Минимальное расстояние редактирования

Лекция 2

Елена Тутубалина

Казанский федеральный университет

15 февраля 2018 г.

Схожесть двух строк

- Исправление ошибок написания
  - Пользователь напечатал "жраф"
    Какие слова наиболее похожи?
    - жираф
    - граф
    - жрав
    - ...

- Также применяется для оценивания методов машинного перевода и распознавание речи

Расстояние редактирования

- Минимальное расстояние редактирования между двумя строками
- – это минимальное число операций:
  - вставка (в)
  - удаление (у)
  - замена (з)
- необходимых для преобразования одной строки в другую.

Выравнивание двух строк

Две строки и их *выравнивание*:

Р	А	З	Р	А	Б	О	Т	К	А
|	|	|	|	|	|	|	|	|	|
*	*	*	Р	Е	Б	Я	Т	*	А

Выравнивание двух строк

Р	А	З	Р	А	Б	О	Т	К	А
|	|	|	|	|	|	|	|	|	|
*	*	*	Р	Е	Б	Я	Т	*	А
у	у	у		з		з		у	

- Если каждая операция стоит 1 балл (расстояние Левенштейна)
  - То расстояние равно 6
- Если каждая замена стоит 2 (Levenshtein)
  - То расстояние равно 8

Другие применения расстояния редактирования в NLP

- Оценивание качества машинного перевода или распознавания речи

Spokesman	confirms		senior	government	adviser	was	shot
Spokesman	said	the	senior		adviser	was	shot
	З	В		У			

- Распознавание именованных сущностей и разрешение кореференции
  - IBM Inc. announced today
  - IBM profits
  - Stanford President John Hennessy announced yesterday
  - for Stanford University President John Hennessy

Как вычислять MPP?

- Ищем путь (последовательность операций) из начальной строки в конечную:

  Начальное состояние : исходное слово

  Операции : вставка, удаление, замена

  Цель : конечное слово

  Цена пути : то, что мы собираемся минимизировать.

Вычисление минимального расстояния как зачада поиска

- Пространство цепочек редактирования огромно!
  - Не можем позволить себе искать “наивно”
  - Множество цепочек приводит к одному и тому же состоянию
    - Не имеет смысла запоминать их все
    - Только самый короткий (дешёвый).

Формальное определение - 1

- Для двух строк
  - X длины n
  - Y длины m
- Определим D(i,j)
  - как расстояние между $X[1..i]$ и $Y[1..j]$
    - т.е., между первыми i символами X и первыми j символами Y
  - Таким образом, искомое расстояние редактирования между X и Y: $D(n,m)$

Динамическое программирование для подсчета MPP

- *Динамическое программирование*: табличный подсчёт $D(n,m)$
- Решаем проблему, комбинируя решения подпроблем.
- “Снизу-вверх”
  - считаем $D(i,j)$ для малых i,j
  - и подсчитываем последующие $D(i,j)$ на основе уже вычисленных значений
  - т.е., вычисляем $D(i,j)$ для всех i $(0 < i < n)$ и j $(0 < j < m)$

Формальная запись алгоритма

- Инициализация
  $D(i, 0) = i$
  $D(0, j) = j$
- Рекурсия:
  для каждого $i = 1 \ldots M$
  для каждого $j = 1 \ldots N$

$$D(i,j) = min = \begin{cases} D(i-1,j)+1, \\ D(i,j-1)+1, \\ D(i-1,j-1)+ \quad 2; \\ \qquad\qquad\qquad\quad 0; \begin{cases} if\ X(i) \neq Y(j) \\ if\ X(i) = Y(j) \end{cases} \end{cases}$$

- Остановка:
  $D(N, M)$ – искомое расстояние

The Edit Distance Table

N	9									
O	8									
I	7									
T	6									
N	5									
E	4									
T	3									
N	2									
I	1									
#	0	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

Edit Distance

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

N	9									
O	8									
I	7									
T	6									
N	5									
E	4									
T	3									
N	2									
I	1									
#	0	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

The Edit Distance Table

N	9									
O	8									
I	7									
T	6									
N	5									
E	4									
T	3	4	5	6	7	8	7	8	9	8
N	2	3	4	5	6	7	8	7	8	7
I	1	2	3	4	5	6	7	6	7	8
#	0	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

