Opinion Mining Based on Sentence Dependency Parsing

HU, Zuying (20451944)

collin.hu@connect.ust.hk

Abstract

Opinion mining is one the main tasks in nature language processing. In this project, Double Propagation(DP) is adopted to do target extraction [1]. Then the result produced by DP is further pruned and processed for getting a more descriptive summary on the reviews on courses.

1. Intoduction

Advances in informationtechnologies have been driving many new scenarios in people's daily life, suchas e-commerce, e-education and etc. One of the typical changes is that Internethas made many resources accessible online regardless the spatial and temporal limitations. For example, people at Asia could by commodities online which aresold in America through amazon, eBay, and students at China could take the course provided by professors at Canadian universities through coursera. After taking the courses on couresa, students could write their reviews on the courses as the feedback to the instructors and references to later learners.One popular course could receive hundreds of reviews from students which could give others a glim of the course before making decision whether to take the course and help the instructors to improve his/her skills and do some adjustments in the future. However, reading the reviews of the course is a tedious job which sometime may even readers confused, so a beforehand summaryof the course review is needed for readers to grasp the most concerned parts ofthe course. Opinion mining could be adopted to address this problem avoiding manual analysis of the review.

Opining mining is one typeof natural language processing which analyses people's opinions on one objects.For one object, people may give an overall review without mentioning one specific aspects or comment on one specific aspect with positive words while giving negative comments on another aspect. To analyse opinions on one object, the job could be subdivided into two tasks, one is target extraction which is to extract aspects which people have comments on and the other one is to judge sentiment that people showed on these aspects.

In this project, I mainly focus on the target extraction as this is the essential part of opinion mining.In one comment which may be composite with many sentences, the aspects of the commented object are call target word or target phrase. For example, *This course is very useful for learning computation theory, but the assignments are meningless* talks about two aspects of this course. One is *computation theory* whichis a target phrase and the other is “assignments” which is a target word. In this project, Double Propagation (DP)[1] is adopted to do target extract. DP is based on the observation that people usually express their opinions by using adjective words and these adjective words are dependency relevant to noun words which are the aspects on which people show opinions.These adjective words are called opinion words. By observation, opinion words are usually relevant to target words through some fixed dependency relations which makes it possible to extract target words from sentences by parsing the sentences based on a list of opinion word, that is, if a noun word or word phrasein relation dependent on an opinion word in one sentence, this noun word or phrase could be a target word with a high likelihood. In turn, target words could be used to extract opinion word. If an adjective word which is not a known opinion word but is used to modify a known target word, then we could add this adjective word to opinion list to grow opinion list. Through this way, target words and opinion words could be used to grow each other by propagating in two lists until reaching a convergence.

2. Related Work

Opinion mining is one of the hot topics in nature language processing field. Many methods have been proposed by researchers to do entity extraction which is called target extraction in this project. Here I list some works which are related to this project with using similar methods. These are all methods are based on parsing sentences with different parsing mechanisms.

Hu and Liu proposed one method which extracts targets by combining tagging words in comment sentences and frequent pattern mining. Hu firstly do part-of-speechtagging(POS) on sentence to tag each word, then noun words are extracted first [2]. After that frequent mining is applied on the extracted noun word by using Apriori algorithm and association rules [5]. This procedure could filter infrequent words and produce target phrases. Then pruning is further applied to remove meaningless words phrase and redundant targets. After extracting targets from sentences, opinion words are extracted based on locality. If one adjective wordis near to one target word, then this adjective word is marked as opinion word.In turn, these opinion words are used to find target words which are infrequent as noun words modified by opinions words are usually target words even they arenot frequent. However, this method doesn't utilize the dependent relations between opinion words and target words even though the relations between opinion words and target words are coarsely considered, so many noises may beinduced.

Another method that extracts target words based on the dependent relations between target words and opinion words is proposed by Qiu. This method uses opinion word listand target word list to grow each other until reaching a convergence which leads to double propagation. The general idea of this DP method has been introducedin Introduction part.

After convergence where both opinion list and target list stop growing by parsing sentences, some further processes are needed to prune target list for reducing non-target words and generating target phrases. After that we could get atarget list based on which we parse each review to extract its target words and phrases.

3. Data Analysis and Cleaning

The aim of this project is to extract target words or phrases form reviews on courses. To do a good target extraction, data analysis is necessary for understanding the data so a suitable method could be selected. The dataset used in this project consists 10,372 available reviews on 36 courses. The most of reviews are written in English but some in other languages. Some reviews may consist of meaningless words such as “-_-”, “!!!” and even un-readable marks. Considering these problems in the dataset, data cleaning is necessary before doing any other process on the data. To clean data, python package NLTK [3] is utilized. Data cleaning mainly includes two parts: step one is removing meaningless words which are formed by Alphabet characters; step two is deleting reviews not written in English language.

