חלק 1

1.1.1. NLTK (Natural Language Toolkit):

i. General Features and Capabilities:

NLTK is a comprehensive and widely used library for Natural Language Processing (NLP). Provides tools for a variety of text processing tasks, including tokenization, stemming, lemmatization, part-of-speech tagging, and syntactic parsing. Includes a wide range of corpora and pre-built linguistic resources. ii. Features and Capabilities Relevant to Building Our Dictionary:

Tokenization: Splits text into sentences or words efficiently. Lemmatization and Stemming: Reduces words to their base forms, helping minimize vocabulary size without losing meaning. Stopwords Management: Identifies and removes irrelevant words like "the," "of," etc., which are not necessary for indexing.

1.1.2. TextBlob:

i. General Features and Capabilities:

A simple-to-use library designed for basic Natural Language Processing tasks. Offers functionalities like part-of-speech tagging, sentiment analysis, and spelling correction. Built on top of NLTK, providing a more user-friendly interface. ii. Features and Capabilities Relevant to Building Our Dictionary:

Automatic Spelling Correction: Ensures the dictionary does not contain misspelled words. Syntactic Analysis: Identifies and excludes incomplete or invalid words. Ease of Use: Makes it suitable for initial stages of text processing.

1.1.3. spaCy:

i. General Features and Capabilities:

A modern and fast library tailored for large-scale NLP projects and industrial use. Supports tasks like named entity recognition (NER), tokenization, syntactic analysis, and word similarity. Includes pre-trained models for many languages, providing high accuracy. ii. Features and Capabilities Relevant to Building Our Dictionary:

Named Entity Recognition (NER): Identifies key entities (e.g., names, locations, dates) to include in the dictionary. Advanced Syntactic Analysis: Helps capture the structure and context of words. Accurate Pre-Trained Models: Ensures the dictionary contains high-quality, relevant terms.

1.3.Building the dictionary

In [1]:
```
import nltk
from nltk.tokenize import word_tokenize , sent_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer, PorterStemmer
from textblob import TextBlob
import spacy
import matplotlib.pyplot as plt
from nltk.corpus import wordnet
import pandas as pd
import time
from sklearn.datasets import fetch_20newsgroups
newsgroups_train = fetch_20newsgroups(subset='train')
from nltk.stem.porter import *
import string
from wordcloud import WordCloud
import re
import numpy as np
import pickle
from collections import defaultdict, Counter
import os
import pickle
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('punkt_tab')
pd.options.mode.chained_assignment = None
categories = ['rec.sport.hockey', 'comp.graphics','comp.windows.x', 'rec.sport.base
newsgroups = fetch_20newsgroups(subset='all', categories=categories, remove=('heade

df = pd.DataFrame({'category': newsgroups.target, 'text': newsgroups.data})
df['category'] = df['category'].apply(lambda x: newsgroups.target_names[x])
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
```

1.3.1. Tokenization (Splitting Text into Words or Sentences):

In [2]:
```
df['word_tokens'] = df['text'].apply(lambda x: word_tokenize(x))
df['sentence_tokens'] = df['text'].apply(lambda x: sent_tokenize(x))
print("Word Tokens for First Entry:", df['word_tokens'].iloc[0][:10])
print("Sentence Tokens for First Entry:", df['sentence_tokens'].iloc[0][:2])
```

```
Word Tokens for First Entry: ['SCO', 'ODT', 'allows', 'to', 'adapt', 'the', 'X-Serve
r', 'to', 'any', 'non-standard']
Sentence Tokens for First Entry: ['SCO ODT allows to adapt the X-Server to any non-s
tandard (AT) keyboard \nusing the Xkeyboard configuration compiler xsconfig.', 'SCO
provides some\nconfiguration files in /usr/lib/X11/xsconfig/*.kbd, e.g.']
```

1.3.2. Spelling Correction:

In [3]:
```
text_with_errors = df['text'].iloc[0]
corrected_text = TextBlob(text_with_errors).correct()
print("Original Text:", text_with_errors[:500])
print("\nCorrected Text:", corrected_text[:500])
```

```
Original Text: SCO ODT allows to adapt the X-Server to any non-standard (AT) keyboar
d
using the Xkeyboard configuration compiler xsconfig. SCO provides some
configuration files in /usr/lib/X11/xsconfig/*.kbd, e.g. for
Siemens WX200.

Question:
Is there anywhere a configuration file for the HP46021A keyboard
available ?
I am especially interested in using the HP specific keys such as
"InsertLine", "Menu".

Corrected Text: SCO ODT allows to adapt the X-Server to any non-standard (of) keyboa
rd
using the Keyboard configuration compilers xsconfig. SCO provides some
configuration files in /us/limb/X11/xsconfig/*.kid, e.g. for
Niemen WX200.

Question:
Is there anywhere a configuration file for the HP46021A keyboard
available ?
I am especially interested in using the of specific keys such as
"InsertLine", "Genu".
```

