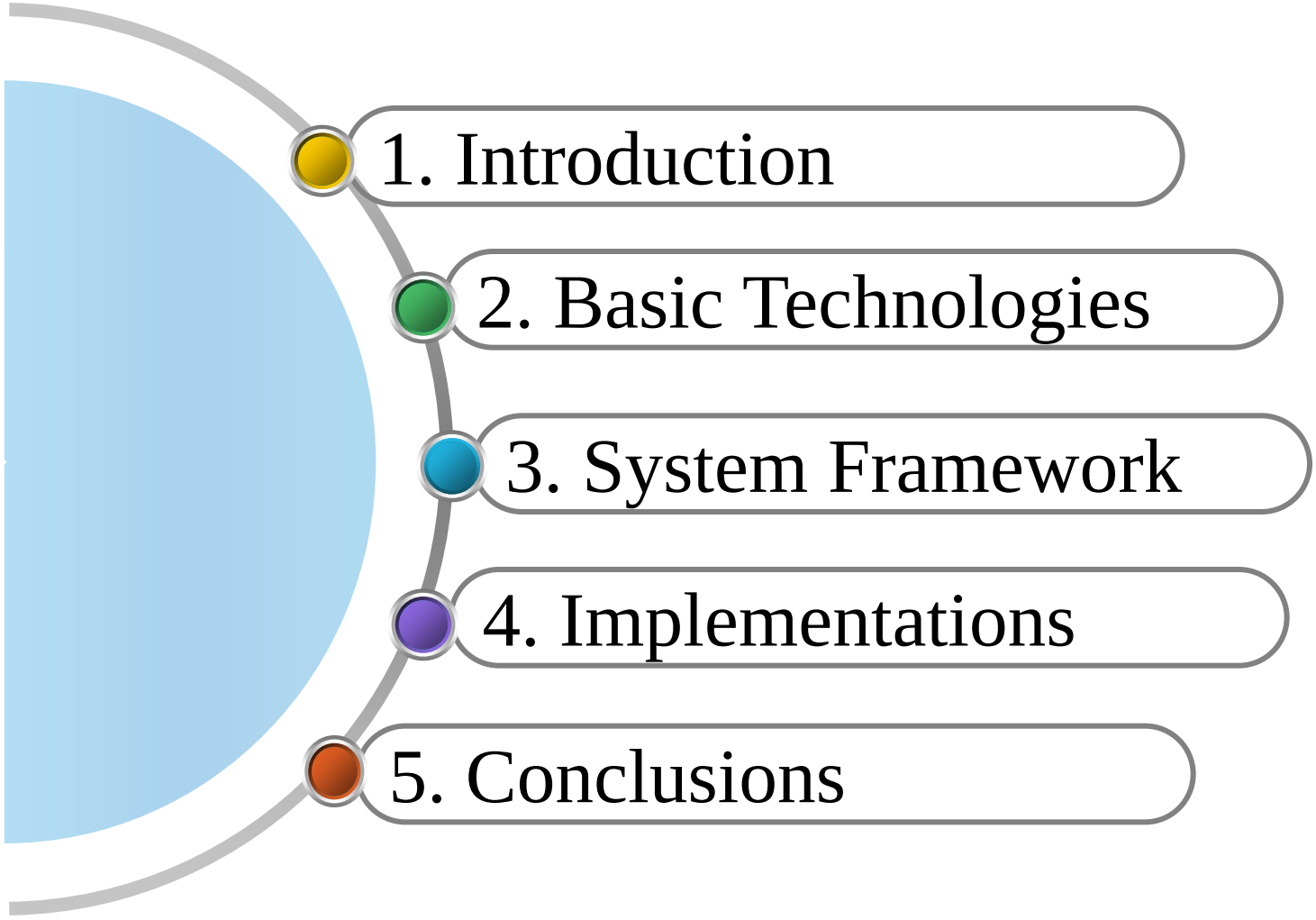
Department of Computer Science and Information Engineering
National Yunlin University of Science and Technology

Web Product Ranking Using Opinion Mining

Presenter: Yin-Fu Huang

2013/05/10

Contents


1. Introduction
2. Basic Technologies
3. System Framework
4. Implementations
5. Conclusions

1. Introduction

1. Introduction(1)

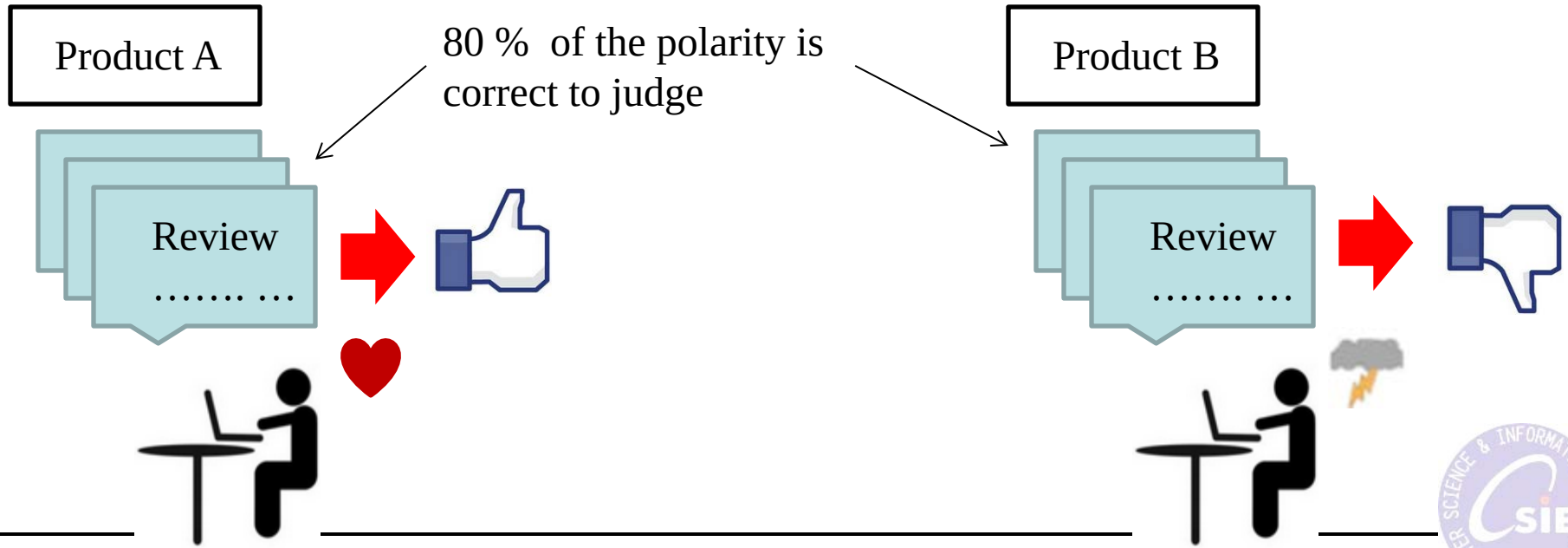
- With the rapid growth of the Web and the convenience of the Internet, more and more people have been changing their shopping habits from traditional to online shopping.

- Tian et al. proposed opinion mining techniques for product ranking, the sentence polarity in product reviews is used as the only one influenced factor.

P. Tian, Y. Liu, M. Liu, and S. Zhu, "Research of product ranking technology based on opinion mining," *Proc. the 2nd International Conference on Intelligent Computation Technology and Automation*, pages. 239-243, 2009.

1. Introduction(2)

- Opinion mining is very suitable for the applications “many users to discuss a single topic” when a large number of sentences and reviews are on the criticism of a single topic or when exactly identifying the correct rate of a bias direction is required to improve accuracy.



1. Introduction(3)

- In this research, a product ranking system using opinion mining techniques to find favorable products for users is presented.

- Here, our product ranking system considers three issues while calculating product scores:
  - 1) product reviews.
  - 2) product popularity.
  - 3) product release month.

1. Introduction(4)

- Eventually, our system would provide users to specify product features in a query, and send back the ranking results of all matched products.

2. Basic Technologies

2.1 XML Path Language (XPath)

- XPath uses path expressions to select nodes or node-sets in an XML document.

```
<html>
        <body>
                <name kind="Camera">Nikon D3000</name>
                <name kind="Laptop">MacBook Air</name>
        </body>
</html>
```

- We can use the path expression “// name[@kind='Laptop']”, then call function text() to get the text "MacBook Air".

2.2 Part of Speech Tagging

- The Part-of-speech tagging (i.e., POS tagging or POST), also called grammatical tagging or word-category disambiguation, is the process for assigning the correct part of speeches (e.g. noun, adjective, verb, adverb, etc.) to each word in a text based on both its definition and context as follows:

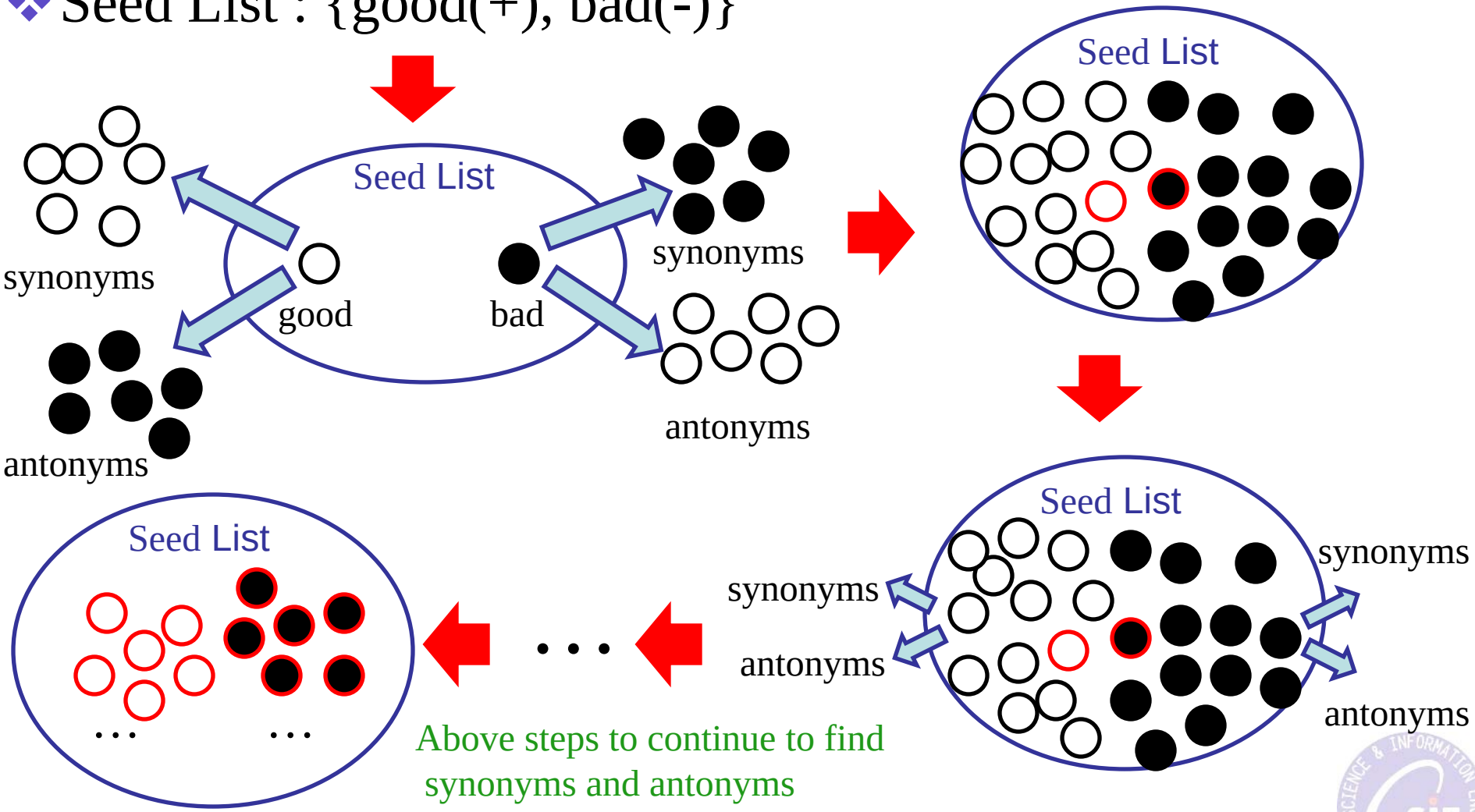
Word	This	article	is	about	the	sport	.
POST	DT	NN	VBZ	RB	DT	NN	SENT

2.3 Semantic Orientation(1)

- The semantic orientation method is to identify the polarity by WordNet.
- We use the procedure **OrientationPrediction** (OP)
  - First, positive and negative opinion words are defined to construct a “seed list”.
  - and then, by the seed list, synonyms and antonyms are found using WordNet.