Computing alignments

- Edit distance isn't sufficient
  - We often need to *align* each character of the two strings to each other
- We do this by keeping a "backtrace"
- Every time we enter a cell, remember where we came from
- When we reach the end,
  - Trace back the path from the upper right corner to read off the alignment

Edit Distance

n	9	↓ 8	↙←↓ 9	↙←↓ 10	↙←↓ 11	↙←↓ 12	↓ 11	↓ 10	↓ 9	↙ **8**
o	8	↓ 7	↙←↓ 8	↙←↓ 9	↙←↓ 10	↙←↓ 11	↓ 10	↓ 9	↙ **8**	← 9
i	7	↓ 6	↙←↓ 7	↙←↓ 8	↙←↓ 9	↙←↓ 10	↓ 9	↙ **8**	← 9	← 10
t	6	↓ 5	↙←↓ 6	↙←↓ 7	↙←↓ 8	↙←↓ 9	↙ **8**	← 9	← 10	←↓ 11
n	5	↓ 4	↙←↓ 5	↙←↓ 6	↙←↓ 7	↙←↓ **8**	↙←↓ 9	↙←↓ 10	↙←↓ 11	↙↓ 10
e	4	↙ 3	← 4	↙← **5**	← **6**	← 7	←↓ 8	↙←↓ 9	↙←↓ 10	↓ 9
t	3	↙←↓ 4	↙←↓ **5**	↙←↓ 6	↙←↓ 7	↙←↓ 8	↙ 7	←↓ 8	↙←↓ 9	↓ 8
n	2	↙←↓ **3**	↙←↓ 4	↙←↓ 5	↙←↓ 6	↙←↓ 7	↙←↓ 8	↓ 7	↙←↓ 8	↙ 7
i	**1**	↙←↓ 2	↙←↓ 3	↙←↓ 4	↙←↓ 5	↙←↓ 6	↙←↓ 7	↙ 6	← 7	← 8
#	**0**	1	2	3	4	5	6	7	8	9
	#	**e**	**x**	**e**	**c**	**u**	**t**	**i**	**o**	**n**

Adding Backtrace to Minimum Edit Distance

- Base conditions: $D(i,0) = i$ $D(0,j) = j$ Termination: $D(N,M)$ is distance
- Recurrence Relation:
  For each $= 1 \ldots M$
  For each $j = 1 \ldots N$

$$D(i,j) = min = \begin{cases} D(i-1,j)+1, & \text{deletion} \\ D(i,j-1)+1, & \text{insertion} \\ D(i-1,j-1)+ \begin{cases} 2; & \\ 0; \begin{cases} if\ X(i) \neq Y(j) & \text{substitutio} \\ if\ X(i) = Y(j) & \end{cases} \end{cases} & \end{cases}$$

$$ptr(i,j) = \begin{cases} LEFT & \text{insertion} \\ DOWN & \text{deletion} \\ DIAG & \text{substitution} \end{cases}$$

Result of Backtrace

- Two strings and their *alignment*:

