SeemGo: Conditional Random Fields Labeling and Maximum Entropy Classification for Aspect Based Sentiment Analysis

Pengfei Liu and Helen Meng
Human-Computer Communications Laboratory
Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong, Hong Kong SAR, China
{pfliu,hmmeng}@se.cuhk.edu.hk

Abstract

This paper describes our SeemGo system for the task of *Aspect Based Sentiment Analysis* in SemEval-2014. The subtask of aspect term extraction is cast as a sequence labeling problem modeled with Conditional Random Fields that obtains the F-score of 0.683 for Laptops and 0.791 for Restaurants by exploiting both word-based features and context features. The other three subtasks are solved by the Maximum Entropy model, with the occurrence counts of unigram and bigram words of each sentence as features. The subtask of aspect category detection obtains the best result when applying the Boosting method on the Maximum Entropy model, with the precision of 0.869 for Restaurants. The Maximum Entropy model also shows good performance in the subtasks of both aspect term and aspect category polarity classification.

1 Introduction

In this paper, we present the SeemGo system developed for the task of *Aspect Based Sentiment Analysis* in SemEval-2014. The task consists of four subtasks: (1) aspect term extraction (identify particular aspects of a given entity, e.g., laptop, restaurant, etc.); (2) aspect category detection (detect the category of a given sentence, e.g., food, service for a restaurant, etc.), (3) aspect term polarity, and (4) aspect category polarity. The polarity of each aspect term or aspect category includes positive, negative, neutral or conflict (i.e., both positive and negative).

In the SeemGo system, the subtask of aspect term extraction is implemented with the CRF model that shows good performance by integrating both word-based features and context features. The other subtasks of aspect category detection, aspect term/category polarity classification are all developed with the MaxEnt model with the occurrence counts of unigram and bigram words of each sentence as features. Experimental results show that MaxEnt obtains good performance in all the three subtasks. For the subtask of aspect category detection, MaxEnt obtains even better performance when combined with the Boosting method.

The rest of this paper is organized as follows: Section 2 discusses related work; Section 3 presents the architecture and the underlying models of the SeemGo system as well as the experimental results. We summarize the paper and propose future work in Section 4.

2 Related Work

The subtask of aspect term extraction is quite similar with Noun Phrase Chunking (NPC) (Sha and Pereira, 2003) and Named Entity Recognition (NER) (Finkel et al., 2005). NPC recognizes noun phrases from sentences, while NER extracts a set of entities such as *Person*, *Place*, and *Organization*. Both NPC and NER are sequential learning problems and they are typically modelled by sequence models such as Hidden Markov Model (HMM) and CRF (Finkel et al., 2005).

For the task of aspect term extraction, some related papers also model it with sequence models. Jin et al. (2009) proposed an HMM-based framework to extract product entities and associated opinion orientations by integrating linguistic features such as part-of-speech tag, lexical patterns and surrounding words/phrases. Choi et al. (2005) proposed a hybrid approach using both CRF and extraction patterns to identify sources of opinions in text. Jakob and Gurevych (2010) described a

CRF-based approach for the opinion target extraction problem in both single- and cross-domain settings. Shariaty and Moghaddam (2011) used CRF for the task of identifying aspects, aspect usages and opinions in review sentences by making use of labeled dataset on aspects, opinions as well as background words in the sentences.

The task of aspect category detection is essentially a text classification problem, for which many techniques exist. Joachims (1998) explored the use of Support Vector Machines (SVM) for text categorization and obtained good performance due to their ability to generalize well in high-dimensional feature spaces. Nigam et al. (1999) proposed the MaxEnt model for document classification by estimating the conditional distribution of the class variable give the document, and showed that MaxEnt is significantly better that Naive Bayes on some datasets.

For polarity classification, Pang et al. (2002) conducted experiments on movie reviews and showed that standard machine learning techniques (e.g., Naive Bayes, SVM and MaxEnt) outperform human-produced baselines.

3 The SeemGo System

We use the CRF model (Lafferty et al., 2001) for the subtask of aspect term extraction, and adopt the MaxEnt model for the other three subtasks with the vectors of *word count* as features. Each entry in the vector represents the occurrence count of each unigram or bigram words in the sentence. Figure 1 shows the architecture and the MaxEnt and CRF models of the SeemGo system. The *label* is denoted in lowercase (e.g. y for sentiment), while *word count*, *label sequence* and *word sequence* are vectors, denoted in bold lowercase (e.g. $\mathbf{y}$ for label sequence). We developed the SeemGo system in Java based on the MALLET Toolkit (McCallum, 2002) for MaxEnt and the Stanford CRFClassifier(Finkel et al., 2005) for CRF.

