Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 8, Issue, 03, pp.27840-27844, March, 2016

INTERNATIONAL JOURNAL
OF CURRENT RESEARCH

ISSN: 0975-833X

RESEARCH ARTICLE

DESIGN OF N-GRAM:SENTIMENT ANALYTICS TAGGER

***Kaushik Halder**

Employee of Accenture, Pursuing FPM from National School of Leadership in Marketing Research and Analytics, Pune – 412105, India

ARTICLE INFO

Article History:
Received 07th December, 2015
Received in revised form
15th January, 2016
Accepted 27th February, 2016
Published online 16th March, 2016

Key words:

Sentiment Analytics, Parser Tokenization
Stemming, Stop Words, Lemmatization
Tagging, Text wrangling, R, Sentiment package.

ABSTRACT

Sentiment Analytics is widely in use in various domains like in Retails for campaign / Recommendation. In Insurance / Financial Sectors for detecting Frauds etc. Sentiment analytics can further be part of other Analytics to enhance model capability by reducing error. It is required, almost in all domains to address various purposes. In general there is no thumb rule to prepare parser which would address almost all need across domain.

Copyright © 2016 Kaushik Halder. *This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.*

Citation: Kaushik Halder, 2016. "Design of n-gram:sentiment analytics tagger", *International Journal of Current Research,* 8, (03), 27840-27844.

INTRODUCTION

An effort has been made to generalize Sentiment Analytics across domain to some extent. Error percentage will always be there as it is hard for a parser to understand 100% all grammar. Here in present document the effort of recognizing part of speech has been confined to "Noun", "Pronoun" (only "it"), "Adjective" and "Negation" (which includes words like No, Not, Never etc and no complex statements). The theory developed was partially based on Finite Automata theory. Sentiment package in the R was referred and it's corpus "AFINN-111"(name of the file) was referred in the development new and generic theory, which is expected to address border dimensions. This particular corpus contains huge list of Adjectives and it's scores.

An N-gram parser is developed for Tagger in much simpler form to address Sentiment Analytics. The algorithm follows below architecture and the details of algorithm discussed in below sections.

Method : Algorithm - Development

The Algorithm, developed taking Retail domain as it's base but the same algorithm can be applied to other domains with little or minimum tailor.

Algorithm: After Text Wrangling, the text is searched from Left to right with an intention of making flag 0 e.g; If first Noun is found then definitely search carries on for next adjective. Likewise if first adjective is found then next search initiated for Noun.

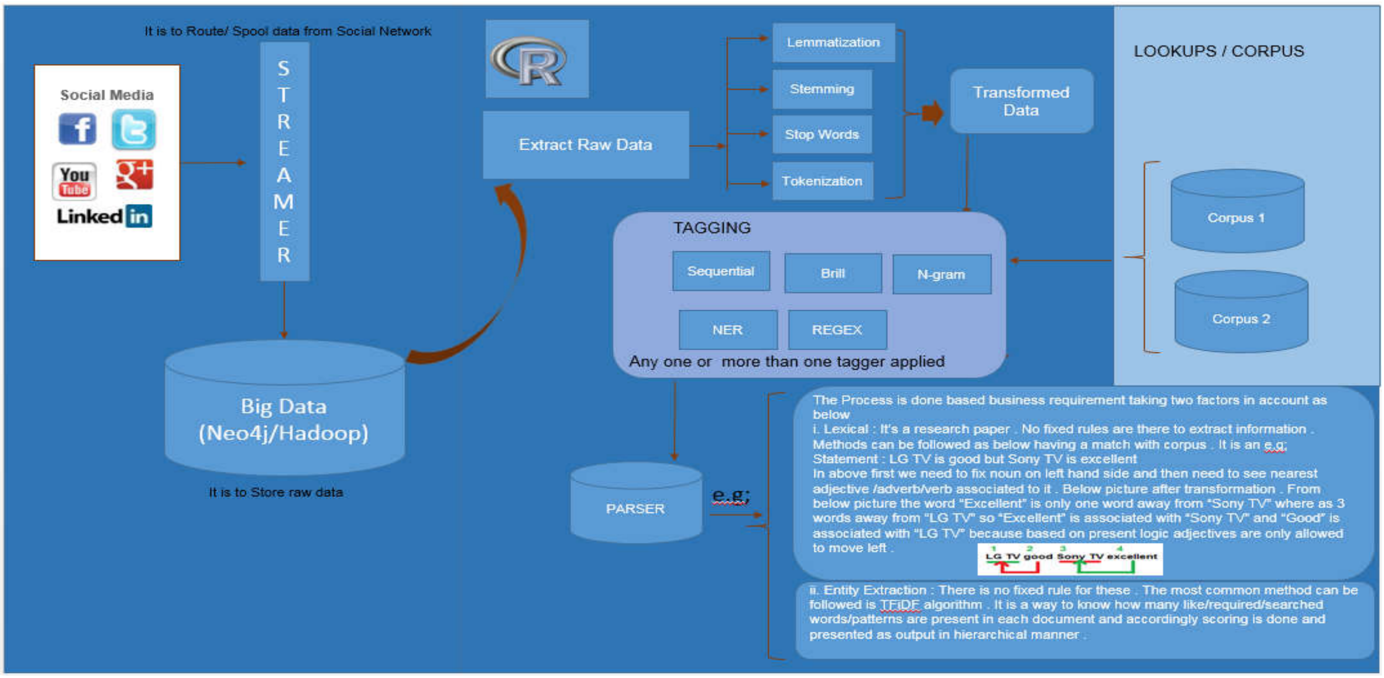
Some time search is made back if in front no opposite flag is found. Pronoun, (e.g;"It") is replaced with nearest Noun only if no Noun defined next.

