Research Paper About HMM-based Learning Framework for Web Opinion Mining

Vitalii Naumov

Eötvös Lorand University, Hungary

Abstract. Nowadays it has become normal to order goods online. With every order most of the companies ask for a feedback of their services or products. With increasing amount of data, the time spent on processing of the data is increasing as well. So, to solve the problem of manually reading the reviews this system was built. It brought my attention because of using modified version of Hidden Markov Models – lexicalized HMMs.

Keywords: HMM, NLP, POS, NER.

1 Introduction

1.1 Hidden Markov Models

The HMM is based on augmenting the Markov chain. A Markov chain is a model that tells us something about the probabilities of sequences of random variables, states, each of which can take on values from some set. These sets can be words, or tags, or symbols representing anything, like the weather. A Markov chain makes a very strong assumption that if we want to predict the future in the sequence, all that matters is the current state. The states before the current state have no impact on the future except via the current state. It's as if to predict tomorrow's weather you could examine today's weather, but you weren't allowed to look at yesterday's weather.

In many cases, however, the events we are interested in are hidden - we don't observe them directly. We call the tags hidden because they are not observed. A hidden Markov model (HMM) allows us to talk about both observed events (like words that we see in the input) and hidden events (like part-of-speech tags) that we think of as causal factors in our probabilistic model. An HMM is specified by the following components:

$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} \ldots a_{ij} \ldots a_{NN}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{N} a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \ldots o_T$	a sequence of T **observations**, each one drawn from a vocabulary $V = v_1, v_2, ..., v_V$
$B = b_i(o_t)$	a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t being generated from a state i
$\pi = \pi_1, \pi_2, ..., \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$

A first-order hidden Markov model instantiates two simplifying assumptions. First, as with a first-order Markov chain, the probability of a particular state depends only on the previous state:

$$\textbf{Markov Assumption:} \quad P(q_i|q_1 ... q_{i-1}) = P(q_i|q_{i-1})$$

Second, the probability of an output observation oi depends only on the state that produced the observation qi and not on any other states or any other observations:

$$\textbf{Output Independence:} \quad P(o_i|q_1 \ldots q_i, \ldots, q_T, o_1, \ldots, o_i, \ldots, o_T) = P(o_i|q_i)$$

Schematic explanation of HMM is shown on the Fig.0:

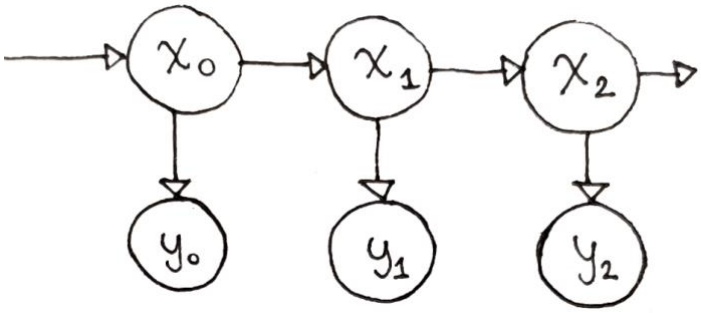

Fig 0. the probability of a particular state (x1) depends only on the previous state (x0); an output observation (y1) o_i depends only on the state that produced the observation (x1)

1.2 Field of study

We live in a time when the segment of goods sold online is constantly growing. It has become a common practice for online merchants to ask their customers to share their opinions and hands-on experiences on products they have purchased. Unfortunately,

reading through all customer reviews is difficult, especially for popular items, the number of reviews can be up to hundreds or even thousands. The goal of this research is to design a framework that is capable of extracting, learning and classifying product related entities from product reviews.

Specifically, given a particular product, the system first identifies potential product related entities and opinion related entities from the reviews, and then extracts opinion sentences which describe each identified product entity, and finally determines opinion orientations (positive or negative) for each recognized product entity. This research proposes a novel framework naturally integrates linguistic features (e.g., part-of-speech, phrases internal formation patterns, and surrounding contextual clues of words/phrases) into automatic learning supported by lexicalized HMMs.

