////////////////////////////////////**DATASETS**////////////////////////////////////

Datasets used are of customer reviews of 5 products.

1. digital camera: Canon G3
2. digital camera: Nikon coolpix 4300
3. celluar phone: Nokia 6610
4. mp3 player: Creative Labs Nomad Jukebox Zen Xtra 40GB
5. dvd player: Apex AD2600 Progressive-scan DVD player

All the reviews are from amazon.com.

////////////////////////////////////**REFERENCE PAPER**////////////////////////////////////

Minqing Hu and Bing Liu. "Mining and summarizing customer reviews".
Proceedings of the ACM SIGKDD International Conference on
Knowledge Discovery & Data Mining (KDD-04), 2004.

Minqing Hu and Bing Liu. "Mining Opinion Features in Customer
Reviews." Proceedings of Nineteeth National Conference on
Artificial Intellgience (AAAI-2004), 2004.

////////////////////////////////////**SYMBOLS USED**////////////////////////////////////

Symbols used in the annotated reviews:

[t]: the title of the review: Each [t] tag starts a review.
We did not use the title information in our papers.

xxxx[+|-n]: xxxx is a product feature.

[+n]: Positive opinion, n is the opinion strength: 3 strongest,
and 1 weakest. Note that the strength is quite subjective.
You may want ignore it, but only considering + and -

[-n]: Negative opinion

: start of each sentence. Each line is a sentence.

[u] : feature not appeared in the sentence.

[p] : feature not appeared in the sentence. Pronoun resolution is needed.

[s] : suggestion or recommendation.

[cc]: comparison with a competing product from a different brand.

[cs]: comparison with a competing product from the same brand.

////////////////////////////////////**PREPROCESSING**////////////////////////////////////

- ❏ POS Tagging
  - ❏ Did POS tagging for each word in review using nltk.
- ❏ Pronoun Resolution
  - ❏ Replaced pronoun in reviews with the corresponding labelled aspects. For example if review is “get it , it's worth every penny ! !” and if there is some labeled aspect (“player”) for the above review, then we are replacing pronoun referencing “it” with the labeled aspect “player”
- ❏ Removed Contraction words
  - ❏ For example :
    - ❏ you’ve - you have
    - ❏ I’m - I am

////////////////////////////////**HANDLING NOUN PHRASES**////////////////////////////////

- ➢ We are supposed to apply algorithm for nouns and noun phrases present in the reviews independently.
- ➢ In order to find out noun phrases we are using POS tagging and grouping the two consecutive nouns as a single noun phrase.
- ➢ Our dataset is classified based on nouns and noun phrases separately.

////////////////////////////////**PARSING USING STANFORD PARSER**////////////////////////////////

- ➢ Each and every review collected from above are passed into Stanford parser.
- ➢ Stanford parser will take each review as an input and parse the review and gives the dependency rules for the review as the output.
  - o Example :
    - If review is “My name is kumar”
    - Stanford Parser’s output will be :
      - o poss(name-2, My-1)
      - o nsubj(kumar-4, name-2)
      - o cop(kumar-4, is-3)

- ➢ Stanford parser will use the EBNF(Extended Backus-Naur form) of standard english grammar and parse the reviews and generate the rules for every possible two words which are related to each other.
- ➢ The number beside each word in the rule implies the position of the that word in the review.

////////////////////////////// **EBNF** //////////////////////////////

- ➢ EBNF is a code that expresses the grammar of a formal language. An EBNF consists of terminal symbols and non-terminal production rules which are the restrictions governing how terminal symbols can be combined into a legal sequence. Examples of terminal symbols include alphanumeric characters, punctuation marks, and whitespace characters.
- ➢ The EBNF defines production rules where sequences of symbols are respectively assigned to a nonterminal:
    - ○ digit excluding zero = "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" ;
    - ○ digit = "0" | digit excluding zero ;
- ➢ This production rule defines the nonterminal *digit* which is on the left side of the assignment. The vertical bar represents an alternative and the terminal symbols are enclosed with quotation marks followed by a semicolon as terminating character. Hence a *digit* is a *0* or a *digit excluding zero* that can be *1* or *2* or *3* and so forth until *9*.
- ➢ A production rule can also include a sequence of terminals or nonterminals, each separated by a comma.