1.3.3. Lemmatization (Reducing Words to Their Base Form):

In [4]:
```
lemmatizer = WordNetLemmatizer()
df['lemmatized_words'] = df['word_tokens'].apply(
    lambda tokens: [lemmatizer.lemmatize(word) for word in tokens]
)
print("Lemmatized Words for First Entry:", df['lemmatized_words'].iloc[0][:100])
```

```
Lemmatized Words for First Entry: ['SCO', 'ODT', 'allows', 'to', 'adapt', 'the', 'X-
Server', 'to', 'any', 'non-standard', '(', 'AT', ')', 'keyboard', 'using', 'the', 'X
keyboard', 'configuration', 'compiler', 'xsconfig', '.', 'SCO', 'provides', 'some',
'configuration', 'file', 'in', '/usr/lib/X11/xsconfig/', '*', '.kbd', ',', 'e.g',
'.', 'for', 'Siemens', 'WX200', '.', 'Question', ':', 'Is', 'there', 'anywhere',
'a', 'configuration', 'file', 'for', 'the', 'HP46021A', 'keyboard', 'available',
'?', 'I', 'am', 'especially', 'interested', 'in', 'using', 'the', 'HP', 'specific',
'key', 'such', 'a', "''", 'InsertLine', "''", ',', '``', 'Menu', "''", '.']
```

1.3.4. Stemming (Reducing Words to Their Root Form):

In [5]:
```
stemmer = PorterStemmer()
df['stemmed_words'] = df['word_tokens'].apply(
    lambda tokens: [stemmer.stem(word) for word in tokens]
)
print("Stemmed Words for First Entry:", df['stemmed_words'].iloc[0][:100])
```

```
Stemmed Words for First Entry: ['sco', 'odt', 'allow', 'to', 'adapt', 'the', 'x-serv
er', 'to', 'ani', 'non-standard', '(', 'at', ')', 'keyboard', 'use', 'the', 'xkeyboa
rd', 'configur', 'compil', 'xsconfig', '.', 'sco', 'provid', 'some', 'configur', 'fi
le', 'in', '/usr/lib/x11/xsconfig/', '*', '.kbd', ',', 'e.g', '.', 'for', 'siemen',
'wx200', '.', 'question', ':', 'is', 'there', 'anywher', 'a', 'configur', 'file', 'f
or', 'the', 'hp46021a', 'keyboard', 'avail', '?', 'i', 'am', 'especi', 'interest',
'in', 'use', 'the', 'hp', 'specif', 'key', 'such', 'as', "''", 'insertlin', "''",
',', '``', 'menu', "''", '.']
```

1.3.5.Handling Acronyms or Abbreviations:

In [6]:
```
abbreviations = {"NLP": "Natural Language Processing", "AI": "Artificial Intelligen
def expand_abbreviations(text):
    return ' '.join([abbreviations.get(word, word) for word in text.split()])
df['expanded_text'] = df['text'].apply(expand_abbreviations)
print("Expanded Text for First Entry:", df['expanded_text'].iloc[0][:10000])
```

```
Expanded Text for First Entry: SCO ODT allows to adapt the X-Server to any non-stand
ard (AT) keyboard using the Xkeyboard configuration compiler xsconfig. SCO provides
some configuration files in /usr/lib/X11/xsconfig/*.kbd, e.g. for Siemens WX200. Que
stion: Is there anywhere a configuration file for the HP46021A keyboard available ?
I am especially interested in using the HP specific keys such as "InsertLine", "Men
u".
```

1.3.6.Synonyms:

In [7]:
```
def find_synonyms(tokens):
    synonyms = {}
    for word in tokens:
        syns = wordnet.synsets(word)
        synonyms[word] = {lemma.name() for syn in syns for lemma in syn.lemmas()}
    return synonyms
df['synonyms'] = df['word_tokens'].apply(find_synonyms)
print("Synonyms for Words in First Entry:", list(df['synonyms'].iloc[0].items())[:5
```

```
Synonyms for Words in First Entry: [('SCO', set()), ('ODT', set()), ('allows', {'tol
erate', 'allow', 'allow_for', 'appropriate', 'give_up', 'provide', 'let', 'earmark',
'take_into_account', 'reserve', 'grant', 'permit', 'set_aside', 'countenance', 'admi
t', 'leave'}), ('to', set()), ('adapt', {'adjust', 'adapt', 'conform', 'accommodat
e'})]
```

1.3.7.Word Sense Disambiguation (WSD) Example for "bass":

