What happens if you vary the size of Hashtable?

The Hash Table stores pointers to link node heads containing goodspeak words. If you have a small hash table, then there will be less index and thus more linked nodes at one index. There will be more node datas stored in the same index. If you have a big Hash table, there will be less chances of a word having the same index. A bigger hash table will mean that we would have to look less at each index. It will take faster time for a bigger hash size because we would have less linked nodes to check from to find the data we are looking for.

Vary Bloom Filter size?

Bloom Filter stores a 1 for the hashed words. If we have a small filter size, then there will be a lot of false positives. We use a bloom filter to save time when compiling and a filter only checks if a word is in there or possibly in there. A small filter size means that there will be more likely for words to have the same hash values and the filter will have more 1's. A bigger filter size means that less hash collisions for the filter and thus, less false positives and saves more time, when checking if a word is there or not.

Move to front rule

Move_to_front rule is not necessary. It only used to save time. If you move it to the front of the linked list then if you search for it again, it would be right there and you wouldn't have to increment all the way down the linked list until you find it. Honestly, it isn't that much faster, but if you have a lot of nodes linked together, searching for the same word again would be more convenient.