Here Noun is flagged as +1 and adjective flagged as -1. After finding of both Noun and Adjective, it gets added up to 0.

The above algorithm was developed based on following hierarchical process having an assumption as follows

Assumption : Text Wrangling is preprocess before the process tagging starts and in all examples/prototypes it is assumed to be done before Tagging.

****Corresponding author: Kaushik Halder,***
Employee of Accenture, Pursuing FPM from National School of Leadership in Marketing Research and Analytics, Pune – 412105, India.



The Process is done based business requirement taking two factors in account as below

i. Lexical : It's a research paper . No fixed rules are there to extract information . Methods can be followed as below having a match with corpus . It is an e.g;
Statement : LG TV is good but Sony TV is excellent
In above first we need to fix noun on left hand side and then need to see nearest adjective /adverb/verb associated to it . Below picture after transformation . From below picture the word "Excellent" is only one word away from "Sony TV" where as 3 words away from "LG TV" so "Excellent" is associated with "Sony TV" and "Good" is associated with "LG TV" because based on present logic adjectives are only allowed to move left .

ii. Entity Extraction : There is no fixed rule for these . The most common method can be followed is TFiDF algorithm . It is a way to know how many like/required/searched words/patterns are present in each document and accordingly scoring is done and presented as output in hierarchical manner .

Tagging – Part of Speech

For simplicity and to make generic only four part of speech is referred "Noun", "Pronoun", "Adjective" and "Negation".

Four corpuses mainly followed for Adjective, for Noun, for Pronoun and for negation. For adjective the corpus "AFINN-111" (present in Sentiment package of R)is followed and the same described in session "Introduction". For Noun a separate corpus is supposed to be prepared, which may be based on product master if the industry referred is Retail etc. For pronoun, similar type corpus prepared having all feasible "Pronouns" and specifically it is also related to domain, like for retail domain the more and only pronoun used widely is "it". For Negation, similar corpus prepared having few words which would negate a statement like "No", "Never" etc. After tokenization, each word in the statement is searched in all corpuses and accordingly tagged as "Noun", "Pronoun", "adjective" or "Negation".

Assigning Adjectives to Nouns

It tries to associate object with right Sentiment by recognizing each word both right and left of Sentiment. Few prototype statements taken below for further sentiment analysis, assuming all stop words removed and only key words retained for analysis.