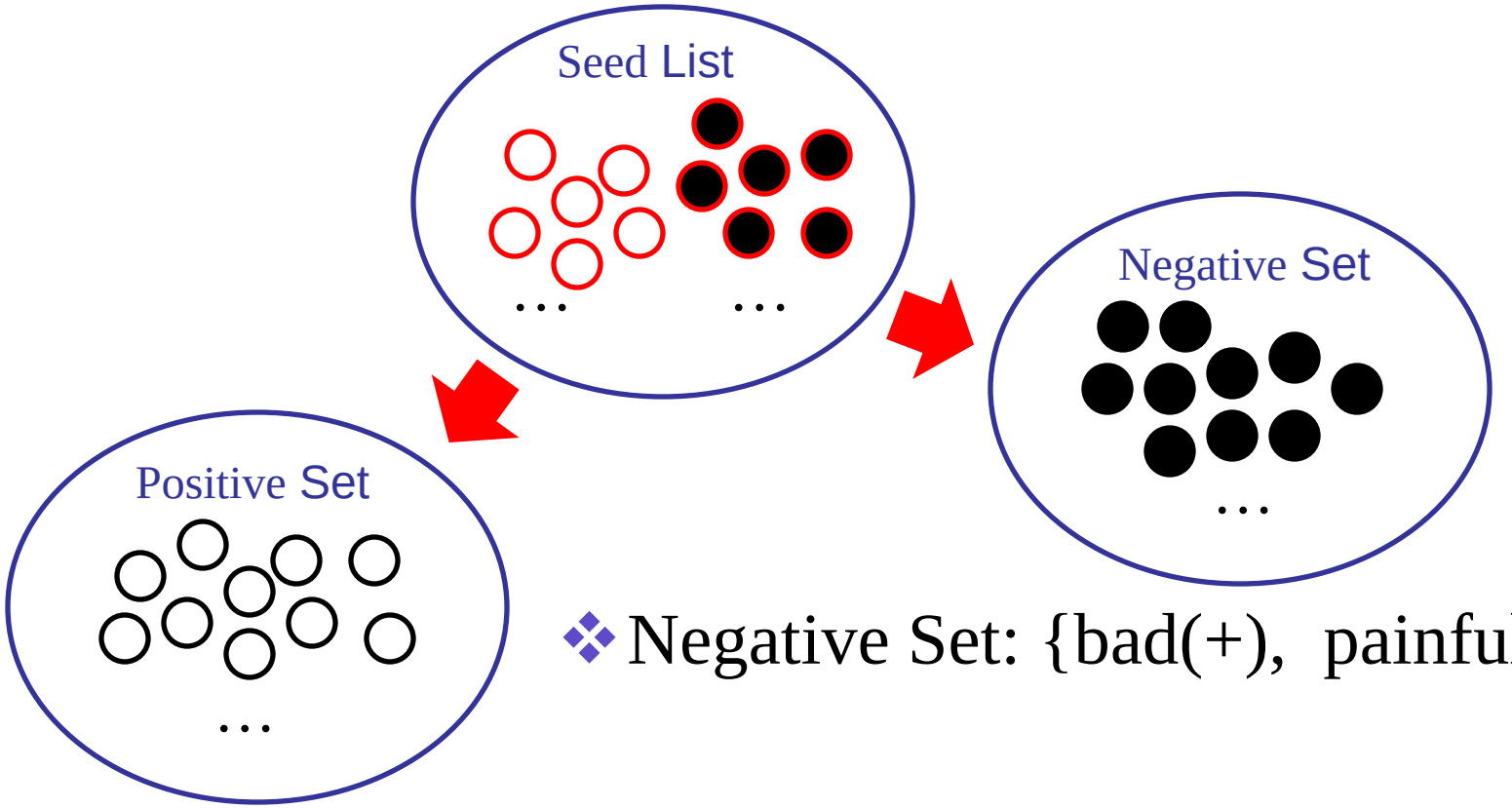
2.3 Semantic Orientation(2)

- Seed List : {good(+), bad(-)}



2.3 Semantic Orientation(3)

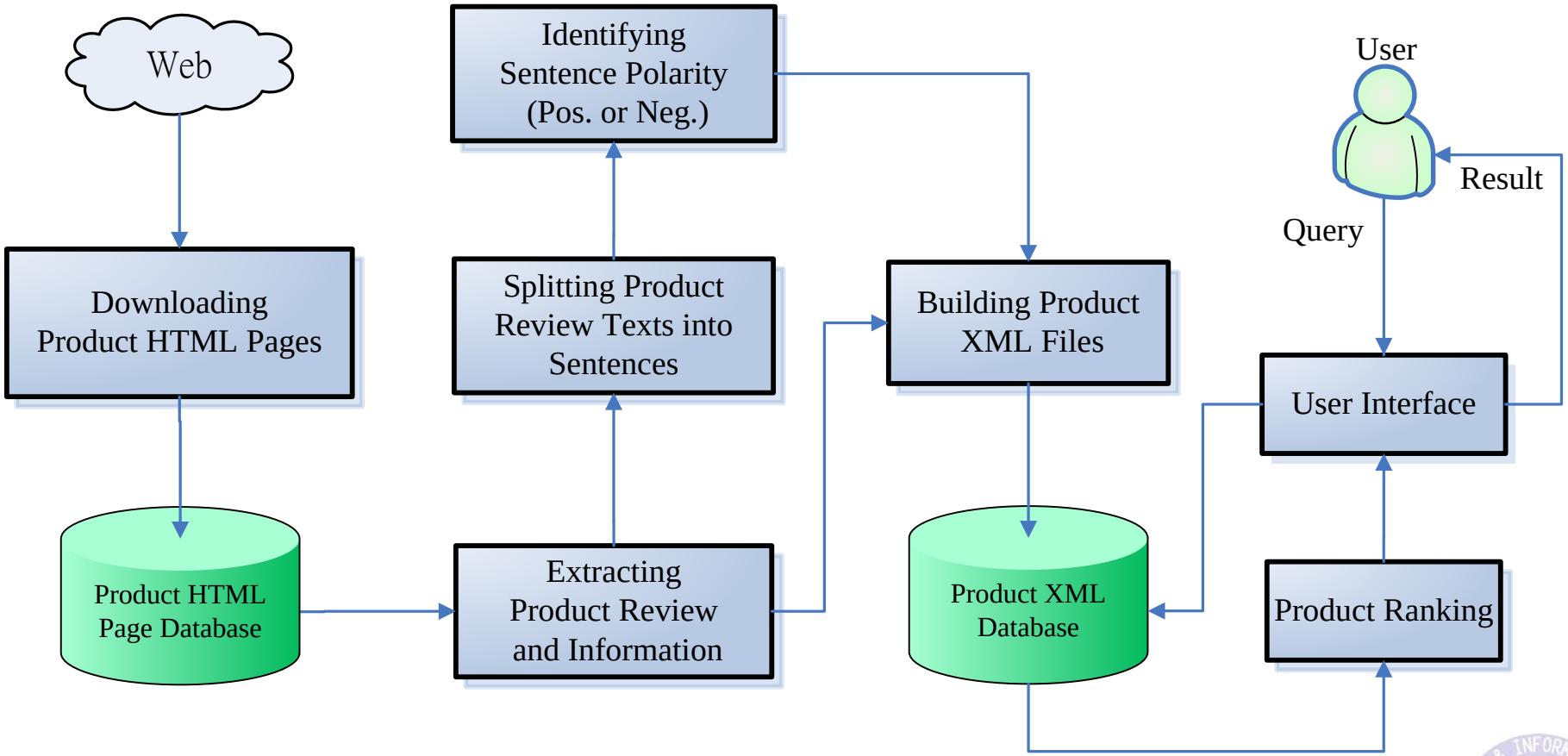
- Seed List : {good(+), great(+) , … , bad(-), painful(-), …}



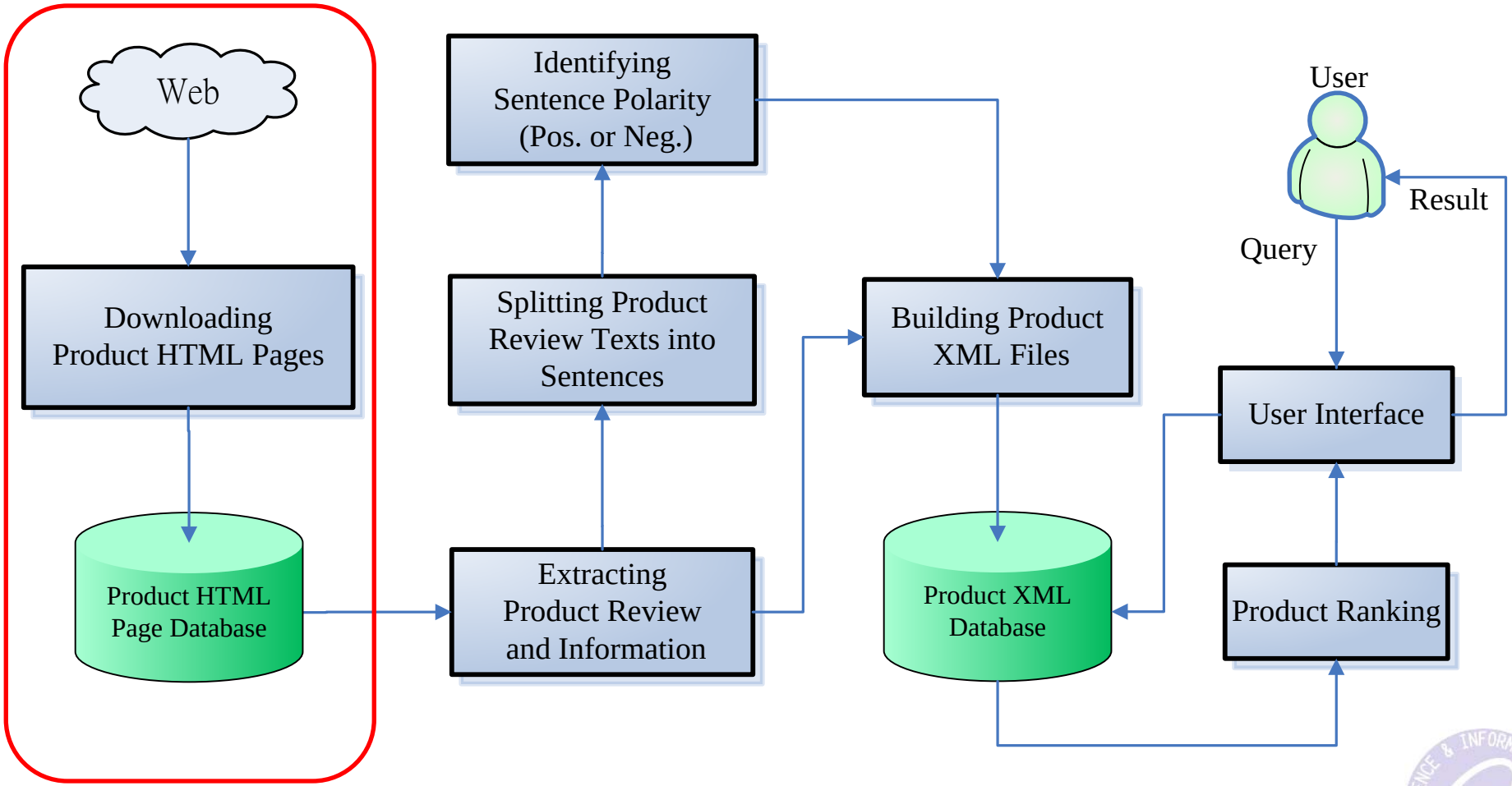
- Negative Set: {bad(+), painful(+) , …}
- Positive Set: {good(+), great(+) , hot(+) …}

3. System Framework

3. System Framework



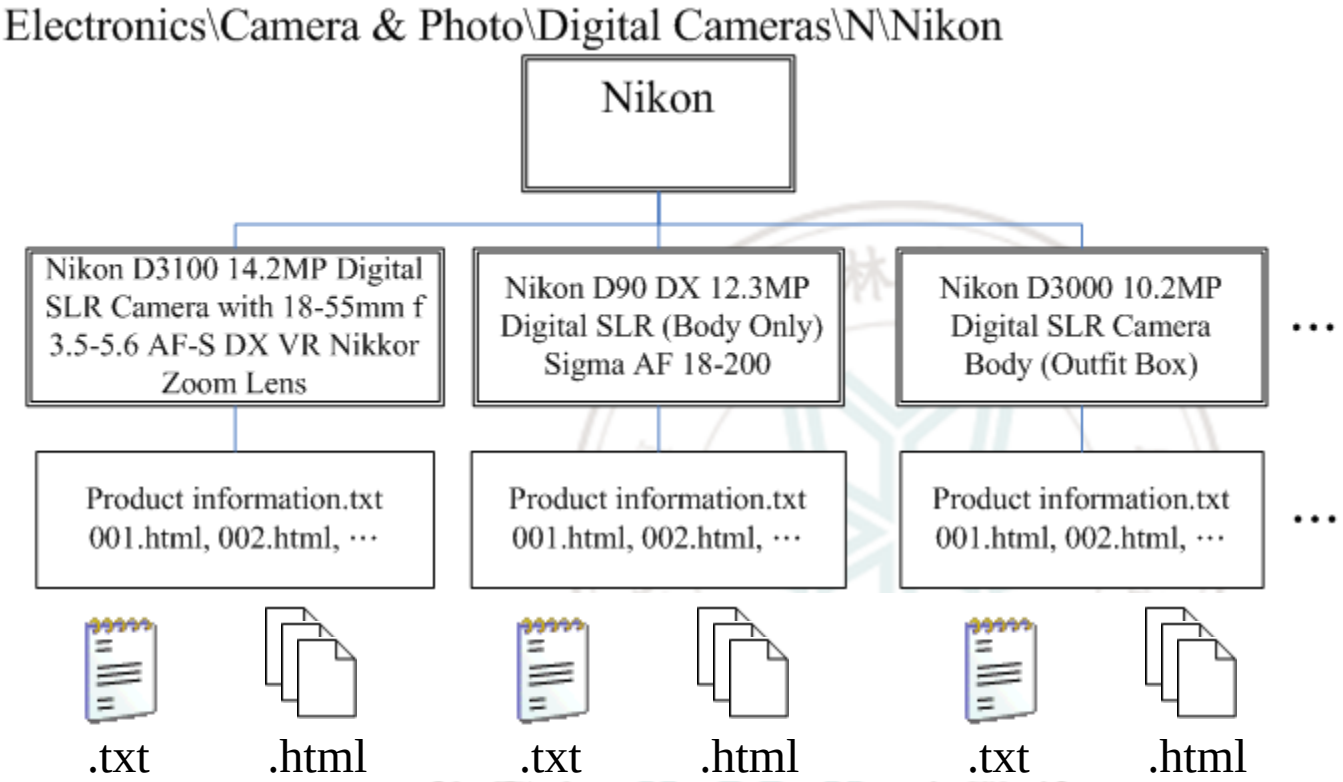
3.1 Downloading Product HTML Pages(1)