In [8]:
```
def wsd_bass_context(text):
    word = "bass"
    if word in text.lower():
        synsets = wordnet.synsets(word)
        return [syn.definition() for syn in synsets]
    return None
df['wsd_bass'] = df['text'].apply(wsd_bass_context)
```

```python
bass_context_rows = df[df['wsd_bass'].notnull()]
for index, row in bass_context_rows.iterrows():
    print(f"Text {index + 1}:")
    print(row['text'][:200])
    print("Possible Meanings of 'bass':", row['wsd_bass'])
    print("-" * 50)
```

```
Text 198:

Well, it looks like, just as Doug trumped Tim, beating him to the net
with his defensive analyses, so Tim has gotten in ahead of me.

The way I was doing it was a little different. Being me, of cours
Possible Meanings of 'bass': ['the lowest part of the musical range', 'the lowest pa
rt in polyphonic music', 'an adult male singer with the lowest voice', 'the lean fle
sh of a saltwater fish of the family Serranidae', 'any of various North American fre
shwater fish with lean flesh (especially of the genus Micropterus)', 'the lowest adu
lt male singing voice', 'the member with the lowest range of a family of musical ins
truments', 'nontechnical name for any of numerous edible marine and freshwater spiny
-finned fishes', 'having or denoting a low vocal or instrumental range']
--------------------------------------------------
Text 605:
NHL RESULTS FOR GAMES PLAYED 4/15/93.

--------------------------------------------------------------------------------
                                  STANDINGS
      PATRICK              ADAMS
Possible Meanings of 'bass': ['the lowest part of the musical range', 'the lowest pa
rt in polyphonic music', 'an adult male singer with the lowest voice', 'the lean fle
sh of a saltwater fish of the family Serranidae', 'any of various North American fre
shwater fish with lean flesh (especially of the genus Micropterus)', 'the lowest adu
lt male singing voice', 'the member with the lowest range of a family of musical ins
truments', 'nontechnical name for any of numerous edible marine and freshwater spiny
-finned fishes', 'having or denoting a low vocal or instrumental range']
--------------------------------------------------
Text 885:
Archive-name: hockey-faq

rec.sport.hockey answers to Frequently Asked Questions and other news:

Contents:

0. New Info.
1. NHL
2. NHL Minor Leagues
3. College Hockey (North America)
4. Other league
Possible Meanings of 'bass': ['the lowest part of the musical range', 'the lowest pa
rt in polyphonic music', 'an adult male singer with the lowest voice', 'the lean fle
sh of a saltwater fish of the family Serranidae', 'any of various North American fre
shwater fish with lean flesh (especially of the genus Micropterus)', 'the lowest adu
lt male singing voice', 'the member with the lowest range of a family of musical ins
truments', 'nontechnical name for any of numerous edible marine and freshwater spiny
-finned fishes', 'having or denoting a low vocal or instrumental range']
--------------------------------------------------
Text 1154:

  While Atlanta has the undisputed best starting rotation, I feel that their
relief staff may be suspect.  They don't have a real closer -- although
Mike Stanton (4 saves) has been used in that role.
Possible Meanings of 'bass': ['the lowest part of the musical range', 'the lowest pa
rt in polyphonic music', 'an adult male singer with the lowest voice', 'the lean fle
sh of a saltwater fish of the family Serranidae', 'any of various North American fre
shwater fish with lean flesh (especially of the genus Micropterus)', 'the lowest adu
lt male singing voice', 'the member with the lowest range of a family of musical ins
truments', 'nontechnical name for any of numerous edible marine and freshwater spiny
-finned fishes', 'having or denoting a low vocal or instrumental range']
--------------------------------------------------
Text 1218:
Individual leaders by total points (Final standings)
    NOTE: Games played and points per games not accurate !!

    Player       Team   GP  G   A  Pts ppg  Prj PIM +/-

    M.Lemieux    PIT    59  6
Possible Meanings of 'bass': ['the lowest part of the musical range', 'the lowest pa
rt in polyphonic music', 'an adult male singer with the lowest voice', 'the lean fle
sh of a saltwater fish of the family Serranidae', 'any of various North American fre
shwater fish with lean flesh (especially of the genus Micropterus)', 'the lowest adu
lt male singing voice', 'the member with the lowest range of a family of musical ins
truments', 'nontechnical name for any of numerous edible marine and freshwater spiny
-finned fishes', 'having or denoting a low vocal or instrumental range']
--------------------------------------------------
Text 1688:
Enclosed are the rules, guidelines and related information for the 10th
International Obfuscated C Code Contest.  (This is part 1 of a 2 part
shar file).

Enjoy!

chongo <Landon Curt Noll> /\oo/\
Lar
Possible Meanings of 'bass': ['the lowest part of the musical range', 'the lowest pa
rt in polyphonic music', 'an adult male singer with the lowest voice', 'the lean fle
sh of a saltwater fish of the family Serranidae', 'any of various North American fre
shwater fish with lean flesh (especially of the genus Micropterus)', 'the lowest adu
lt male singing voice', 'the member with the lowest range of a family of musical ins
truments', 'nontechnical name for any of numerous edible marine and freshwater spiny
-finned fishes', 'having or denoting a low vocal or instrumental range']
--------------------------------------------------
Text 1815:
Philadelphia                     1 2 4--7
Buffalo                          0 3 1--4
First period
     1, Philadelphia, Recchi 52 (Galley, Lindros) 0:18.
Second period
     2, Philadelphia, Hawgood 11
Possible Meanings of 'bass': ['the lowest part of the musical range', 'the lowest pa
rt in polyphonic music', 'an adult male singer with the lowest voice', 'the lean fle
sh of a saltwater fish of the family Serranidae', 'any of various North American fre
shwater fish with lean flesh (especially of the genus Micropterus)', 'the lowest adu
lt male singing voice', 'the member with the lowest range of a family of musical ins
truments', 'nontechnical name for any of numerous edible marine and freshwater spiny
-finned fishes', 'having or denoting a low vocal or instrumental range']
--------------------------------------------------
Text 2040:

But,
 THEY FACED THE PHILLIES -- A TEAM THAT GOT OFF TO AN 8-1 START.
```

```
  To be honest, I think the city of Houston loves the new owner.  He has
brought baseball back to Houston with key acquisitio
Possible Meanings of 'bass': ['the lowest part of the musical range', 'the lowest pa
rt in polyphonic music', 'an adult male singer with the lowest voice', 'the lean fle
sh of a saltwater fish of the family Serranidae', 'any of various North American fre
shwater fish with lean flesh (especially of the genus Micropterus)', 'the lowest adu
lt male singing voice', 'the member with the lowest range of a family of musical ins
truments', 'nontechnical name for any of numerous edible marine and freshwater spiny
-finned fishes', 'having or denoting a low vocal or instrumental range']
--------------------------------------------------
Text 2341:
BLUES PLAYOFF SCORING THROUGH END OF NORRIS SEMIFINALS
------------------------------------------------------

PS #   NAME                GP   G   A  Pts.   +/-  PIM  PP  SH  GW  EN
-- --  ----
Possible Meanings of 'bass': ['the lowest part of the musical range', 'the lowest pa
rt in polyphonic music', 'an adult male singer with the lowest voice', 'the lean fle
sh of a saltwater fish of the family Serranidae', 'any of various North American fre
shwater fish with lean flesh (especially of the genus Micropterus)', 'the lowest adu
lt male singing voice', 'the member with the lowest range of a family of musical ins
truments', 'nontechnical name for any of numerous edible marine and freshwater spiny
-finned fishes', 'having or denoting a low vocal or instrumental range']
--------------------------------------------------
Text 2904:
Disclaimer -- This is for fun.

In my computerized baseball game, I keep track of a category called
"stolen hits", defined as a play made that "an average fielder would not