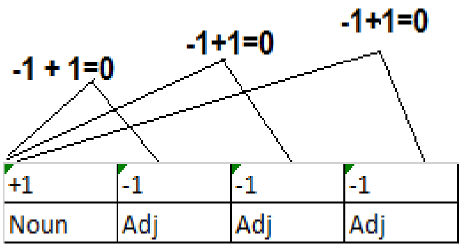
i. <Noun>, <Adj> , <Adj>, <Adj>,<Adj>, <Adj><Noun>
ii. <Noun>, <Adj> , <Adj>, <Adj>, <Adj>, <Adj><Noun>, <Adj>, <Adj>, <Adj>, <Noun>

iii. <Noun>, <Noun>, <Adj>, <Adj>, <Adj>, <Adj>, <Adj><Noun>, <Adj>, <Adj>, <Adj>, <Noun>

iv. <Noun>, <Noun>, <Adj>, <Adj>, <Adj>,<Noun>, <Adj>, <Adj>, <Adj>, <Noun>, <Adj>, <Adj>

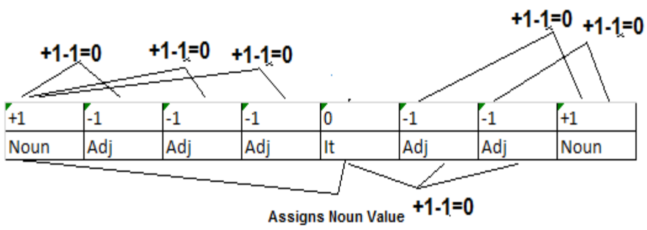
After Stop words processing, where Adjectives, pronouns, Nouns, adverbs are supposed to be retained. Each Noun and pronoun are assigned with +1 flag. Adjective and Adverbs are assigned with -1 flag. Pronouns are assigned with 0 flag. Each Noun/Pronoun gets added with nearest adjective/adverb to get sum 0. Few cases discussed below

Case I :



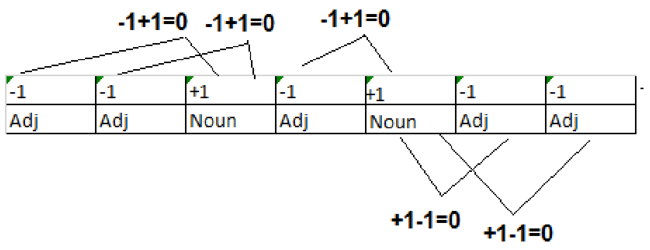
Technical Note: The search found Noun first and assigned +1 flag and next it went on assigning all adjectives/adverbs until it found next Pronoun/Noun. **Intention is to make all sum "ZERO"**. Hence, sum same Noun flag with all Adjectives and make the sum 0 and to which ever Adjectives it gets sum it gets attached to that Adjective.

Case II :



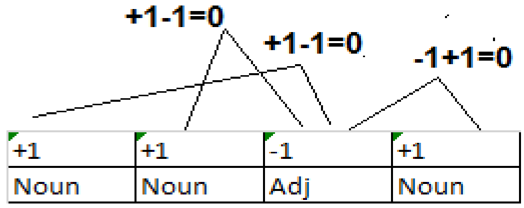
Technical Note : It is same as in Case I, except the fact that it will stop search for any more adjective after getting “It”. “It” gets replaced with previous Noun. Adjectives present after pronoun gets assign to pronoun. Further, Process flows as in Case I, keeping same intension of making sum 0.

Case III :



Technical Note : It is same as in Case II , except the fact that it will stop search for any more adjective after getting first Noun. Second search gets initiated and continues till it finds Noun/Pronoun. Third search starts, but here no Noun is found hence it moves back to find nearest Noun and the process of summation 0 carries on, keeping same intension of making sum 0.

Case IV :

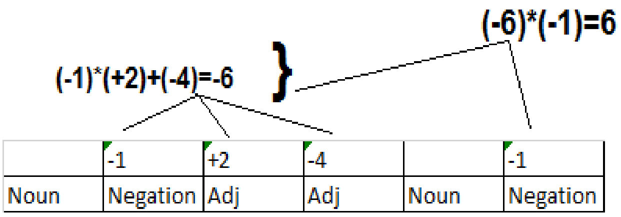


Technical Note : It is same as in Case III and in other cases ,where with intension on making Sum 0 search starts with first Noun and continues till next Noun. No adjective found in between two Nouns hence no -1 found hence search continues till it finds Adjective. First adjective got associated with both first Noun to make their summation 0. The last Noun had to move back to make it’s sum 0.