3.1 Downloading Product HTML Pages(2)

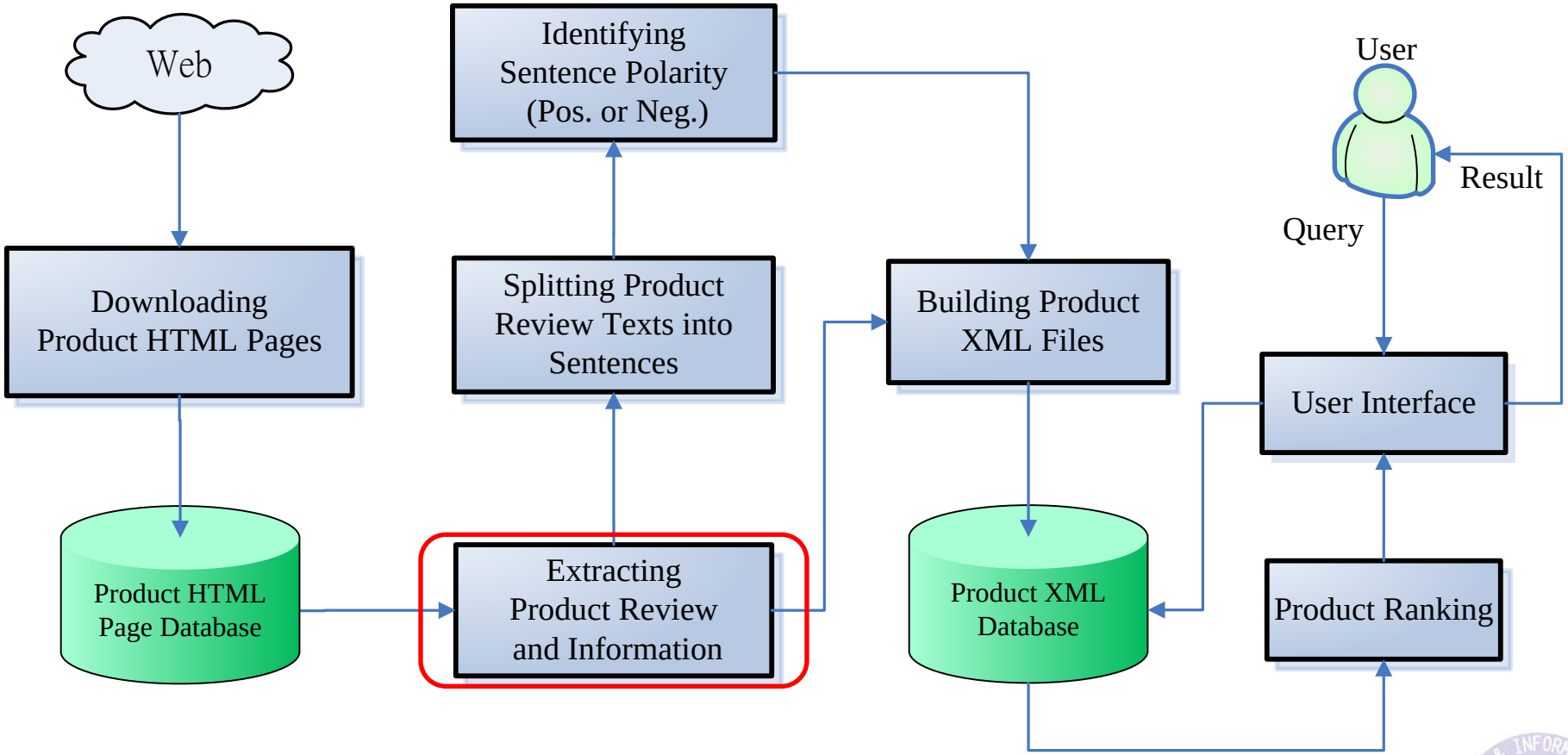
- For the Amazon product home page, users can specify product categories and brands to download the specified product information.

3.1 Downloading Product HTML Pages(3)



Kind: Camera
Brand: Nikon
Product Name: Nikon_D3100_Digital_SLR_Camera_Body
Date first available at Amazon.com: August 17, 2010
Price: $599.00

3.2 Extracting Product Review Information(1)

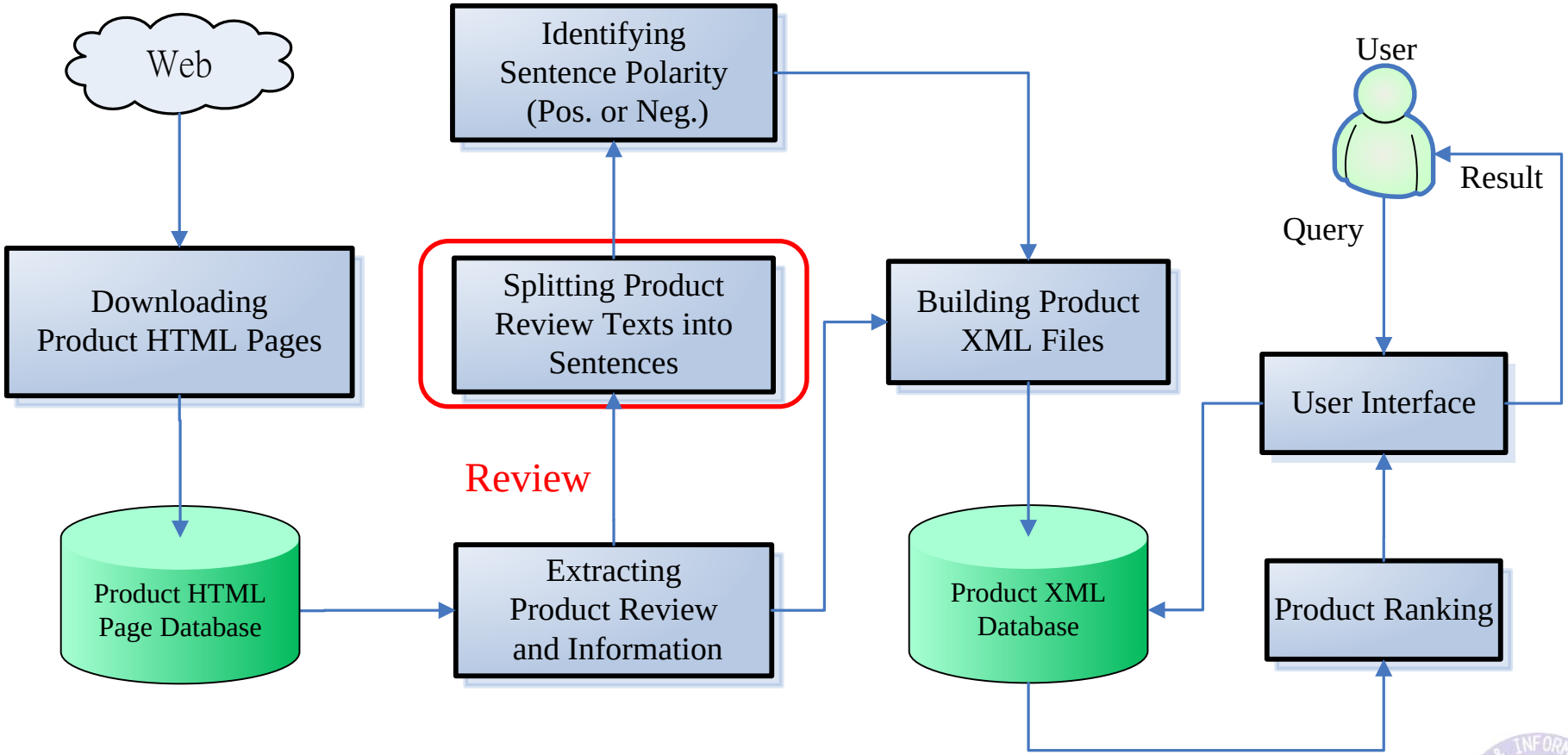


3.2 Extracting Product Review Information(2)

- We can use XPath to extract five kinds of information.

1) Helpful: 0 of 2 people found the following review helpful
2) Star: 5.0 out of 5 stars
3) Title: Great camera
4) Date: March 15, 2011
5) Review: For many years, I used a Nikon FG, which was a mid-level camera. Took great pictures if you knew what you were doing, but of course it was a film camera. Then came digital cameras. For the last 10 years, I was basically using point and shoot cameras, which were great for general landscapes.....

3.3 Splitting Product Review Texts into Sentences(1)

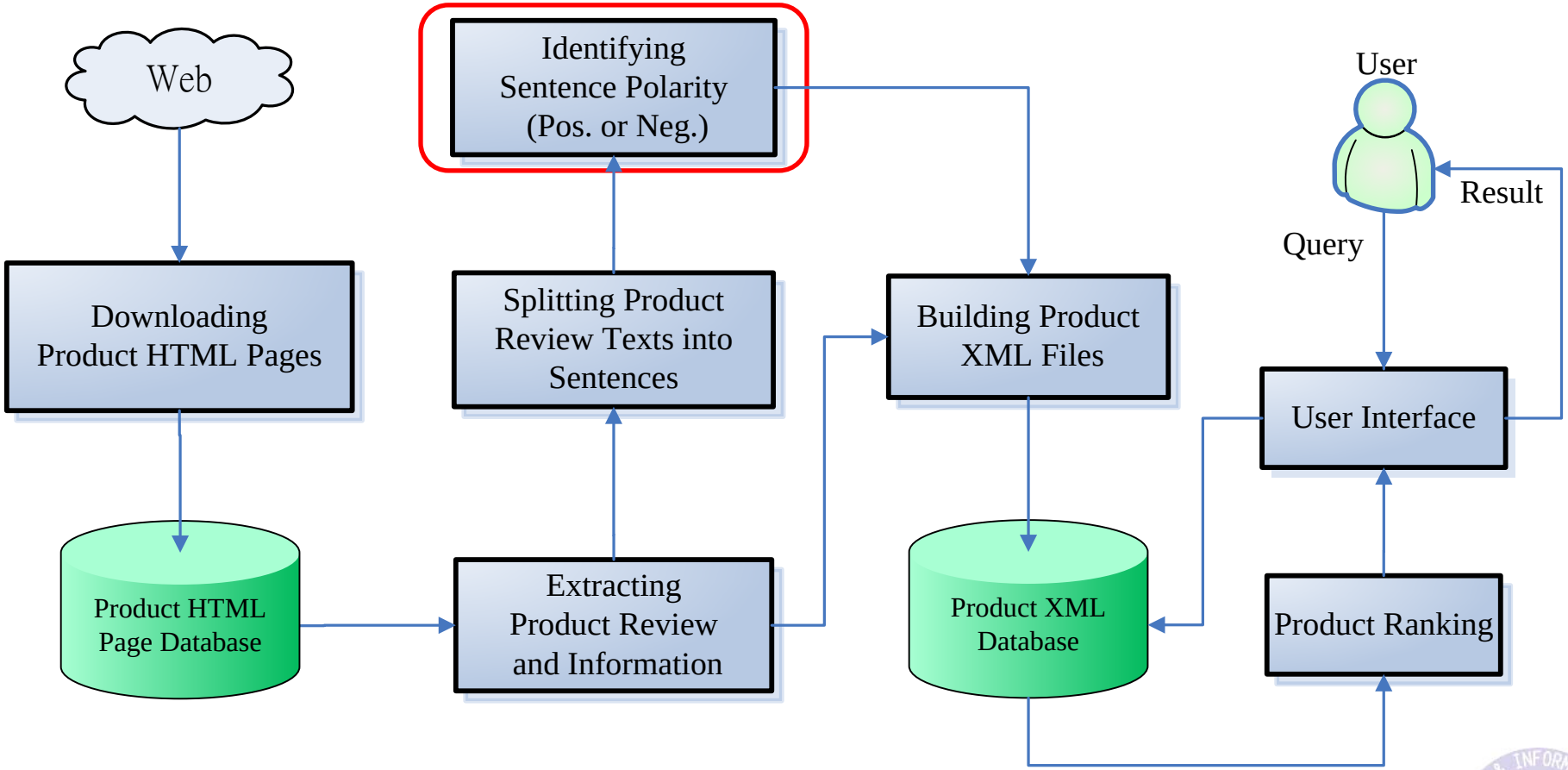


3.3 Splitting Product Review Texts into Sentences(2)

- We parse a product review to split texts into sentences, and produce POS (part of speech) tags, such as noun, verb, adjective, etc., for each word.