//////////////////////////////**ASPECT EXTRACTION USING 3 RULES**//////////////////////////////

- ➢ **Type 1 Rules :**
    - o Given a set of seed opinion words, we are extracting aspects from them.
    - o It can be done by the following two rule patterns,

1. if a word A, whose POS is NN, is depended by an opinion word O through Dep, where Dep is one of the dependency relations amod, prep, nsubj, csubj, xsubj, dobj and iobj, then A is an aspect.

   For Example,
   - **“The phone has a good screen ,”** we can extract the aspect **“screen”** as there is a amod relation between “good” and “screen,” given that “good” is an opinion word, and “screen” is a noun

2. if an opinion word O and a word A, whose POS is NN, depend on a third word H through dependency relations Dep_i and Dep_j respectively, where Dep_i and Dep_j are one of the relations amod, prep, nsubj, csubj, xsubj, dobj, and iobj, then A is an aspect.

For Example,

- **“iPod is the best mp3 player ,”** we can extract the aspect **“iPod”** as there is a amod relation between “best” and “player,” a nsubj relation between “player” and “iPod,” given that “best” is an opinion word, and “iPod” is a noun.

➢ **Type 2 Rules**

- Using the known aspects that are extracted in the type 1 rules, new aspects are extracted.
- It can be done by the following two rule patterns

.

1. if a word A_j whose POS is NN, directly depends on an aspect A_i through conj, then A_j is an aspect.

For Example,

- **“Does the player play dvd with audio and video?** ” we can extract “**video**” as an aspect as there is a conj relation between “audio” and “video,” given that “audio” is a known aspect.

2. if a word A_j whose POS is NN, and an aspect A_i , directly depend on a third word H through the dependency relations Dep_i and Dep_j , where Dep_i and Dep_j are one of the relations in amod, prep, nsubj, csubj, xsubj, dobj, and conj, then A_j is an aspect.

For Example,

- **“Canon G3 has a great lens ,”** we can extract **“G3”** as an aspect as there is a nsubj relation between “has” and “G3,” a dobj relation between “has” and “lens,” given that “lens” is a known aspect

➢ **Type 3 Rules**

- Here, we are using known aspects and seed opinion words to extract new opinion words. The known aspects are the one that extracted from the above two rules.
- It can be done by the following four rule patterns,
  1. if a word O, whose POS is JJ (adjective), directly depends on an aspect A through dependency relation Dep, where Dep is one of the dependency relations amod, prep, nsubj, csubj, xsubj, dobj, and iobj, then O is an opinion word.

For Example,

- **“The phone has a good screen ,”** we can extract the **opinion word “good”** as there is a amod relation between “good” and “screen,” given that “screen” is an aspect, and “good” is an adjective.

2. if a word O, whose POS is JJ, and an aspect A, directly depend on a third word H through relations Dep_i and Dep_j respectively, where Dep_i and Dep_j are one of the relations amod, prep, nsubj, csubj, xsubj, dobj,conj, and iobj, then O is an opinion word.

For Example,

- "**iPod is the best mp3 player ,**" we can extract the opinion word "best" as there is a amod relation between "best" and "player," a nsubj relation between "player" and "iPod," given that "iPod" is an aspect, and "best" is an adjective.

3. if a word O_j , whose POS is JJ, directly depends on an opinion word O_i through conj, then O_j is an opinion word.

For Example,

- **"The camera is amazing but expensive**," we can extract "expensive" as an opinion word, as there is a conj relation between "amazing" and "expensive," given that "amazing" is an known opinion word.

4. if a word O_j , whose POS is JJ, and an opinion word O_i , directly depend on a third word H through the dependance relations Dep_i and Dep_j , where Dep_i and Dep_j are one of the relations in amod, prep, nsubj, csubj, xsubj, dobj, and conj, then O_j is an opinion word.

For Example,

- "**If you want to buy a sexy, cool mp3 player, you can choose iPod,"** we can extract "cool" as an opinion word, as there is a amod relation between "sexy" and "player," a amod relation between "cool" and "player," given that "sexy" is an opinion word.

//////////////////////**RULE SET SELECTION USING GREEDY ALGORITHM**//////////////////////

As analyzed in the introduction, finding the optimal subset of rules is NP-hard, we thus propose a greedy algorithm to perform the task, which has three steps, i.e., rule evaluation, rule ranking and rule selection

////////////////////////////////////**STEP1: RULE EVALUATION**////////////////////////////////////

- We apply each rule in type 1 with seed opinion words to extract all aspects and calculate precision, recall.

- If the set of correct aspects in A is denoted by T, then the precision is defined by $|T| / |A|$, and the recall is defined by $|T| / |All_{lab}|$, where All_{lab} is the set of all labeled aspects in D.