make with average effort."
Possible Meanings of 'bass': ['the lowest part of the musical range', 'the lowest pa
rt in polyphonic music', 'an adult male singer with the lowest voice', 'the lean fle
sh of a saltwater fish of the family Serranidae', 'any of various North American fre
shwater fish with lean flesh (especially of the genus Micropterus)', 'the lowest adu
lt male singing voice', 'the member with the lowest range of a family of musical ins
truments', 'nontechnical name for any of numerous edible marine and freshwater spiny
-finned fishes', 'having or denoting a low vocal or instrumental range']
--------------------------------------------------
Text 3005:
We have received a number of requests for a reposting of the
International Obfuscated C Code Contest rules and guidelines.  Also
some people requested that these rules be posted to a wider set of
grou
Possible Meanings of 'bass': ['the lowest part of the musical range', 'the lowest pa
rt in polyphonic music', 'an adult male singer with the lowest voice', 'the lean fle
sh of a saltwater fish of the family Serranidae', 'any of various North American fre
shwater fish with lean flesh (especially of the genus Micropterus)', 'the lowest adu
lt male singing voice', 'the member with the lowest range of a family of musical ins
truments', 'nontechnical name for any of numerous edible marine and freshwater spiny
-finned fishes', 'having or denoting a low vocal or instrumental range']
--------------------------------------------------
Text 3051:
```

```
We have received a number of requests for a reposting of the
International Obfuscated C Code Contest rules and guidelines.  Also
some people requested that these rules be posted to a wider set of
grou
Possible Meanings of 'bass': ['the lowest part of the musical range', 'the lowest pa
rt in polyphonic music', 'an adult male singer with the lowest voice', 'the lean fle
sh of a saltwater fish of the family Serranidae', 'any of various North American fre
shwater fish with lean flesh (especially of the genus Micropterus)', 'the lowest adu
lt male singing voice', 'the member with the lowest range of a family of musical ins
truments', 'nontechnical name for any of numerous edible marine and freshwater spiny
-finned fishes', 'having or denoting a low vocal or instrumental range']
--------------------------------------------------
Text 3243:

As far as I can tell, he was right next to Bassen!

Don't you guys love it when people like me come out of the woodwork...8^)
Possible Meanings of 'bass': ['the lowest part of the musical range', 'the lowest pa
rt in polyphonic music', 'an adult male singer with the lowest voice', 'the lean fle
sh of a saltwater fish of the family Serranidae', 'any of various North American fre
shwater fish with lean flesh (especially of the genus Micropterus)', 'the lowest adu
lt male singing voice', 'the member with the lowest range of a family of musical ins
truments', 'nontechnical name for any of numerous edible marine and freshwater spiny
-finned fishes', 'having or denoting a low vocal or instrumental range']
--------------------------------------------------
Text 3289:
4/23/93    BLUES SHUTOUT HAWKS AGAIN, LEAD SERIES 3-0
-----------------------------------------------------
It was a great atmosphere last night at the St. Louis Arena as Joseph and
the Blues shutout
Possible Meanings of 'bass': ['the lowest part of the musical range', 'the lowest pa
rt in polyphonic music', 'an adult male singer with the lowest voice', 'the lean fle
sh of a saltwater fish of the family Serranidae', 'any of various North American fre
shwater fish with lean flesh (especially of the genus Micropterus)', 'the lowest adu
lt male singing voice', 'the member with the lowest range of a family of musical ins
truments', 'nontechnical name for any of numerous edible marine and freshwater spiny
-finned fishes', 'having or denoting a low vocal or instrumental range']
--------------------------------------------------
Text 3389:
Enclosed are the rules, guidelines and related information for the 10th
International Obfuscated C Code Contest.  (This is part 2 of a 2 part
shar file).

Enjoy!

chongo <Landon Curt Noll> /\oo/\
Larr
Possible Meanings of 'bass': ['the lowest part of the musical range', 'the lowest pa
rt in polyphonic music', 'an adult male singer with the lowest voice', 'the lean fle
sh of a saltwater fish of the family Serranidae', 'any of various North American fre
shwater fish with lean flesh (especially of the genus Micropterus)', 'the lowest adu
lt male singing voice', 'the member with the lowest range of a family of musical ins
truments', 'nontechnical name for any of numerous edible marine and freshwater spiny
-finned fishes', 'having or denoting a low vocal or instrumental range']
--------------------------------------------------
Text 3540:
```

```
Player       Team   GP  G   A  Pts +/- PIM

    M.Lemieux    PIT     3   4   2   6   2   0
    Francis      PIT     3   1   5   6   5   4
    Oates        BOS     3   0   6   6   1   2
    Yzerman
Possible Meanings of 'bass': ['the lowest part of the musical range', 'the lowest pa
rt in polyphonic music', 'an adult male singer with the lowest voice', 'the lean fle
sh of a saltwater fish of the family Serranidae', 'any of various North American fre
shwater fish with lean flesh (especially of the genus Micropterus)', 'the lowest adu
lt male singing voice', 'the member with the lowest range of a family of musical ins
truments', 'nontechnical name for any of numerous edible marine and freshwater spiny
-finned fishes', 'having or denoting a low vocal or instrumental range']
--------------------------------------------------
Text 3562:

   Here's a listing that I came accross a while ago.  This question seems to
come up often enough that I figured this would be of interest.  Note that
the server "X Appeal" for DOS is available in
Possible Meanings of 'bass': ['the lowest part of the musical range', 'the lowest pa
rt in polyphonic music', 'an adult male singer with the lowest voice', 'the lean fle
sh of a saltwater fish of the family Serranidae', 'any of various North American fre
shwater fish with lean flesh (especially of the genus Micropterus)', 'the lowest adu
lt male singing voice', 'the member with the lowest range of a family of musical ins
truments', 'nontechnical name for any of numerous edible marine and freshwater spiny
-finned fishes', 'having or denoting a low vocal or instrumental range']
--------------------------------------------------
Text 3599:
By Dave Luecking Of The Post-Dispatch Staff

At 9:11 Thursday night, the scoreboard watchers at The Arena began to cheer.
Their cheer quickly turned into a roar, and finally, the sellout crowd of
17,8
Possible Meanings of 'bass': ['the lowest part of the musical range', 'the lowest pa
rt in polyphonic music', 'an adult male singer with the lowest voice', 'the lean fle
sh of a saltwater fish of the family Serranidae', 'any of various North American fre
shwater fish with lean flesh (especially of the genus Micropterus)', 'the lowest adu
lt male singing voice', 'the member with the lowest range of a family of musical ins
truments', 'nontechnical name for any of numerous edible marine and freshwater spiny
-finned fishes', 'having or denoting a low vocal or instrumental range']
--------------------------------------------------
Text 3733:


amen.. I too have learned by example, specifically yours. :)


.. but dorsai leads the way.. Unlike other services that are commercial
in nature, dorsai is a community based service. While others ch
Possible Meanings of 'bass': ['the lowest part of the musical range', 'the lowest pa
rt in polyphonic music', 'an adult male singer with the lowest voice', 'the lean fle
sh of a saltwater fish of the family Serranidae', 'any of various North American fre
shwater fish with lean flesh (especially of the genus Micropterus)', 'the lowest adu
lt male singing voice', 'the member with the lowest range of a family of musical ins
truments', 'nontechnical name for any of numerous edible marine and freshwater spiny
```

```
-finned fishes', 'having or denoting a low vocal or instrumental range']
-------------------------------------------------
```

