```
import nltk
import pickle
from nltk.tokenize import word_tokenize
from nltk.util import ngrams
from collections import Counter
import string

# Download NLTK data for tokenization
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     /Users/charanlokku/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
True
```

```
def build_language_model(file_path):
    with open(file_path, 'r', encoding='utf-8') as file:
        content = file.read().replace('\n', ' ')

    # Convert to lowercase
    content = content.lower()

    # Tokenize
    tokens = nltk.word_tokenize(content)

    # Filter out punctuation and digits
    tokens = [token for token in tokens if token.isalnum()]

    # Create bigrams and unigrams
    bigrams = [(tokens[i], tokens[i+1]) for i in range(len(tokens)-1)]
    unigrams = tokens

    # Count occurrences of bigrams and unigrams
    bigram_freq = Counter(bigrams)
    unigram_freq = Counter(unigrams)

    # Create dictionaries for bigrams and unigrams
    bigram_dict = {' '.join(bigram): count for bigram, count in bigram_freq.items()}
    unigram_dict = {unigram: count for unigram, count in unigram_freq.items()}

    return unigram_dict, bigram_dict

# Process English training data
english_unigrams, english_bigrams = build_language_model('LangId.train.English.txt')
# Process French training data
french_unigrams, french_bigrams = build_language_model('LangId.train.French.txt')
# Process Italian training data
italian_unigrams, italian_bigrams = build_language_model('LangId.train.Italian.txt')

def save_model(filename, unigrams, bigrams):
    with open(filename, 'wb') as file:
        pickle.dump((unigrams, bigrams), file)

save_model('english_lang_model.pkl', english_unigrams, english_bigrams)
save_model('french_lang_model.pkl', french_unigrams, french_bigrams)
save_model('italian_lang_model.pkl', italian_unigrams, italian_bigrams)

# Function to print the first 3 items of a dictionary or list
def first_3_lines(data, data_name):
    print(f"First 3 lines of {data_name}:")
    count = 0

    if isinstance(data, list):
        for item in data[:3]:
            print(item)
            count += 1
    else:  # If the input data is a dictionary
        for key, value in list(data.items())[:3]:
            print(key, value)
            count += 1

    print()
```

```
first_3_lines(english_unigrams, 'English unigrams')
first_3_lines(english_bigrams, 'English bigrams')
first_3_lines(french_unigrams, 'French unigrams')
first_3_lines(french_bigrams, 'French bigrams')
first_3_lines(italian_unigrams, 'Italian unigrams')
first_3_lines(italian_bigrams, 'Italian bigrams')
```

```
First 3 lines of English unigrams:
approval 3
of 2769
the 5698

First 3 lines of English bigrams:
approval of 3
of the 904
the minutes 6

First 3 lines of French unigrams:
approbation 4
du 961
procès 11

First 3 lines of French bigrams:
approbation du 2
du procès 1
procès verbal 6

First 3 lines of Italian unigrams:
approvazione 7
del 906
processo 26

First 3 lines of Italian bigrams:
approvazione del 2
del processo 5
processo verbale 6
```