- The POS tagger (or called TreeTagger) developed at the University of Stuttgart in annotating words with POS tags is employed. TreeTagger has been verified to achieve 96.36% accuracy on Penn-Treebank data.

3.4 Identifying Sentence Polarity(1)



3.4 Identifying Sentence Polarity(2)

- This step is to determine the polarity of opinion words, and then to identify sentence polarity.

- In the system, only the adjectives in product reviews are used as opinion words.

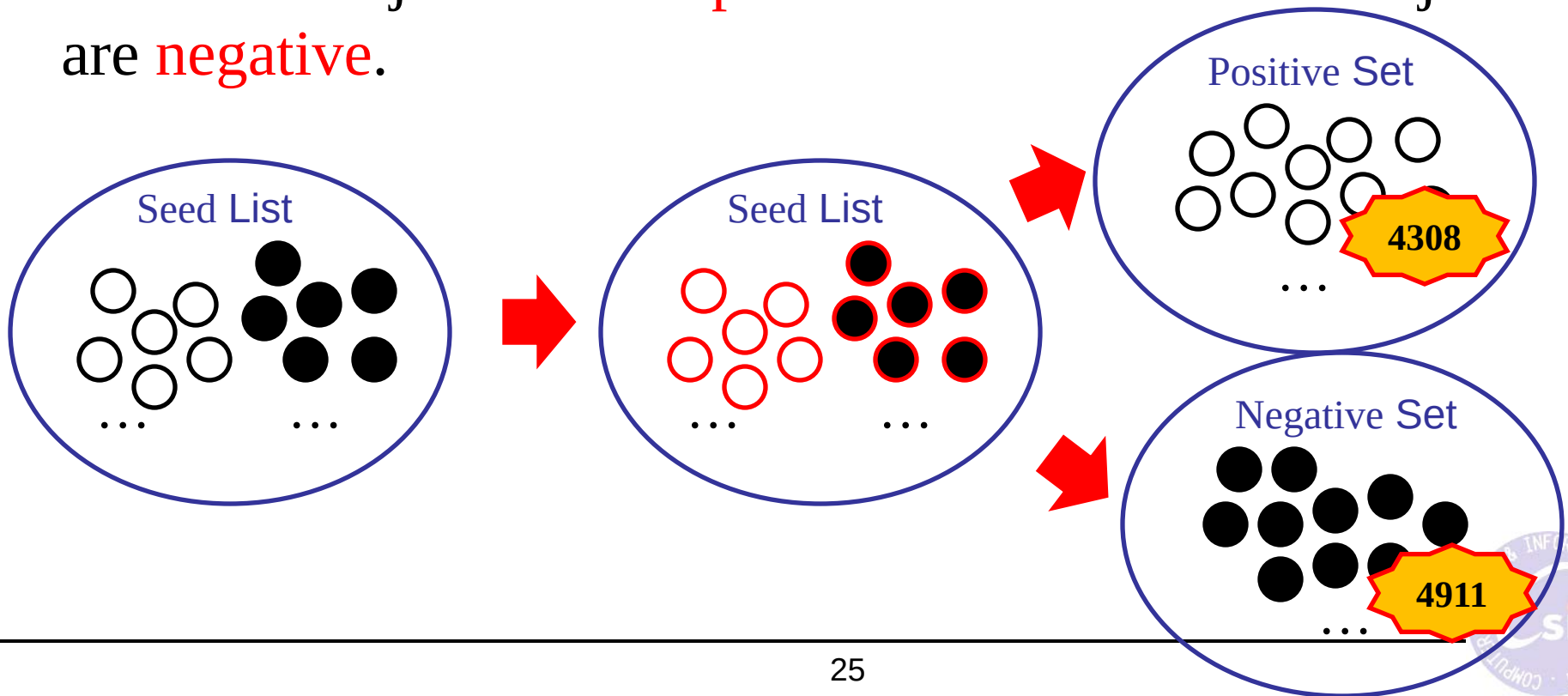
- We extract opinion words using a dictionary-based approach (i.e., by procedure **OrientationPrediction** (OP) proposed by Hu and Liu to determine the polarity).

  M. Hu and B. Liu, "Mining and summarizing customer reviews," *Proc. the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages. 168-177, 2004.

3.4.1 Extracting Opinion Words

- Here, OP is used to collect the positive set and negative set.

- First, we define 30 common adjectives in a seed list, of which 15 adjectives are positive and another 15 adjectives are negative.



3.4.2 Calculating the Opinion Strength(1)

- The polarity strength of opinion words could be calculated as follows:

$$OS_p = Sign(SET(p))\frac{|CS(p)|}{|Set(p)|}$$

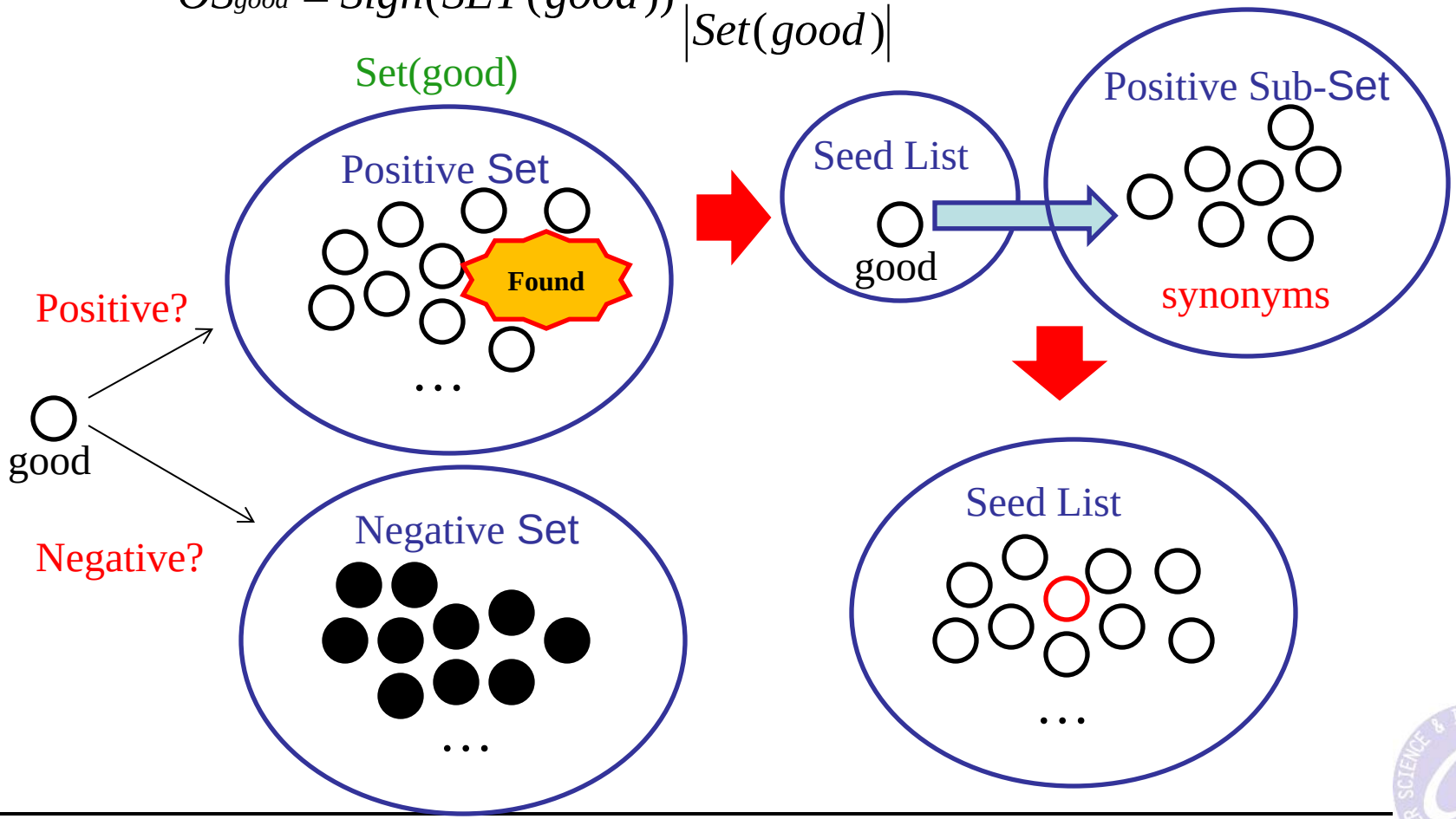
where *p* is an adjective, *Set(p)* is the positive set or negative set based on the polarity of *p*, *CS(p)* is the closed set extended by *p* using synonyms.

OS_p is the opinion strength of *p* in the range [-1, 1].

3.4.2 Calculating the Opinion Strength(2)

- Taking the adjective “good” as an example.

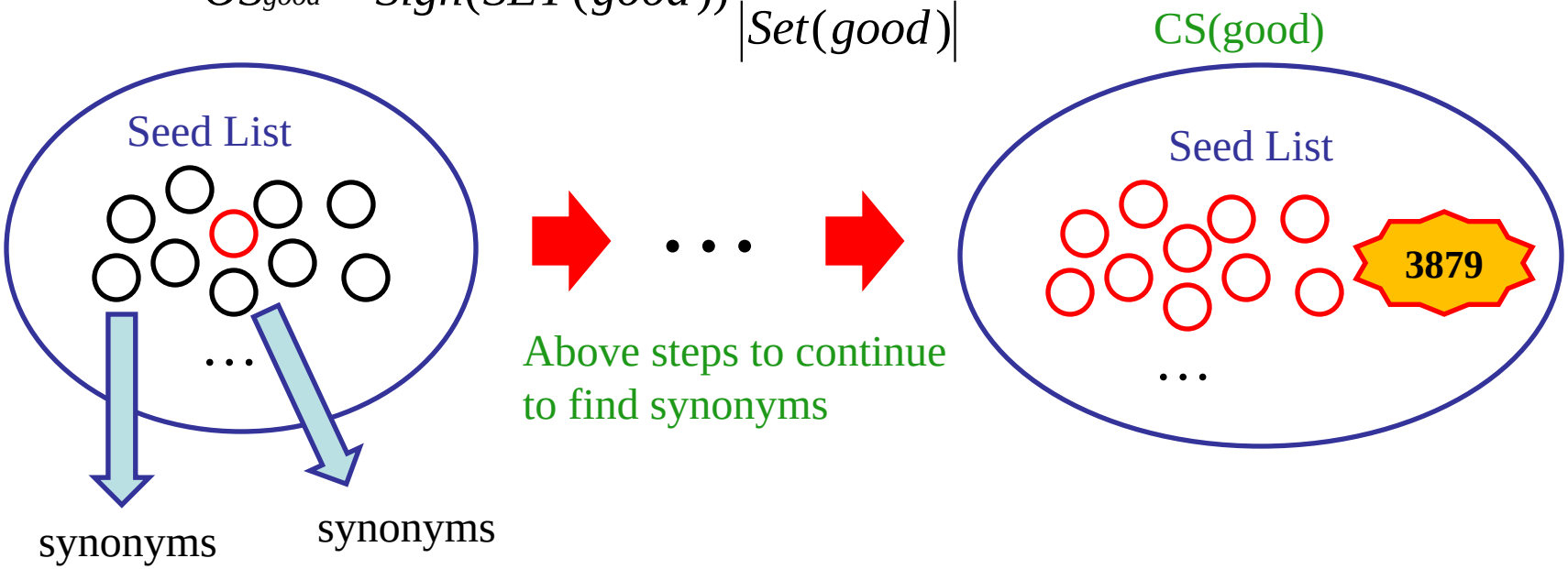
$$OS_{good} = Sign(SET(good))\frac{|CS(good)|}{|Set(good)|}$$



3.4.2 Calculating the Opinion Strength(3)

- Taking the adjective "good" as an example.

$$OS_{good} = Sign(SET(good))\frac{|CS(good)|}{|Set(good)|}$$



- Then, OS_{good} can be calculated as 3879/4308= +0.900418

3.4.3 Inverse Document Frequency

- Usually, the opinion strength of p (i.e., OS_p) is relatively high if p is a common adjective.

- Here, we use IDF to reduce the effects of common adjectives and enhance important adjectives. IDF could be calculated as follows.

$$IDF_p = \ln\left(\frac{R}{RCA_p}\right) \times \gamma, \qquad \gamma = \frac{1}{\ln(R)}$$

where RCA_p is the number of product reviews containing p, R is the number of all product reviews, and γ is a normalization formula making IDF value in the range [0, 1].