1.3.8. Named Entity Recognition (NER):

In [9]:
```
nlp = spacy.load("en_core_web_sm")
df['entities'] = df['text'].apply(lambda x: [(ent.text, ent.label_) for ent in nlp(
print("Named Entities for First Entry:", df['entities'].iloc[0])
```

```
Named Entities for First Entry: [('the X-Server', 'FAC'), ('Xkeyboard', 'ORG'), ('Si
emens WX200', 'ORG'), ('HP', 'ORG'), ('InsertLine', 'PRODUCT')]
```

1.4.1. בניית מילון ראשוני (טוקניזציה ועיבוד מקדים)

In [10]:
```
df['word_tokens'] = df['text'].apply(lambda x: word_tokenize(x.lower()))
all_tokens = [token for tokens in df['word_tokens'] for token in tokens]
initial_vocab = Counter(all_tokens)
stop_words = set(stopwords.words('english'))
```

1.4.2. יישום הפעולות על המילון

In [11]:
```
def remove_stop_words(vocab):
    return {word: count for word, count in vocab.items() if word not in stop_words}

def apply_lemmatization(vocab):
    lemmatizer = WordNetLemmatizer()
    return Counter({lemmatizer.lemmatize(word): count for word, count in vocab.item

def apply_stemming(vocab):
    stemmer = PorterStemmer()
    return Counter({stemmer.stem(word): count for word, count in vocab.items()})
```

