The idea of stemming is a sort of normalizing method. Many variations of words carry the same meaning, other than when tense is involved.

The reason why we stem is to shorten the lookup, and normalize sentences.

Consider:

I was taking a ride in the car.  
I was riding in the car.

This sentence means the same thing. in the car is the same. I was is the same. the ing denotes a clear past-tense in both cases, so is it truly necessary to differentiate between ride and riding, in the case of just trying to figure out the meaning of what this past-tense activity was?

No, not really.

This is just one minor example, but imagine every word in the English language, every possible tense and affix you can put on a word. Having individual dictionary entries per version would be highly redundant and inefficient, especially since, once we convert to numbers, the "value" is going to be identical.

One of the most popular stemming algorithms is the Porter stemmer, which has been around since 1979.

First, we're going to grab and define our stemmer:

from nltk.stem import PorterStemmer

from nltk.tokenize import sent\_tokenize, word\_tokenize

ps = PorterStemmer()

Now, let's choose some words with a similar stem, like:

example\_words = ["python","pythoner","pythoning","pythoned","pythonly"]

Next, we can easily stem by doing something like:

for w in example\_words:

print(ps.stem(w))

Our output:

python

python

python

python

pythonli

Now let's try stemming a typical sentence, rather than some words:

new\_text = "It is important to by very pythonly while you are pythoning with python. All pythoners have pythoned poorly at least once."

words = word\_tokenize(new\_text)

for w in words:

print(ps.stem(w))

Now our result is:

It

is

import

to

by

veri

pythonli

while

you

are

python

with

python

.

All

python

have

python

poorli

at

least

onc

.