- We are using the extracted aspects generated from type 1 rule and applying each sub rule of type 2 rules and calculating precision, recall independently for each sub rule.

- Each rule in type 3 are applied to extract opinion word and then type 1 and type 2 rules are applied on those extracted opinion words and extracted aspects to extract further new aspects and finally in a similar fashion precision and recall is calculated for each rule.

////////////////////////////////////**STEP2: RULE RANKING**////////////////////////////////////

After evaluating all rules, they are ranked according to their precisions and recalls.The three types of rules are ranked separately, i.e., rules of each type are ranked among themselves.

We have rank(r_i) > rank(r_j), on the basis of following

1. The precision of r_i is greater than that of r_j ,
2. Their precisions are the same, but the recall of r_i is greater than that of r_j ,

Both the precisions and recalls of r_i and r_j are the same, but the initial rule ID of r_i (i.e., i), which is unique, is smaller than that of r_j

////////////////////////////////////**STEP3: RULE SELECTION**////////////////////////////////////

Then we can select rules based on the ranking of the rules of each type. The proposed algorithm is called RSG, shorthand for Rule Selection using a Greedy algorithm

Algorithm 1 RSG($\mathcal{R}$, $\mathcal{D}$, $\mathcal{O}$)

Input: Rule set $\mathcal{R}$ including three ranked rule sets $\mathcal{R}^1$, $\mathcal{R}^2$ and $\mathcal{R}^3$, training data $\mathcal{D}$, seed opinion words $\mathcal{O}$
Output: The best subset $\mathcal{S}$ of rules

1: $\mathcal{S}$ =<>; // $\mathcal{S}$ is represented as a sequence
2: Discard all the rules in $\mathcal{R}$ whose precision and recall are both zero;
3: **for** each rule $r_i \in \mathcal{R}^1$ in descending order **do**
4: insert r_i at the end of $\mathcal{S}$;
5: compute F_1-score x of $\mathcal{S}$ on $\mathcal{D}$ and $\mathcal{O}$, $score(r_i) = x$;
6: **end for**
7: Find the first rule p in $\mathcal{S}$ with the highest derived F_1-score among all the rules of $\mathcal{S}$, drop all the rules after p in $\mathcal{S}$;
8: **for** each rule $r_j \in \mathcal{R}^2$ in descending order **do**
9: insert r_j at the end of $\mathcal{S}$;
10: compute F_1-score x of $\mathcal{S}$ on $\mathcal{D}$ and $\mathcal{O}$, $score(r_j) = x$;
11: **end for**
12: Find the first rule q in $\mathcal{S}$ with the highest derived F_1-score and drop all the rules after q in $\mathcal{S}$, let $\max F = score(q)$;
13: **for** each rule $r_k \in \mathcal{R}^3$ in descending order **do**
14: compute F_1-score x of $S \cup \{r_k\}$ on $\mathcal{D}$ and $\mathcal{O}$;
15: **if** $x > \max F$ **then**
16: insert r_k at the end of $\mathcal{S}$, $\max F = x$;
17: **end if**
18: **end for**
19: Output $\mathcal{S}$ as the final rule set.

Above algorithm consists of 4 sub-steps, each steps is described below:-

- **Sub-step 1**
  - (lines 1-2): Initialize the sequence S = <> (line 1), and discard the rules in the rule set whose precision and recall are both zero (line 2). It guarantees that the remaining rules can correctly extract at least one aspect.
- **Sub-step 2** (lines 3-7)
  - Since rules in R^2 *and* R^3 depends on rules in R^1 , thus rules in will be selected and added into S first.
  - Then rules in R^2 *and* R^3 will be added depending on the ranking.
  - F1 score is calculated after addition of each rule into S.
  - After inserting all rules in S, pruning step will be performed by removing