3.4.4 Degree of Adverbs

- An adverb can modify an adjective (i.e., Adverb + Adjective) and enhance or weaken the adjective strength or even change the polarity (i.e., not+Adjective).

- $Degree_p$ represents the degree of an adverb modifying an adjective *p*.

High level (0.6)	Medium level (0.5)	Low level (0.4)	Negative level (-1)
very, incredibly, much, so, too, completely, …, etc.	fairly, pretty, rather, as, almost, partly, half, …, etc.	slightly, a little, a bit, somewhat, …, etc.	not

- If no adverb is used to modify an adjective, the weight would be 0.5.

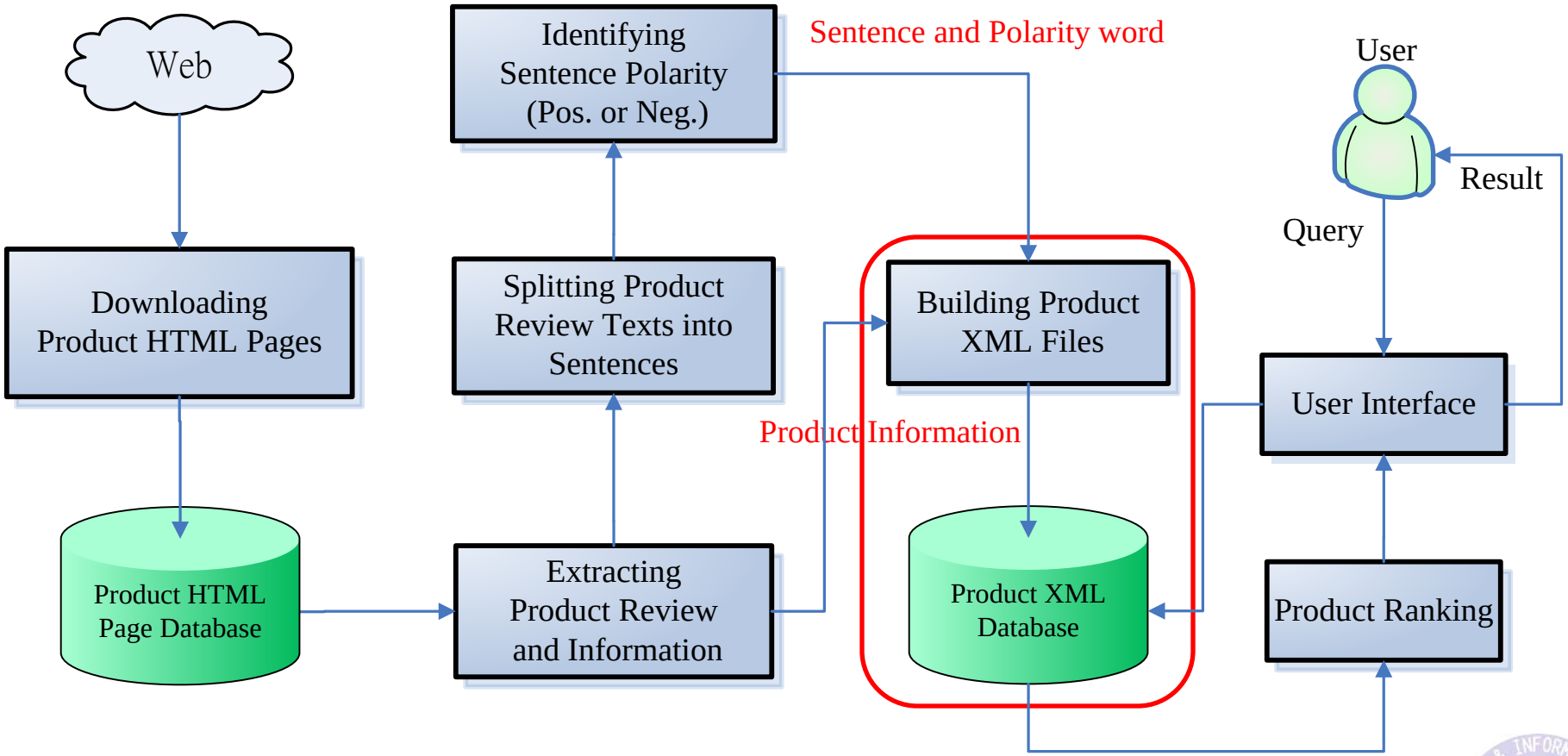
3.4.5 Calculating Sentence Polarity

- We use three weights described above to calculate sentence polarity as follows:

$$Sentence_p = OS_p \times IDF_p \times Degree_p$$

where p is an adjective, OS_p is the opinion strength of p, IDF_p is inverse document frequency of p, and $Degree_p$ is the degree of adverbs modifying p.

3.5 Building Product XML Files(1)



3.5 Building Product XML Files(2)

- The product information and sentence polarity would be integrated into an XML file.

- This file consists of three parts:
  - 1) product information.
  - 2) review section describing review information and sentence polarity.
  - 3) specific section describing other information.

3.5 Building Product XML Files(3)

- File structure

3.5 Building Product XML Files(4)

- Review section

```
<review reviewNumber="review number">
        <helpful>? of ? people found the following review helpful</helpful>
        <star>Posting people for product ratings</star>
        <title>Review title</title>
        <reviewPostDate>Review post date (RPD)</reviewPostDate>
        <originalReview>Original review</originalReview>
```

3.5 Building Product XML Files(5)

- Review section

```
<allSentence>
    <sentence sentenceNumber="sentence number">
    Sentence
        <adv tag="POS tag">
        Adverbs modify adjectives
            <advValue>Strength value</advValue>
        </adv>
        <adj tag="POS tag">
        adjective word
            <adjValue>
                <adjPolarity>positive or negtive</adjPolarity>
                <adjStrength>strength value</adjStrength>
            </adjValue>
        <numberOfReviewsThatContainTheAdj>
        Number of product review contains the adjectives
        (RCA)
        <numberOfReviewsThatContainTheAdj>
    </adj>
    </sentence>
    ⋮
</allSentence>
</review>
⋮
```

$$IDF_p = \ln\left(\frac{R}{RCA_p}\right) \times \gamma$$

Degree of Adverbs

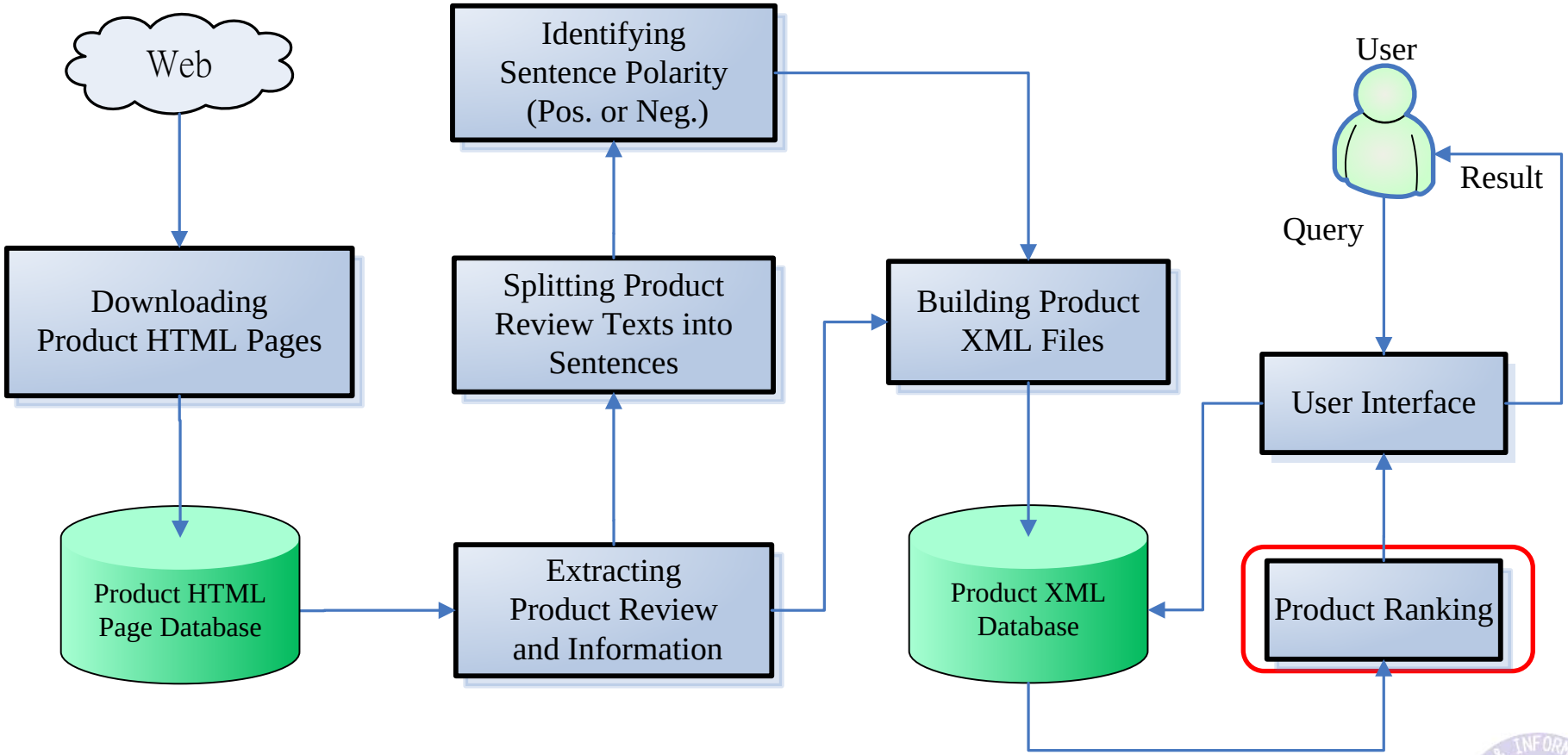
Opinion Strength

3.5 Building Product XML Files(6)

- Specific section

<Search for specific information>
<maximumNumberOfReview>
The maximum number of reviews ⟶ ranking
</maximumNumberOfReview>
<totalNumberOfReview>
All reviews in the number of the product kind ⟶ $IDF_p = \ln(\frac{R}{RCA_p}) \times \gamma,$
</totalNumberOfReview>
<earliestProductDateFirstAvailableValue>
The first product release date of the product kind ⟶ ranking
</earliestProductDateFirstAvailableValue>
<earliestReviewPostDateValue>
Posting date of the earliest reviews ⟶ ranking
</earliestReviewPostDateValue>
</Search for specific information>

3.6 Product Ranking(1)



3.6 Product Ranking(2)

- Our product ranking system is to rank the products specified by users, not all the products.

- Users can specify product features such as kind, brand, available date, and price as they need. Then, the system searches matched products and ranks them according to their scores.

3.6 Product Ranking(3)

- The product scores could be calculated as follows:

$$Score_i = APR_i \times PW_i \times WPRM_i$$

where i is product i, APR_i is the Average Polarity of Reviews, PW_i is the Popularity Weight, and $WPRM_i$ is the Weight of Product Release Month.

3.6.1 Average Polarity of Reviews (APR)(1)

- Since a product contains a number of reviews, the average polarity of all reviews should be considered for the product.
- *APR* could be calculated as follows:

$$APR_i = \frac{\sum_{j=1}^{n} [(Polarity_j + HF_j) \times WRPD_j]}{n}$$

where n is the number of reviews, $Polarity_j$ is the polarity of review j in the range [-1, 1].

HF_j is the extent of clicking "helpful" in the range (-0.99, 0.99).

$WRPD_j$ is the Weight of Review Post Date in the range (0.36, 0.99).

3.6.1 Average Polarity of Reviews (APR)(2)

- 1) *Polarity* could be calculated as follows:

$$Polarity_j = \frac{\sum_{p=1}^{k} Sentence_p}{k}$$, where k is the number of sentences in review j.

- 2) *HF* could be calculated as follows:

$$HF_j = \frac{\alpha \times [\ln(|Help_j - NotHelp_j| + 2)]}{10}$$

$$\alpha = \begin{cases} -1, \text{ if } 1)\ Polarity_j > 0 \text{ and } Help_j - NotHelp_j < 0 \\ \quad 2)\ Polarity_j < 0 \text{ and } Help_j - NotHelp_j > 0 \\ 1, \text{ if } 3)\ Polarity_j > 0 \text{ and } Help_j - NotHelp_j > 0 \\ \quad 4)\ Polarity_j < 0 \text{ and } Help_j - NotHelp_j < 0 \end{cases}$$

HF value is in the range (-0.99, 0.99).

Here, HF is used to adjust the polarity.

3.6.1 Average Polarity of Reviews (APR)(3)

- 3) *WRPD* could be calculated as follow:

$$WRPD_j = \exp\left(\frac{RPD_j - t}{30 \times \beta}\right), \quad \beta = \left\lceil \frac{t - \min(RPD_j)}{30} \right\rceil$$

where RPD_j is the post date of review j, t is the current date, and β normalizes *WRPD* value in the range (0.36, 0.99).

$$Score_i = APR_i \times PW_i \times WPRM_i$$

3.6.2 Popularity Weight (PW)

- The more popular a product is, the more discussion it has. Therefore, we use the number of product reviews to represent the product popularity.
- *PW* could be calculated as follows:

$$PW_i = \frac{\ln(m_i + 1)}{\ln(\max(m) + 1)}$$

where m_i is the number of reviews for product i, and $max(m)$ is the maximum number of reviews among all products.

