The problem is that 'tables' does not exist in the lex

Tutorial 2: Text Normalization

Solutions: 1. Fallback to single characters or substrings
2. Do segmentation based on maximal probability from word frequency and dynamic programming
3. Include common concatenations or compounds words in lexicon
4. Use a statistical or ML based segmenter

Q1. Consider the following word segmentation algorithm in the lecture notes:

Given a lexicon of Chinese, and a string

1) Start a pointer at the beginning of the string
2) Find the longest word in dictionary that matches the string starting at pointer
3) Move the pointer over the word in string
4) Goto2

If matched,
run morphological analysis from beginning of the word
move the pointer over the morphologically matched part of the string

If no word matched, skip to next pointer

Strictly following the algorithm, you perhaps end up with failing to segment a string, if you cannot find a matching. For example, consider segmenting the following string using the given lexicon.

String: thetablesdownthere

Lexicon: the table down there bled own.

Discuss how to fix the above problem.

Q2. Try the tokenization demo on https://textanalysisonline.com/ (or you may use other tokenizer APIs, e.g., http://text-processing.com/demo/tokenize/). Discuss your findings based on the output of different tokenizers.

punctuation is separated from words but dashed words as kept as they are

Q3. Try the stemmer demo on https://textanalysisonline.com/ (or you may use other stemmer APIs, e.g., http://text-processing.com/demo/stem/). Discuss your findings based on the output of the stemmers.

some words have -e and -s removed while others do not, forming non existent words

Q4. Write a program to do the following tasks:

1. Download the Web page of a given link and extract the text content of the page
2. Split the text into sentences and count sentences
3. Split the text into tokens and count token types
4. Find lemmas (or stems) of the tokens and count lemma types
5. Do stemming on the tokens and count unique 'stemmed' tokens

You may use any tools, including nltk, LingPipe, and any other NLP packages. Verify the results using example URLs, e.g., a random Wikipedia page.

Q5. **Open-ended Question, for discussion only**.

In social media (e.g., forums), online users often use informal names or references when mentioning products. Below are example sentences discussing mobile phones, where the words highlighted in bold are the phones being referred to, and the [bracketed text] indicates their official product names.

1. True, **Desire** [HTC Desire] might be better if compared to **X10** [Sony Ericsson Xperia X10] but since I am using **HD2** [HTC HD2], it will be a little boring to use back HTC ...
2. I just wanna know what problems do users face on the **OneX** [HTC One X]... of course I know that knowing the problems on **one x** [HTC One X] doesn't mean knowing the problems on **s3** [Samsung Galaxy SIII]
3. Still prefer **ip 5** [Apple iPhone 5] then **note 2** [Samsung Galaxy Note II]...
4. oh, the mono rich recording at **920** [Nokia Lumia 920] no better than stereo rich recording at **808** [Nokia 808 PureView].

The table below shows the number of users who have mentioned a phone using a specific name in a forum.

Name variation	#users	Name variation	#users
1. galaxy s3	553	14. lte s3	46
2. s3 lte	343	15. galaxy s3 lte	45
3. samsung galaxy s3	284	16. s3 non lte	32
4. s iii	242	17. samsung galaxy siii	32
5. galaxy s iii	225	18. sgs 3	27
6. samsung s3	219	19. samsung galaxy s3 lte	22
7. sgs3	187	20. sg3	21
8. siii	149	21. gsiii	16
9. samsung galaxy s iii	145	22. samsung galaxy s3 i9300	15
10. i9300	120	23. samsung i9300 galaxy s iii	13
11. gs3	82	24. s3 4g	11
12. galaxy siii	61	25. 3g s3	11
13. i9305	52	–	

Task: Assume that we have successfully identified the phone mentions (e.g., 's3 lte,' 'sgs3'). How can we normalize these mentions to their formal names?

Use a dictionary to map them to their respective formal names

- Either choose stemming or lemmatization
- choosing is application dependent
- lemmatization is slower than stemming

- some tokenizer require training data to develop