```
# Import libraries
import pandas as pd
from sklearn.datasets import fetch_20newsgroups
from nltk.corpus import stopwords
import nltk
nltk.download('stopwords')
from sklearn.feature_extraction.text import TfidfVectorizer
from gensim.models import Word2Vec
import gensim.downloader as api
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
# Load the Word2Vec model
model = api.load('word2vec-google-news-300')
```

```
[==================================================] 100.0% 1662.8/1662.8MB downloaded
```

```
# Load the 20newsgroups dataset
newsgroups_train = fetch_20newsgroups(subset='train')
```

```
# Takes lists of texts and their titles then finds the most unique word (keyword) in each text according to TF-IDF
def get_top_tfidf_word(texts, category_name):
    stop_words = stopwords.words('english')
    vectorizer = TfidfVectorizer(stop_words=stop_words) # Creates vectorizer including stopwords to be ignored
    vectors = vectorizer.fit_transform(texts) # Fits text to vectorizer by transforming to TF-IDF matrix
    feature_names = vectorizer.get_feature_names_out() # Gets words of associated TF-IDF scores
    dense = vectors.todense() # Turns sparse TF-IDF matrix into dense
    denselist = dense.tolist() # Makes dense matrix a list of lists
    df = pd.DataFrame(denselist, columns=feature_names) # Creates dataframe with words from dataset as columns
    top_word_idx = df.sum().argmax() # Finds the index position with the highest TF-IDF score summed up across all documents in the category
    top_word = feature_names[top_word_idx] # Gets the word at that position
    top_tfidf = df.sum().max() # Gets the TF-IDF score for that word
    return top_word, top_tfidf
```

```
top_words = [] # Empty list to store keywords
# Loops through each category in dataset and outputs the unique keyword with its TF-IDF score and category it belongs to
for category in newsgroups_train.target_names:
    category_index = newsgroups_train.target_names.index(category) # Locates index position of category
    category_mask = (newsgroups_train.target == category_index) # Specifies the documents that belong to category using boolean mask
    category_texts = [text for text, mask in zip(newsgroups_train.data, category_mask) if mask] # Filters out the documents that do not belong to category
    top_word, top_tfidf = get_top_tfidf_word(category_texts, category) # Calls keyword function
    top_words.append((top_word, top_tfidf, category)) # Adds the word, its TF-IDF score, and category to list of keywords
    print(f"Top word for category '{category}': {top_word} (TF-IDF score: {top_tfidf:.4f})")
```

```
Top word for category 'alt.atheism': edu (TF-IDF score: 24.0282)
Top word for category 'comp.graphics': edu (TF-IDF score: 22.0857)
Top word for category 'comp.os.ms-windows.misc': edu (TF-IDF score: 23.7632)
Top word for category 'comp.sys.ibm.pc.hardware': edu (TF-IDF score: 20.7596)
Top word for category 'comp.sys.mac.hardware': edu (TF-IDF score: 24.8660)
Top word for category 'comp.windows.x': edu (TF-IDF score: 19.3980)
Top word for category 'misc.forsale': edu (TF-IDF score: 23.9920)
Top word for category 'rec.autos': edu (TF-IDF score: 23.1517)
Top word for category 'rec.motorcycles': com (TF-IDF score: 23.2428)
Top word for category 'rec.sport.baseball': edu (TF-IDF score: 29.9561)
Top word for category 'rec.sport.hockey': edu (TF-IDF score: 25.9939)
Top word for category 'sci.crypt': key (TF-IDF score: 22.3453)
Top word for category 'sci.electronics': edu (TF-IDF score: 21.5093)
Top word for category 'sci.med': edu (TF-IDF score: 26.1149)
Top word for category 'sci.space': edu (TF-IDF score: 24.3572)
Top word for category 'soc.religion.christian': god (TF-IDF score: 26.0375)
Top word for category 'talk.politics.guns': edu (TF-IDF score: 24.1822)
Top word for category 'talk.politics.mideast': edu (TF-IDF score: 22.9387)
Top word for category 'talk.politics.misc': edu (TF-IDF score: 20.3259)
Top word for category 'talk.religion.misc': edu (TF-IDF score: 14.6385)
```

```
print("\nKeyword Similarities:")
# Calculate similarities between top words
for i in range(len(top_words)):
    word1, tfidf1, category1 = top_words[i]
    most_similar_word = None
    max_similarity = -1
    for j in range(len(top_words)):
        if i != j: # Makes sure word is from different category
            word2, tfidf2, category2 = top_words[j]
            try:
                similarity = model.similarity(word1, word2) # Calculates similarity using word2vec representations
                if similarity > max_similarity: # Checks to find the most similar keyword in list of keywords
                    max_similarity = similarity
                    most_similar_word = (word2, category2)
            except KeyError:
                continue
    if most_similar_word:
        print(f"'{word1}' ({category1}) is most similar to '{most_similar_word[0]}' ({most_similar_word[1]}) with similarity: {max_similarity:.4f}")
    else:
        print(f"No similar word found for '{word1}' ({category1})")
```

```
Keyword Similarities:
'edu' (alt.atheism) is most similar to 'edu' (comp.graphics) with similarity: 1.0000
'edu' (comp.graphics) is most similar to 'edu' (alt.atheism) with similarity: 1.0000
'edu' (comp.os.ms-windows.misc) is most similar to 'edu' (alt.atheism) with similarity: 1.0000
'edu' (comp.sys.ibm.pc.hardware) is most similar to 'edu' (alt.atheism) with similarity: 1.0000
'edu' (comp.sys.mac.hardware) is most similar to 'edu' (alt.atheism) with similarity: 1.0000
'edu' (comp.windows.x) is most similar to 'edu' (alt.atheism) with similarity: 1.0000
'edu' (misc.forsale) is most similar to 'edu' (alt.atheism) with similarity: 1.0000
'edu' (rec.autos) is most similar to 'edu' (alt.atheism) with similarity: 1.0000
'com' (rec.motorcycles) is most similar to 'edu' (alt.atheism) with similarity: 0.5552
'edu' (rec.sport.baseball) is most similar to 'edu' (alt.atheism) with similarity: 1.0000
```

```
'edu' (rec.sport.hockey) is most similar to 'edu' (alt.atheism) with similarity: 1.0000
'key' (sci.crypt) is most similar to 'edu' (alt.atheism) with similarity: 0.0109
'edu' (sci.electronics) is most similar to 'edu' (alt.atheism) with similarity: 1.0000
'edu' (sci.med) is most similar to 'edu' (alt.atheism) with similarity: 1.0000
'edu' (sci.space) is most similar to 'edu' (alt.atheism) with similarity: 1.0000
'god' (soc.religion.christian) is most similar to 'edu' (alt.atheism) with similarity: 0.1593
'edu' (talk.politics.guns) is most similar to 'edu' (alt.atheism) with similarity: 1.0000
'edu' (talk.politics.mideast) is most similar to 'edu' (alt.atheism) with similarity: 1.0000
'edu' (talk.politics.misc) is most similar to 'edu' (alt.atheism) with similarity: 1.0000
'edu' (talk.religion.misc) is most similar to 'edu' (alt.atheism) with similarity: 1.0000
```