Tools Used in Assignment

* Tokenizer

The tokenizer used was the PTBTokenizer from the stanford-corenlp library. It is an efficient, fast, deterministic tokenizer developed by Christopher Manning, Tim Grow, Teg Grenager, Jenny Finkel, and John Bauer. It produces Penn Treebank style tokenization of English text.. PTBTokenizer is best at tokenizing formal English writing rather than colloquial and abbreviated writing. Also stanford-corenlp DocumentPreprocessor was used to separate the data into sentences. It also uses PTBTokenizer.

* Spell Correctors

The isolated Spellchecker used was the SpellChecker from lucene-spellchecker library. It uses the provideddirectory as a spell checker index with a LevensteinDistance as the default StringDistance. The dictonary referenced for this task was the Google’s 10,000 most frequent english words. (<https://github.com/first20hours/google-10000-english/blob/master/google-10000-english.txt>).

For contextual spell checking Microsoft Bing Spellcheck API was used. (<https://api.cognitive.microsoft.com/bing/v5.0/spellcheck>)

* Stemmer

For stemming, PorterStemmer from Apache opennlp.tools libary was used.

* Lemmatizer

For Lemmatizing, SimpleLemmatizer from Apache opennlp.tools libary was used. This implemented a Dictionary based Lemmatization technique. Before the Lemmatization, the sentences were tokenized and pos tagged using POSTaggerME. For the POS Stream, en-pos-maxent.bin from OpenNLP 1.5 models was used. And for the dictionary, en-lemmatizer.dict from (<https://raw.githubusercontent.com/richardwilly98/elasticsearch-opennlp-auto-tagging/master/src/main/resources/models/en-lemmatizer.dict>) was used.

Accuracy of the tokenizer

When considering the tokenized output of the three texts, we can see that they are in the order of Research Paper, Student Feedback and Twitter Data from most accurate to least accurate. This matches with the expectation as the PTBTokenizer is better at tokenizing more formal English writing. It can also be the tokenizer had more difficulty with Twitter Data due to presence of special characters and abbreviated forms.

Impact of Spell Correction

When it comes to spell correction, both Research paper and Student Feedback seems to have fared well. There not many word which were classified as unknown. There is a small amount of false positives due to the limited size of the dictionary used (10,000 most frequent words). But when it comes to Twitter Data, there are many words classified as unknown due to presence of special characters and language features unique to Twitter (eg. Hashtag). Using a tokenizer designed for Twitter, could help in this regard.

Suitability of stemming and lemmatizing for retrieving base forms of words

Since stemming algorithms work by cutting off the end or the beginning of the word, taking into account a list of common prefixes and suffixes that can be found in an inflected word, the performance of Stemming doesn’t depend on a dictionary. In contrast lemmatizing depends on dictionaries that lemmatizing algorithms look through to link the words to lemma. Therefore, to get accurate results from lemmatization, there is a requirement for highly detailed dictionaries.

Since the dictionary used in lemmatization for this assignment is reasonably detailed, lemmatization seems to be able to retrieve base form much more accurately.