2 Related work

Opinion mining can be divided into two categories, document level and feature level. Document level aims to classify the overall sentiment orientation of a document; feature level is interested in finding product features being commented on and the opinion polarity for each feature. In this paper, focus is on feature level opinion mining and propose it involves two major tasks, recognition and classification. Recognition is the task of recognizing sentences expressing opinions; classification is the task of classifying elements in an opinion sentence into different categories such as opinion words/phrases and product features. Determining the polarity of opinion words (positive or negative) is also a classification task.

In this work, the framework naturally integrates linguistic features into automatic learning. The system can identify complex product-specific features (which are possible low frequency phrases in the reviews). The system can also self-learn new vocabularies based on the patterns it has seen from the training data. Therefore, the system is able to predict potential features in the test dataset even without seeing them in the training set. These capabilities differ the project from all other previous researches.

3 The proposed techniques

Lexicalized HMMs was previously used in Part-of- Speech (POS) Tagging and Named Entity Recognition (NER) problem. The task of POS tagging is the process of marking up the words in a text (corpus) as corresponding to a particular part-of-speech, such as noun and verb. The task of NER is identifying and classifying person names, location names, organization names, and etc. I believe that the lexicalization technique with POS information will be able to handle more information of tags to known words, including contextual words and contextual tags under the HMM. Figure 1 gives the architectural overview of the opinion mining system.

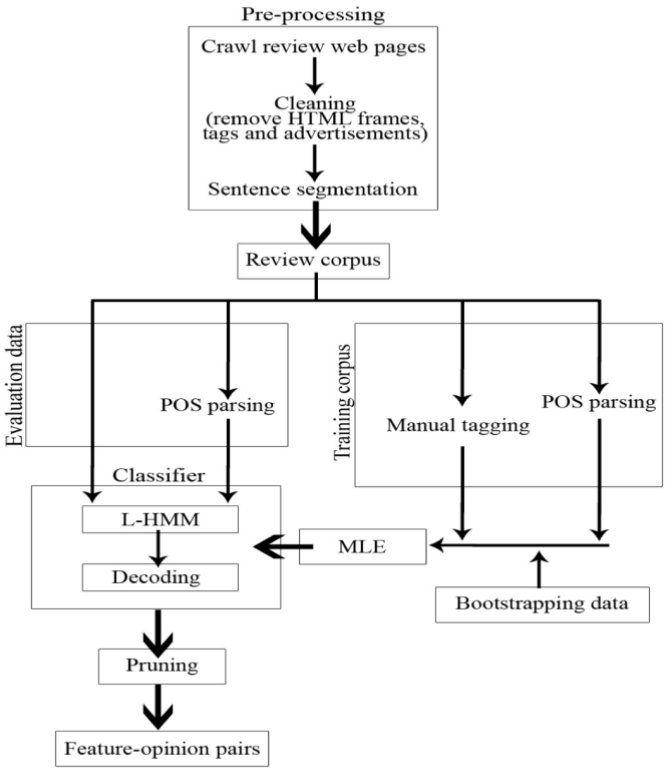

Figure 1. The system framework

3.1 Entity categories and tag sets

Authors used four entity categories as shown in table 1:

Table 1. Definitions of entity categories and examples

COMPONENTS	Physical objects of a camera including the camera itself, e.g., LCD, viewfinder, battery
FUNCTIONS	Capabilities provided by a camera, e.g., movie playback, zoom, automatic fill-flash, auto focus
FEATURES	Properties of components or functions, e.g., color, speed, size, weight, clarity
OPINIONS	Ideas and thoughts expressed by reviewers on product features / components / functions.