Scoring Adjectives

Scoring of Sentiments / adjectives carries by referring the corpus referred in section “Introduction” except few followings i.

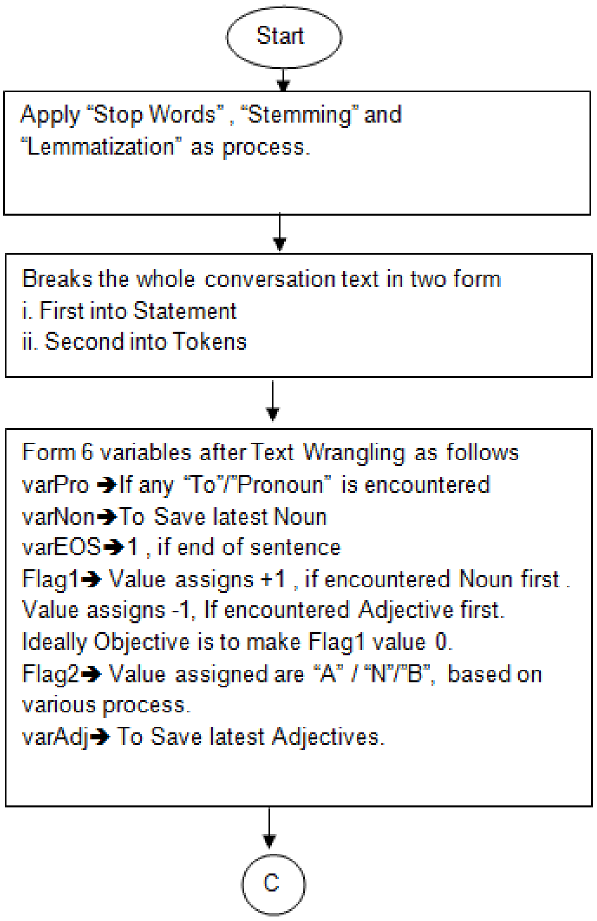
If there is any negation word e.g; “No”, “Never”, “Not” present, then next adjective gets multiplied with -1 or previous all adjectives gets multiplied with -1.



Result : Development - Process Flow

After development of N-Gram Tagger, a parser was developed where the same algorithm was applied. The Process flow for Parser is described below

Note : The process flow developed is more aligned in using R/Revo R


Start
Apply “Stop Words” , “Stemming” and “Lemmatization” as process.
Breaks the whole conversation text in two form
i. First into Statement
ii. Second into Tokens
Form 6 variables after Text Wrangling as follows
varPro ➔If any “To”/”Pronoun” is encountered
varNon➔To Save latest Noun
varEOS➔1 , if end of sentence
Flag1➔ Value assigns +1 , if encountered Noun first .
Value assigns -1, If encountered Adjective first.
Ideally Objective is to make Flag1 value 0.
Flag2➔ Value assigned are “A” / “N”/”B”, based on various process.
varAdj➔ To Save latest Adjectives.
C