3.1 Background

3.1.1 Maximum Entropy Classifier

The MaxEnt model defines the conditional distribution of the class (y) given an observation vector $\mathbf{x}$ as the exponential form in Formula 1:

$$\mathrm{P}(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\exp\left(\sum_{k=1}^{K}\theta_k f_k(\mathbf{x}, y)\right) \quad (1)$$

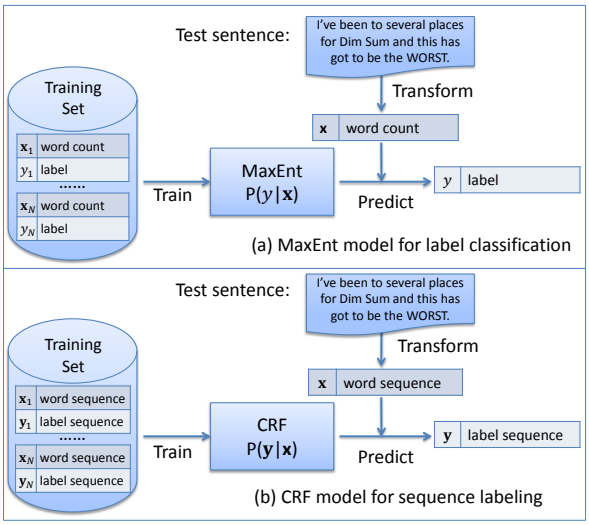

Figure 1: The Architecture, the MaxEnt and CRF Models of the SeemGo System.

where θ_k is a weight parameter to be estimated for the corresponding feature function $f_k(x, y)$, and $Z(\mathbf{x})$ is a normalizing factor over all classes to ensure a proper probability. K is the total number of feature functions.

3.1.2 Conditional Random Fields

CRF is an extension to the MaxEnt model for handling sequence data. The linear-chain CRF is a special case of CRF that obeys the Markov property between its neighbouring labels. Following McCallum and Li (2003), Formula 2 defines the linear-chain CRF: $\mathbf{y} = \{y_t\}_{t=1}^{T}$, $\mathbf{x} = \{x_t\}_{t=1}^{T}$ are label sequence and observation sequence respectively, and there are K arbitrary feature functions $\{f_k\}_{1\leq k\leq K}$ and the corresponding weight parameters $\{\theta_k\}_{1\leq k\leq K}$. $Z(\mathbf{x})$ is a normalizing factor over all label sequences.

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\exp\left(\sum_{t=1}^{T}\sum_{k=1}^{K}\theta_k f_k(y_t, y_{t-1}, \mathbf{x}, t)\right) \quad (2)$$

In the labeling phase, the *Viterbi* decoding algorithm is applied to find the best label sequence $\mathbf{y}*$ for the observation sequence $\mathbf{x}$.

3.2 Subtask 1: Aspect Term Extraction

The datasets (Laptops and Restaurants) are provided in XML format, with each sentence and its annotations consisting of a training instance. For each instance, SeemGo first transform the sentence into a word sequence $\mathbf{x}$, and converts the corresponding annotations into the label sequence $\mathbf{y}$. SeemGo then learns a CRF model $P(\mathbf{y}|\mathbf{x})$ based on the N the training instances $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{N}$.

3.2.1 IOB Labeling

Since an aspect term can contain multiple words (e.g., *hard disk*), we define the label **B-TERM** for the beginning of an aspect term, the label **I-TERM** for the subsequent inside words or end word of an aspect term and the label **O** for all other words. This definition follows the *Inside, Outside, Beginning* (IOB) labeling scheme (Ramshaw and Marcus, 1999). The subtask 1 can be viewed as a sequence labeling problem by labeling each word either as B-TERM, I-TERM or O. Figure 2 shows two example sentences labeled with the IOB2 scheme [1].

I	liked	the	service	and	the	staff.
O	**O**	**O**	**B-TERM**	**O**	**O**	**B-TERM**

The	hard	disk	is	very	noisy.
O	**B-TERM**	**I-TERM**	**O**	**O**	**O**

Figure 2: Example Sentences with IOB2 Labels.

3.2.2 Features for the CRF Model

In CRF, features typically refer to feature functions $\{f_k\}$, which can be arbitrary functions. In text applications, CRF features are typically binary (Sutton and McCallum, 2012). As an example for "virus protection", a binary feature function may have value 1 if and only if the label for "virus" is *B-TERM* and the current word "protection" has the suffix of "tion", and otherwise 0. Similar to the features used in Finkel et al. (2005) for the NER task, Table 1 summarizes the features for the aspect term extraction task. We call the features derived from the current word *word-based* features such as $w_{id}, w_{character}$, and the features from the surrounding words and the previous label the *contex* features ($context$).