The basic tag set is given in table 2:

Table 2. Basic tag set and its corresponding entities

TAGS	CORRESPONDING ENTITIES
<PROD_FEAT>	Feature Entities
<PROD_COMP>	Component Entities
<PROD_FUNCTION>	Function Entities
<OPINION_POS_EXP>	Explicit Positive Opinion Entities
<OPINION_NEG_EXP>	Explicit Negative Opinion Entities
<OPINION_POS_IMP>	Implicit Positive Opinion Entities
<OPINION_NEG_IMP>	Implicit Negative Opinion Entities
<BG>	Background Words

A word w in an entity may take one of the following four patterns to present itself:

- *w* is an independent entity;
- *w* is the beginning component of an entity;
- *w* is at the middle of an entity;
- *w* is at the end of an entity.

So, for each of the patterns mentioned above they have created 4 types of tags given in table 3:

Table 3. Pattern tag set and its corresponding patterns

PATTERN TAGS	CORRESPONDING PATTERNS
<>	Independent Entities (Single Words)
<BOE>	The Beginning Component of an Entity
<MOE>	The Middle Component of an Entity
<EOE>	The End of an Entity

The following example illustrates the hybrid tag and basic tag representations of an opinion sentence "I love the ease of transferring the pictures to my computer." Patterns of background words are considered as independent entities.

Hybrid tags:

<BG>I</BG><OPINION_POS_EXP>love</OPINION_POS_EXP><BG>the</BG><PROD_FEAT-BOE>ease</PROD_FEAT-BOE> <PROD_FEAT-MOE>of</PROD_FEAT-MOE><PROD_FEAT-MOE>transferring</PROD_FEAT-MOE> <PROD_FEAT-MOE>the</PROD_FEAT-MOE> <PROD_FEAT-EOE>pictures</PROD_FEAT-EOE> <BG>to</BG><BG>my</BG><BG>computer</BG>

Basic tags:

<BG>I</BG><OPINION_POS_EXP>love</OPINION_POS_EXP> <BG> the </BG> <PROD_FEAT>ease of transferring the pictures </PROD_FEAT> <BG> to </BG><BG>my</BG><BG>computer</BG>

3.2 Lexicalized HMMs

In this project scientists decided to use not just a traditional HMM but HMM with integrated linguistic features such as part-of-speech and lexical patterns. I believe this was one of the most important decisions which caused great results they mentioned in the original report, because all this method was never used before in exactly this field such as opinion mining.

An observable state is represented by a pair ($word_i$, $POS(word_i)$) where $POS(word_i)$ represents the part-of-speech of $word_i$. Given a sequence of words $W = w_1w_2w_3...w_n$ and corresponding parts-of-speech $S = s_1s_2s_3...s_n$, the task is to find an appropriate sequence of hybrid tags $T = t_1t_2t_3...t_n$ that maximize the conditional probability P(T|W,S), namely

$$\hat{T} = \arg\max_T P(T \mid W, S) = \arg\max_T \frac{P(W, S \mid T)P(T)}{P(W, S)} \qquad (1)$$

Since the probability P(W, S) remains unchanged for all candidate tag sequences, we can disregard it. Thus, we have a general statistical model as follows:

$$\hat{T} = \underset{T}{\operatorname{argmax}} P(W,S \mid T)P(T) = \underset{T}{\operatorname{argmax}} P(S \mid T)P(W \mid T,S)p(T)$$
$$= \underset{T}{\operatorname{argmax}} \prod_{i=1}^{n} \begin{pmatrix} P(s_i \mid w_1 ... w_{i-1}, s_1 .. s_{i-1}, t_1 .. t_{i-1} t_i) \times \\ P(w_i \mid w_1 ... w_{i-1}, s_1 .. s_{i-1} s_i, t_1 .. t_{i-1} t_i) \times \\ P(t_i \mid w_1 ... w_{i-1}, s_1 .. s_{i-1}, t_1 .. t_{i-1}) \end{pmatrix} \quad (2)$$

The general model described above usually is not computable since it involves too many parameters. So, to resolve this problem there were implemented two approximations to simplify the model.