Another preprocessing is to group reviews based oncourse id as reviews are highly related to course contents. Reviews on different courses may vary significantly in contents; on the other hand, it is meaningless to extract target words from a mixture of reviews from different courses.

After cleaning and grouping the data, target extraction could be done on the processed data.

4. Target Extraction

In this part, the details of the whole process of target extraction of which DP is the core part. To extract targets from sentences, a list of seed opinion words should be prepared beforehand that works as seeds for extracting target words. Then the extracted target words are to grow opinion words in turn and so on so forth. Through the mechanism of double propagation, opinion words and target words could be used to grow each other until convergence is reached where size of the target words list and size of opinion words could not grow any more. After getting the target words, targets pruning and target phrase generation should be done over the extracted target words, so infrequent words, which are usually non-target words but wrongly extracted by applying extracting rules, and target phrases, which are compound of noun words. After that, these words could be grouped so that target words or phrases describing same aspector feature could be aggregated together and give a more descriptive and clearer summary on the reviews.

4.1 Dependency Parsing

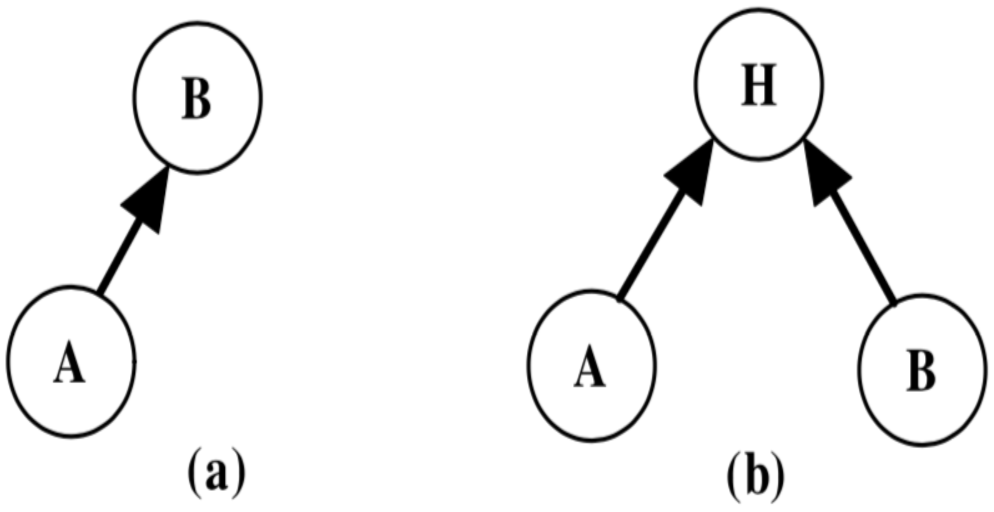
As mentioned above, opinion words and target words are extracted based on the dependency relations between each other, therefore, denpendcies between words should be analysed.

Dependency Relations

There are various types of dependencies including *"det","nmod","nsubj", "dobj"* and etc. Two categories to summarize all possible dependencies between two words in sentences are defined by Qiu [1].

Definition one (Direcct Dependency (DD)):

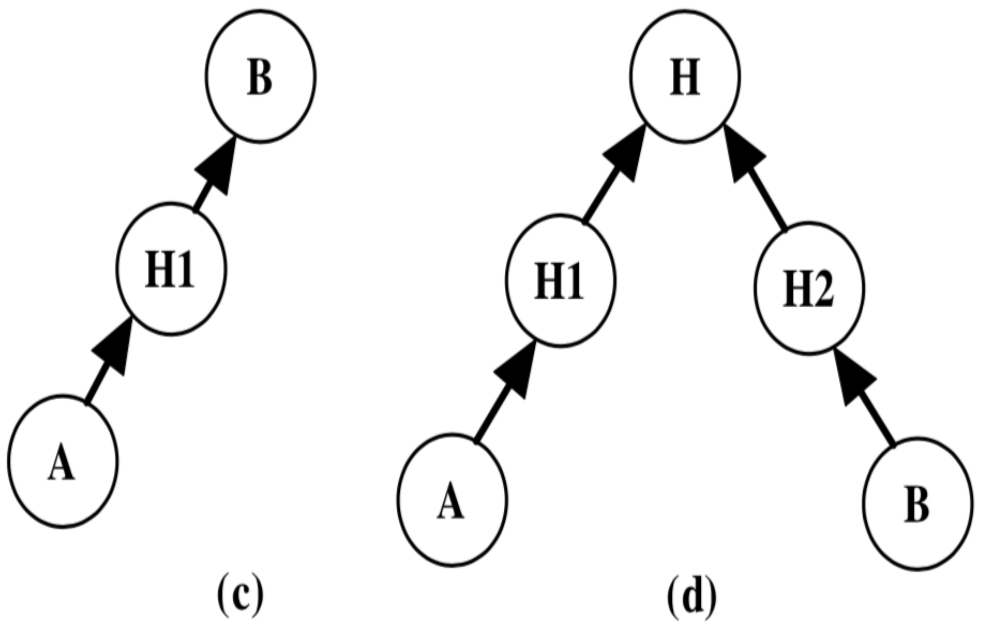
A Direct dependency indicates that one word dependson the other word without any additional words in their dependency path (i.e.,directly) or they both depend on a third word directly.



