DIVING INTO THE WORLD OF SPELL CHECKS!

Niharika Krishnan

Cloud Community Days Conference - 19 June 2020

Before we get started....

NIHARIKA KRISHNAN

- Machine Learning Engineer, TCS
  - Build Chatbots for a living!
- Founding Member of PyLadies Chennai
  - Community of 100+ women tech enthusiasts
- Speaker
  - PyCon Canada'19, India'19
  - Google Women Techmakers, Global Diversity CFP
- AI and NLP enthusiast

269 Billion Emails in 2019

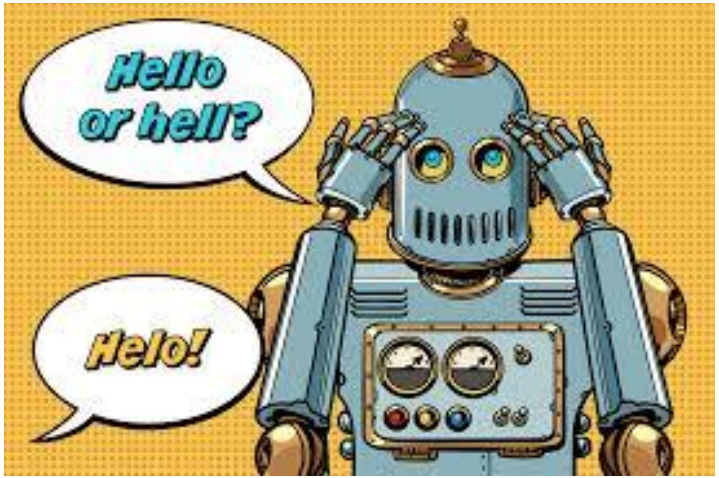
~ 20 mails in a day

naturral lanugageee processing

All News Images Books Videos More Settings Tools

About 53,40,00,000 results (0.64 seconds)

Showing results for ***natural language*** processing
Search instead for naturral lanugageee processing



The elephante enjoyed the peanuts.

elephant
elephants
telephone
elephantine
More...

Ignore
Ignore All
Change All
Add
Auto Correct

Cancel

18 June 8:47 PM | 33 characters

This is cloud community days conferenv

conferenv conferences conferencing

q w e r t y u i o p
a s d f g h j k l
z x c v b n m
?123

Python Packages

Natural Language ToolKit

python™

```
>>>from nltk.metrics import edit_distance
>>>edit_distance("rain","shine")
3
```

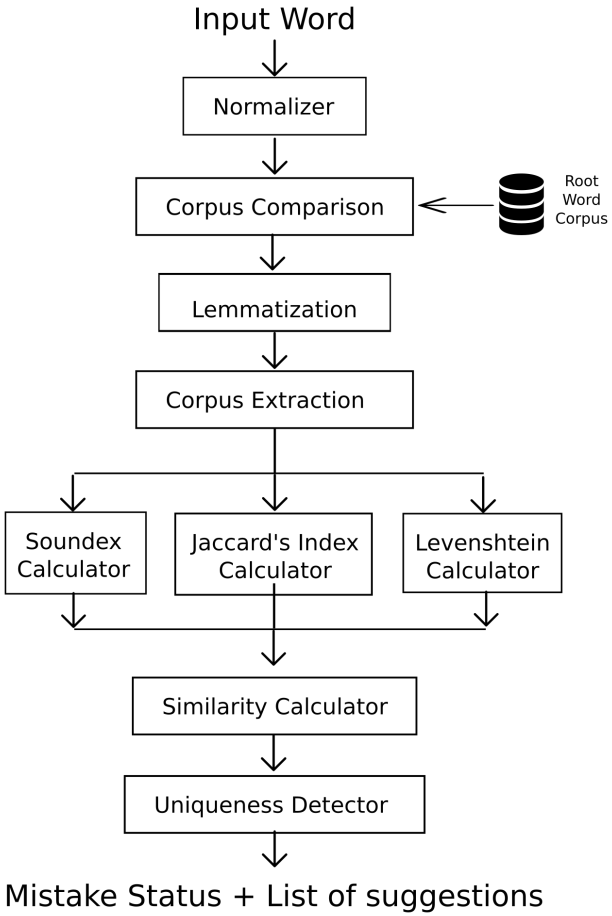

```
>>>b = TextBlob("I havv goood speling")
>>>print(b.correct())
I have good spelling!
```


```
>>>from spellchecker import SpellChecker
>>>spell = SpellChecker()
>>>misspelled = spell.unknown(["cmputr", "study","watr"])
>>>for word in misspelled:
>>>        print(spell.correction(word))
>>>        print(spell.candidates(word))
computer
{'caput', 'caputs', 'compute', 'computor', 'impute', 'computer'}
water
{'water', 'watt', 'warr', 'wart', 'war', 'wath', 'wat'}
```

What happens under the hood ?