The first approximation is based on the independent hypothesis used in standard HMMs. First-order HMMs is used in view of data sparseness, i.e.,

$$P(t_i \mid t_{i-K} \ldots t_{i-1}) \approx P(t_i \mid t_{i-1}).$$

The second approximation combines the POS information with the lexicalization technique where three main hypotheses are made:

- The assignment of current tag t_i is supposed to depend not only on its previous tag t_{i1} but also previous J ($1 \le J \le i-1$) words $w_{i-J} ... w_{i-1}$.
- The appearance of current word w_i is assumed to depend not only on the current tag t_i, current POS s_i, but also the previous K($1 \le K \le i-1$) words $w_{i-K} ... w_{i-1}$.
- The appearance of current POS s_i is supposed to depend both on the current tag t_i and previous L($1 \le L \le i-1$) words $w_{i-L} ... w_{i-1}$.

To avoid the issue of data sparseness, they set J=K=L=1. Based on these assumptions, the general model in equation (2) was rewritten as:

$$\hat{T} = \arg \max_{T} \prod_{i=1}^{n} \begin{pmatrix} P(s_i \mid w_{i-1}, t_i) \times \\ P(w_i \mid w_{i-1}, s_i, t_i) \times \\ P(t_i \mid w_{i-1}, t_{i-1}) \end{pmatrix} \quad (3)$$

Maximum Likelihood Estimation (MLE) is used to estimate the parameters in equation (3). For instance, $P(s_i \mid w_{i-1}, t_i)$ can be estimated as:

$$P(s_i \mid w_{i-1}, t_i) = \frac{C(w_{i-1}, t_i, s_i)}{\sum_s C(w_{i-1}, t_i, s)} = \frac{C(w_{i-1}, t_i, s_i)}{C(w_{i-1}, t_i)} \quad (4)$$

The sum of counts of $C(w_{i-1}, t_i, s)$ for all s is equivalent to the count of $C(w_{i-1}, t_i)$. MLE values for other estimations in equation (3) can be computed similarly.

However, MLE will yield zero probabilities for any cases that are not observed in the training data. To solve this problem, they employ the linear interpolation smoothing technique to smooth higher-order models with their relevant lower-order models, or to smooth the lexicalized parameters using the related non-lexicalized probabilities, namely

$$
\begin{aligned}
P'(s_i \mid w_{i-1}, t_i) &= \lambda P(s_i \mid w_{i-1}, t_i) + (1-\lambda) P(s_i \mid t_i) \\
P'(w_i \mid w_{i-1}, s_i, t_i) &= \begin{pmatrix} \beta P(w_i \mid w_{i-1}, s_i, t_i) + \\ (1-\beta) P(w_i \mid s_i, t_i) \end{pmatrix} \\
P'(t_i \mid w_{i-1}, t_{i-1}) &= \alpha P(t_i \mid w_{i-1}, t_{i-1}) + (1-\alpha) P(t_i \mid t_{i-1})
\end{aligned} \tag{5}
$$

Where λ, β and α denote the interpolation coefficients (In terms of F- score, the settings 0.7 for λ, β and α achieved the best performance in their experiments).

3.3 Tagging

The algorithm contains three major steps as follows:

1. The generation of candidate tags
2. The decoding of the best tag sequence
3. The conversion of the results

3.4 Opinion sentence extraction

This step identifies opinion sentences in the reviews. In the pruning step, the following two types of sentences are not considered as effective opinion sentences:

1. Sentences that describe product related entities without expressing reviewers' opinions.
2. Sentences that express opinions on another product model's entities.

3.5 Determining opinion orientation

The pseudocode is shown in Algorithm 1. Shortly, the steps are following: conversion the hybrid tagged sentences to basic tagged sentences, then looking a matching opinion entity for each recognized product entity, which is defined as the nearest opinion word/phrase identified by the tagger and only then apply natural language rules to deal with specific cases.

