NOTES ON LEVENSHTEIN DISTANCE

ERIC MARTIN

1. LEVENSHTEIN_DISTANCE(DEPART, LEOPARD)

$\leftarrow$ Deletion (cost 1) of a letter x in first word: $\begin{smallmatrix} x \\ * \end{smallmatrix}$
$\downarrow$ Insertion (cost 1) of a letter x in second word: $\begin{smallmatrix} * \\ x \end{smallmatrix}$
$\swarrow$ Match (cost 0) of the same letter x in both words: $\begin{smallmatrix} x \\ x \end{smallmatrix}$
Substitution (cost 2) of a letter x in first word by a different letter y in second word: $\begin{smallmatrix} x \\ y \end{smallmatrix}$

d	7		6	←	7		6		5		4	← 5
	↓	↙			↓		↓		↓		↓	↙ ↓
r	6	←	7		6		5		4		3	← 4
	↓	↙	↓		↓		↓		↓	↙		
a	5	←	6		5		4		3	←	4	← 5
	↓	↙	↓		↓		↓	↙				
p	4	←	5		4		3	←	4	←	5	← 6
	↓	↙	↓		↓	↙						
o	3	←	4		3	←	4	←	5	←	6	← 7
	↓	↙	↓		↓	↙	↓	↙	↓	↙	↓	↙ ↓
e	2	←	3		2	←	3	←	4	←	5	← 6
	↓	↙	↓	↙								
l	1	←	2	←	3	←	4	←	5	←	6	← 7
	↓	↙	↓	↙	↓	↙	↓	↙	↓	↙	↓	↙ ↓
·	0	←	1	←	2	←	3	←	4	←	5	← 6
	·		d		e		p		a		r	t

*	d	e	*	p	a	r	*	t		d	*	e	*	p	a	r	*	t		d	e	*	p	a	r	*	t
l	*	e	o	p	a	r	d	*		*	l	e	o	p	a	r	d	*		l	e	o	p	a	r	d	*
*	d	e	*	p	a	r	t	*		d	*	e	*	p	a	r	t	*		d	e	*	p	a	r	t	*
l	*	e	o	p	a	r	*	d		*	l	e	o	p	a	r	*	d		l	e	o	p	a	r	*	d
*	d	e	*	p	a	r	t			d	*	e	*	p	a	r	t			d	e	*	p	a	r	t	
l	*	e	o	p	a	r	d			*	l	e	o	p	a	r	d			l	e	o	p	a	r	d	

Date: Session 2, 2017.

2. LEVENSHTEIN_DISTANCE(PAPER, POPE)

4	e	4 ↓		3 ↓	← ↙	4 ↓		3 ↑		2 ↙	←	3
3	p	3 ↓	↙	2 ↓	← ↙	3 ↓	↙	2	←	3	←	4
2	o	2 ↓		1 ↓	← ↙	2 ↑	← ↙	3 ↑	← ↙	4 ↑	← ↙	5 ↑
1	p	1 ↓	↙	0	←	1	← ↙	2	←	3	←	4
0	·	0	←	1	←	2	←	3	←	4	←	5
		·		p		a		p		e		r
		0		1		2		3		4		5

Table computed by _get_distances_and_backtraces_table():

```
[[[0, []], [1, ['V']], [2, ['V']], [3, ['V']], [4, ['V']]],
 [[1, ['H']], [0, ['D']], [1, ['V']], [2, ['D', 'V']], [3, ['V']]],
 [[2, ['H']], [1, ['H']], [2, ['D', 'H', 'V']], [3, ['D', 'H', 'V']], [4, ['D', 'H', 'V']]],
 [[3, ['H']], [2, ['D', 'H']], [3, ['D', 'H', 'V']], [2, ['D']], [3, ['V']]],
 [[4, ['H']], [3, ['H']], [4, ['D', 'H', 'V']], [3, ['H']], [2, ['D']]],
 [[5, ['H']], [4, ['H']], [5, ['D', 'H', 'V']], [4, ['H']], [3, ['H']]]]
```

Alignments computed by _compute_alignments() for given i, j and dir:

```
1 1 D ('p', 'p')
      2 2 D ('pa', 'po')
1 1 D ('p', 'p')
    1 2 V ('p*', 'po')
      2 2 H ('p*a', 'po*')
1 1 D ('p', 'p')
    2 1 H ('pa', 'p*')
      2 2 V ('pa*', 'p*o')
        3 3 D ('pap', 'pop'),
              ('p*ap', 'po*p'),
              ('pa*p', 'p*op')
          4 4 D ('pape', 'pope'),
                ('p*ape', 'po*pe'),
                ('pa*pe', 'p*ope')
            5 4 H ('paper', 'pope*'),
                  ('p*aper', 'po*pe*'),
                  ('pa*per', 'p*ope*')
```