_2	***S***	C_3	C_4	C_5	***S***	C_6	C_7		C_8	C_9	***S***	C_{10}	***S***	C_{11}	C_{12}	C_{13}	***S***	C_{14}
C_1	C_2	***S***	C_3	C_4	C_5	***S***	C_6	C_7	***S***	C_8	C_9	***S***	C_{10}		C_{11}	C_{12}	C_{13}	***S***	C_{14}
		Сов				**Сов**			**Удл**			**Сов**		**Вст**				**Сов**	

В третьей строке таблицы приведены операции редактирования, которые надо выполнить, для модификации второй строки в первую. В этом примере имеем совпадения (Сов) после C_1, C_2, C_9 и C_{13}; вставку (Вст) после C_{10}; удаление (Удл) после C_7. Точность (precision) и полнота (recall) являются показателями, которые используются при оценке большей части алгоритмов извлечения информации.

Результаты сравнения параметров различных сегментаторов приведены в таблице 3, в столбце 1 которой показаны виды тестируемых сегментаторов, в столбце 2 – тип корпуса (новостной, из инструментального набора, или словарный, отобранный из словаря), использованный при тестировании. В 3-5 столбцах таблицы приведены количества операций редактирования (совпадения, вставки, удаления), которые необходимо было выполнить для модификации текста после сегментатора в стандартный текст. В последних, 6-8 столбцах представлены значения показателей, рассчитанные по формулам (1) – (3) при проведении тестирования. Для алгоритма Е. Петерсона тестирование было проведено с использованием словарей двух размеров - 348982 слов (один из словарей сегментатора Jieba) и 119804 слов (словарь WordList).

Таблица 3. Результаты тестирования

Вид сегментатора	**Тестовый корпус**	**Сов.**	**Вст.**	**Удл.**	**Precision**	**Recall**	**F-1**
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Jieba	**новостной**	**19581**	**2567**	**1233**	**94.08%**	**88.41%**	**0.91**
	словарный	**9966**	**33**	**365**	**96.47%**	**99.67%**	**0.98**
Online	**новостной**	**21334**	**814**	**1911**	**91.78%**	**96.32%**	**0.94**
	словарный	**9686**	**313**	**3508**	**73.41%**	**96.87%**	**0.84**
Алгоритм с поиском кратчайшего пути в графе (словарь 348982 слов)	**новостной**	**19313**	**2835**	**1699**	**91.91%**	**87.20%**	**0.89**
	словарный	**9988**	**11**	**464**	**95.56%**	**99.89%**	**0.98**
Алгоритм Е. Петерсона (словарь 348982 слов)	**новостной**	**20673**	**1475**	**1073**	**95.07%**	**93.34%**	**0.94**
	словарный	**9882**	**117**	**2940**	**77.07%**	**98.83%**	**0.87**
Алгоритм Е. Петерсона (словарь 119804 слов)	**новостной**	**20524**	**1624**	**1452**	**93.39%**	**92.67%**	**0.93**
	словарный	**9696**	**303**	**4850**	**66.66%**	**96.97%**	**0.79**
IrSegmenter	**новостной**	**21345**	**803**	**3145**	**87.16%**	**96.37%**	**0.92**

2 5 марта считается датой основания белорусской народной республики представители белорусской оппозиции отмечают этот день национальным праздником независимости белоруссии еще в среду один из оппозиционных лидеров николай стат к е вич передал в минский горисполком заявление что он собирается провести в субботу в центре столицы массовую акцию в связи сч е мв целях безопасности людей предложил освободить от движения транспорта две крайние правые полосы проезжей части проспекта независимости накануне марша он через хартию 9 7 распространил обращение что сегодня люди выйдут на улицы не за миску похлебки

Рис. 3 – Пример сегментации новостного текста на русском языке

Кроме тестирования алгоритмов сегментации на основе китайских текстов, для алгоритма максимального соответствия FMM и алгоритма с поиском кратчайшего пути в графе для оценки разбиения проводилось также тестирование с использованием текстов и

словарей на английском и русском языке. Оценивался процент слов, сегментированных с ошибками для массивов данных из различных видов информационных ресурсов - новостных сообщений, технических и художественных текстов. При этом использовались два вида словарей – словарь русского языка и словарь, сформированный путем накопления новостных сообщений. На Рис. 3 показан пример сегментации новостного текста на русском языке с помощью алгоритма поиска кратчайшего пути в графе. Из исследуемого текста в этом случае были предварительно удалены пробелы и разделители, а прописные буквы преобразованы в строчные. Были выявлены характерные ошибки сегментации: в числах и в фамилии, которые отсутствуют в словаре. Следовательно, для чисел в алгоритм сегментатора необходимо ввести соответствующее правило. Одна ошибка в фрагменте была связана с особенностями самого алгоритма.