For example, A depends on B directly in figure (a), and A and B both are directly dependent on H so A and B have direct dependency.

Definition two(Indirect Dependency (IDD)):

An indirect dependency indicates that one worddepends on the other word through some additional words (i.e. indirectly) or they both depend on a third word through additional words.



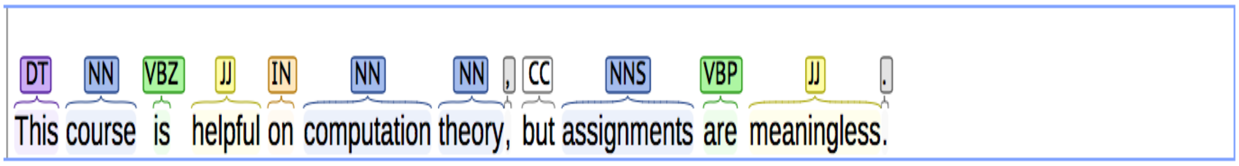
In figure (c), A is indirectly dependent on B as A depends on B through B. In figure (d), A depends on H through H1 and B depends on H through H2, so A and B are indirect dependent to each other.

In the propagation, only DD s are considered as inreviews reviewers tends to express their opinions in a direct way and thereby,the direct dependencies between opinion words and target words predominate thereviews. IDDs won't be used in this project. These DDs used for targetextraction include *'nsubj', 'dobj', 'xsubj', 'csubj', 'nmod', 'iobj', 'xcomp','amod' and 'conj'.*

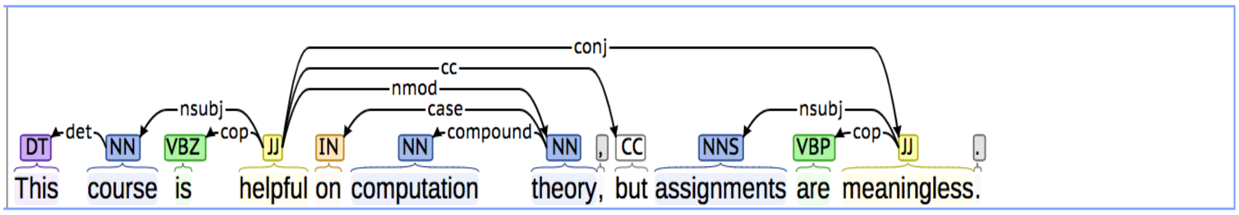
Dependency Parse Tree

To identify the dependency relations between words in sentences, dependency parse tree is adopted to represent the dependency relations. Dependency parse tree is a tree structure where two words are siblings if there is dependency between them. In following figure, a dependency parse tree is illustrated to show the dependencies between words in sentence *This course is helpful on computation theory, but assignments are meaningless.*

POS Tagging on words


DT
NN
VBZ
JJ
IN
NN
NN
,
CC
NNS
VBP
JJ
.
This course is helpful on computation theory, but assignments are meaningless.

Dependency Parse Tree of Sentence


conj
cc
nmod
case
compound
det
nsubj
cop
nsubj
cop
DT
NN
VBZ
JJ
IN
NN
NN
,
CC
NNS
VBP
JJ
.
This course is helpful on computation theory, but assignments are meaningless.

In dependency parse tree, words are connected through dependency relations with relations indicating the braches and words placed in the leaf nodes. In this project, I use Stanford corenlp dependency parser [4] provided by Stanford NLP group to get the dependency relations between words in sentence. Then based on the dependency relations, further analysis could be done.

It should be noted that the tags of words play a vital role in dependency relation analysis. To identify dependency relations between words, the tags of words should be labeled beforehand as the tags and their orders inone sentence determine the relations between words. Besides that, tags could also be useful to find the target words and opinion words because target words are usually noun words and opinion words are usually adjective words. Independency relations, noun words are labels as $NN*$ where $*$ may be any other alphabet character or just doesn't exist, and adjective words are labeled as $JJ*$.

4.2 Target Extraction algorithm

Target Extraction is though propagation according to extraction rules. In this part, I will talk about the details of extraction rules and propagation algorithm.