3.6.3 Weight of Product Release Month (WPRM)

- *WPRM* is similar to *WRPD* in the calculation as follows:

$$WPRM_i = \exp\left(\frac{PRM_i - t}{12 \times \beta}\right), \quad \beta = \left\lceil \frac{t - \min(PRM_i)}{12} \right\rceil$$

where PRM_i is the release month of product i, t is the current month, and β normalizes *WPRM* value in the range (0.36, 0.99).

Building Product XML Files

- Specific section

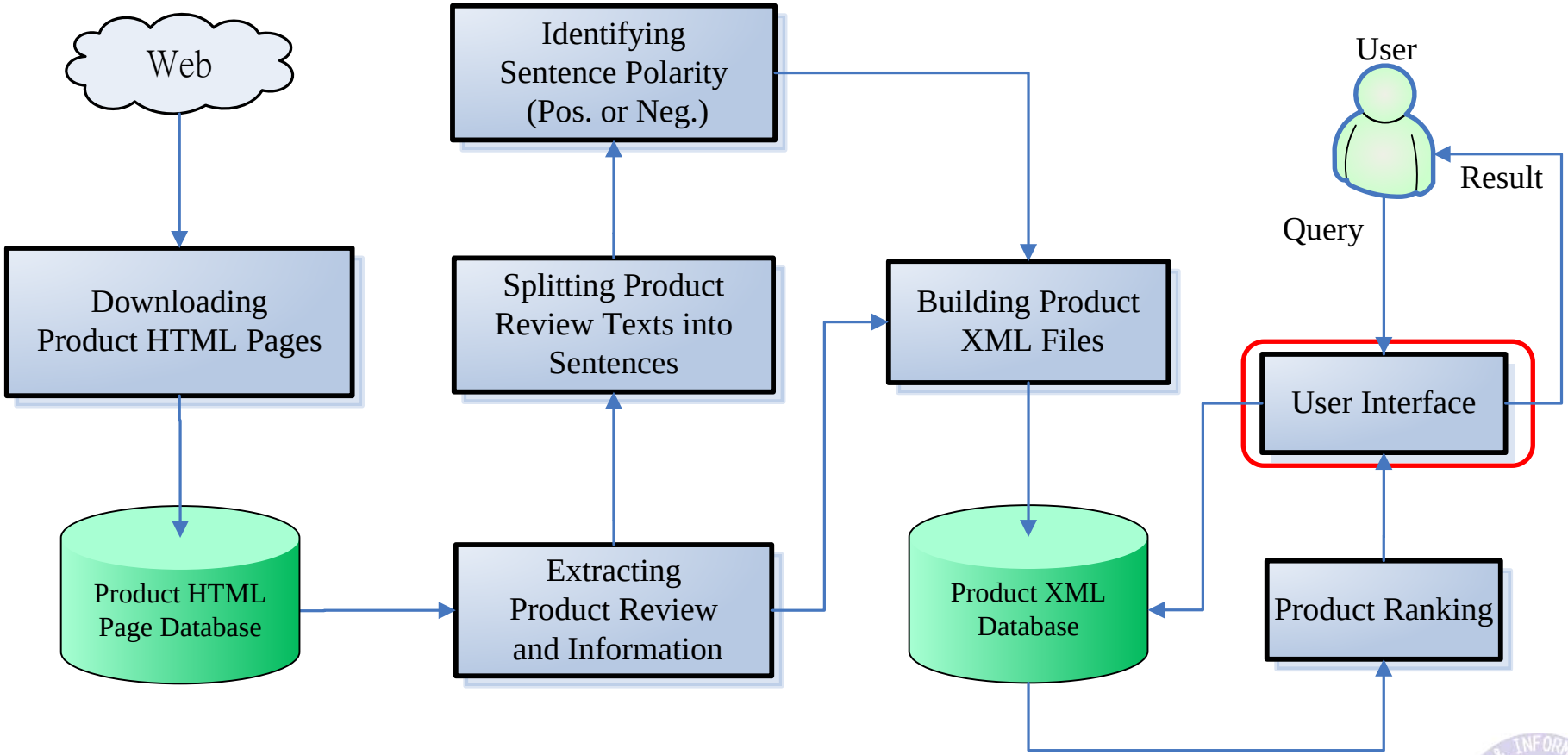
<Search for specific information>
<maximumNumberOfReview>
The maximum number of reviews
</maximumNumberOfReview>
<totalNumberOfReview>
All reviews in the number of the product kind
</totalNumberOfReview>
<earliestProductDateFirstAvailableValue>
The first product release date of the product kind
</earliestProductDateFirstAvailableValue>
<earliestReviewPostDateValue>
Posting date of the earliest reviews
</earliestReviewPostDateValue>
</Search for specific information>

$$PW_i = \frac{\ln(m_i + 1)}{\ln(\max(m) + 1)}$$

$$\beta = \left\lceil \frac{t - \min(PRM_i)}{12} \right\rceil$$

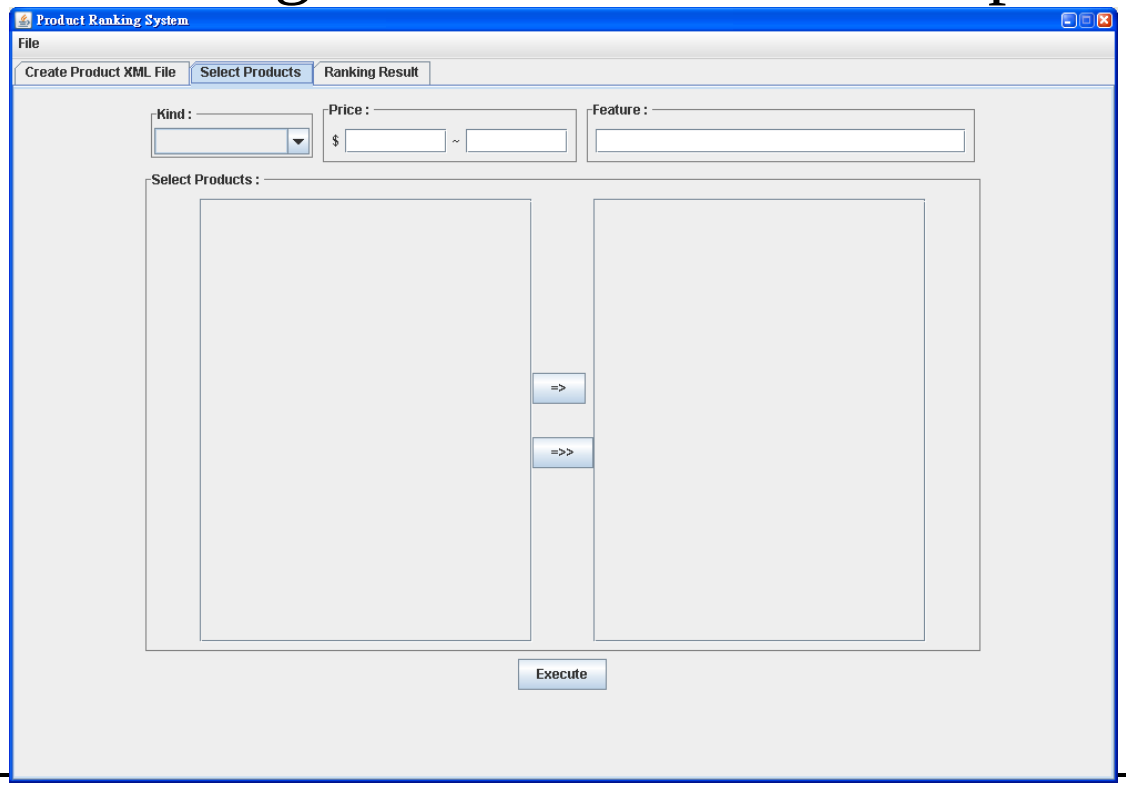
$$\beta = \left\lceil \frac{t - \min(RPD_j)}{30} \right\rceil$$

3.7 User Interface(1)



3.7 User Interface(2)

- We have implemented a user interface through which users can specify products and/or their features (e.g. quality, power, etc.) Through the product ranking system, users can get back the ranking results of all matched products.



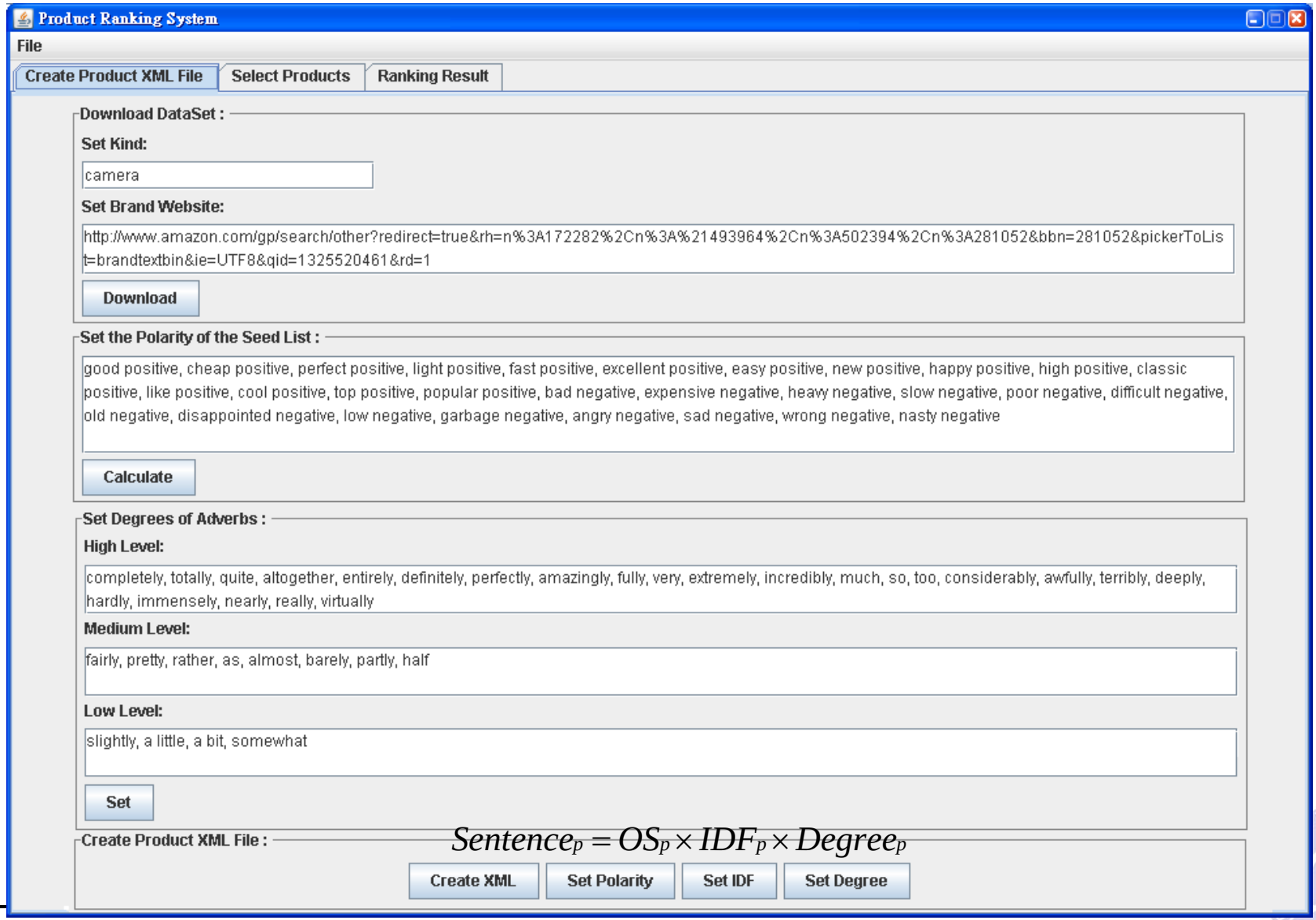
4. Implementations

4.1 Functional Design

- The product ranking system was implemented in Java and conducted on an Intel Core 2 Duo E7200 2533MHz CPU with 2G main memory in Window XP professional.

