Incorporating Conditional Random Fields and Active Learning to Improve Sentiment Identification

Kunpeng Zhang[12] Yusheng Xie[13] Yi Yang[3] Aaron Sun[4] Hengchang Liu[5]
Alok Choudhary[3]
[1]: Authors contributed equally
[2]: kzhang6@uic.edu
[3]: {yxi389,yya518,choudhar}@eecs.northwestern.edu
[4]: a.sun@samsung.com
[5]: hcliu@ustc.edu.cn

[2]: *University of Illinois at Chicago, Chicago, IL USA*
[3]: *Northwestern University, Evanston, IL USA*
[4]: *Cloud Research Lab, Samsung Research America, San Jose, CA USA*
[5]: *University of Science and Technology of China, Hefei, China*

Abstract

Many machine learning, statistical, and computational linguistic methods have been developed to identify sentiment of sentences in documents, yielding promising results. However, most of state-of-the-art methods focus on individual sentences and ignore the impact of context on the meaning of a sentence. In this paper, we propose a method based on conditional random fields to incorporate sentence structure and context information in addition to syntactic information for improving sentiment identification. We also investigate how human interaction affects the accuracy of sentiment labeling using limited training data. We propose and evaluate two different active learning strategies for labeling sentiment data. Our

experiments with the proposed approach demonstrate a 5-15% improvement in accuracy on Amazon customer reviews compared to existing supervised learning and rule-based methods.
Keywords: conditional random fields, active learning, customer reviews, sentiment analysis

---

1. Introduction

The rapid proliferation of Internet connectivity has led to increasingly large volumes of electronic commerce, resulting in a huge amount of social media data in various forms such as online customer reviews, blog articles, social network comments, microblog messages (e.g., tweets in Twitter). Analyzing and mining useful information from these data using computational techniques, including social network analysis and web information retrieval, has become an important task. With the current trend, more and more people express their opinions publicly via social media platforms. According to two surveys conducted on more than $2,000$ American adults (Pang & Lee, 2008), we note that:

1. 81% of Internet users (60% of American users) have researched a consumer product online for at least once;
2. Among readers of online reviews of restaurants, hotels, and other services (including travel agencies or doctors), between 73% and 87% participants report that previous reviews had a significant influence on their decision to purchase;

3. 32% have provided a rating on a product or service via an online ratings system; and
4. 30% have posted an online comment or review regarding a product or service, including 18% of online senior citizens.

Understanding the sentiment of sentences allows us to summarize online opinions which could help people make informed decisions. Automated sentiment identification has seen huge research efforts for many years and has achieved some promising results. On one hand, different machine learning techniques, statistical learning methods, and computational linguistic methods have been developed to recognize sentiments (Xie et al., 2013) (Hu & Liu, 2004). On the other hand, some researchers have also proposed rule-based (and unsupervised) methods to improve sentiment classification (Cambria et al., 2013) and (Hu et al., 2013). All of the state-of-the-art algorithms perform well on individual sentences without considering any context information, but their accuracy is dramatically lower on the document level because they fail to consider context. New algorithms are needed to analyze sentiment in longer documents.

There are many difficulties owing to the special characteristics and diversity in sentence structure, in which people express their opinions. For example, one sentence may express multiple sentiments though the speaker may emphasize one part, as in "*The color of this camera is pretty good, but it is too expensive comparing to similar products from other manufactures.*"

Also, sarcastic sentences express opinions differently from what texts would suggest in literal, and many sentences express their author's opinions indirectly through comparison. For example, the sentence "*In terms of customer service, Nikon wins over Canon, hands down.*" expresses the reviewer's preference over Nikon cameras, which can be positive or negative depending on whether Nikon or Canon is the main subject of the document to which the sentence belongs. Such sentences explicitly show positive or negative sentiments, but their implicit sentiments are different if they are placed into a particular context. Some typical representatives are sarcastic sentences. Capturing relationships among such sentences in a document is therefore a particular challenge.

In addition, complicated sentence structure and Internet slang make sentiment analysis even more challenging. In this paper, we not only consider syntax that may influence the sentiment, including newly emerged Internet language, emoticons, positive words, negative words, and negation words, but we also incorporate information about sentence structure, like conjunction words and comparisons. The context around a sentence plays an important role in determining the sentiment; e.g., a compound sentence is more likely to be positive if both sentences before and after are positive. Therefore, we employ a conditional random fields (CRF) method to capture syntactic, structural, and contextual features of sentences. Our experimental results on Amazon customer reviews and Facebook comments show improved accuracy compared to supervised and rule-based methods.

