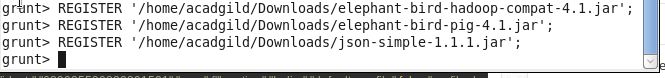
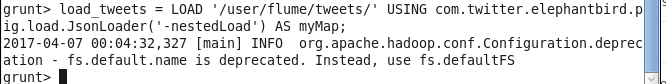
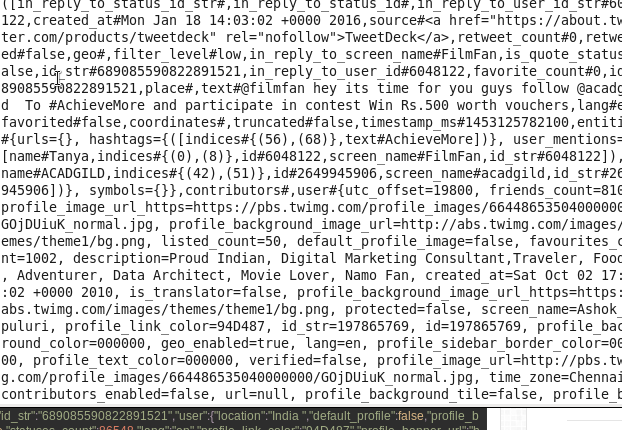
* **REGISTERED REQUIRED JARS**



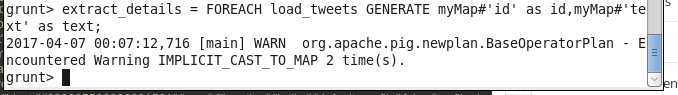
* **load\_tweets = LOAD '/user/flume/tweets/' USING com.twitter.elephantbird.pig.load.JsonLoader('-nestedLoad') AS myMap;**



* **dump load\_tweets;**



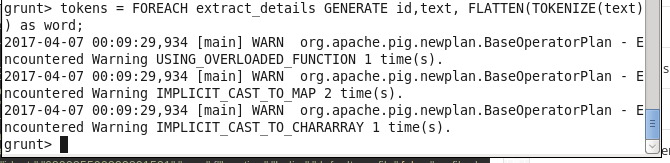
* **extract\_details = FOREACH load\_tweets GENERATE myMap#'id' as id,myMap#'text' as text;**