Spell-checks

- Spell Checker points to spelling errors and possibly suggests alternatives
- Autocorrector automatically picks the most likely word
- Types:
  - PHONETICS
  - EDIT DISTANCE (Peter Norvig)
  - SYMMETRIC DELETE SPELLING CORRECTION (SymSpell)
- Real word Errors vs Non-Word Errors



Phonetics

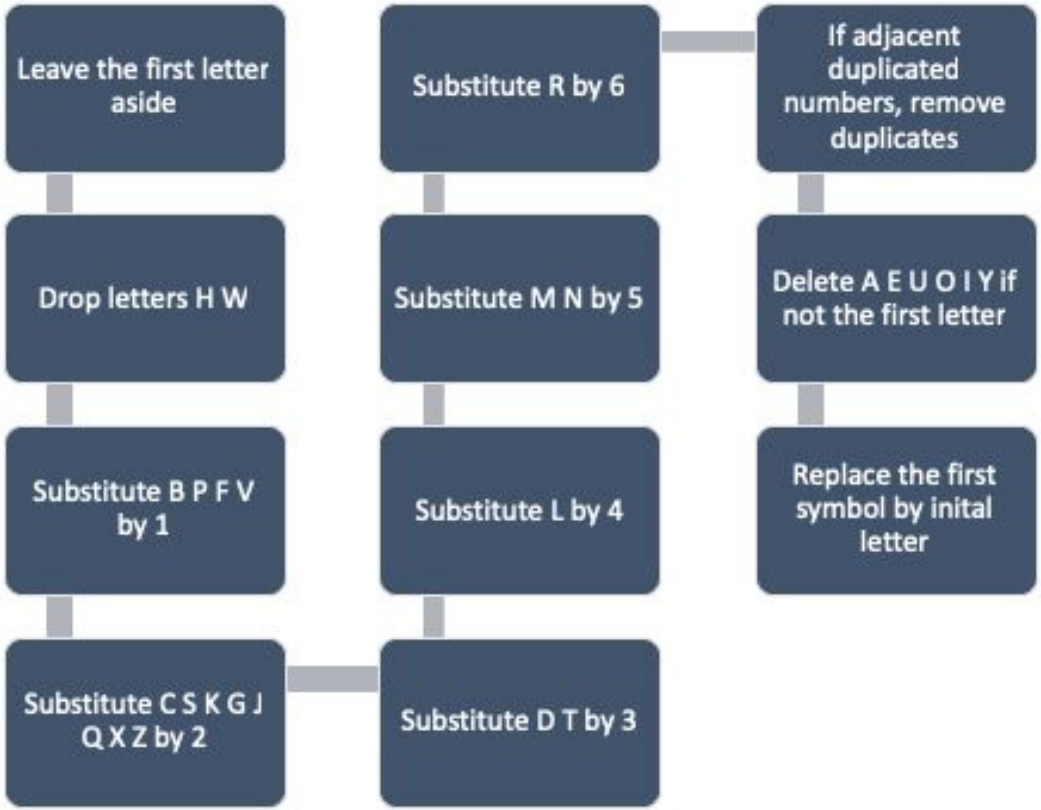
- Detect similar-sounding words even if they are spelt differently like Smith & Schmidt
- Creates a specific phonetic representation of a single word
- **Algorithms:**
  - SOUNDEX
  - METAPHONE

```
>>> import jellyfish
>>> jellyfish.soundex('Break')
'B620'
>>> jellyfish.soundex('Brake')
'B620'
>>> jellyfish.metaphone('Break')
'BRK'
>>> jellyfish.metaphone('Brake')
'BRK'
```