Furthermore, labeling sentiments manually is expensive. Often a large number of labels are necessary when training a probabilistic sentiment model with realistic complexity. Therefore, we apply active learning to tag sequences of unlabeled sentences that are most informationally valuable to the model. We propose two different strategies to select "label-worthy" data with high uncertainty for human beings to label, and our experimental results on customer reviews demonstrate faster convergence compared to baselines. This active learning strategy is especially useful when human effort, compared to data availability (i.e., big data), becomes a scarce resource.

2. Literature

Recently, there has been a lot of research on sentiment analysis using techniques ranging from rule-based, bag-of-words approaches to machine learning techniques. The analyzed subjects range from long documents to short sentences.

2.1. Classifying Document Sentiment

Document sentiment classification is the analysis/classification of text sentiment on a multi-sentence document (e.g., product reviews or blog articles) as positive or negative (Pang et al., 2002) (Turney, 2002) (Mcdonald et al., 2007). In (Choi & Cardie, 2008), the authors present a

novel approach based on compositional semantics that incorporates sentence structure into the learning procedure. They also find that "content-word negators" play an important role in determining expression-level polarity. Further in (Xie et al., 2013), the authors expand the rules and consider the specialty of social media data to improve sentiment classification. Machine learning has also been widely used to identify sentiments of sentences. In (Pang et al., 2002), the authors employ machine learning techniques to classify movie reviews by overall sentiment. Their results show that three machine learning methods (Naïve Bayes, maximum entropy classification, and support vector machine) do not perform as well on sentiment classification as on traditional topic-based categorization. However, an important aspect in document sentiment is the consideration of inter-sentence dynamics, which has not yet been systematically handled in previous works other than several specific rules proposed in (Xie et al., 2013) and (Choi & Cardie, 2008).

2.2. Classifying Sentence Sentiment

Another important research direction is classifying sentences as positive subjective, negative subjective, or objective (Wiebe et al., 1999) (Wiebe & Wilson, 2002) (Yu & Hatzivassiloglou, 2003) (Wilson et al., 2004) (Kim & Hovy, 2004) (Riloff & Wiebe, 2003). In (Narayanan et al., 2009), the authors present linguistic analysis of conditional sentences, and build some supervised learning models to determine if sentiments expressed on different

topics in a conditional sentence are positive, negative or neutral. The more general problem of rating inference, where one must determine the authors' evaluation with respect to a multi-point scale (e.g., one to five "stars" for a review) can be viewed simply as a multi-class text categorization problem. Predicting degree of positivity provides more fine-grained rating information. At the same time, it is an interesting learning problem in itself.

There have been studies on building sentiment lexicons to define the strength of word sentiment, which is largely the foundation for accurate sentence sentiment. Esuli & Sebastiani (2006) constructed a lexical resource, SentiWordNet, a WordNet-like lexicon emphasizing sentiment orientation of words and providing numerical scores of how objective, positive, and negative these words are. However, lexicon-based methods can be tedious and inefficient and may not be accurate due to the complex cross-referencing in dictionaries like WordNet. The sentiment scoring approach in (Liu & Seneff, 2009) makes use of collective data such as user star ratings in reviews. By associating user star ratings and frequency with each phrase extracted from review texts, they can easily associate numeric scores with textual sentiment. They propose an approach for extracting *adverb-adjective-noun* phrases based on clause structure obtained by parsing sentences into a hierarchical representation. They also propose a robust general solution for modeling the contribution of adverbials and negation to the score for degree of sentiment.

2.3. Applications of Sentiment Analysis

Several researchers have also focused on the applications of sentiment analysis, for example, feature/topic-based sentiment analysis (Popescu & Etzioni, 2005) (Mei et al., 2007) (Ku et al., 2006) (Ding et al., 2008) (Stoyanov & Cardie, 2008). Their objective is to extract topics or product features in sentences and determine whether the sentiments expressed on them are positive or negative. In (Hu & Liu, 2004), the authors aim to summarize all customer reviews of a product by mining the features of the product on which customers have expressed their opinions and whether the opinions are positive or negative. In (Zhang & Liu, 2011), the authors use feature-based opinion mining model to identify noun product features that imply opinions. It is mainly focusing on the problem of objective nouns and sentences with implied opinions. Lastly, Cambira & White (2014) provides a compact review and outlook in the Natural Language Processing field, which sentiment analysis is founded upon.