В работах [22-23] отмечается, что скорость обработки китайских текстов современными сегментаторами составляет от десятков до сотен и тысяч килобайт в секунду. Кроме оценки правильности сегментации слов, для предложенной АСС оценивается скорость обработки текстов. Для поиска кратчайшего пути в графе в предложенном АСС используется модифицированный волновой алгоритм. В отличие от классического волнового алгоритма, описанного выше, вычислительная сложность которого выражается квадратичной зависимостью от количества ячеек ДРП (вершин графа), вычислительная сложность предложенного алгоритма от объема входных данных соответствует линейной. АСС учитывает особенности входных данных (текст, со словами без пробелов, без разделителей) и построен так, что необходимые вычисления выполняются за один проход. Поэтому его вычислительная сложность зависит от количества букв во входной строке и числа слов разбиения, начинающихся на данную букву и найденных в словаре. Зависимость времени выполнения сегментации текста от объема текста при использовании предложенного алгоритма сегментации на выбранных примерах приведена на рис. 4, на котором верхняя кривая соответствует сегментации русского текста, представленного множеством строк (абзацев) по несколько сотен символов, которые обрабатываются независимо друг от друга. Средняя зависимость соответствует тому же тексту, представленному одной строкой. На нижней, пунктирной кривой, показана разница между двумя предыдущими графиками. (Данные получены при выполнении сегментации слов на компьютере с процессором Celeron 1,5 GHz и памятью 2 Gb).

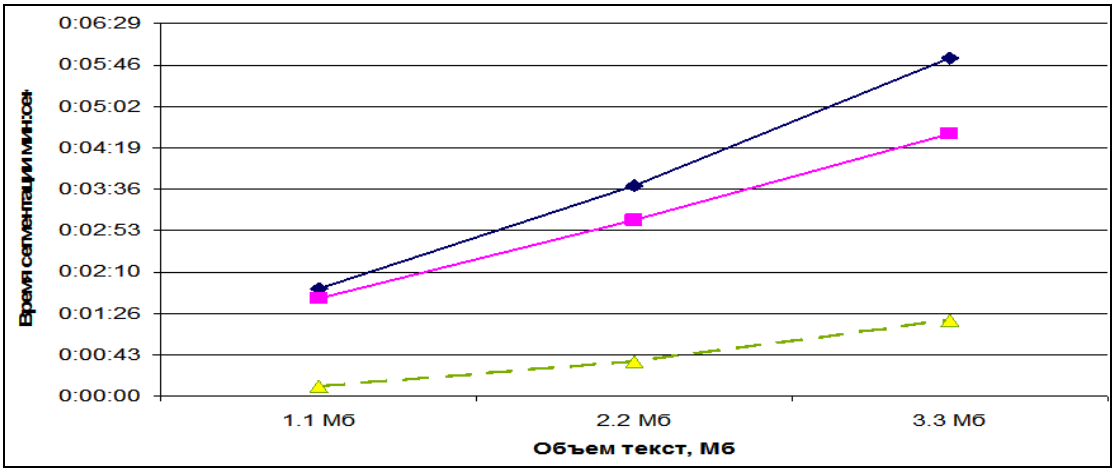

Рис. 4 – Зависимость времени выполнения сегментации текста от объема текста при использовании АСС на основе поиска кратчайшего пути в графе

Программные средства, созданные на основе предложенной АСС использовались для построения поискового индекса в результатах мониторинга информационных ресурсов китайского сегмента Интернет. На Рис. 5. показаны результаты выявления значимых (ключевых) слов, полученных после сегментации в сообщении на китайском языке о завершении строительства скоростной железной дороги с помощью предложенного алгоритма.

Рис. 5. Информационное сообщение и выделенные значимые слова

Выводы

Рост количества информационных ресурсов китайского сегмента сети Интернет обуславливает необходимость создания глобальных информационно-поисковых систем, для реализации поисковых индексов которых необходима быстрая, точная и полная сегментация

слов в текстах сетевых документов. Именно этим объясняется актуальность задачи сегментации слов в настоящее время. В работе приведены варианты алгоритмов сегментации слов, которые могут быть использованы для формирования индекса поисковых систем, показана применимость моделей на основе словарей.

Также в статье предложена и проанализирована новая модель FMM для ACC на основе словаря, включающая алгоритм сегментации с поиском кратчайшего пути в графе, для чего разработано программное обеспечение.

Получены оценки качества сегментации и результаты использования модели ACC при формировании индекса поисковой системы, которые показывают возможность использования алгоритма для информационных ресурсов китайского сегмента сети Интернет при достаточном объеме словаря.

1. Ландэ Д.В., Березин Б.А., Додонов В.А. Обзор особенностей и возможности контент-мониторинга национального сегмента сети Интернет // Реєстрація, зберігання і обробка даних, 2016. – Т. 18, № 3. – С. 20-38.