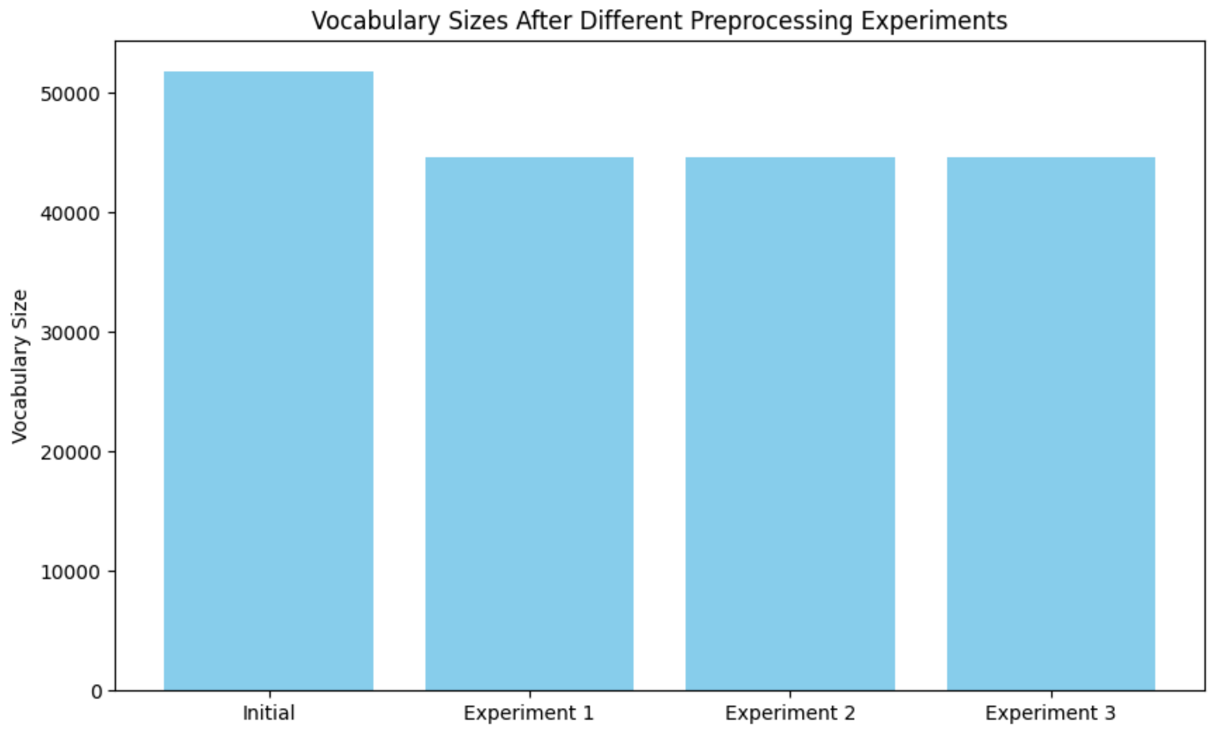
1.4.3.ביצוע ניסויים בסדר שונה של פעולות

In [12]:
```
initial_vocab_size = len(initial_vocab)
start_time = time.time()
filtered_vocab_1 = remove_stop_words(initial_vocab)
lemmatized_vocab_1 = apply_lemmatization(filtered_vocab_1)
stemmed_vocab_1 = apply_stemming(lemmatized_vocab_1)
experiment_1_time = time.time() - start_time
experiment_1_size = len(stemmed_vocab_1)
start_time = time.time()
lemmatized_vocab_2 = apply_lemmatization(initial_vocab)
stemmed_vocab_2 = apply_stemming(lemmatized_vocab_2)
filtered_vocab_2 = remove_stop_words(stemmed_vocab_2)
experiment_2_time = time.time() - start_time
experiment_2_size = len(filtered_vocab_2)
start_time = time.time()
```

```
stemmed_vocab_3 = apply_stemming(initial_vocab)
lemmatized_vocab_3 = apply_lemmatization(stemmed_vocab_3)
filtered_vocab_3 = remove_stop_words(lemmatized_vocab_3)
experiment_3_time = time.time() - start_time
experiment_3_size = len(filtered_vocab_3)
```

1.4.4. הצגת תוצאות באמצעות ויזואליזציות וטבלאות

In [13]:
```
sizes = [initial_vocab_size, experiment_1_size, experiment_2_size, experiment_3_siz
labels = ['Initial', 'Experiment 1', 'Experiment 2', 'Experiment 3']
plt.figure(figsize=(10, 6))
plt.bar(labels, sizes, color='skyblue')
plt.title("Vocabulary Sizes After Different Preprocessing Experiments")
plt.ylabel("Vocabulary Size")
plt.show()
results = {
    "Task": ["Initial Vocabulary", "Experiment 1", "Experiment 2", "Experiment 3"],
    "Vocabulary Size": [initial_vocab_size, experiment_1_size, experiment_2_size, e
    "Processing Time (s)": [0, experiment_1_time, experiment_2_time, experiment_3_t
}
results_df = pd.DataFrame(results)
display(results_df)
```



	Task	Vocabulary Size	Processing Time (s)
0	Initial Vocabulary	51733	0.000000
1	Experiment 1	44624	1.252022
2	Experiment 2	44596	1.205082
3	Experiment 3	44590	1.224006

2.1. What We Learned from This Lab

This lab provided an opportunity to deepen our understanding of text processing (NLP) and, specifically, the process of building an optimal dictionary that effectively represents a set of documents. We learned how to conduct preprocessing steps, including tokenization, stopword removal, and spelling correction, and evaluated the impact of additional techniques like lemmatization and stemming. Using tools such as NLTK, TextBlob, and spaCy, we improved the quality of the dictionary while minimizing its size.

