```
import nltk
from utilities import forEachQuestion
import re

class Preprocessor:


    @staticmethod
    def preprocessQuestions(questions):
        print("\nPreprocessor: remove punctuation")
        forEachQuestion(questions, Preprocessor.removePunctuation)
        print("Preprocessor: words")
        forEachQuestion(questions, Preprocessor.addWords)
        print("Preprocessor: stopwords")
        forEachQuestion(questions, Preprocessor.removeStopwords)
        print("Preprocessor: parts of speech")
        forEachQuestion(questions, Preprocessor.addPartOfSpeech)
        print("Preprocessor: bigrams")
        forEachQuestion(questions, Preprocessor.addBigrams)
        print("Preprocessor: trigrams")
        forEachQuestion(questions, Preprocessor.addTrigrams)


    # This should augment the QA tree with bigram distributions for each question
    @staticmethod
    def removePunctuation(question):
        question['question_clean'] = re.sub('[^\w\s]', ' ', question['question'])
        question['question_clean'] = re.sub('[\s+]', ' ', question['question'])

    @staticmethod
    def addBigrams(question):
        question['question_bigram_list'] = list(nltk.bigrams(question['question_words']))
        question['question_bigram_list_nostopwords'] = list(nltk.bigrams(question['questio


    @staticmethod
    def addTrigrams(question):
        question['question_trigram_list'] = list(nltk.trigrams(question['question_words'])
        question['question_trigram_list_nostopwords'] = list(nltk.trigrams(question['quest


    @staticmethod
    def addPartOfSpeech(question):
        question['question_words_pos'] = nltk.pos_tag(question['question_words'])
        question['question_words_pos_nostopwords'] = nltk.pos_tag(question['question_words


    @staticmethod
    def stopwordsList():
        stopwords = nltk.corpus.stopwords.words('english')
        return stopwords


    @staticmethod
```

```
    @staticmethod
    def removeStopwords(question):
        stopwords = Preprocessor.stopwordsList()
        question['question_words_nostopwords'] = [i for i in question['question_words'] if


    @staticmethod
    def addWords(question):
        question['question_words'] = nltk.word_tokenize(question['question'])
```