Line 8 to line 23 checks the presence of any negation words within five-word distance in front of an opinion word/phrase and changes opinion orientation accordingly, except

- A negation word appears in front of a coordinating conjunction (e.g., and, or, but). (line 10 – 13)
- A negation word appears after the appearance of a product entity during the backward search within the five-word window. (line 14 -17)

Line 27 to 32 handles the coordinating conjunction "*but*" and prepositions such as "*except*" and "apart from". The purpose of this procedure is to resolve the true opinion polarity for product entities when:

- The opinion expression in the “but” clause is unsolvable.
- The opinion is expressed on a bunch of product entities except some.

Algorithm 1 Determining Opinion Orientation

```
RESOLVE_OPINION_ORI(tagged OpinionSentence)
1. FOR each product related entity f_i in OpinionSentence
2.   corresponding opinion entity o_i = f_i's matching
3.   opinion word/phrase
4.  f_i's initial opinion orientation = o_i's orientation
5.
6.   // look backwards and search for negation words
7.  done = FALSE
8.  FOR (distance = 1; distance <= 5 && !done;
9.  distance++)
10.    IF ((o_i's position - distance) is a coordinating
11.    conjunction)
12.       done = TRUE
13.    END IF
14.    IF ((o_i's position - distance) is in front of f_i's
15     position)
16.       done = TRUE
17.    END IF
18.    IF ((o_i's position - distance) is a negation word)
19.         done = TRUE
20.         f_i's opinion orientation = opposite(f_i's initial
21          opinion orientation)
22.    END IF
23.  END FOR
24.
25. // handling the conjunctions such as “but” and
26     prepositions such as “except”
27.  IF o_i is in front of f_i
28.     IF “but/except” appears between o_i and f_i
29.        f_i's opinion orientation = opposite(f_i's initial
30.        opinion orientation)
31.     END IF
32.  END IF
33. END FOR
```

4 Experiments

Authors used Amazon’s digital camera reviews as the evaluation dataset. The reviews for the first 16 unique cameras listed on Amazon.com. POS parsing was applied to each review document. Then they used the Part-of-Speech tagger designed by the Stanford NLP Group1 and default settings of the tagger were used.

4.1 Training Design

After downloading and pre-processing, there were 1728 review documents obtained. They split the documents into 2 sets. One set (293 documents for 6 cameras) were manually tagged by experts who put all the tags manually. The remaining documents (1435 documents for 10 cameras) were used by the bootstrapping process to self-learn new vocabularies (described next). The second main interesting thing in the project for me, was enabling self-directed learning, which can be employed in situations where collecting a large training set could be expensive and difficult to accomplish.

4.2 Bootstrapping

Labeling training documents would be too intensive task - they thought and designed a self-learning procedure. The process is shown in Fig. 2 with the following steps:

- First, the bootstrapping program creates two child processes. The parent process acts as master and the rest acts as workers. Master is responsible for coordinating the bootstrapping process, extracting and distributing high confidence data to each worker.
- Split the training documents into two halves, t_1 and t_2 by random selection. Each half is used as seeds for each worker's HMM.
- At the initial stage (0^{th} iteration), each worker trains its own HMM classifier based on its training set, and then each worker trained HMM is used to tag the documents in the bootstrap document set and produces a new set of tagged review documents.
- Master inspects each sentence tagged by each HMM classifier and only extracts opinion sentences that are agreed upon by both classifiers.
- A hash value is then calculated for each extracted opinion sentence from step 4 and compared with those of the sentences already stored in the database. If it is a newly discovered sentence, master stores it into the database.
- Master randomly splits the newly discovered data from the database into two halves. This bootstrap process is repeated until no newer data being discovered.