We consider the sentence *"I've been to several places for Dim Sum and this has got to be the WORST."* as an example to explain why we choose these features: (a) word-based features: the word "Sum" is located in the middle of the sentence, with the first character capitalized. (b) context features: the previous word "Dim" is also capitalized in the first character and the label of "Dim" is assumed to be "B-TERM". By combining the word-based features and the context features, the Viterbi decoding algorithm will then label "Sum" as "I-TERM" with high degree of confidence, which is a part of the multi-word term "Dim Sum", instead of a mathematical function in some other context.

[1] With IOB2, every aspect term begins with the B label.

Table 1: Features for the CRF Model.

Feature	Description
w_{id}	word identity
$w_{character}$	whether the word characters are capitalized, hyphenated, numeric, e.g., built-in camera, BIOS, Dim Sum, Windows 7
$w_{location}$	word index in the word sequence $\mathbf{x}$
w_{ngram}	n-gram character sequences of each word with maximum length of 6, including prefixes and suffixes, e.g., **"tion"** in specification, navigation
$context$	current word w_t, its neighbouring words (w_{t-2},...,w_{t+2}) and previous label y_{t-1}
w_{pos}	part-of-speech tag of each word

3.2.3 Experimental Results

We trained the CRF model with different feature set on the training set provided by the SemEval2014 organizers, and reported the experimental results on the testing set by the evaluation tool *eval.jar*. The detailed experimental results are listed in Table 2. The *basic* feature set consists of w_{id}, $w_{character}$ and $w_{location}$. The results from one of the best systems on each dataset are also listed, marked with the star (*).

Table 2: Experimental Results on Different Feature Set for Aspect Term Extraction.

	Feature Set	Precision	Recall	F-score
Lap	$basic$	0.780 (263/337)	0.402 (263/654)	0.531
	$basic + w_{ngram}$	0.781 (375/480)	0.573 (375/654)	0.661 **(+0.13)**
	$basic + w_{context}$	0.827 (296/358)	0.453 (296/654)	0.585 **(+0.054)**
	$basic$+w_{ngram}+ $context$	0.830 (380/458)	**0.581 (380/654)**	**0.683 (+0.152)**
	$basic$+w_{ngram}+ $context + w_{pos}$	**0.837 (365/436)**	0.558 (365/654)	0.670 **(-0.013)**
	IHS_RD_Belarus*	0.848	0.665	0.746
Res	$basic$	0.862 (692/803)	0.610 (692/1134)	0.715
	$basic + w_{ngram}$	0.838 (804/959)	0.709 (804/1134)	0.768 **(+0.053)**
	$basic + w_{context}$	0.856 (704/822)	0.621 (704/1134)	0.720 **(+0.05)**
	$basic$+w_{ngram}+ $context$	0.865 (827/956)	**0.729 (827/1134)**	**0.791 (+0.076)**
	$basic$+w_{ngram}+ $context + w_{pos}$	**0.870 (806/926)**	0.711 (806/1134)	0.783 **(-0.08)**
	XRCE*	0.909	0.818	0.840

We have the following observations:

(1) Compared with using only the *basic* features, adding the feature of w_{n-gram} contributes the

greatest performance improvement, with the absolute increase of F-score by 13% for Laptops and 5.3% for Restaurants; while adding the $w_{context}$ feature *improves* the F-score by around 5% for both datasets.

(2) Combining the word-based features ($basic$ and w_{ngram}) and the context-based features ($w_{context}$) lead to the best performance for both datasets in terms of recall and F-score.

(3) The POS tags lead to a *decrease* in both recall and F-score, with the absolute decrease of F-score by 1.3% for Laptops and 8% for Restaurants. The same observation is also reported by Tkachenko and Simanovsky (2012) for NER.

3.3 Subtask 3: Aspect Category Detection

We encode each sentence as a feature vector $\mathbf{x}$ with each entry representing occurrence count of each unigram word and bigram words (i.e., *word count*). All words are lowercased, while keeping the stopwords as most sentences in the datasets are short. Using the provided training set, We trained a MaxEnt classifier (ME) $P(y|\mathbf{x})$ with a Gaussian prior variance of 20 to prevent overfitting.

We also tried the Bagging (Breiman, 1996) on MaxEnt (BaggingME) and the Boosting (Freund et al., 1996) on MaxEnt (BoostME). Table 3 shows the experimental results on the provided testing set. It shows that the Boosting method on MaxEnt improves both precision and recall as well as the F-score by 1.1%. The best evaluation result is by the NRC-Canada team.

Table 3: Performance of Different Classifiers for Aspect Category Detection.

