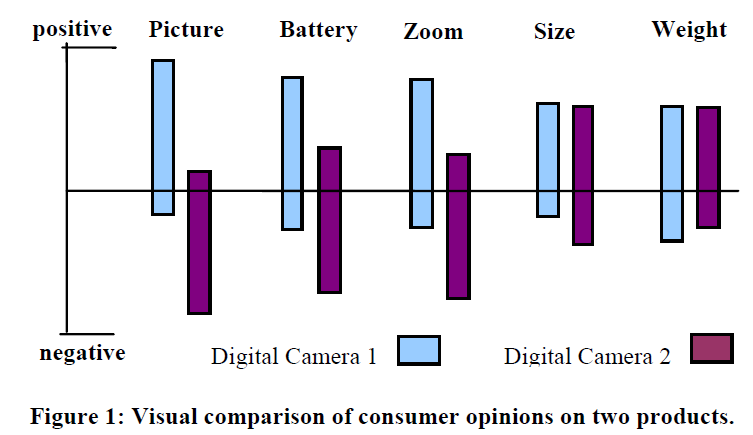
**Opinion Observer Analyzing and Comparing Opinions**

**1. INTRODUCTION**



To enable the above visualization, two challenging technical tasks need to be performed:

1. Identifying product features that customers have expressed their (positive or negative) opinions on.

2. For each feature, identifying whether the opinion from each reviewer is positive or negative, if any. Negative opinions typically represent complains/problems about some features.

**2. RELATED WORK**

**Terminology finding and entity extraction**

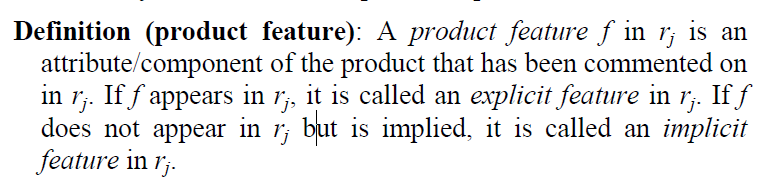
There are basically two techniques for terminology finding: symbolic approaches that rely on noun phrases, and statistical approaches that exploit the fact that words composing a term tend to be found close to each other and reoccurring.

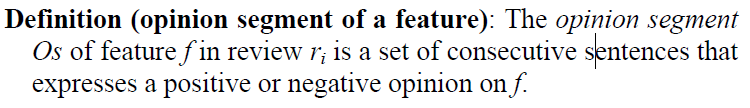
**Sentiment classification**

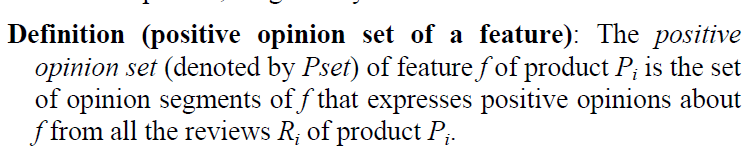
Sentiment classification classifies opinion texts or sentences as positive or negative.

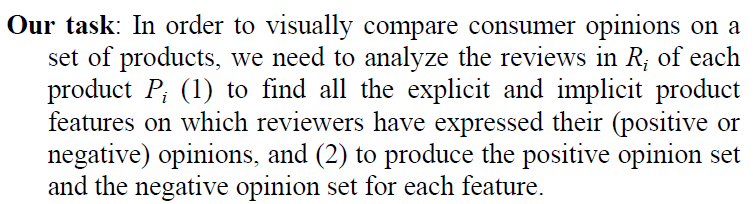
**3. OPINION OBSERVER**

**3.1 Problem Statement**









**3.2 Visualizing Opinion Comparison**

**3.3 Automated Opinion Analysis**

*3.3.1 Extracting Product Features*

We use supervised rule discovery to perform this task. We firstprepare a training dataset by manually labeling (or tagging) a large number of reviews. The steps are as follows:

1. Perform Part-Of-Speech (POS) tagging and remove digits: We use the NLProcessor linguistic parser [28] to generate the POS tag of each word (whether the word is a noun, verb, adjective, etc). POS tagging is important as it allows us to generate general language patterns. We also remove digits in sentences, e.g., changing “16MB” to “MB”. Digits often represent concepts that are too specific to be used in rule discovery, which aims to generalize. We use two examples from above to illustrate the results of this step: “<N> Battery <N> usage” “<V> included <N> MB <V>is <Adj> stingy” <N> indicates a noun, <V> a verb, and <Adj> an adjective. Each POS tag appears right before the corresponding word(s).