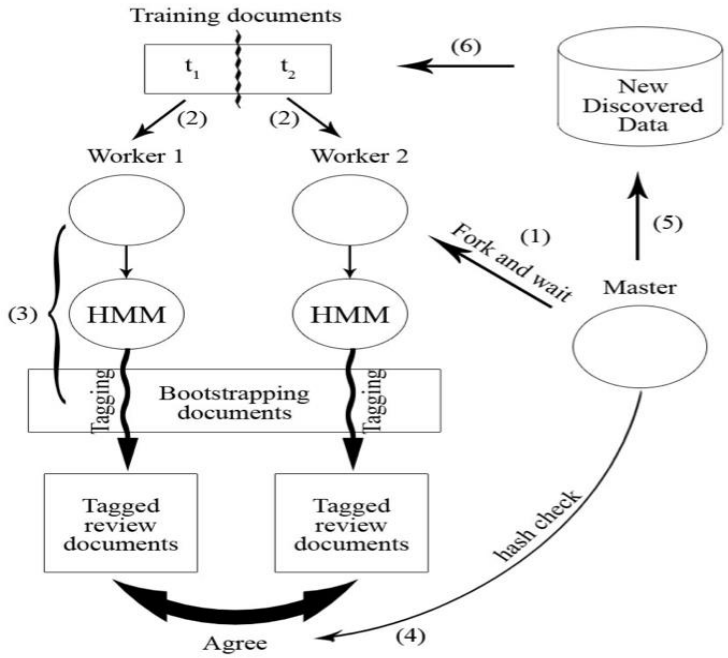

Figure 2. The bootstrapping process

Figure 3 and 4 demonstrate the experimental results obtained from each bootstrap cycle regarding one of the products used in our experiments

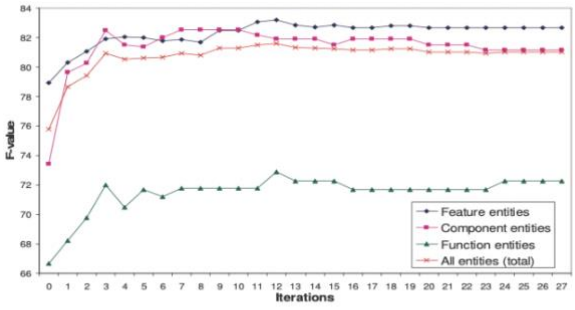

Figure 3. Bootstrapping results for entity extraction

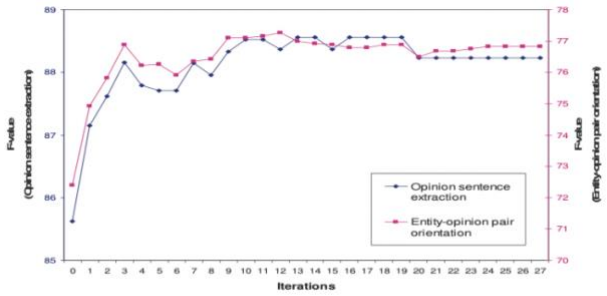

Figure 4. Bootstrapping results for opinion extraction and entity-opinion pair orientation

5 Evaluation

To evaluate the effectiveness of the proposed framework, recall, precision and F-score of extracted entities were calculated, opinion sentences and opinion orientations, respectively as well. The system performance is evaluated by comparing the results tagged by the system with the manually tagged truth data. Only an exact match is considered as a correct recognition in our evaluation.

The detailed evaluation results are presented in Table 5, 6 and 7. As a post analysis, the proposed machine learning approach performs significantly better than the rule-based baseline system in terms of entity extraction, opinion sentence recognition and opinion polarity classification.

6 Conclusion

To conclude my research about research, I would like to say that the proposed approach performs much better than the rule-based systems when we are talking about opinion recognition, opinion classification or entity extraction. I strongly believe that present results were achieved due to using HMM and a self-learning procedure of auto-creation of a vocabulary. I included HMM in the list because it helps to find the tags of the words and with these tags, we can realize which class it will belong to. Self-learning algorithm helps to collect a large training set when it could be expensive and difficult to accomplish.

Relying on my humble knowledge in NLP, I would say the field of opinion mining is quite competitive and further this implementation might get stack with several problems such as sarcasm detection, complex construction reviews with opposite opinions or not a proper language constructs, where will be difficult for the system to detect a mood.