Target Extraction Rules

After parsing the dependency relations between words in sentences, the attention should be paid on the noun words and adjective words. There are several relations should be considered between words, and they are *'nsubj', 'dobj', 'xsubj', 'csubj','nmod', 'iobj', 'xcomp','amod' and 'conj'.*

In propagation to grow target list and opinion list, there are four scenarios could be considered to extracted words based on the dependency relations [1].

One: extracting target words using opinion words;

Two: extracting targets using the extracted target;

Three: extracting opinion words using the extracted targets;

Four: extracted opinion words using both the given and the extracted opinion words.

For scenarios one and three, the target and opinion dependency should be considered. For scenario two, dependency between targets should considered. For scenario four, dependency between opinion words should be considered. In this target extraction task, only direct dependency is considered. As defined above, DD consists two types of dependencies so these four scenarios could be extended to eight rules during double propagation. These eight rules are listed in Table 1 [1].

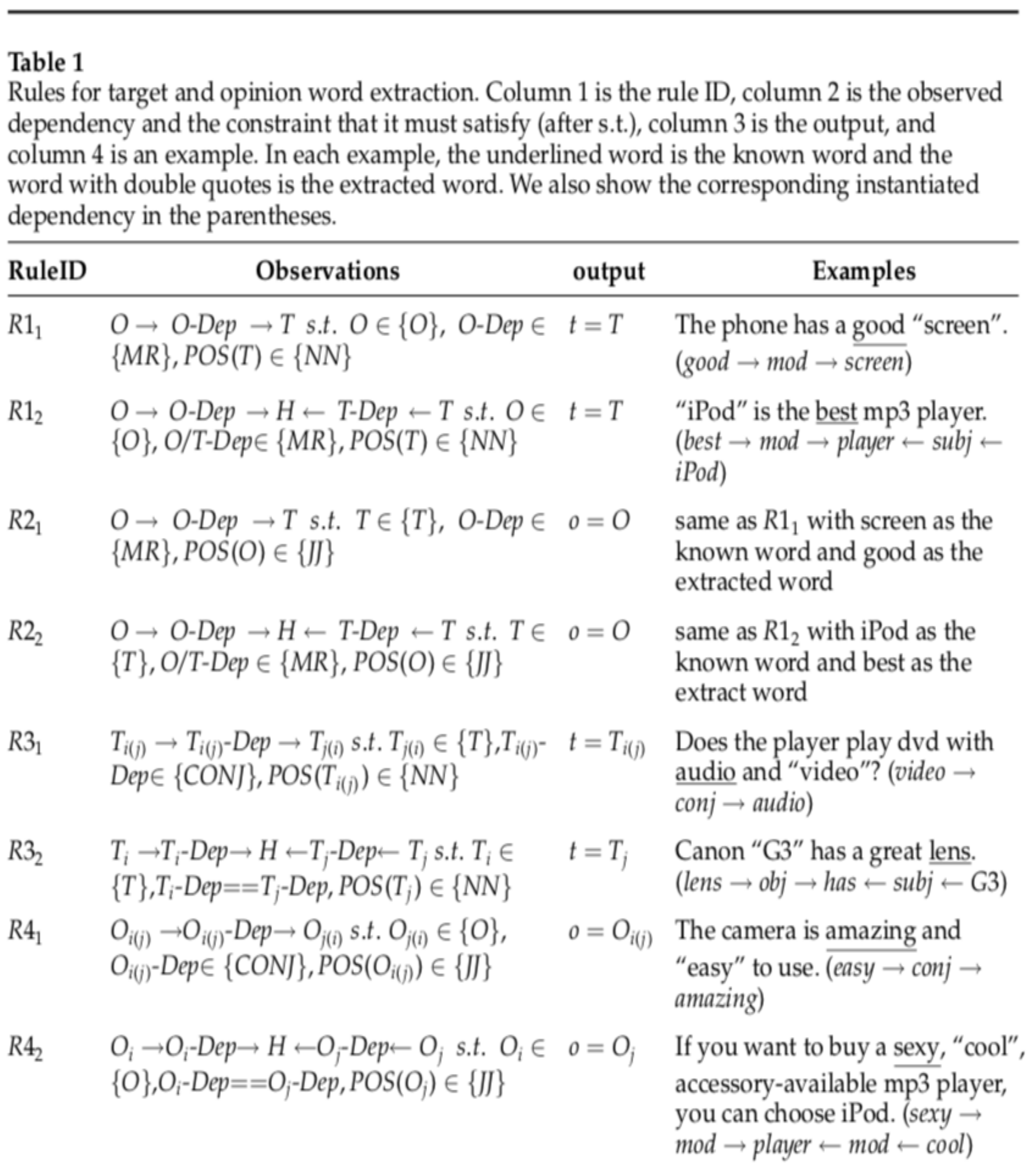


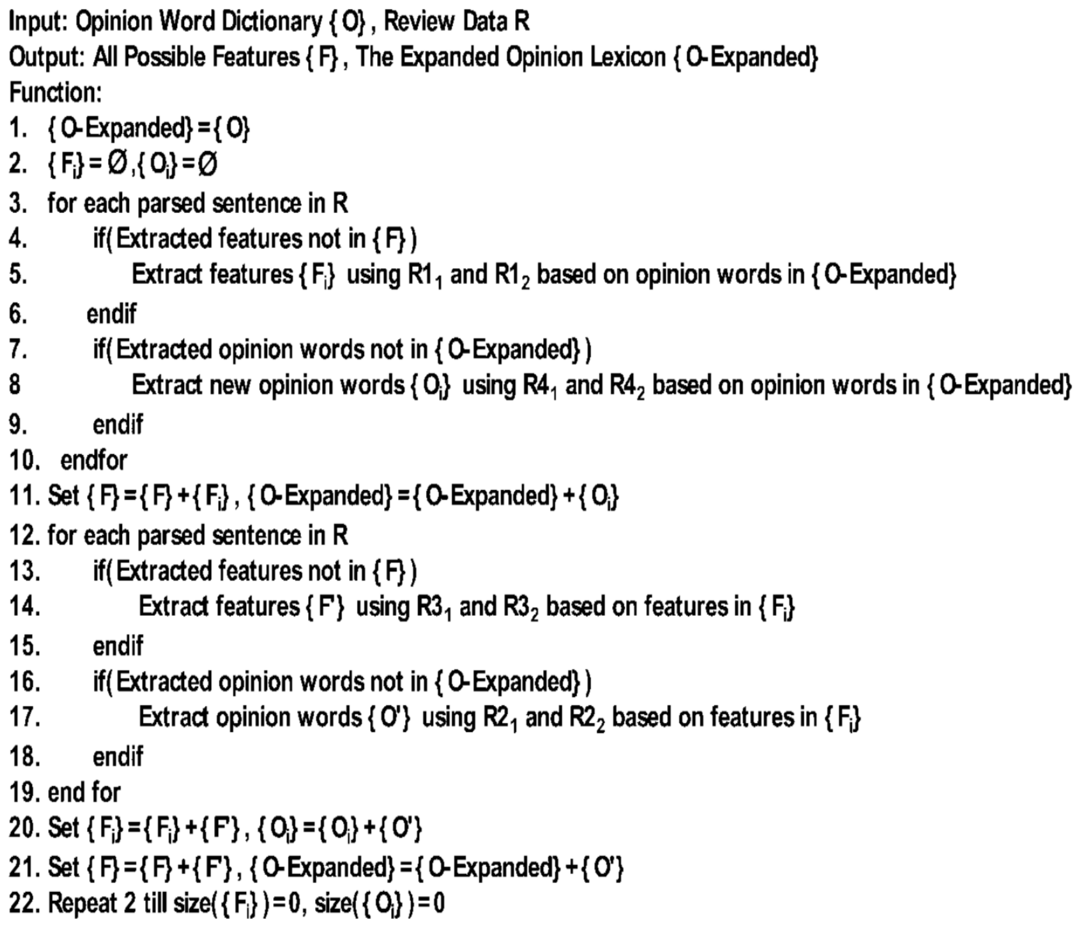
Table 1
Rules for target and opinion word extraction. Column 1 is the rule ID, column 2 is the observed dependency and the constraint that it must satisfy (after s.t.), column 3 is the output, and column 4 is an example. In each example, the underlined word is the known word and the word with double quotes is the extracted word. We also show the corresponding instantiated dependency in the parentheses.