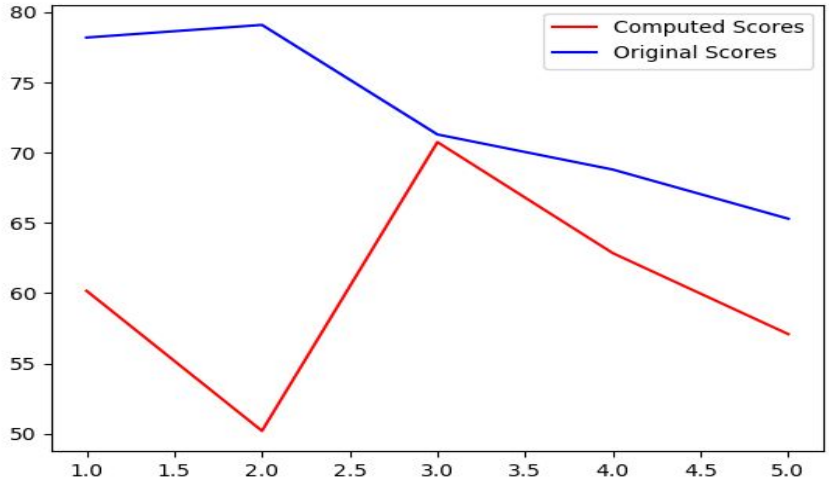
those rules which do not improve F1-score.
    - After attaining highest F1-score, all the rules of that type will be rejected as that may introduce error.

➢ **Sub-step 3** (lines 8-12)
    - Select the rules from R^2 and following the descending order defined by the ranking
    - Similarly add each rule of R^2 in S and compute F1-score as before.
    - After inserting all the rules in R^2 into S, line 12 performs rule pruning in a similar way as done in Sub-step 2.

➢ **Sub-step 4**(line 13-19)
    - Select the rules from R^3 and following the descending order defined by the ranking.
    - Unlike Sub-step 2 and Sub-step 3, not all the rule in R^3 is inserted into S as it can result in many useless rules.
    - Only those rules will be selected which improves the performance of the S.
    - Pruning is not required as we are not inserting all the rules first as in case of Sub-step 2 and Su-step 3. Thus every added rule can improve the results in this Sub-step.

////////////////////////////////////// **Result Graph** //////////////////////////////////////

- Original results vs computed result



- Scores :

Product *********	Original F1 Score **********	Computed F1 Score **********
Nokia 6610	71.3	70.7547169811
Nikon coolpix 4300	79.1	50.1960784314
Canon G3	78.2	60.1671309192
Creative Labs Nomad Jukebox Zen Xtra 40GB	68.8	62.8495339547
Apex AD2600 Progressive-scan DVD player	65.3	57.0841889117

////////////////////////////////////// **Obtained Top Sub Rules** //////////////////////////////////

- ➢ For Type 1 Rule :
  - Rules with dependency relations [“nsubj”, “dobj”, “prep”, “amod”] are obtained as top rules.
- ➢ For Type 2 Rule :
  - Rules with dependency relations [“amod”, “conj”] are obtained as top rules.
- ➢ For Type 3 Rule :
  - Rules with dependency relations [“amod”, “conj”, “nsubj”] are obtained as top rules.

////////////////////////////////////// **Reasons for obtained sub rules** //////////////////////////////////

- ➢ We can see that “amod” , “conj” , “nsubj” are obtained as top sub rules in the results.
- ➢ The reviews in the dataset are subjective.
- ➢ Reviews mostly describes about the particular product.
- ➢ An adjectival modifier(amod) of an NP is any adjectival phrase that serves to modify the meaning of the NP.
  - Example : “Sam eats red meat” - amod(meat, red)
- ➢ A conjunct(conj) is the relation between two elements connected by a coordinating conjunction, such as “and”, “or”, etc.
  - Example : “Bill is big and honest” - conj(big , honest)

- ➢ A nominal subject(nsubj) is a noun phrase which is the syntactic subject of a clause. The governor of this relation might not always be a verb: when the verb is a copular verb, the root of the clause is the complement of the copular verb, which can be an adjective or noun.
    - Example : “Clinton defeated Dole” - nsubj(defeated , clinton)
- ➢ In the above 3 dependency relations, the aspect(noun) is closely related to adjective (or) some other aspect itself.
- ➢ In our dataset, all reviews mainly describe about the product which are associated with some opinion.
- ➢ Reviews dataset is mostly subjective in nature and users express their direct opinion related to the product. And those direct expressions are mostly related to “amod” , “conj” , “nsubj” dependency rules.
- ➢ So, those rules are obtained as top rules in reviews data.
- ➢ In the same way the other rules are obtained as top rules.
- ➢ If we change to some other dataset, then we may not obtain same dependency relations as top rules.

////////////////////////////////////// **References** //////////////////////////////////////

1. Dependencies manual
2. POS_tagging manual
3. NLTK POS tagging tutorial
4. Stanford Parser Manual
5. Sentiment analysis and Opinion Mining book by Bing Liu
6. NLP with python nltk