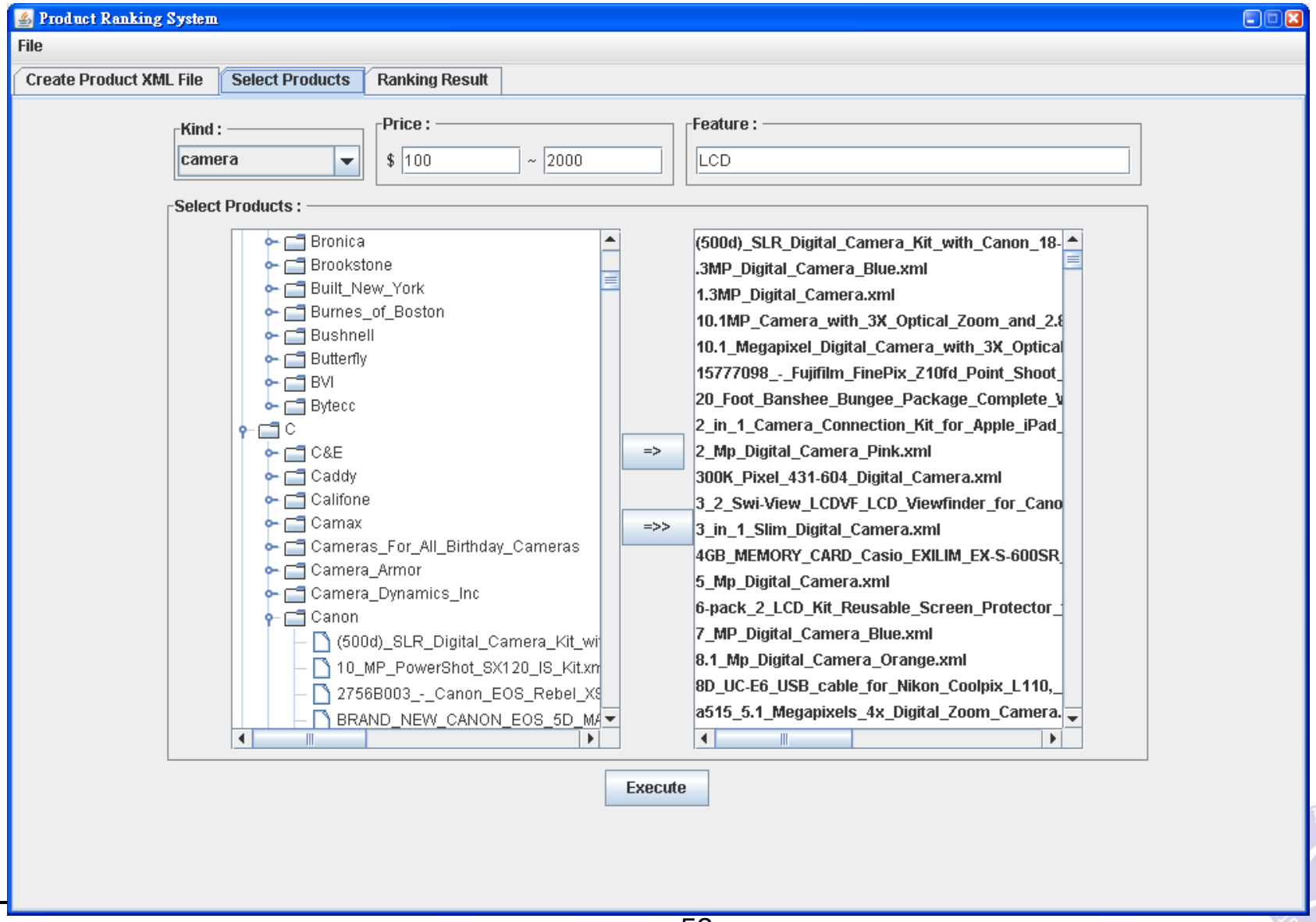
- In this system, we provide three functional pages for users:
  - 1) creating product XML files.
  - 2) selecting products.
  - 3) ranking results.

4.1.1 Creating Product XML Files

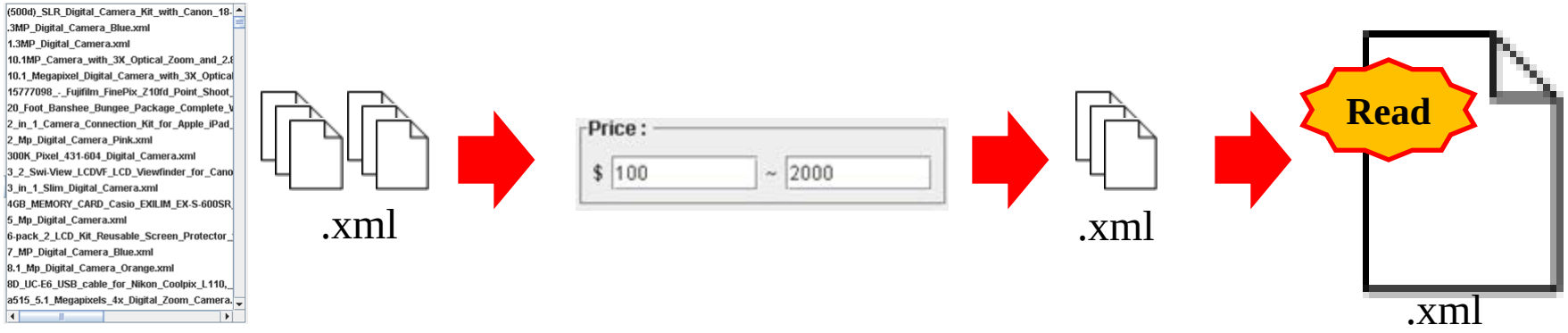


$$Sentence_p = OS_p \times IDF_p \times Degree_p$$

4.1.2 Selecting Products(1)



4.1.2 Selecting Products(2)



$$Polarity_j = \frac{\sum_{p=1}^{k} Sentence_p}{k}$$

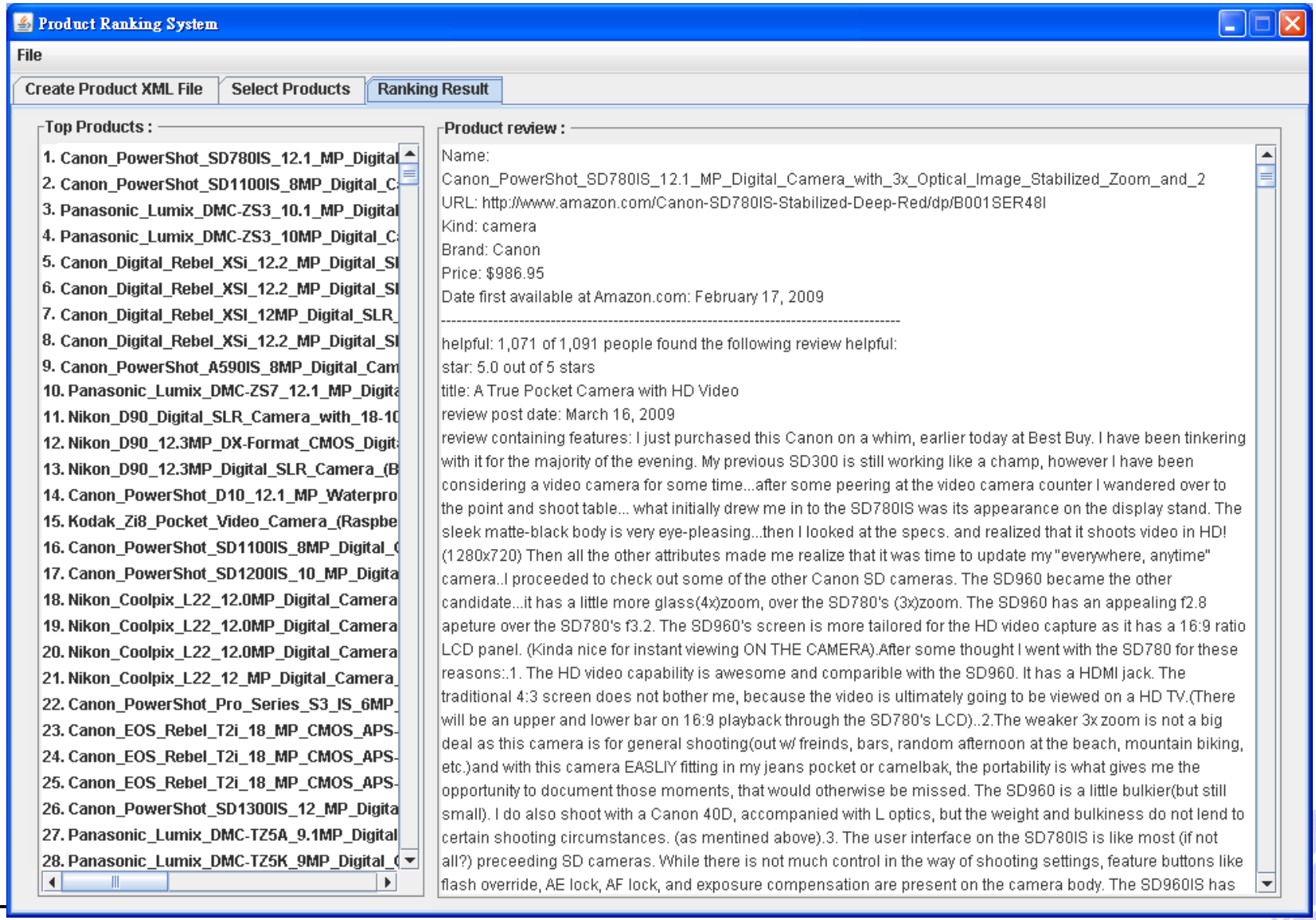
Sentence contains the feature to calculate

$$Sentence_p = OS_p \times IDF_p \times Degree_p$$

$$APR_i = \frac{\sum_{j=1}^{n} [(Polarity_j + HF_j) \times WRPD_j]}{n}$$

$$Score_i = APR_i \times PW_i \times WPRM_i$$

4.1.3 Ranking Results



4.2 Experimental Results(1)

- Here, the dataset used in the experiment is downloaded from the Amazon product home page; three kinds of products tested in the system are “camera”, “laptop”, and “mobile phone”.

Kind of product	camera	laptop	mobile phone
Number of products	3,205	270	1,847
Number of reviews	168,285	2,935	29,041
Number of sentences	1,452,391	43,247	97,743

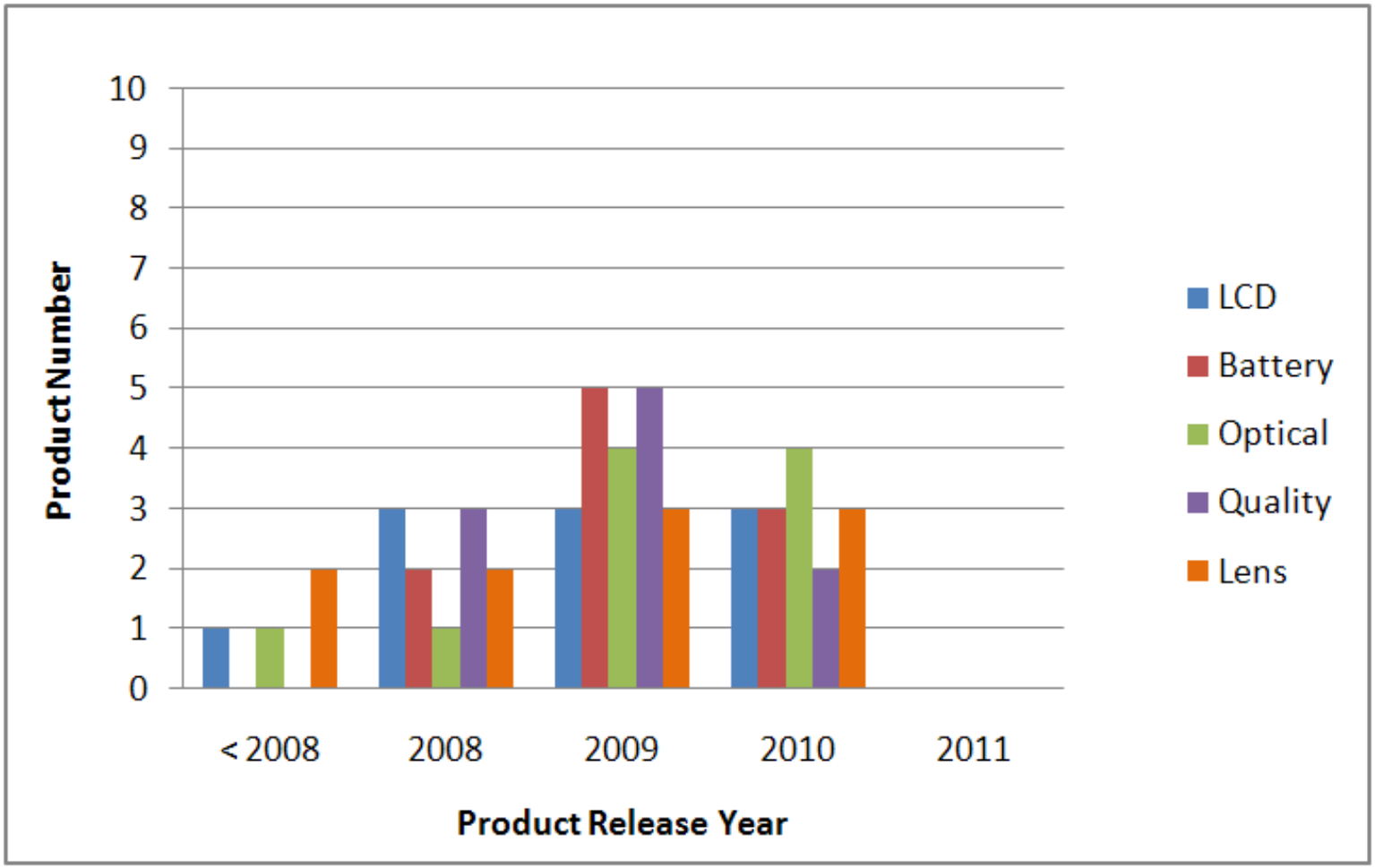
4.2 Experimental Results(2)

- In the dataset from the Amazon product home page, the release months of products (i.e., W*PRM*) are only provided for “camera” and “laptop”, but not for “mobile phone”.