RuleID	Observations	output	Examples
$R1_1$	$O \rightarrow$ *O-Dep* $\rightarrow T$ s.t. $O \in \{O\}$, *O-Dep* $\in \{MR\}$, $POS(T) \in \{NN\}$	$t = T$	The phone has a good "screen". (*good → mod → screen*)
$R1_2$	$O \rightarrow$ *O-Dep* $\rightarrow H \leftarrow$ *T-Dep* $\leftarrow T$ s.t. $O \in \{O\}$, *O/T-Dep* $\in \{MR\}$, $POS(T) \in \{NN\}$	$t = T$	"iPod" is the best mp3 player. (*best → mod → player ← subj ← iPod*)
$R2_1$	$O \rightarrow$ *O-Dep* $\rightarrow T$ s.t. $T \in \{T\}$, *O-Dep* $\in \{MR\}$, $POS(O) \in \{JJ\}$	$o = O$	same as $R1_1$ with screen as the known word and good as the extracted word
$R2_2$	$O \rightarrow$ *O-Dep* $\rightarrow H \leftarrow$ *T-Dep* $\leftarrow T$ s.t. $T \in \{T\}$, *O/T-Dep* $\in \{MR\}$, $POS(O) \in \{JJ\}$	$o = O$	same as $R1_2$ with iPod as the known word and best as the extract word
$R3_1$	$T_{i(j)} \rightarrow T_{i(j)}$-*Dep* $\rightarrow T_{j(i)}$ s.t. $T_{j(i)} \in \{T\}$, $T_{i(j)}$-*Dep* $\in \{CONJ\}$, $POS(T_{i(j)}) \in \{NN\}$	$t = T_{i(j)}$	Does the player play dvd with audio and "video"? (*video → conj → audio*)
$R3_2$	$T_i \rightarrow T_i$-*Dep* $\rightarrow H \leftarrow T_j$-*Dep* $\leftarrow T_j$ s.t. $T_i \in \{T\}$, T_i-*Dep* $==$ T_j-*Dep*, $POS(T_j) \in \{NN\}$	$t = T_j$	Canon "G3" has a great lens. (*lens → obj → has ← subj ← G3*)
$R4_1$	$O_{i(j)} \rightarrow O_{i(j)}$-*Dep* $\rightarrow O_{j(i)}$ s.t. $O_{j(i)} \in \{O\}$, $O_{i(j)}$-*Dep* $\in \{CONJ\}$, $POS(O_{i(j)}) \in \{JJ\}$	$o = O_{i(j)}$	The camera is amazing and "easy" to use. (*easy → conj → amazing*)
$R4_2$	$O_i \rightarrow O_i$-*Dep* $\rightarrow H \leftarrow O_j$-*Dep* $\leftarrow O_j$ s.t. $O_i \in \{O\}$, O_i-*Dep* $==$ O_j-*Dep*, $POS(O_j) \in \{JJ\}$	$o = O_j$	If you want to buy a sexy, "cool", accessory-available mp3 player, you can choose iPod. (*sexy → mod → player ← mod ← cool*)

In above table, *"O"* stands for opinion words and *"T"* stands for target word. *"POS(W)"* is the tag of the word *W* where *"W"* could be *"T"* or *"O"*.And *"W-Dep"* is the dependency relation for word *"W"* with other words. *MR* is the collection of several dependency relations including *'nsubj', 'dobj', 'xsubj', 'csubj','nmod', 'iobj', 'xcomp','amod'* , and *CONJ* refers to the dependency relation *"conj"*.

Propagation Algorithm

Progation alogrithm is used to extract opinion words and target words according to extraction rules based on the dependency relations between words in sentences. In this popropation algorithm, it is only needed to prepare a list of opinion words beforehand which could be easily got. The procedure of the popagation algorithm is stated as following pseudo codes [1].



```
Input: Opinion Word Dictionary { O} , Review Data R
Output: All Possible Features { F} , The Expanded Opinion Lexicon { O-Expanded}
Function:
1.  { O-Expanded} = { O}
2.  { $F_i$} = Ø ,{ $O_i$} = Ø
3.  for each parsed sentence in R
4.        if( Extracted features not in { F} )
5.             Extract features { $F_i$}  using $R1_1$ and $R1_2$ based on opinion words in { O-Expanded}
6.        endif
7.        if( Extracted opinion words not in { O-Expanded} )
8              Extract new opinion words { $O_i$}  using $R4_1$ and $R4_2$ based on opinion words in { O-Expanded}
9.        endif
10.  endfor
11. Set { F} = { F} + { $F_i$} , { O-Expanded} = { O-Expanded} + { $O_i$}
12. for each parsed sentence in R
13.       if( Extracted features not in { F} )
14.            Extract features { F'}  using $R3_1$ and $R3_2$ based on features in { $F_i$}
15.       endif
16.       if( Extracted opinion words not in { O-Expanded} )
17.            Extract opinion words { O'}  using $R2_1$ and $R2_2$ based on features in { $F_i$}
18.       endif
19. end for
20. Set { $F_i$} = { $F_i$} + { F'} , { $O_i$} = { $O_i$} + { O'}
21. Set { F} = { F} + { F'} , { O-Expanded} = { O-Expanded} + { O'}
22. Repeat 2 till size( { $F_i$} ) = 0, size( { $O_i$} ) = 0
```

Word Stemming

One problem in target extraction is that one same word may have various forms but refers to same meaning. If not processed, these different word may be treat as different target words. To reduce the word variant, word stemming should be done on the extracted words. If two different words have same stem, they will be treat as the same target, for example, *"assignments"* and *"assignment "* both have the same stem *"assign"*, so the target will be stored as *"assign"* for both *"assignment"* and *"assignments"*. Word stemming is one option to reduce word variant, and other choices may could be considered such as word embedding where words with different variants will be converted to almost same word vector. If using word vector, the words with similar word vector will be treat as same target.