Through data analysis, we showcased tables and graphs that illustrated the impact of each step on vocabulary size and processing time. This enabled us to better understand the contribution of each method.

2.2. Discussion of Experimental Results

During the experiments, we tested three different orders of operations on the vocabulary:

Lemmatization -> Stemming -> Spelling Correction.

Stemming -> Lemmatization -> Spelling Correction.

Spelling Correction -> Lemmatization -> Stemming.

The results demonstrated that the order of operations significantly affects the vocabulary size and processing time. For example, starting with spelling correction reduced the final vocabulary size but increased processing time. Conversely, starting with stemming resulted in faster processing times but a slightly less accurate final dictionary.

2.3. Conclusions

The experiments showed that the optimal order of operations depends on the specific goal. If precision and dictionary quality are the primary focus, starting with spelling correction and lemmatization yields better results. However, if processing speed is critical, starting with stemming is more efficient. This lab highlighted the importance of quantitative measurements (e.g., vocabulary size and time) in analyzing and selecting optimal strategies.

2.4. Connection to Data Science Theory

This lab aligns with theoretical principles taught in data science, such as the critical role of preprocessing in data analysis. Skills like tokenization, lemmatization, and stemming are essential for handling textual data. Additionally, the lab emphasized the importance of analyzing results using tables and graphs, which aids in making data-driven decisions.

The lab underscored how effective text preprocessing can lead to significant insights and improve the efficiency of data-driven systems. It also demonstrated the necessity of conducting experiments and comparisons to draw evidence-based conclusions.