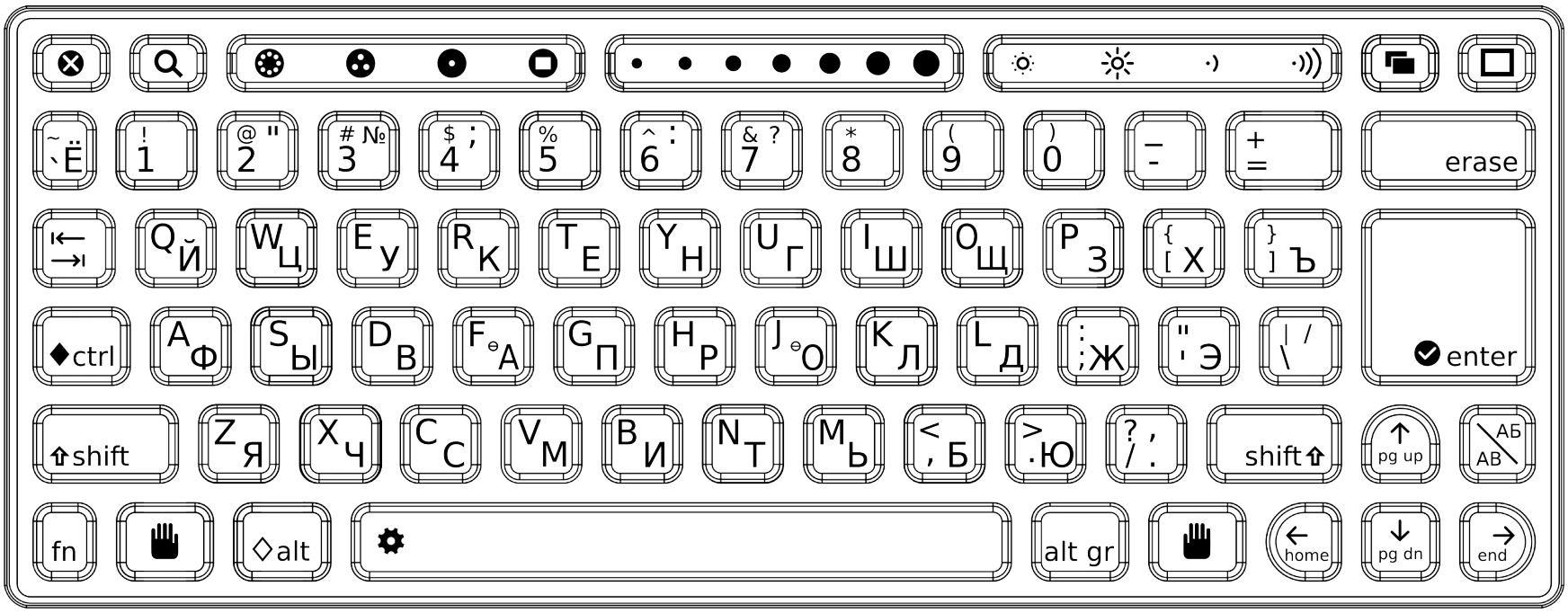
```
I  N  T  E  *  N  T  I  O  N
|  |  |  |  |  |  |  |  |  |
*  E  X  E  C  U  T  I  O  N
```

Performance

- Time: $O(nm)$
- Space: $O(nm)$
- Backtrace: $O(n + m)$

Weighted Edit Distance

- Why would we add weights to the computation?
  - Spell Correction: some letters are more likely to be mistyped than others
  - Biology: certain kinds of deletions or insertion are more likely than others


~ ` Ё
! 1
@ " 2
# № 3
$; 4
% 5
^ : 6
& ? 7
* 8
(9
) 0
_ -
+ =
erase
Q Й
W Ц
E У
R К
T Е
Y Н
U Г
I Ш
O Щ
P З
{ [Х
}] Ъ
ctrl
A Ф
S Ы
D В
F Э A
G П
H Р
J О
K Л
L Д
: ; Ж
" ' Э
| \ /
enter
shift
Z Я
X Ч
C С
V М
B И
N Т
M Ь
< , Б
> . Ю
? / ,
shift
pg up
АБ
АВ
fn
alt
alt gr
home
pg dn
end

Weighted Min Edit Distance

- Initialization:
  $D(0,0) = 0$
  $D(i,0) = D(i-1,0) + del[x(i)]; \qquad 1 < i \leq N$
  $D(0,j) = D(0,j-1) + ins[y(j)]; \qquad 1 < j \leq M$
- Recurrence Relation:

$$D(i,j) = min = \begin{cases} D(i-1,j) + del[x(i)], \\ D(i,j-1) + ins[y(j)], \\ D(i-1,j-1) + sub[x(i), y(j)]; \end{cases}$$

- Termination:
  $D(N,M)$ is distance

Ссылки

- https://web.stanford.edu/ jurafsky/slp3/2.pdf
- https://www.youtube.com/playlist?list=PL6397E4B26D00A269
- https://compscicenter.ru/courses/introduction-nlp/2017-autumn/classes/

Задание 2

- Подсчитать (на бумаге) расстояние для пар *drive* и *brief*, *drive* и *divers*.
- Запрограммировать вычисление минимального расстояния и сравнить с подсчетами на бумаге.
- Добавить в реализацию вывод Backtrace.