I did a lot of custom text analytics. Like I said, many in the top probably built simpler models than mine, but here are some things I tried:

- made a subset of the popular posts and built a word cloud from headline & snippet (abstract is redundant to snippet), then explored and extracted some words with grepl. I did this for a bunch of patterns of words, created about 30 different variables. Here's an example of one, some of the words came from the word clouds, some from my own logical guesses (which i cross-checked with prop.table and table):

NewsTrain$isPoliticsH = as.factor(as.numeric(grepl("senat | vote | voting | politic | republican | democrat | president | gov | obama | candidate | candidacy | G.O.P. | Hillary | Clinton | Warren | Biden | Pelosi | Jeb | elect", NewsTrain$Headline, ignore.case=TRUE)))

NewsTest$isPoliticsH = as.factor(as.numeric(grepl("senat | vote | voting | politic | republican | democrat | president | gov | obama | candidate | candidacy | G.O.P. | Hillary | Clinton | Warren | Biden | Pelosi | Jeb | elect", NewsTest$Headline, ignore.case=TRUE)))

- i explored various words and groups of words in my popular subset and i looked at word clouds for the testing set of both headline and snippet. when looking at those, i explored frequent words with this basic line of code (but also look at table() without the prop.table function to see how many observations you are actually capturing, sometimes it could be just 1 or 2 and therefore not useful, want to catch things in the 100s at least):

prop.table(table(NewsTrain$Popular, grepl("word | otherword", NewsTrain$Headline, ignore.case=TRUE)),2)

- made HeadlineIsQuestion and SnippetIsQuestion variables, and they both seemed to be significant, similar to what others have mentioned, like so:

NewsTrain$HeadlineIsQuestion = as.factor(as.numeric(grepl("\\?", NewsTrain$Headline)))  
NewsTest$HeadlineIsQuestion = as.factor(as.numeric(grepl("\\?", NewsTest$Headline)))

NewsTrain$SnippetIsQuestion = as.factor(as.numeric(grepl("\\?", NewsTrain$Snippet)))  
NewsTest$SnippetIsQuestion = as.factor(as.numeric(grepl("\\?", NewsTest$Snippet)))

FYI, when creating these variables as factors (and all the other 30 variables I made from text extraction) you have to make sure they match up in the training and testing sets:

NewsTest$HeadlineIsQuestion = factor(NewsTest$HeadlineIsQuestion, levels=levels(NewsTrain$HeadlineIsQuestion))

NewsTest$SnippetIsQuestion = factor(NewsTest$SnippetIsQuestion, levels=levels(NewsTrain$SnippetIsQuestion))

- added variables for some typical stopwords extracted from the headline/snippet – "why" "where" and "how" seemed to be significant.

- did some tabling of NewsDesk vs SectionName and made some logical guesses as to where certain things should be categorized. Then assigned them, like this:

NewsTrain[,1][NewsTrain[,2] == "Opinion"] = "OpEd"  
NewsTest[,1][NewsTest[,2] == "Opinion"] = "OpEd"

NewsTrain[,2][NewsTrain[,1] == "OpEd"] = "Opinion"  
NewsTest[,2][NewsTest[,1] == "OpEd"] = "Opinion"

NewsTrain[,1][NewsTrain[,2] == "Travel"] = "Travel"  
NewsTest[,1][NewsTest[,2] == "Travel"] = "Travel"

NewsTrain[,1][NewsTrain[,2] == "Multimedia"] = "Multimedia"  
NewsTest[,1][NewsTest[,2] == "Multimedia"] = "Multimedia"

NewsTrain[,1][NewsTrain[,2] == "U.S." & NewsTrain[,1] == ""] = "National"  
NewsTest[,1][NewsTest[,2] == "U.S." & NewsTest[,1] == ""] = "National"

NewsTrain[,1][NewsTrain[,2] == "Business Day" & NewsTrain[,1] == ""] = "Business"  
NewsTest[,1][NewsTest[,2] == "Business Day" & NewsTest[,1] == ""] = "Business"

NewsTrain[,1][NewsTrain[,2] == "Crosswords/Games" & NewsTrain[,1] == ""] = "Business"  
NewsTest[,1][NewsTest[,2] == "Crosswords/Games" & NewsTest[,1] == ""] = "Business"

NewsTrain[,2][NewsTrain[,1] == "TStyle"] = "Magazine"  
NewsTest[,2][NewsTest[,1] == "TStyle"] = "Magazine"

NewsTrain[,2][NewsTrain[,1] == "Styles" & NewsTrain[,2] == ""] = "Style"  
NewsTest[,2][NewsTest[,1] == "Styles" & NewsTest[,2] == ""] = "Style"

NewsTest[,2][NewsTest[,1] == "Culture" & NewsTest[,2] == ""] = "Arts"

NewsTest[,1][NewsTest[,2] == "Arts" & NewsTest[,1] == ""] = "Culture"

NewsTest[,1][NewsTest[,2] == "Technology" & NewsTest[,1] == ""] = "Business"

NewsTest[,1][NewsTest[,2] == "N.Y. / Region" & NewsTest[,1] == ""] = "Metro"

NewsTrain[,2][NewsTrain[,1] == "Foreign"] = "World"  
NewsTest[,2][NewsTest[,1] == "Foreign"] = "World"

NewsTrain[,1][NewsTrain[,1] == ""] = "NewsDesk Missing"  
NewsTrain[,2][NewsTrain[,2] == ""] = "SectionName Missing"  
NewsTrain[,3][NewsTrain[,3] == ""] = "SubsectionName Missing"

NewsTest[,1][NewsTest[,1] == ""] = "NewsDesk Missing"  
NewsTest[,2][NewsTest[,2] == ""] = "SectionName Missing"  
NewsTest[,3][NewsTest[,3] == ""] = "SubsectionName Missing"

- also tried clustering on different variables (time of day, news desk, section name, other things, and made sure i preprocessed), but only found that clustering hurt my model, so didn't use in my final models.

- i DID use a corpus for headline AND for snippet, but set my sparse values so i would catch very few words and threw out others that i found insignificant:

WordsToRemoveH = c("can", "get", "make", "say", "take", "will", "another", "good", "like", "new", "york")  
CorpusHeadline = tm\_map(CorpusHeadline, removeWords, c(WordsToRemoveH, stopwords("english")))

In the end I made a random forest from all the many variables i built plus the provided variables (about 93 overall) and then tuned the ntree, nodesize, and mtry. I didn't cross-validate, but simply made predictions on my training set and calculated AUC, and i expected this to get me ~-.02 on the LB from my predicted AUC.

Probably more complicated than some of the winning models, but that's why I want to see what they did. others have posted about things like GBM and SVM, and i don't know what those are, (YET!). i read the forums but didn't explore those.

And perhaps my edge is that i'm a New Yorker and i read the NYT almost every day (except Saturday and Sunday, duh!), so that may have helped me in my assumptions with text extraction, but i never comment, so maybe i don't know.