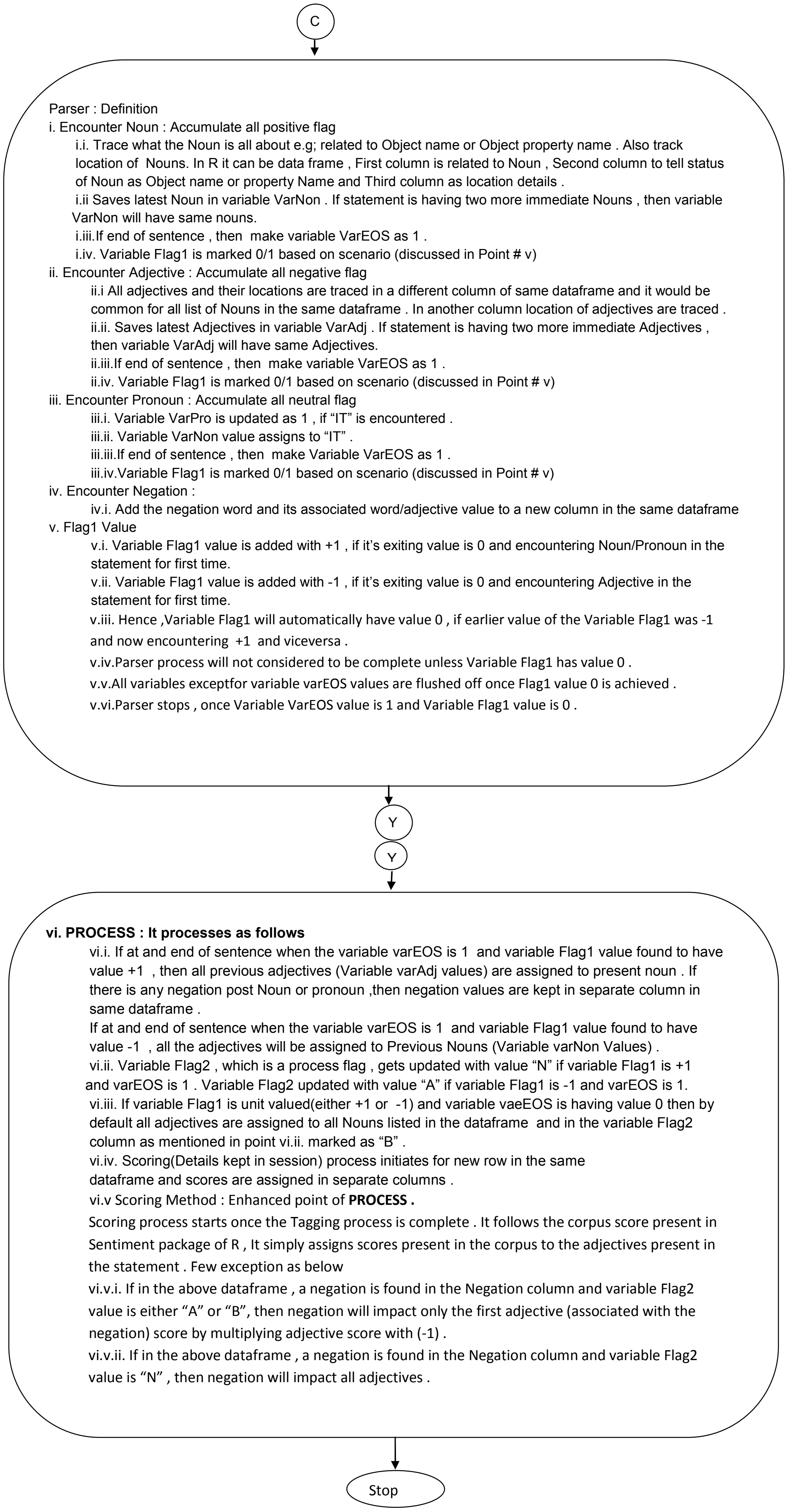
DISCUSSION

Taking an example “XYZ is excellent camera and ABC as well found to be NOT BAD”. The example is broken down in matrix format as mentioned below based on the parser defined earlier

Noun	Adjective	Pronoun	Negation	Flag	Total Sentiment
XYZ	Excellent		Not bad	N	4
ABC	Bad		Not	N	(-1)*(-2)=2
Total Sentiment					6

Explanation

After Text Wrangling, the parser is applied where the adjective “Excellent” (assumed score to be 4) got associated with Noun “XYZ” . The adjective “Bad” (assumed score to be -2) and a “Negation” got associated with Noun “ABC”. The negation score (negation score is -1) gets multiplied with the score of “Bad” and thus gives total sentiment of Noun “ABC”.



