

I found two different ways to split up the sentences, however in the process realized splitting the entire answer into subparts seems like the wrong first step in automating the logic components, in my opinion, for these reasons:

1. Finding the optimal subsentences is quite messy due to poor sentence structure in responses.
  - a. Take the two responses "I multiple 16 and 45 and i got 720 and than add 140 and i got 860." and "I miltiplieded 16 and 45 and i got 720 and added 140 and got 860" found in the Steven's Laptop data.

However, their structure is marked differently. The subsections become:  
"I multiple 16 and 45 and" "i got 720 and" "than add 140" "and i got 860"

"I miltiplieded 16 and 45 and" "i got 720 and" "added 140 and got 860"

Both end with a version of "add 140 and get 860." In the second example the "got" is marked as a conjugate connecting it to the "added 140 and" while in the second example "got" is not marked as a conjugate, and so the phrase gets split in two.

2. Subsentences are not optimized to find components.
  - a. When splitting sentences into parts, often times extraneous information is added to the front and end of phrases. For example "However, they are asking, how many hours and not how much money." should become "they are asking, how many hours."

Instead, I propose we extract only the subportions of the response that are deemed potentially matching to a component, and then have the graders manually check if it matches or not. This would increase the accuracy around key phrases and make the subsections much more clean. My process would be:

1. Find key words and phrases
  - a. We can do this first manually, for example "money" with steven's laptop
  - b. And then we can find edge cases with TF-IDF
2. Find the important words surrounding the key
  - a. Use spacy, knowing the key word will allow us to extract important words surrounding the key.
3. Store all the new phrases and the component they correspond to.
4. At this point we could create a machine learning model to predict if a phrase should get the component or not

In order to make this happen, I would need a dataset with the individual component scores, as Steven's Laptop only has holistic scores. Do you have one of those available?