In [29]:
```
import nltk
import string
from nltk import pos_tag
from collections import Counter
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer, WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [30]:
```
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
```

```
[nltk_data] Downloading package punkt to /home/student/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     /home/student/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to /home/student/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

Out[30]: True

In [31]:
```
data=pd.read_csv("/home/student/Downloads/HN_posts_year_to_Sep_26_2016.csv.zip")
data
```

Out[31]:

	id	title	url	num_points	num_comments	au
0	12579008	You have two days to comment if you want stem ...	http://www.regulations.gov/document?D=FDA-2015...	1	0	a
1	12579005	SQLAR the SQLite Archiver	https://www.sqlite.org/sqlar/doc/trunk/README.md	1	0	bla
2	12578997	What if we just printed a flatscreen televisio...	https://medium.com/vanmoof/our-secrets-out-f21...	1	0	pavel_
3	12578989	algorithmic music	http://cacm.acm.org/magazines/2011/7/109891-al...	1	0	poindon
4	12578979	How the Data Vault Enables the Next-Gen Data W...	https://www.talend.com/blog/2016/05/12/talend-...	1	0	markga
...	...	...	...	...	...	
293114	10176919	Ask HN: What is/are your favorite quote(s)?	NaN	15	20	kum
293115	10176917	Attention and awareness in stage magic: turnin...	http://people.cs.uchicago.edu/~luitien/nrn2473...	14	0	st
293116	10176908	Dying vets fuck you letter (2013)	http://dangerousminds.net/comments/dying_vets_...	10	2	mycodeb
293117	10176907	PHP 7 Coolest Features: Space Ships, Type Hint...	https://www.zend.com/en/resources/php-7	2	0	Gar
293118	10176903	Toyota Establishes Research Centers with MIT a...	http://newsroom.toyota.co.jp/en/detail/9233109/	4	0	tir

293119 rows × 7 columns

In [32]:
```
data.head()
```

Out[32]:

	id	title	url	num_points	num_comments	author	cr
0	12579008	You have two days to comment if you want stem ...	http://www.regulations.gov/document?D=FDA-2015...	1	0	altstar	9
1	12579005	SQLAR the SQLite Archiver	https://www.sqlite.org/sqlar/doc/trunk/README.md	1	0	blacksqr	9
2	12578997	What if we just printed a flatscreen televisio...	https://medium.com/vanmoof/our-secrets-out-f21...	1	0	pavel_lishin	9
3	12578989	algorithmic music	http://cacm.acm.org/magazines/2011/7/109891-al...	1	0	poindontcare	9
4	12578979	How the Data Vault Enables the Next-Gen Data W...	https://www.talend.com/blog/2016/05/12/talend-...	1	0	markgainor1	9

In [33]:
```
doc="You have two days to comment if you want stem cells to be classified as your own
```

In [34]:
```
nltk.download('punkt_tab')
nltk.download('averaged_perceptron_tagger_eng')
```

```
[nltk_data] Downloading package punkt_tab to
[nltk_data]     /home/student/nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger_eng to
[nltk_data]     /home/student/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger_eng is already up-to-
[nltk_data]       date!
```

Out[34]: True

In [35]:
```
tokens = word_tokenize(doc)
print("Tokens:", tokens)
```

```
Tokens: ['You', 'have', 'two', 'days', 'to', 'comment', 'if', 'you', 'want', 'stem',
'cells', 'to', 'be', 'classified', 'as', 'your', 'own', '.']
```

In [36]:
```
pos_tags=pos_tag(tokens)
print("PDS Tags:",pos_tags)
```

```
PDS Tags: [('You', 'PRP'), ('have', 'VBP'), ('two', 'CD'), ('days', 'NNS'), ('to',
'TO'), ('comment', 'VB'), ('if', 'IN'), ('you', 'PRP'), ('want', 'VBP'), ('stem', 'J
J'), ('cells', 'NNS'), ('to', 'TO'), ('be', 'VB'), ('classified', 'VBN'), ('as', 'I
N'), ('your', 'PRP$'), ('own', 'JJ'), ('.', '.')]
```

In [37]:
```
stop_words = set(stopwords.words('english'))
filtered_tokens = [word for word in tokens if word.lower() not in stop_words]
print("Filtered Tokens:", filtered_tokens)
```

```
Filtered Tokens: ['two', 'days', 'comment', 'want', 'stem', 'cells', 'classified',
'.']
```

In [38]:
```
tokens=[word for word in tokens if word not in string.punctuation]
print("Tokens after removing punctuation:",filtered_tokens)
```

```
Tokens after removing punctuation: ['two', 'days', 'comment', 'want', 'stem', 'cell
s', 'classified', '.']
```


In [39]:
```
stemmer = PorterStemmer()
stemmed_words = [stemmer.stem(word) for word in tokens]

print("Original Tokens:", filtered_tokens)
print("Stemmed Tokens:", stemmed_words)
```

```
Original Tokens: ['two', 'days', 'comment', 'want', 'stem', 'cells', 'classified',
'.']
Stemmed Tokens: ['you', 'have', 'two', 'day', 'to', 'comment', 'if', 'you', 'want',
'stem', 'cell', 'to', 'be', 'classifi', 'as', 'your', 'own']
```

In [40]:
```
lemmatizer = WordNetLemmatizer()
lemmatized_words = [lemmatizer.lemmatize(word) for word in tokens]

print("Original Tokens:", filtered_tokens)
print("Lemmatized Tokens:", lemmatized_words)
```

```
Original Tokens: ['two', 'days', 'comment', 'want', 'stem', 'cells', 'classified',
'.']
Lemmatized Tokens: ['You', 'have', 'two', 'day', 'to', 'comment', 'if', 'you', 'wan
t', 'stem', 'cell', 'to', 'be', 'classified', 'a', 'your', 'own']
```

In [41]:
```
tf = Counter(tokens)

print("Term Frequency (TF):")
for word, freq in tf.items():
    print(f"{word}: {freq}")
```

```
Term Frequency (TF):
You: 1
have: 1
two: 1
days: 1
to: 2
comment: 1
if: 1
you: 1
want: 1
stem: 1
cells: 1
be: 1
classified: 1
as: 1
your: 1
own: 1
```

In [42]:
```
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer

docs = [
    "NLP helps machines understand human language.",
    "Chatbots use NLP for better interactions.",
    "Machine translation is an NLP application."
]

vectorizer = TfidfVectorizer(use_idf=True)
vectorizer.fit_transform(docs)

idf_values = dict(zip(vectorizer.get_feature_names_out(), vectorizer.idf_))

print("Inverse Document Frequency (IDF):")
for word, idf in idf_values.items():
    print(f"{word}: {idf:.4f}")
```

```
Inverse Document Frequency (IDF):
an: 1.6931
application: 1.6931
better: 1.6931
chatbots: 1.6931
for: 1.6931
helps: 1.6931
human: 1.6931
interactions: 1.6931
is: 1.6931
language: 1.6931
machine: 1.6931
machines: 1.6931
nlp: 1.0000
translation: 1.6931
understand: 1.6931
use: 1.6931
```

In [ ]:
```
```