2.4. Our Contribution

Compared to existing algorithms of sentiment analysis both on sentence and document levels, our proposed algorithm attempts to improve sentiment classification in the following ways:

1. Taking full advantage of the sentence structure;

2. Using context information to capture the relationship among sentences and to improve document-level sentiment classification;
3. Accounting for Internet language word set and emoticons; and
4. Incorporating human interaction to improve sentiment identification accuracy and construct a large training dataset.

3. Methodology

In this section, we discuss the Conditional Random Fields (CRF) based model and its corresponding features for identifying sentence sentiments. CRF are a class of discriminative undirected probabilistic graphical model generally applied in pattern recognition and machine learning, where they are specified designed to optimize structured prediction. A "generic" classifier predicts a label for a single sample without regard to "neighboring/connected" samples, a (linear-chain) CRF can take context into account. For this reason, CRF is popular in natural language processing where a document could be regarded as a sequence of sentences. And the sentences can be split into a couple of segments, where each segment is relatively linked. The features we extract are based on two perspectives. Semantic characteristics (including sentiment words, emoticons, etc.) of a sentence will definitely help in determining its sentiment. Negation words, conjunction words, and other syntactic features also significantly influence the sentiment. Further, the structure of a

sentence would also affect its sentiment to some extent.

3.1. Problem Definition

The input of the algorithm includes specified subjects $SUB := \{sub_1, sub_2, \ldots, sub_m\}$ and a set of corresponding documents $D := \{d_1, d_2, \ldots, d_m\}$, where m is the number of documents. Note that SUB is an essential part of input because different subjectivity can generate different and reversed sentiments for sentences. For example, there is a online customer review sentence for a Canon digital camera, but it gives positive opinions on Nikon's digital camera. In this case, we should label this sentence as positive if the subject is Nikon, but negative or objective for Canon.

Each document $d_i \in D$ further contains multiple sentences $S_i := \{s_1^i, s_2^i, \ldots, s_{n_i}^i\}$, where $n_i \geq 1$ is the number of sentences in document d_i. In this paper, we consider the sentiment recognition as a discrete classification problem. Therefore, the algorithm will give the output for all documents in the following format. For the jth sentence in the ith document, s_j^i, the model will assign a sentiment value $o_j^i \in \{P, N, O\}$, where P represents positive, N represents negative, and O represents neutral.

3.2. The Conditional Random Fields Model

We want to capture the context information (e.g., neighboring sentences or sentences connected by transition words) among sentences in a

document. The procedure of sentiment identification therefore becomes a kind of sequence labeling. The goal of the model is to give a label to each sentence corresponding to the sentence sequence. In this paper, we use CRF as a tool to model this sequence labeling problem.

Conditional Random Fields (CRF) provide a probabilistic framework for calculating the probability of Y globally conditioned on X (Lafferty et al., 2001), where X is a random variable/vector over sequence data to be labeled, and Y is a random variable/vector over corresponding label sequences. X and Y could have a natural and/or complicated graph structure. Two commonly used structures are linear-chain and skip-chain structures. In this paper, we use linear chain structure because it is more suitable for sequence tagging. Because we only consider the sentiment interaction among neighboring sentences rather than sentences with long distances. In addition, it has been widely used in text labeling domain(McCallum, 2003) (Zhang et al., 2012). Linear chain structure is popular because it is CRF model in the simplest form. CRF examples given in seminal works such as (Lafferty et al., 2001) (McCallum, 2003) assumed linear chain structures as well. A further observation is that there is a one-to-one correspondence between states and labels. Figure 1 gives a simple visualization of how a CRF model looks like and how it differs from similar Hidden Markov Models.

Figure 1 should be inserted here.

Given an observation sequence (i.e., a document containing multiple sentences) $X := (x_1, x_2, \ldots, x_m)$ and the corresponding label sequence (each sentence is tagged as a label) $Y := (y_1, y_2, \ldots, y_m)$, the probability of Y conditioned on X defined in CRF, $\Pr(Y|X)$, is expressed as follows:

$$\begin{aligned} \Pr(Y|X) &= \frac{1}{Z_X} \exp\left(\sum_{j=1}^{K\times L} F_j(Y, X)\right) \\ &= \frac{1}{Z_X} \exp\left(\sum_{i=1,k=1}^{m,K} \lambda_k f_k(y_{i-1}, y_i, X) + \sum_{i=1,l=1}^{m,L} \mu_l g_l(y_i, X)\right), \end{aligned} \tag{1}$$

where Z_X is the normalization constant that makes the probability of all state sequences sum to one. Equation 1 contains two types of feature indicator functions:

1. $f_k(y_{i-1}, y_i, X)$ is an arbitrary feature function over the entire observation sequence X and state positions i and $i-1$;
2. $g_l(y_i, X)$ is a feature function of state at position i and the observation sequence X.