Table 5. Experimental results on entity extraction for each category

Products	Methods	Feature Entity			Component Entity			Function Entity		
		R(%)	P(%)	F(%)	R(%)	P(%)	F(%)	R(%)	P(%)	F(%)
Camera A	L-HMM+POS+Bootstrapping	85.81	80.78	83.22	83.33	83.33	83.33	70.31	75.00	72.58
	L-HMM+POS	82.01	77.70	79.80	73.08	73.08	73.08	65.63	70.00	67.74
	L-HMM	80.74	75.86	78.22	70.36	70.30	70.33	60.19	67.19	63.50
Camera B	L-HMM+POS+Bootstrapping	89.12	75.72	81.88	77.66	79.35	78.49	60.87	82.35	70.00
	L-HMM+POS	84.35	72.09	77.74	74.47	70.92	72.65	52.17	75.00	61.54
	L-HMM	80.67	71.51	75.81	71.66	70.46	71.05	47.83	64.71	55.00
Camera C	L-HMM+POS+Bootstrapping	76.62	81.94	79.19	100.0	83.67	91.11	63.64	87.50	73.69
	L-HMM+POS	74.03	80.28	77.03	97.56	78.43	86.96	63.64	80.50	71.08
	L-HMM	70.32	77.33	73.66	97.56	74.07	84.21	63.64	70.00	66.67

Table 6. Experimental results on entity extraction for all categories

Products	Methods	All Entities (Total)		
		R(%)	P(%)	F(%)
Camera A	L-HMM+POS+Bootstrapping	83.10	80.88	81.98
	L-HMM+POS	77.21	75.43	76.31
	L-HMM	75.78	73.18	74.46
	Baseline	20.43	29.97	24.30
Camera B	L-HMM+POS+Bootstrapping	82.58	77.30	79.85
	L-HMM+POS	78.03	71.84	74.81
	L-HMM	74.81	70.87	72.78
	Baseline	15.53	24.26	18.94
Camera C	L-HMM+POS+Bootstrapping	82.95	82.95	82.95
	L-HMM+POS	80.62	79.60	80.11
	L-HMM	78.23	75.54	76.86
	Baseline	17.05	23.66	19.82

Table 7. Experimental results on opinion sentence identification and opinion orientation classification

Products	Methods	Opinion Sentence Extraction (Sentence Level)			Entity-Opinion Pairs Orientation (Feature Level)		
		R(%)	P(%)	F(%)	R(%)	P(%)	F(%)
Camera A	L-HMM+POS+Bootstrapping	90.72	85.71	88.15	78.98	76.86	77.91
	L-HMM+POS	87.63	83.88	85.71	74.26	72.55	73.40
	L-HMM	86.32	82.11	84.16	73.25	69.89	71.53
	Baseline	51.89	60.64	55.93	19.65	28.82	23.36
Camera B	L-HMM+POS+Bootstrapping	87.95	82.95	85.38	75.00	70.21	72.53
	L-HMM+POS	86.75	81.82	84.21	69.70	65.95	67.77
	L-HMM	85.14	80.29	82.64	68.45	65.02	66.69
	Baseline	46.39	57.04	51.56	13.26	20.71	16.17
Camera C	L-HMM+POS+Bootstrapping	83.91	82.95	83.43	77.52	77.52	77.52
	L-HMM+POS	80.46	80.34	80.40	72.87	72.31	72.59
	L-HMM	79.76	78.82	79.29	72.09	66.91	69.40
	Baseline	43.68	54.29	48.41	17.05	23.66	19.82

7 References

1. Daniel Jurafsky & James H. Martin :Speech and Language Processing. October 2, 2019
2. Wei Jin , Hung Hay Ho : A Novel Lexicalized HMM-based Learning Framework for Web Opinion Mining, 2007
3. Ethem Alpaydin : Introduction to Machine Learning, Second edition, 2010