2. Replace the actual feature words in a sentence with [feature]: This replacement is necessary because different products have different features. The replacement ensures that we can find general language patterns which can be used for any product. After replacement, the above two examples become: “<N> [feature] <N> usage” “<V> included <N> [feature] <V> is <Adj> stingy” For implicit features, we replace the words that indicate such features with [feature]. For example, “MB” above is replaced with [feature] as it indicates implicit feature <memory>.

3. Use n-gram to produce shorter segments from long ones: For example, “<V> included <N> [feature] <V> is <Adj> stingy” will generate 2 smaller segments: “<V> included <N> [feature] <V> is” “<N> [feature] <V> is <Adj> stingy” We only use 3-grams (3 words with their POS tags) here, which works well. The reason for using n-gram rather than full sentences is because most product features can be found based on local information and POS tagging. Using long sentences tend to generate a large number of spurious rules.

4. Distinguish duplicate tags: When there are duplicate tags in a segment, we distinguish them with a sequence number, e.g.: “<N1> [feature] <N2> usage”

5. Perform word stemming: This is commonly performed in information retrieval tasks to reduce a word to its stem.

**Extraction of product features:** The resulting patterns are used to match and identify *candidate features* from new reviews after POS tagging. There are a few situations that need to be handled.

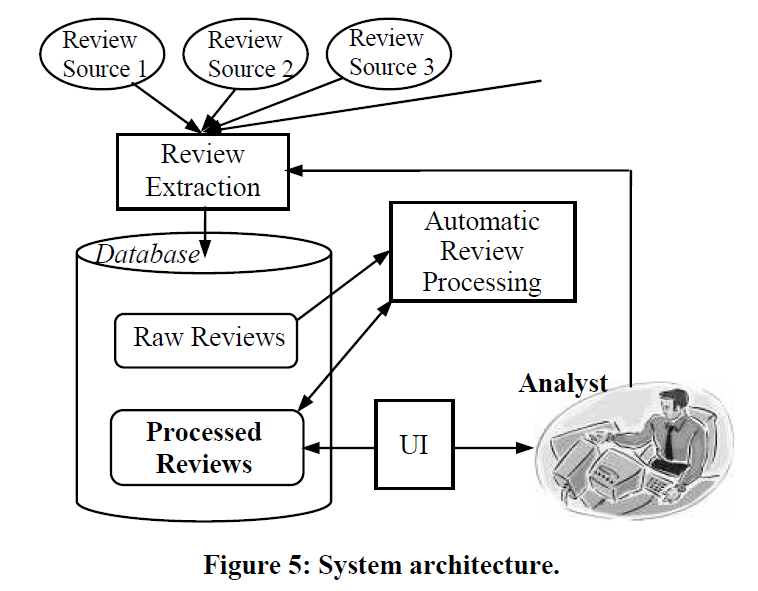
*3.3.2 Grouping Synonyms*

The basic idea is to employ WordNet to check if any synonym groups/sets exist among the features.

**3.4 Semi-Automated Tagging of Reviews**

**3.5 Extracting Reviews from Web Pages**

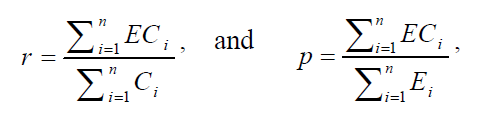
**4. SYSTEM ARCHITECTURE**



**5. EXPERIMENT RESULTS**

**Training and test review data**

**Evaluation measures**: We use recall (*r*) and precision (*p*) as evaluation measures:



**Semi-automatic tagging**

**6. CONCLUSIONS**