2.Ландэ Д., Березин Б., Павленко О. Построение модели информационного сервиса на базе национального сегмента Интернет // Информационные технологии и безопасность. Материалы XVI Международной научно-практической конференции ИТБ-2016. - К.: ИПРИ НАН Украины, 2017. – С. 48-57.

3.Boisen, S. Chinese information extraction and retrieval / S. Boisen, M. Crystal, E. Peterson, R. Weischedel, J. Broglio, J. Callan, M. E. Okurowski // Proceedings of a workshop on held at Vienna, Virginia. Association for Computational Linguistics, 1996. – P. 109-119.

4.Загибалов Т.Е. Автоматический анализ текстов на китайском языке. Проблема выбора базовой единицы // Труды международной конференции “ Диалог”, 2005. – С. 31-37.

5.Durrani, N., Hussain, S. Urdu word segmentation // Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 2010. – P. 528-536.

6. Zhao H., Utiyama M., Sumita E., Lu B. L. An empirical study on word segmentation for chinese machine translation // International Conference on Intelligent Text Processing and Computational Linguistics. Springer Berlin Heidelberg, 2013. – P. 248-263.

7. Jia Z., Wang P., Zhao H. Graph model for Chinese spell checking // Proceedings of the 7th SIGHAN Workshop on Chinese Language Processing (SIGHAN’13), 2013. – P. 88-92.

8. Антонова, А. Ю.; Соловьев, А. Н. Метод условных случайных полей в задачах обработки русскоязычных текстов // Информационные технологии и системы. Труды международной научной конференции. Калининград, 2013. – С. 321-325.

9. Zhang, M. Y., Lu, Z. D., Zou, C. Y. A Chinese word segmentation based on language situation in processing ambiguous words // Information Sciences, 2004. - 162(3). – P. 275-285.

10. Liu, C. Research on words segmentation technology in Chinese full text retrieval system // Applied Mechanics and Materials. Trans Tech Publications, 2013. - Vol. 411, - P. 313-316.

11. Xu, L. X., Fu, X. L., Zhang, C. H. Research on Full-text Retrieval based on Lucene in Enterprise Content Management System // Applied Mechanics and Materials. Trans Tech Publications, 2014. – Vol. 644. – P. 1950-1953.

12.Yang M., Li J., Gou X. The research of Chinese word segmentation strategy in educational resources search engine based on lucene // Advanced Intelligence and Awareness Internet (AIAI 2011), 2011 International Conference on. IET, 2011. – P. 136-140.

13. Peterson Erik. A Chinese named entity extraction system // Proceedings of the 8th Annual Conference of the International Association of Chinese Linguistics, Melbourne, Australia. 1999. – P. 47-58.

14. Ландэ Д.В., Березин Б.А., Павленко О.Ю. Разработка алгоритма сегментации слов для систем мониторинга национальных интернет-ресурсов //Міжнародна науково-практична конференція "Інтелектуальні технології лінгвістичного аналізу": Тези доповідей. – Київ: НАУ, 2017. – С. 11.

15. Березин Б., Ландэ Д., Павленко О. Разработка, оценка и использование алгоритма сегментации слов для систем мониторинга национальных интернет-ресурсов // Информационные технологии и безопасность. Материалы XVII Международной научно-практической конференции ИТБ-2017. - К.: ООО "Инжиниринг", 2017. – С. 22-31.

16. Rubin F. The Lee path connection algorithm // IEEE Transactions on Computers. — 1974. – P. 907-914.

17. Изотова Т. Ю. Обзор алгоритмов поиска кратчайшего пути в графе // Новые информационные технологии в автоматизированных системах, 2016. – № 19. – С. 341-344.

18. Алексеев В. Е., Таланов, В. А. Графы. Модели вычислений. Структуры данных // Нижний Новгород: Изд-во ННГУ, 2005. – С. 169.

19. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы. Построение и анализ. – М: Издательский дом Вильямс, 2009. – С. 637.

20. Fung R., Bigi B. Automatic word segmentation for spoken Cantonese // Oriental COCOSDA held jointly with 2015 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE), 2015 International Conference IEEE. – P. 196-201.

21. Chea V., Thu Y.K., Ding C., Utiyama M. Khmer word segmentation using conditional random fields // Khmer Natural Language Processing, 2015. – pp. 62-69.

22. Peng H., Cambria E., Hussain, A. A review of sentiment analysis research in chinese language // *Cognitive Computation*. - 2017. – P. 1-13.

23. Li-guo D., Peng D., Ai-ping L. A new naive Bayes text classification algorithm // Indonesian Journal of Electrical Engineering and Computer Science, 2014. – 12(2). – P. 947-952.