Dmitrii Obideiko
Suraj Janakiraman

How we created a knowledge base:
We created a knowledge base by first clearing up all the text in files. For example all empty lines were removed as well as any spaced between the lines. Then all lines from every text were combined into one big list of lines. A dictionary of sets was used to keep all occurrences in a line for a specific word. (The key was a word and the value was a set of lines where that word appeared.)

```
# creates a search dict for 1 webstie only
def build_searchable_knowledge_base(texts):
    searchDict = defaultdict(set)
    for text in texts:
        # break the text into lines
        lines = text.split('\n')
        for line in lines:
            # break the line into words
            words = word_tokenize(line.lower())
            for word in words:
                searchDict[word].add(line)

    return searchDict
```

Sample Screenshots of knowledge base (text):

form - Open in app Home Notifications Lists Stories Write Published in Go Into The Story Scott Myers Follow Oct 6 · 9 min read

You will see at the bottom that the words “form” and “read are part of the 10 terms listed.

But even once they work through this period, steady-state streaming margins will likely be much lower than traditional TV.

You will see at the bottom that the word “ work” is one of the 10 terms listed. There are multiple usages of the word “work” in the knowledge base.

I have certainly been a design groupie, wanting to meet some of my design heroes at a conference or speaking engagement,

Notice the word sign most typically appears in the word **“design.”** You will see at the bottom that the word “sign” is one of the 10 terms listed.

Indication of the top 10 terms
We created a function that gets the mostly frequent words/tokens from each text in the list of urls.

```python
# returns top n frequent token from text
def extract_most_freq_terms(texts, n):
    texts = ' '.join(texts)
    tokens = word_tokenize(texts.lower())
    # removes all tokens whose length is <= 2
    tokens = [token for token in tokens if len(token) > 2]
    # remove stop words
    stop_words = set(stopwords.words('english'))
    filtered_tokens = [w for w in tokens if w not in stop_words]
    # create a frequency dictionary where the key is a word and the value if the frequency
    freqDict = {token : texts.count(token) for token in set(filtered_tokens)}
    # sort the frequency dictionary in ascending order and return n most frequent words
    sortedFreqDict = sorted(freqDict.items(), key = lambda x: x[1], reverse=True)
    print(len(sortedFreqDict))
    return [el[0] for el in sortedFreqDict[:n]]
```

Output:
['pro', 'ted', 'per', 'min', 'art', 'act', 'low', 'one', 'form', 'men', 'eat', 'use', 'tim', 'work', 'sign', 'app', 'age', 'end', 'able', 'ming', 'format', 'king', 'formation', 'read', 'inc']

From this function, we can choose the top 10 terms which can be classified as actual words:

1) form
2) act
3) work
4) sign
5) app
6) age
7) end
8) able
9) format
10) read

Now we create a dialog based on the knowledge base:

- our story needs the best idea
- think about them
- share ideas
- easy our story is the best