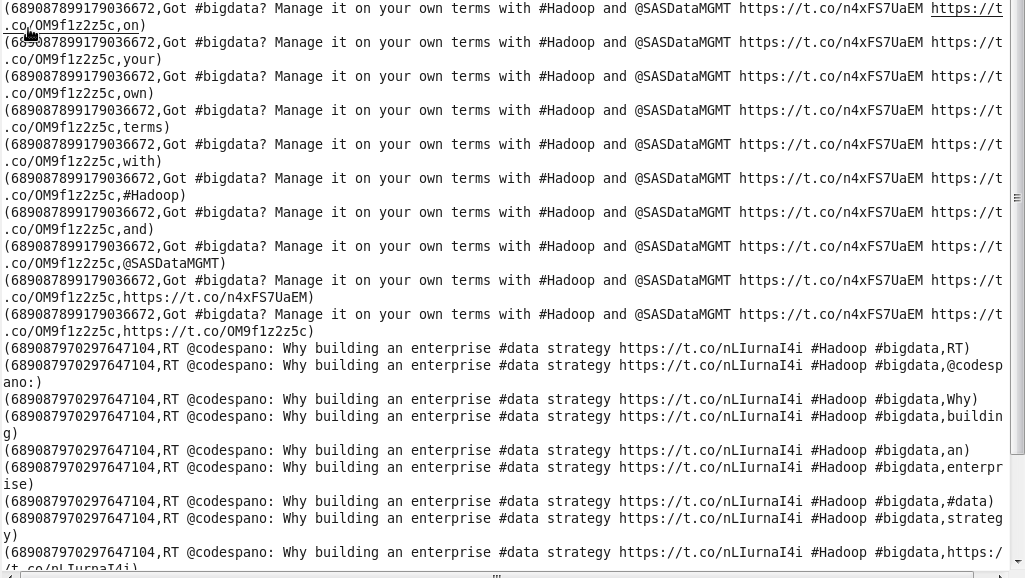
* **dump extract\_details;**



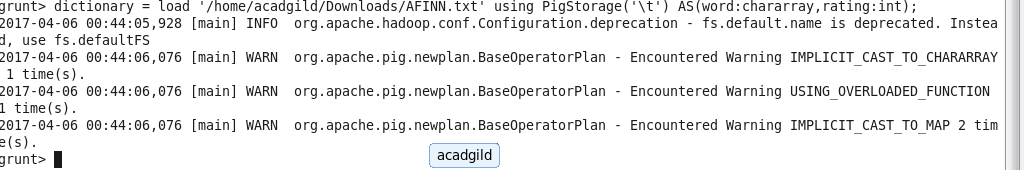
* **tokens = foreach extract\_details generate id,text, FLATTEN(TOKENIZE(text)) As word;**



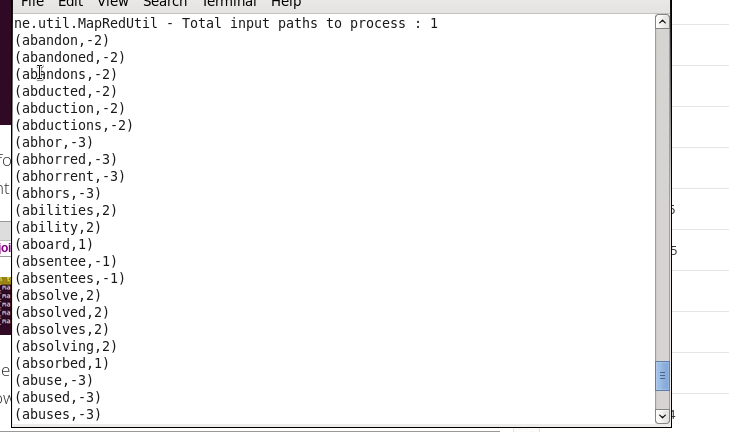
* **dump tokens;**



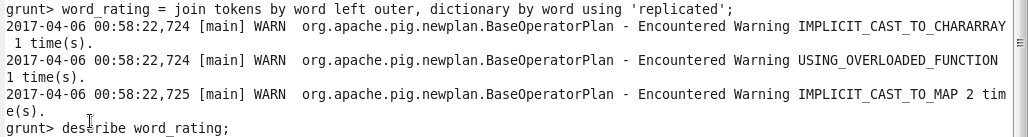
* **dictionary = load '/user/acadgild/hadoop/AFINN.txt' using PigStorage('\t') AS(word:chararray,rating:int);**



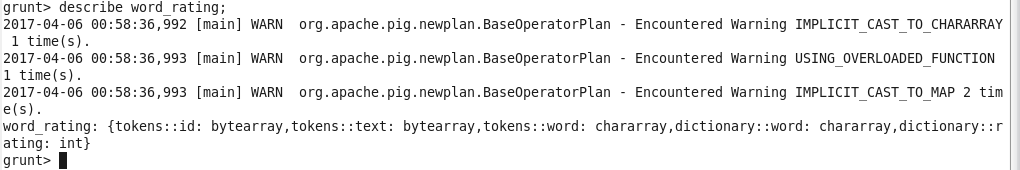
* **dump dictionary(LIMITED by 25);**



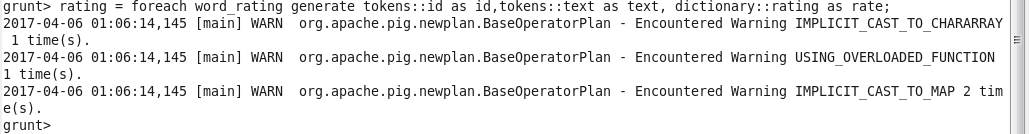
* **word\_rating = join tokens by word left outer, dictionary by word using 'replicated';**