- According to the formula used to calculate product scores, the popularity weight (i.e., *PW*) is favorable for old products whereas the weight of product release month (i.e., *WPRM*) is favorable for new products. Next, we would like to observe what ranking happens to new and old products in these three kinds.

4.2 Experimental Results(3)

- Release year distribution of top-ten cameras with specified features.

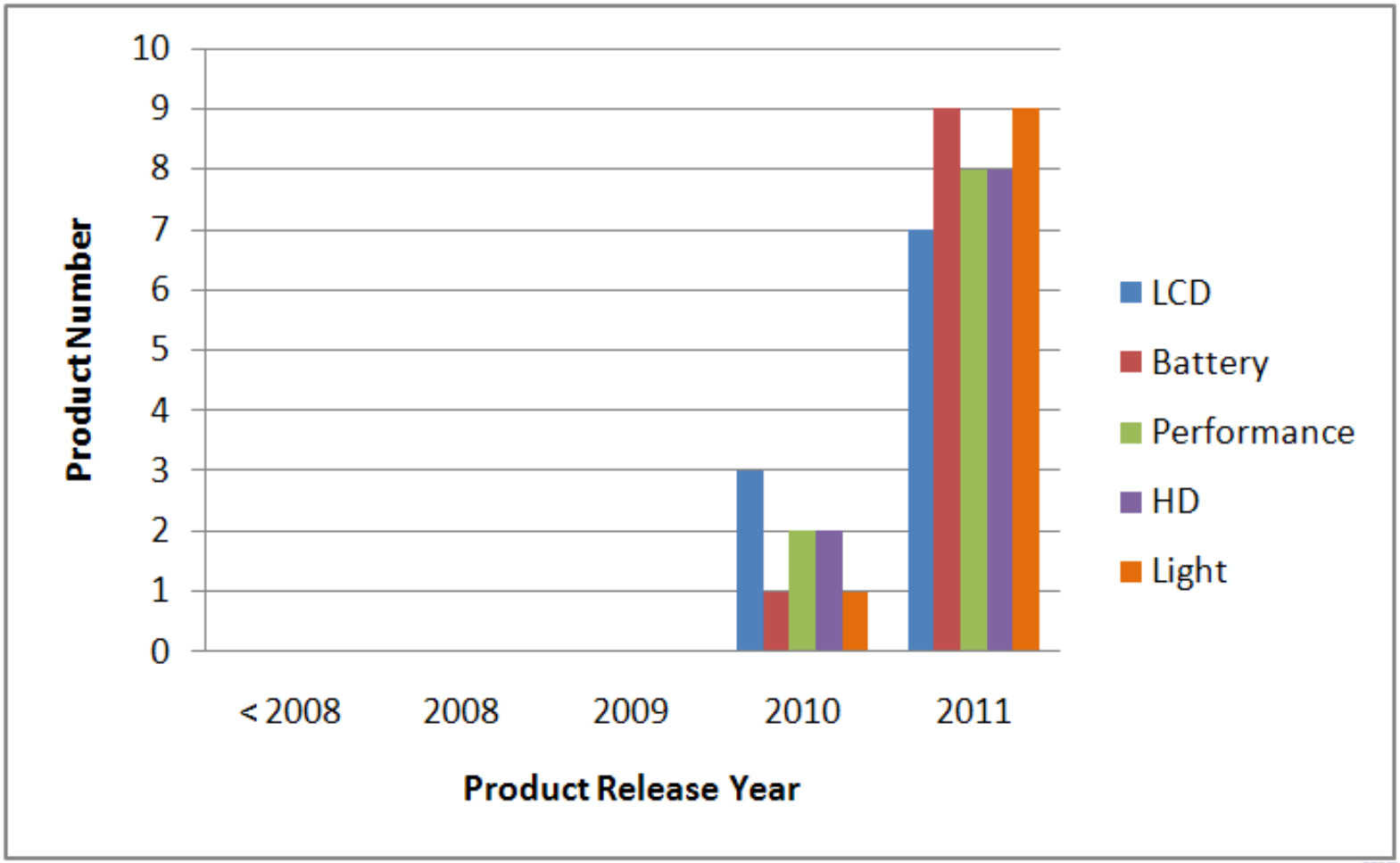


4.2 Experimental Results(4)

Kind of product	camera	laptop	mobile phone
Number of products	3,205	270	1,847
Number of reviews	168,285	2,935	29,041
Number of sentences	1,452,391	43,247	97,743

4.2 Experimental Results(5)

- Release year distribution of top-ten laptops with specified features.

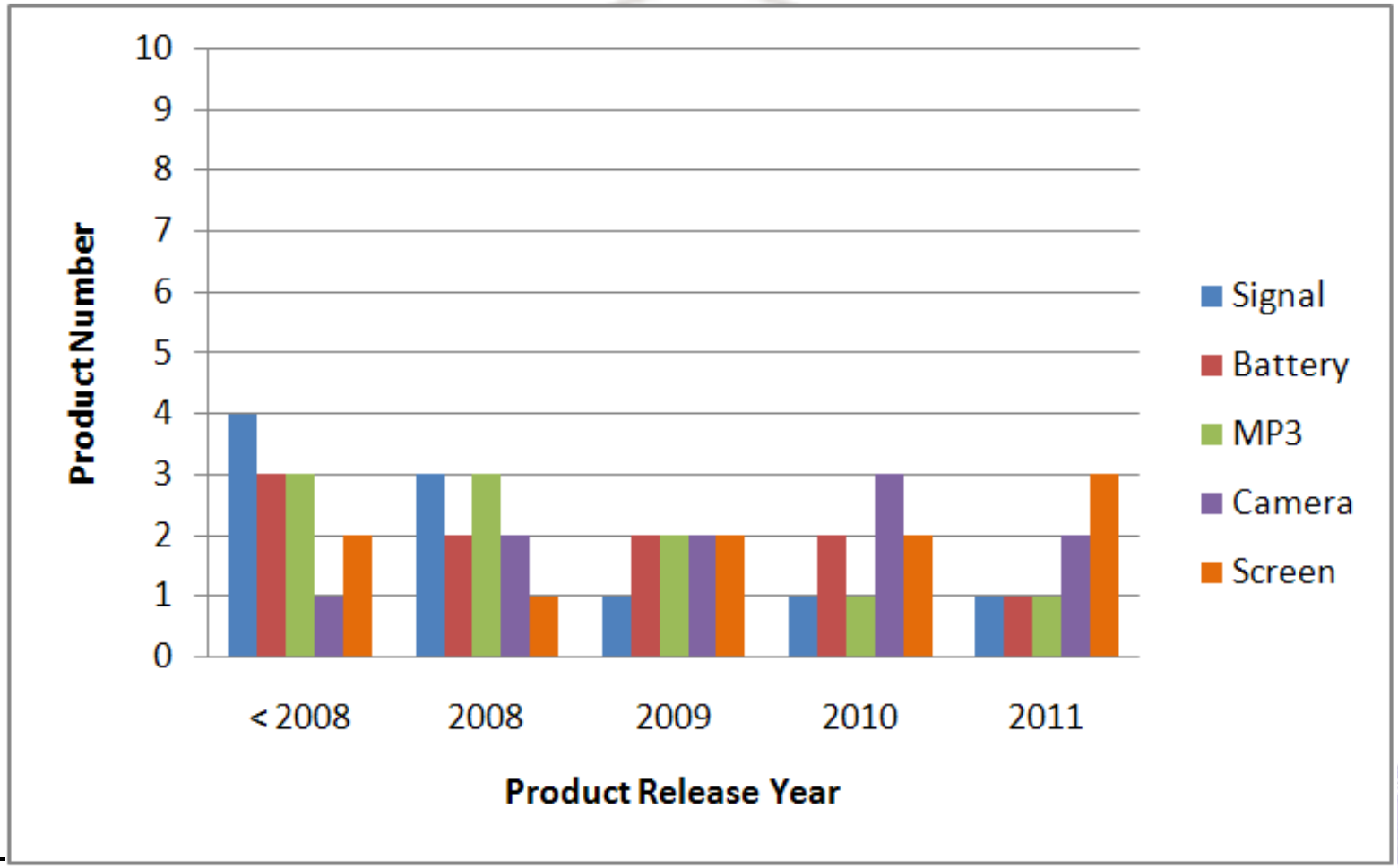


4.2 Experimental Results(6)

Kind of product	camera	laptop	mobile phone
Number of products	3,205	270	1,847
Number of reviews	168,285	2,935	29,041
Number of sentences	1,452,391	43,247	97,743

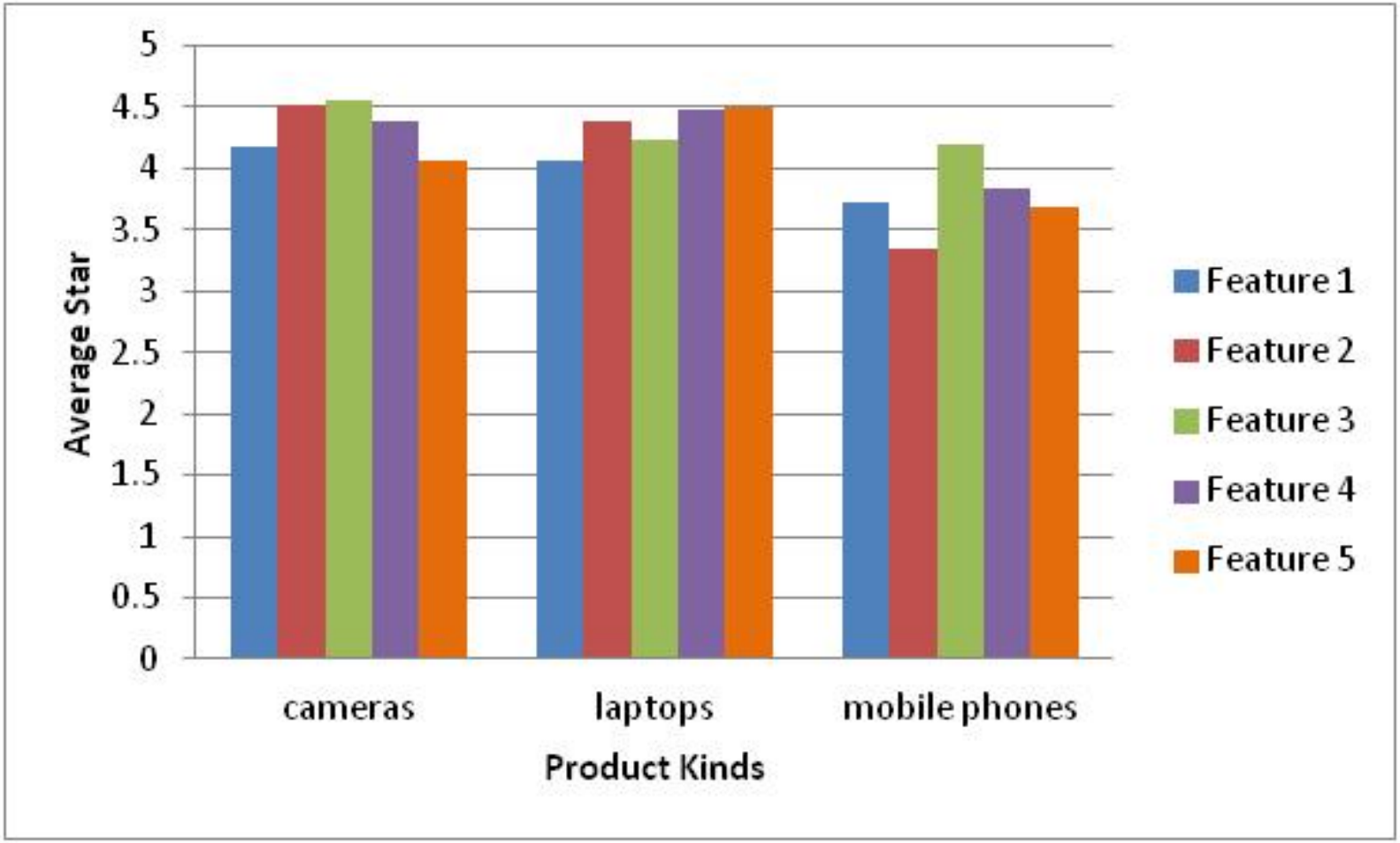
4.2 Experimental Results(7)

- Release year distribution of top-ten mobile phones with specified feature, and no WPRM is counted in.



4.2 Experimental Results(8)

- Average stars of top-ten cameras laptops, and mobile phones with specified features.



5. Conclusions

5. Conclusions

- In this research, we propose a product ranking system where users can specify product features to get back the ranking results of all matched products.
- The experimental results show that the system is practical and the ranking results are interesting.
- Especially, the system can be used to find the release year distribution of top-ten products with specified features. The results reveal that new products are not always more favorable than old products.

Demo

7 Core Skills You'll need

- Cloud Computing
- Social Media
- Information Security
- JQuery – a JavaScript library
- Mobile App Development
- HTML5
- Database Management

國立雲林科技大學
National Yunlin University of Science & Technology

Thank You !