School of Computing and Information Systems
The University of Melbourne
COMP90049
Knowledge Technologies (Semester 1, 2017)
Workshop sample solutionss: Week 3

1. Following on from last week, write a **regular expression** which will match a string according to whether it contains a price (like $20 or $0.99, but not 11.30 or 0$1an).

   - Depends on how stringently we want to handle cases like $001.230 — one possible solution:

     `/\$(0|[1-9][0-9]*)(\.\d{1,2})?/`

&

Suppose that we have observed the token `lended`, and we have a dictionary as follows:

```
addendum
blenders
commodity
deaden
end
leader
leant
lent
lemonade
pleading
```

2. Which, if any, of the above dictionary entries be returned using a Neighbourhood Search with a neighbourhood of 1? 2? 3?

   - There aren't any items in the dictionary requiring only a single change from `lended`.
   - With a neighbourhood size of 2, there is a dictionary entry:
     - `leader`, by Replacing the `n` with `a`, and the second `d` with `r`
   - Along with the above, the following are also within a neighbourhood of 3:
     - `blenders`, by Inserting the `b`, Replacing the second `d` with `r`, and Inserting the `s`
     - `deaden` (three Replaces)
     - `end` (three Deletions)
     - `lent` (one Replace and two Deletions)

3. With respect to the input string `lended` and the dictionary entry `deaden`, calculate the following:

   (a) the Global Edit Distance, using the parameter $[m, i, d, r] = [+1, -1, -1, -1]$

   - From the first table overleaf, we can observe that the Global Edit Distance is 0, corresponding to the following sequence of operations: Replace, Match, Replace, Match, Match, Replace, which I will abbreviate as `rmrmmr`. (You can follow along with the highlighted back-pointers.)

   (b) the Local Edit Distance, using the parameter $[m, i, d, r] = [+1, -1, -1, -1]$

   - From the second table overleaf, we can observe that the Local Edit Distance is 2 (highlighted); there are five equivalent-scoring substring matches that it corresponds to:
     - Align `-de-` in `lended` with the first `de-` in `deaden`: `mm`
     - Align `-ded` with `dead-`: `mmim`
     - Align `-de-` in `lended` with the second `-de-` in `deaden`: `mm`

    - Align `-ende-` with `-eade-`: `mrmm`
    - Align `-en-` with `-en`: `mm`

   (c) the N-Gram Distance, using $n = 2$

    - We begin by generating the 2-grams of the two strings; I will use the terminal marker (#) here:
        - `lended`: `#l`, `le`, `en`, `nd`, `de`, `ed`, `d#`
        - `deaden`: `#d`, `de`, `ea`, `ad`, `de`, `en`, `n#`
    - Recall that the N-Gram Distance is defined as follows:

$$D(s,t) = \mid G_n(s) \mid + \mid G_n(t) \mid -2\times \mid G_n(s) \cap G_n(t) \mid$$

    - Here we have 7 2-grams in `lended`, as well as 7 in `deaden`. Also, the two sets share 2 2-grams: `de` and `en`. (Note that we don't double-count the `des` in `deaden`, because there is only a single `de` in `lended`)
    - Consequently, the 2-gram Distance is $7 + 7 - 2 \times 2 = 10$

4. Find the best approximate match (or matches, if there are ties) in the dictionary for the string `lended`, based on the following methods; consider different parameters where necessary:

   (a) the Global Edit Distance

    - Using the above scoring parameter, the closest matches are `blenders` (+2) and `leader` (+2)
    - You might like to try some other parameter setting(s), to see if they give different results.

   (b) the Local Edit Distance

    - Using the above scoring parameter, the closest match is `blenders` (+5)
    - In this case, changing the parameter is unlikely to result in a different answer. (Why?)

   (c) the N-Gram Distance

    - If we are using n is 2 and padding with #, the best dictionary entry is `lent`, with a 2-Gram Distance of 6.
    - You might find that removing the padding characters or changing n will give different results.

   (d) Soundex

    - The Soundex code of `lended` is `l533`.
    - None of the dictionary entries have this exact code; however, if we permit mismatches in the Soundex code, then the best matches are `commodity` (`c533`), `leant` (`l53`), `lent` (`l53`), and `lemonade` (`l553`)

(a)	ε		l		e		n		d		e		d
ε	0	←	-1	←	-2	←	-3	←	-4	←	-5	←	-6
	↑	↖		↖		↖		↖				↖	
d	-1		-1	←	-2	←	-3		-2	←	-3	←	-4
	↑	↖	↑	↖						↖			
e	-2		-2		0	←	-1	←	-2		-1	←	-2
	↑	↖	↑		↑	↖		↖			↑	↖	
a	-3		-3		-1		-1	←	-2		-2		-2
	↑	↖	↑		↑	↖	↑	↖				↖	
d	-4		-4		-2		-2		0	←	-1		-1
	↑	↖	↑	↖	↑	↖	↑		↑	↖			
e	-5		-5		-3		-3		-1		1	←	0
	↑	↖	↑		↑	↖			↑		↑	↖	
n	-6		-6		-4		-2		-2		0		0

(b)	ε	l		e		n		d		e		d
ε	0	0		0		0		0		0		0
							↖				↖	
d	0	0		0		0		1	←	0		1
			↖						↖			
e	0	0		1	←	0		0		2	←	1
				↑	↖					↑	↖	
a	0	0		0		0		0		1		1
							↖				↖	
d	0	0		0		0		1	←	0		2
			↖					↑	↖			↑
e	0	0		1	←	0		0		2	←	1
				↑	↖					↑	↖	
n	0	0		0		2	←	1		1		1