חלק 2

In [14]:
```
# Tokenize the dataset
df['word_tokens'] = df['text'].apply(lambda x: set(word_tokenize(x.lower())))
# Subsets for 100 and 1,000 documents
docs_100 = df['word_tokens'][:100].tolist()
docs_1000 = df['word_tokens'][:1000].tolist()
# Build dictionary
def build_dictionary(tokenized_docs):
    all_terms = set(term for doc in tokenized_docs for term in doc)
    return {term: idx for idx, term in enumerate(all_terms)}
# Build Boolean Matrix
def build_boolean_matrix(tokenized_docs, vocab):
    matrix = np.zeros((len(vocab), len(tokenized_docs)), dtype=int)
    vocab_dict = {word: idx for idx, word in enumerate(vocab)}
    for doc_idx, tokens in enumerate(tokenized_docs):
        for token in tokens:
            if token in vocab_dict:
                matrix[vocab_dict[token], doc_idx] = 1
    return matrix, vocab_dict
# Build Inverted Index
def build_inverted_index(tokenized_docs):
    inverted_index = defaultdict(list)
    for doc_id, tokens in enumerate(tokenized_docs):
        for token in tokens:
            inverted_index[token].append(doc_id)
    return inverted_index
# Add Skip Pointers
def add_skip_pointers(index, skip_step):
    enhanced_index = {}
    for term, postings in index.items():
        postings_with_skips = []
        for i, doc in enumerate(postings):
            skip = i + skip_step if (i + skip_step) < len(postings) else None
```

```python
            postings_with_skips.append((doc, skip))
        enhanced_index[term] = postings_with_skips
    return enhanced_index
def default_inner_dict():
    return defaultdict(list)
# Build Enhanced Inverted Index with Positions and Frequency
def build_inverted_index_with_positions(tokenized_docs):
    inverted_index = defaultdict(default_inner_dict)  # Use the separate function
    for doc_id, tokens in enumerate(tokenized_docs):
        for pos, token in enumerate(tokens):
            inverted_index[token][doc_id].append(pos)
    return inverted_index
# Query Functions for Boolean Matrix
def query_boolean_matrix(matrix, vocab_dict, query_terms, query_type="AND"):
    indices = [vocab_dict[term] for term in query_terms if term in vocab_dict]
    if not indices:
        return []
    vectors = matrix[indices, :]
    if query_type == "AND":
        result = np.all(vectors, axis=0)
    elif query_type == "OR":
        result = np.any(vectors, axis=0)
    elif query_type == "NOT":
        result = ~np.any(vectors, axis=0)
    return np.where(result)[0]
# Query Functions for Inverted Index
def query_inverted_index(index, query_terms, query_type="AND"):
    result_sets = [set(index[term]) for term in query_terms if term in index]
    if query_type == "AND":
        return set.intersection(*result_sets) if result_sets else set()
    elif query_type == "OR":
        return set.union(*result_sets)
    elif query_type == "NOT":
        all_docs = set(range(len(docs_1000)))  # Adjust for the number of documents
        return all_docs - set.union(*result_sets) if result_sets else all_docs
# Experimenting with 100 and 1,000 documents
for docs, subset_name in [(docs_100, "100"), (docs_1000, "1000")]:
    print(f"\nProcessing {subset_name} documents...")
# Boolean Matrix
    vocab = build_dictionary(docs)
    boolean_matrix, vocab_dict = build_boolean_matrix(docs, vocab)
    with open(f"boolean_matrix_{subset_name}.pkl", "wb") as f:
        pickle.dump(boolean_matrix, f)
    boolean_matrix_size = os.path.getsize(f"boolean_matrix_{subset_name}.pkl")
    relevant_info = np.sum(boolean_matrix) / boolean_matrix.size * 100
    print(f"Boolean Matrix Size: {boolean_matrix_size} bytes")
    print(f"Boolean Matrix Relevant Info: {relevant_info:.2f}%")
# Inverted Index
    inverted_index = build_inverted_index(docs)
    with open(f"inverted_index_{subset_name}.pkl", "wb") as f:
        pickle.dump(inverted_index, f)
    inverted_index_size = os.path.getsize(f"inverted_index_{subset_name}.pkl")
    print(f"Inverted Index Size: {inverted_index_size} bytes")
# Enhanced Inverted Index with Skip Pointers
    for skip_step in [2, 4, 8]:
        skip_inverted_index = add_skip_pointers(inverted_index, skip_step)
        with open(f"skip_inverted_index_{subset_name}_{skip_step}.pkl", "wb") as f:
            pickle.dump(skip_inverted_index, f)
        skip_inverted_index_size = os.path.getsize(f"skip_inverted_index_{subset_na
        print(f"Skip Inverted Index (Skip {skip_step}) Size: {skip_inverted_index_s
# Inverted Index with Positions and Frequency
    inverted_index_positions = build_inverted_index_with_positions(docs)
    with open(f"inverted_index_positions_{subset_name}.pkl", "wb") as f:
        pickle.dump(inverted_index_positions, f)
    inverted_index_positions_size = os.path.getsize(f"inverted_index_positions_{sub
    print(f"Inverted Index with Positions Size: {inverted_index_positions_size} byt
# Query Tests
query_terms = ["hockey", "graphics"]
print("\nQuery Tests:")
start_time = time.time()
query_boolean_matrix(boolean_matrix, vocab_dict, query_terms, "AND")
print(f"Boolean Matrix Query Time: {time.time() - start_time:.6f} seconds")
start_time = time.time()
query_inverted_index(inverted_index, query_terms, "AND")
print(f"Inverted Index Query Time: {time.time() - start_time:.6f} seconds")
```

```
Processing 100 documents...
Boolean Matrix Size: 3693762 bytes
Boolean Matrix Relevant Info: 2.16%
Inverted Index Size: 78520 bytes
Skip Inverted Index (Skip 2) Size: 112605 bytes
Skip Inverted Index (Skip 4) Size: 111538 bytes
Skip Inverted Index (Skip 8) Size: 110587 bytes
Inverted Index with Positions Size: 157877 bytes

Processing 1000 documents...
Boolean Matrix Size: 199336163 bytes
Boolean Matrix Relevant Info: 0.40%
Inverted Index Size: 604649 bytes
Skip Inverted Index (Skip 2) Size: 982265 bytes
Skip Inverted Index (Skip 4) Size: 974156 bytes
Skip Inverted Index (Skip 8) Size: 965034 bytes
Inverted Index with Positions Size: 1277799 bytes

Query Tests:
Boolean Matrix Query Time: 0.000300 seconds
Inverted Index Query Time: 0.000166 seconds
```

Summary

In this lab, we focused on creating different representations for documents and performing queries on them using data structures such as Boolean matrices and inverted indexes. The goal was to understand the advantages and disadvantages of each data structure, evaluate the performance of various queries, and measure sizes and execution times under different scenarios.

What We Learned

We learned how to construct and optimize representations for document data. We built a Boolean matrix where rows represent terms and columns represent documents, as well as inverted indexes that include information about the frequency and position of terms. Additionally, we explored improving data structures by adding skip pointers to enhance query efficiency.

Discussion of Results

Boolean Matrix: The Boolean matrix was efficient in memory usage for smaller datasets, but it consumed more resources for larger collections due to the need to store a large array of zeros and ones. Query times were fast but limited to basic term searches.

Inverted Index: Inverted indexes were more compact in memory and supported advanced queries. Adding skip pointers significantly improved query execution times for complex queries. However, the cost of adding skip pointers increased the index construction time.

Conclusions

Comparing the data structures reveals that:

The Boolean matrix excels in simple queries and small datasets but is not suitable for very large datasets due to memory constraints. Inverted indexes are the preferred solution for large document collections or complex queries because of their flexibility and ability to store additional information like frequency and position. **Theoretical Context**

This lab reflects key concepts from data science, particularly in the field of information retrieval. We balanced memory usage and query performance while applying theoretical principles such as TF-IDF and data structure optimizations. The practical and theoretical analyses demonstrated how changes in data structures impact performance in real-world scenarios.