Classifier	Precision	Recall	F-score
ME	0.858 (686/800)	0.669 (686/1025)	0.752
BagME	0.843 (674/800)	0.658 (674/1025)	0.739
BoostME	**0.869 (695/800)**	**0.678 (695/1025)**	**0.762**
Best*	0.910	0.862	0.886

3.4 Subtask 2 & 4: Aspect Term & Category Polarity Classification

Similar to subtask-3, we also used MaxEnt for the subtasks of 2 and 4, with word count as features. For category polarity classification, we count the words from both the sentence and the category name. For example, we count the sentence "The Dim Sum is delicious." and its category "Food" as features. This improves performance compared with counting the sentence only.

Table 4 shows the accuracy of each classifier for the subtasks of 2 and 4 on Laptops and Restaurants, including the best results from NRC-Canada (a) and DCU (b). In both datasets, the distributions of aspect term/category polarities are very imbalanced with very few sentences on *conflict* but with most sentences on *positive* or *negative*. This leads to very low classification performance for the *conflict* class, with the F-score less than 0.2. In this case, the Boosting method does not necessarily improve the performance.

Table 4: Accuracy of Different Classifiers for Aspect Term & Category Polarity Classification.

Classifier	Term		Category
	Laptops	Restaurants	(Restaurants)
ME	**0.648 (424/654)**	0.729 (827/1134)	**0.752 (771/1025)**
BagME	0.635 (415/654)	**0.732 (830/1134)**	**0.752 (771/1025)**
BoostME	0.642 (420/654)	0.730 (828/1134)	0.747 (766/1025)
Best*	0.705 (a,b) (461/654)	0.810 (b) (918/1134)	0.829 (a) (850/1025)

3.5 Evaluation Ranks

Table 5 shows the official ranks (and the new ranks in braces of the revised version after evaluation) of the SeemGo system on the two datasets. The evaluation metrics are Precision, Recall and F-score for the subtasks of 1 and 3, and Accuracy (Acc) for the subtasks of 2 and 4.

Table 5: Ranks of SeemGo on the Constrained Run (Using only the Provided Datasets).

	Subtask	Precision	Recall	F-score	Acc
Lap	1	4	12 (8)	8 (4)	-
	2	-	-	-	12 (6)
Res	1	3	11 (7)	5	-
	2	-	-	-	8 (6)
	3	3 (2)	12	8 (7)	-
	4	-	-	-	4

4 Conclusions

This paper presents the architecture, the CRF and MaxEnt models of our SeemGo system for the task of *Aspect Based Sentiment Analysis* in

SemEval-2014. For the subtask of aspect term extraction, CRF is trained with both the word-based features and the context features. For the other three subtasks, MaxEnt is trained with the features of the occurrence counts of unigram and bigram words in the sentence. The subtask of aspect category detection obtains the best performance when applying the Boosting method on MaxEnt. MaxEnt also shows good average accuracy for polarity classification, but obtains low performance for the *conflict* class due to very few training sentences.This leaves us the future work to improve classification performance for imbalanced datasets (He and Garcia, 2009).

Acknowledgements

We thank the organizers for their hard work in organizing this evaluation, and the two anonymous reviewers for their helpful comments.

References

Leo Breiman. 1996. Bagging predictors. *Machine learning*, 24(2):123–140.

Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 355–362.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370.

Yoav Freund, Robert E Schapire, et al. 1996. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, volume 96, pages 148–156.

Haibo He and Edwardo A Garcia. 2009. Learning from imbalanced data. *Knowledge and Data Engineering, IEEE Transactions on*, 21(9):1263–1284.

Niklas Jakob and Iryna Gurevych. 2010. Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1035–1045.

Wei Jin, Hung Hay Ho, and Rohini K Srihari. 2009. A novel lexicalized HMM-based learning framework for web opinion mining. In *Proceedings of the International Conference on Machine Learning*, pages 465–472. Citeseer.

Thorsten Joachims. 1998. *Text categorization with support vector machines: Learning with many relevant features*. Springer.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 188–191.

Andrew Kachites McCallum. 2002. MALLET: A Machine Learning for Language Toolkit.

Kamal Nigam, John Lafferty, and Andrew McCallum. 1999. Using maximum entropy for text classification. In *IJCAI-99 workshop on machine learning for information filtering*, volume 1, pages 61–67.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 79–86.

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 134–141.

Shabnam Shariaty and Samaneh Moghaddam. 2011. Fine-grained opinion mining using conditional random fields. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, pages 109–114. IEEE.

Charles Sutton and Andrew McCallum. 2012. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373.

Maksim Tkachenko and Andrey Simanovsky. 2012. Named entity recognition: Exploring features. In *Proceedings of KONVENS*, volume 2012, pages 118–127.