Break

Brake

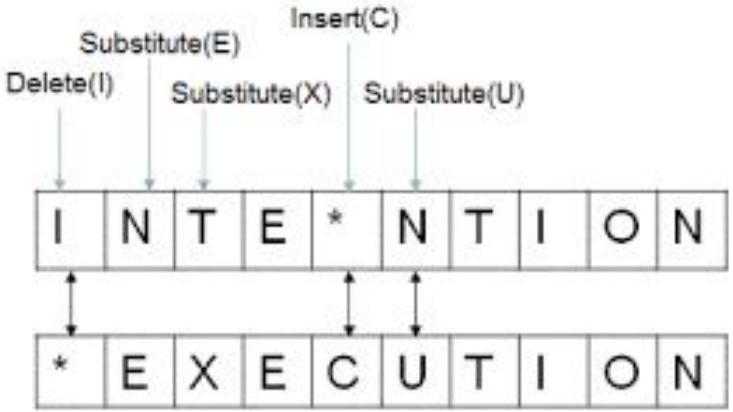
Soundex



Name	Soundex Key
Smith	S530
Schmidt	S530

Edit Distance

- Quantifying how dissimilar two strings are to one another
- Minimum number of edit operations required to transform s1 into s2
  - Insertion, Deletion
  - Substitution, Transposition




```
>>> import jellyfish
>>> jellyfish.levenshtein_distance('jellyfish', 'smellyfish')
2
>>> jellyfish.damerau_levenshtein_distance('jellyfish', 'jellyfihs')
1
>>> jellyfish.hamming_distance('jellyfish', 'jellyfihs')
2
>>> jellyfish.jaro_similarity('jellyfish', 'jellyfihs')
0.9629629629629629
>>> jellyfish.jaro_winkler_similarity('jellyfish', 'smellyfish')
0.8962962962962964
```

Algorithms

- **LEVENSHTEIN**
  - Insertion + Deletion + Substitution
  - RECIEVE → RECEIVE → Edit Distance = 2
  - RECEIVE → RECEIPT → Edit Distance = 2
  - Very different semantically and context

- **DAMERAU - LEVENSHTEIN**
  - Insertion + Deletion + Substitution + Transposition
  - Character swapping

- **LEAST COMMON SUBSEQUENCE**
  - Insertion + Deletion

Levenshtein vs Longest Common Sequence

Kitten → **S**itten (substitute "s" for "k")	**K**itten → itten (delete "k" at 0)
sitt**E**n → sitt**I**n (substitute "i" for "e")	itten → **S**itten (insert "s" at 0)
sittin → sittin**G** (insert "g" at the end)	sitt**E**n → sittn (delete "e" at 4)
	sittn → sitt**I**n (insert "i" at 4)
	sittin → sittin**G** (insert "g" at 6)

Algorithms

- **HAMMING DISTANCE**
  - SUBSTITUTION
  - Only applies to strings of the same length
- **JARO**
  - TRANSPOSITION + Matching Characters
  - Range [0,1] : 0 - Least Similar, 1 - Most Similar
- **JARO-WINKLER**
  - TRANSPOSITION + Matching Characters + Prefix
  - Uses a prefix scale of ‘p’ which gives more favourable ratings to strings that match from the beginning for a set prefix length

Symmetric Delete Spelling Correction

- ➢ Delete-only edit candidate generation
- ➢ 5 letter word → **3 Million Possibilities vs 25 Possibilities (Edit Distance: 3)**

1 Million times faster

INSERTION	delete (dictionary entry,edit_distance)	input entry
goa	**delete(goal,1), delete(goat,1)**	**goa**
DELETION	dictionary entry	delete(input entry,edit_distance)
goall	**goal**	**delete(goall,1)**
SUBSTITUTION & TRANSPOSITION	delete(dictionary entry,edit_distance)	delete(input entry,edit_distance)
goal	**delete(goal,1), delete(goat,1)**	**delete(goak,1)**

Symspellpy

- **Verbosity parameter:**
  - **Top:** highest term frequency + smallest edit distance
  - **Closest:** smallest edit distance found, ordered by term frequency
  - **All:** All suggestions within maxEditDistance, ordered by edit distance, term frequency
- **maxEditDistance**
- **Word frequency dictionary:**
  - LoadDictionary
  - CreateDictionary (Customize it for your use-case!)

Let's see how symspell works!

QUESTIONS

Want to explore further? Let's connect!

 linkedin.com/in/niharikakrishnan

 @Nihaaarika

 niharikakrishnan

Slide Deck: https://github.com/niharikakrishnan/Talks