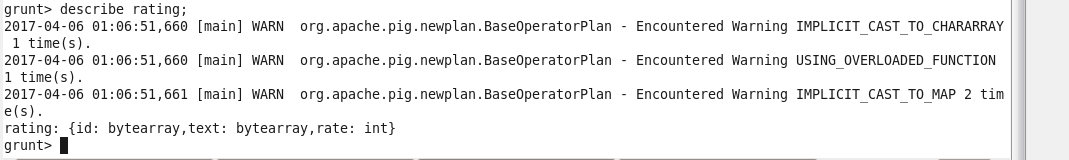
* **describe word\_rating;**



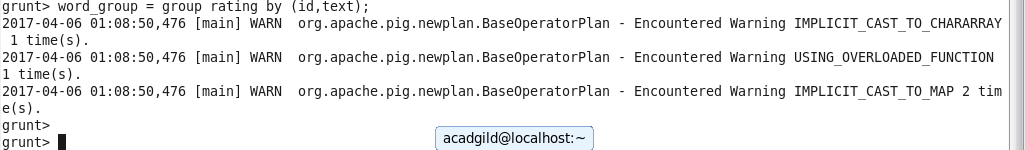
* **rating = foreach word\_rating generate tokens::id as id,tokens::text as text, dictionary::rating as rate;**



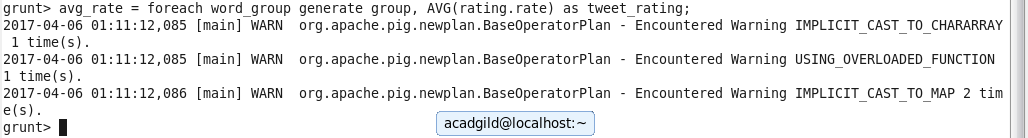
* **describe rating;**



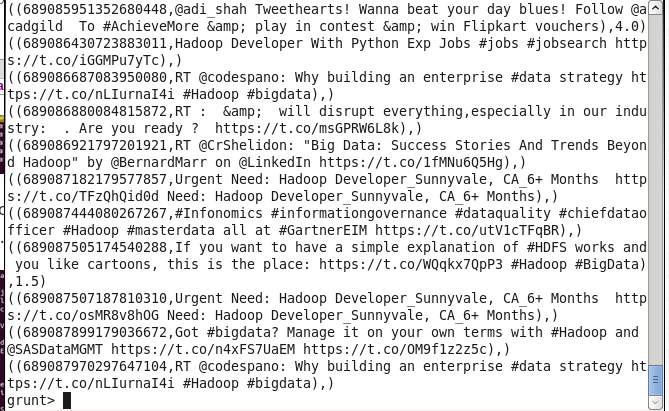
* **word\_group = group rating by (id,text);**



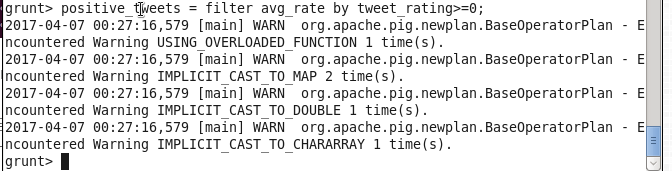
* **avg\_rate = foreach word\_group generate group, AVG(rating.rate) as tweet\_rating;**



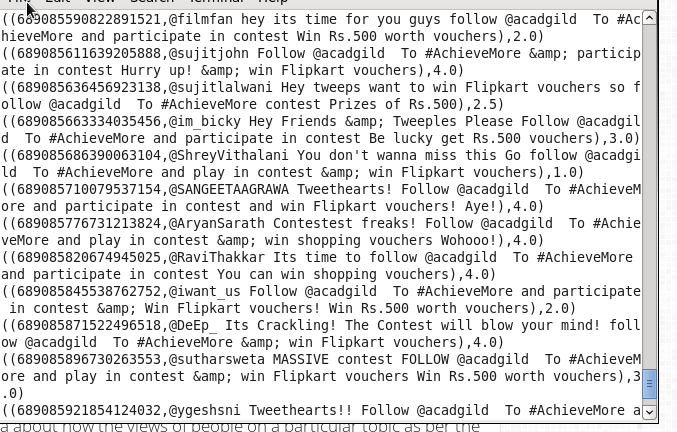
* **dump avg\_rate;**



* **positive\_tweets = filter avg\_rate by tweet\_rating>=0;**



* **dump positive\_tweets;**

****