Parser : Definition

i. Encounter Noun : Accumulate all positive flag

i.i. Trace what the Noun is all about e.g; related to Object name or Object property name . Also track location of Nouns. In R it can be data frame , First column is related to Noun , Second column to tell status of Noun as Object name or property Name and Third column as location details .

i.ii Saves latest Noun in variable VarNon . If statement is having two more immediate Nouns , then variable VarNon will have same nouns.

i.iii.If end of sentence , then make variable VarEOS as 1 .

i.iv. Variable Flag1 is marked 0/1 based on scenario (discussed in Point # v)

ii. Encounter Adjective : Accumulate all negative flag

ii.i All adjectives and their locations are traced in a different column of same dataframe and it would be common for all list of Nouns in the same dataframe . In another column location of adjectives are traced .

ii.ii. Saves latest Adjectives in variable VarAdj . If statement is having two more immediate Adjectives , then variable VarAdj will have same Adjectives.

ii.iii.If end of sentence , then make variable VarEOS as 1 .

ii.iv. Variable Flag1 is marked 0/1 based on scenario (discussed in Point # v)

iii. Encounter Pronoun : Accumulate all neutral flag

iii.i. Variable VarPro is updated as 1 , if "IT" is encountered .

iii.ii. Variable VarNon value assigns to "IT" .

iii.iii.If end of sentence , then make Variable VarEOS as 1 .

iii.iv.Variable Flag1 is marked 0/1 based on scenario (discussed in Point # v)

iv. Encounter Negation :

iv.i. Add the negation word and its associated word/adjective value to a new column in the same dataframe

v. Flag1 Value

v.i. Variable Flag1 value is added with +1 , if it's exiting value is 0 and encountering Noun/Pronoun in the statement for first time.

v.ii. Variable Flag1 value is added with -1 , if it's exiting value is 0 and encountering Adjective in the statement for first time.

v.iii. Hence ,Variable Flag1 will automatically have value 0 , if earlier value of the Variable Flag1 was -1 and now encountering +1 and viceversa .

v.iv.Parser process will not considered to be complete unless Variable Flag1 has value 0 .

v.v.All variables exceptfor variable varEOS values are flushed off once Flag1 value 0 is achieved .

v.vi.Parser stops , once Variable VarEOS value is 1 and Variable Flag1 value is 0 .

vi. PROCESS : It processes as follows

vi.i. If at and end of sentence when the variable varEOS is 1 and variable Flag1 value found to have value +1 , then all previous adjectives (Variable varAdj values) are assigned to present noun . If there is any negation post Noun or pronoun ,then negation values are kept in separate column in same dataframe .

If at and end of sentence when the variable varEOS is 1 and variable Flag1 value found to have value -1 , all the adjectives will be assigned to Previous Nouns (Variable varNon Values) .

vi.ii. Variable Flag2 , which is a process flag , gets updated with value "N" if variable Flag1 is +1 and varEOS is 1 . Variable Flag2 updated with value "A" if variable Flag1 is -1 and varEOS is 1.

vi.iii. If variable Flag1 is unit valued(either +1 or -1) and variable vaeEOS is having value 0 then by default all adjectives are assigned to all Nouns listed in the dataframe and in the variable Flag2 column as mentioned in point vi.ii. marked as "B" .

vi.iv. Scoring(Details kept in session) process initiates for new row in the same dataframe and scores are assigned in separate columns .

vi.v Scoring Method : Enhanced point of **PROCESS .**

Scoring process starts once the Tagging process is complete . It follows the corpus score present in Sentiment package of R , It simply assigns scores present in the corpus to the adjectives present in the statement . Few exception as below

vi.v.i. If in the above dataframe , a negation is found in the Negation column and variable Flag2 value is either "A" or "B", then negation will impact only the first adjective (associated with the negation) score by multiplying adjective score with (-1) .

vi.v.ii. If in the above dataframe , a negation is found in the Negation column and variable Flag2 value is "N" , then negation will impact all adjectives .

Acknowledgement

Author is much grateful to Mr. Susmit Yenkar an employee of Accenture and Mr. Ashish Kumar Nayak an employee of Accenture, who developed the parser using R and tested sentiment under various scenarios. Author is grateful to his Supervisor Mr. Shantanu Biswas and his younger Brother Soumik Halder who motivated and inspired him for publication

REFERENCES

NLTK Essentials-Build cool NLP and machine learning applications using NLTK and other Python libraries - Nitin Hardeniya

Theory of Computer science–K.L.P Mishra & N.Chandrasekaran

Wikipedia : https://en.wikipedia.org/wiki/Parsing, 8-Feb-2016: 10 .16 AM

www.it-ebooks.info

****