The main benefit of the implementation in the textbook is it's run time efficiency. All the insertions, removals, and return value methods in the positional implementation are in constant time, while for our implementation, these methods, except for append, operate in linear time.

There are multiple benefits for our implementation though. First off, our doubly linked list looks easier to implement than the positional list. And although it is less efficient in run time, one major drawback of the positional list is that you must keep track of the positional abstraction. Although this is not a major drawback, especially when working with small amounts of data, it is still a drawback compared to our doubly linked list. However, I suspect that this is worthwhile due to the runtime efficiency. Another possible issue with the positional implementation is that in order to insert or retrieve an element you must know the position of at least one of the surrounding nodes. Again, for a small amount of data I imagine this isn't really an issue, but for a large amount I could see this being a problem.