Second Scanning

After the terminationof this algorithm, a list of opinion words and a list of target words could beproduced. Based on this target list, the reviews will be parsed one by one again to extract target word and opinion words for each review, this process is called second scanning. During the second scanning, there is no need to obey the extraction rules as it is assumed that once a target word appears in the review the reviewer is commenting on this aspect even though this target word may not satisfy the extraction rules. This assumption is made based on the observation that reviews sometimes may be expressed in an informal way so some target may not be found in the review even though it is mentioned, but thetarget should emerge somewhere that satisfies the extractions. Therefore, a target word could be found by parsing all reviews but may not be found in aspecific review. Based on this fact, the extraction rules are not followed toextract target for a specific review.

During the second scanning, a target word should be extracted as long as it appear in the target list and it is tagged as noun words. After the second scanning, targets foreach review could be collected for latter pruning.

4.3 Pruning and Target Phrase generation

The result got by applying propagation algorithm is a list of individual words and noise may be introduced in it. However, the target could be a compound of noun words, for example, in sentence *"This course is helpful on computation theory, but the assignments are meaningless."*, the targets should be *"computation theory"* and *"assignment"*. In the sentence *"I am so busy at my own work that I could finish the assignment."*, the word *"work"* is not a target word but could be wrongly extracted from this sentence.

To address these problems, pruning and target phrase generation are necessary. It could be assumed that one wrongly extracted word is an infrequent word overall and a target word as one aspect will be mentioned many times over different reviews, so pruning could be done based on frequency.

Pruning Rule:If the frequency of an extracted word in target list over all the reviews is lower than the threshold, this word should be regarded as noise and be removed.

In this case, the real target word but with a low frequency could also be removed but that doesn't affect a lot since only few reviewers mentioned that aspect which is not important enough to pay attention on.

Another problem is that some targets should be in form of word phrase. To extracted word phrase, one dependency relation could be utilized, which is *"compound"*.

Target Phrase Generation: If one word of the noun phrase is a target word, then this noun phrase should be extracted as a target phrase. One limitation is also set for target phrase extraction, which is that the length of a target phrase is no more than 3.

After that we could filter the targets again based on frequency, but the threshold could be lower as a noun phrase usually appear in one review to refer to some aspect even its frequency is low.

4.4 Aggregation

In this part, I will talk about aggregation of target words or phrases but this part is not done perfectly until the submission of this report. To maintain the completeness of the whole target extraction, target aggregation is stated generally in this part and several possible methods are listed and discussed.

One basic aggregation is performed. For a three-word target phrase, if it has two common conseuctive words with a two-phrase target word, the three-phrase word will be aggregated to two-phrase word. For example, *"machine learning course"* and *"machine learning"* should be aggregated into *"machine learning"*.

To get a clear summarisation of reviews, aggregation is a necessary procedure. For example, *"machine learning specification"*, *"machine learning course"*, *"machine learning"*, these three word phrase all refer to aspects *"machine learning"* and should be aggregated as *"machine learning"*.

One possible solution is fuzzy matching. By fuzzy matching, targets with slight differences could be aggregated to one. But there is a problem could not solve the meaning similarity problem. For example, *"Assignment"* and *"homework"* both refer to aspect *"homework"*, but these two words can't be aggregated by fuzzy matching as they are very different in composition of characters. Another solution , word embedding, could be considered. By training, words with similar meaning will be represented by word vectors with slight differences, that is, the distance between the two word vectors is small.

5. Experiment and Discussion

In this part, the algorithm stated in part 4 is applied on course comments data discuss the potential problems in current results.

First the course comments data is cleaned to remove meaningless marks then grouped based on course id. The target extraction will be done each course reviews. To extract target, a list of seed opinion words is needed. In this project, the seed opinion words are got from Liu's collection[6].

As there are 10,372 reviews but 36 courses, each course have around 300 reviews on average. The number of target phrased could be usually small. In experiments, if the number of reviews is no more than 300, pruning is only applied on the individual target words not phrases to maitain the target phrases, otherwise, the pruning is applied on both individual target words with minimum support rate 0.01 and target phrases with threshold 1.

There are some noises which could not be removed even after pruning. For one course, targets list and their corresponding frequencies are listed after pruning. It could be found that some noise may also have high frequency.

```
{"learner": 2, "lot": 3, "function": 2, "cours": 30, "program": 4, "busi": 2, "major": 1,
"topic": 7, "techniqu": 4, "program assign": 8, "thank": 5, "teacher": 2, "knowleg": 2,
"question": 1, "lesson": 1, "theori": 2, "professor": 3, "someth": 2, "ir": 1, "problem": 4,
"lack": 2, "respons": 2, "meta support": 2, "segment fault": 2, "assign": 2, "materi": 3,
"explan": 3, "quizz": 2, "subject": 2, "week": 2, "amount": 2, "tool": 2, "basic": 4,
"inform retriev": 2, "search engin": 7, "rerun": 2, "help": 2, "forum": 2, "exercis": 2,
"star": 1, "languag": 1, "python": 1, "survey": 2, "overview": 2, "coverag": 2, "video": 2,
"student": 2, "user": 2, "time": 4, "assign problem": 2, "textbook": 2, "aptitud": 2,
"detail": 3, "unit": 2, "special": 2, "introduct": 2, "retriev": 2, "understand": 1, "tr": 1,
"area": 1, "context": 1, "engin": 1, "class": 2, "text retriev": 5, "sens": 2, "world": 2,
"way": 2, "someon": 2, "softwar": 1, "text explan": 1, "diagram": 1, "machin": 1,
"analysi": 1, "background": 1, "opinion": 1, "data analyst": 1, "insight": 2, "lectur": 1,
"assess": 1, "collater materi": 1, "slide handout": 1, "machin learn system": 2}
```

For example, *"lot"* and *"thank"* is absolutely not target words but have high frequencies and be kept. So some other pruning method should be explored to removed these non-target words.

6. Future Work

Next work after target extraction is sentiment judgement for each target in a review. This a subsequent step of target extraction but could utilise the opinion words extracted in propagation. To do this task, sentiment polarity of the opinion words should be judged. For example, *"intelligent"* should be marked as positive and *"boring"* should be marked as negative. After that the sentiment that reviewers show on one target could be judged by the opinion words which are used to modify the target words.

Another work should be done is to measure the performance. The measurement could be very hard as there are rare available annotated dataset which are similar to the dataset and different project may give different definition on target words, for example, in this project targets are defined as noun words or noun phrases that reviewers comment on while the target may be a compound of adjective word and noun word from the view of other researchers. To address this problem, fuzzy matching or word vector similarity to could be used to compare the results produced by your methods and the annotated targets by dataset providers. Some possible available datasets that could be used to measure the performance of the method stated in this report are provided by Liu[7], which are annotated electronic products reviews from customers.

References

[1]. Qiu, G.; Liu, B.; Bu, J.; and Chen, C. 2011. *Opinion word expansion and target extraction through double propagation*. Computa-tional Linguistics 37(1):9–27.

[2]. Hu, M., and Liu, B. 2004. *Mining and summarizing customer reviews*. In KDD ’04, 168–177.

[3]. NLTK Package: http://www.nltk.org/

[4]. Stanford NLP corenlp: http://nlp.stanford.edu:8080/corenlp/process

[5]. Agrawal, R. and Srikant, R. 1994. *Fast algorithm for mining association rules*. VLDB’94, 1994.

[6]. Seed Opinion Words:https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon

[7]. Measure Dataset: https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#dataset

Appendix

README

Environment

1. python 3.6
2. python packages:

   NLTK, re, pandas, json.
3. java package:

   stanford-corenlp-full-2017-06-09/stanford-corenlp-3.8.0.jar,

   stanford-corenlp-full-2017-06-09/stanford-corenlp-3.8.0-models.jar

Code Explanation

As stated in report, the whole process includes data cleaning, target extraction

Data Cleaning

the codes are sotred in directory *data_process*

1. *clean_data.py* : this code is used to remove meaningless marks

   input: *raw_data.csv*, output: *step1_data.csv*
2. *remove_non_english_sents.py* : this code is used to remove reviews not writen in English

   input: *step1_data.csv*, output: *step2_data.csv*
3. *groupData_base_on_id.py*: group data based on course id.

   input: *step2_data.csv*. output: 36 course review files

All data used in above are stored in directory *data*, and 36 course review files are stored in the subdirectory *course* under *data*. These 36 files are named in form of *course_"course_id".csv*

Target Extraction

The codes are stored in directory *extract_target*.

1. *extract_target_list.py*: this code is to grow target and opinion list by propagation

   input: *course_"course_id".csv*, output: *course_"course_id"_target_list.txt*

2. *extract_target_phrase.py*: this code is to extract target word or phrase for each review.

   input: *course_"course_id".csv*, *course_"course_id"_target_list.txt*

   output: *course_"course_id"_transaction.csv*

3. *filter_target.py*: this coude is used to prun the targets.

   inuput: *course_"course_id"_transaction.csv*

   output: *course_"course_id"_target_filtered.txt*

   All output files produced by this part are stored under the directory *result*.

Result

Final result stored in the files names as *course_"course_id"_target_filtered.txt*.

Other code files

other files not mentioned above are used for testing or experiments coulde be ignored.