λ_k and μ_l are positive weights learned (from training data) for feature functions f_k and g_l, reflecting the model's confidence of the corresponding f_k and g_l. These feature functions can describe any aspect of a transition from y_{i-1} to y_i as well as y_i and the global characteristics of X. To give an concrete example, f_k may be conceivably evaluated to:

1. 1 when y_{i-1} has label P;

2. 1 when y_i has label N;
3. 1 when x_{i-1} contains positive emoticons;
4. 1 when x_i contains conjunction words in the beginning of its sentence;
5. 0 elsewhere.

On the other hand, g_l may be conceivably evaluated to:

1. 1 when y_i has label P;
2. 1 when x_i contains positive adjectives and no negation words;
3. 0 elsewhere.

3.3. Parameter Estimation

The goal of parameter estimation is to learn the set of weights/parameters in a CRF model. Let $\Theta := \{\lambda_k, \mu_l | 1 \leq k \leq K; 1 \leq l \leq L\}$ be the parameter for our CRF model. It is commonly seen that Θ is estimated by the principle of maximum likelihood, that is, by maximizing the conditional log-likelihood function of the labeled sequences in the training data $D := (\mathbf{X}, \mathbf{Y}) = \{(X^{(1)}, Y^{(1)}), \ldots, (X^{(M)}, Y^{(M)})\}$, where M is the number of

training samples. The log-likelihood function then is defined as follows.

$$\begin{aligned}\mathcal{L}(\Theta) &= \log \Pr(\mathbf{Y}|\mathbf{X};\Theta) \\ &= \log \prod_{j\in\{1,\ldots,M\}} \Pr(Y^{(j)}|X^{(j)};\Theta) \\ &= \sum_{j\in\{1,\ldots,M\}} \log \Pr(Y^{(j)}|X^{(j)};\Theta).\end{aligned} \tag{2}$$

To avoid over-fitting in the training process, regularization methods (Peng & McCallum, 2006) are often added to $\mathcal{L}(\Theta)$ from Equation 2. A very common way is to add a Gaussian prior over the parameters.

$$\begin{aligned}\mathcal{L}'(\Theta) = &\sum_{j\in\{1,\ldots,M\}} \log \Pr(Y^{(j)}|X^{(j)};\Theta) \\ &- \sum_k \frac{\lambda_k^2}{2\sigma_k^2} - \sum_l \frac{\mu_l^2}{2\sigma_l^2},\end{aligned} \tag{3}$$

where σ_k^2 and σ_l^2 are the variances of the Gaussian priors (typically set to 1).

The last step in maximum likelihood estimation is to differentiate the regularized log-likelihood function with respect to each parameter λ_k or μ_l.

$$\begin{aligned}\frac{\partial \mathcal{L}'(\Theta)}{\partial \lambda_k} =&\mathbf{E}_{\tilde{p}(Y,X)}\left[F_k(Y,X)\right] \\ &- \sum_j \mathbf{E}_{Y|X^{(j)};\lambda}\left[F_k(Y,X^{(j)})\right],\end{aligned} \tag{4}$$

where $\tilde{p}(Y,X)$ is the empirical distribution of training data. Note that setting this derivative to zero yields the maximum entropy model constraint

for parameter λ_k. The expectation of each feature with respect to the model distribution is equal to the expected value under the empirical distribution of the training data. Many Iterative Scaling algorithms (e.g., Generalized Iterative Scaling (GIS) and Improved Iterative Scaling (IIS) (Lafferty et al., 2001)) can be used to optimize $\mathcal{L}'(\Theta)$. In addition, some stochastic gradient methods can also be used to optimize parameters (Sha & Pereira, 2003). In this paper, we use a quasi-Newton method called limited memory Broyden–Fletcher–Goldfarb–Shanno algorithm (L-BFGS) (Liu & Nocedal, 1989) which converges significantly faster than the original BFGS algorithm.

3.4. Inference

Given the conditional probability of the state sequence defined by a CRF model in Equation 1 and the estimated parameters $\hat{\Theta}$, the label prediction of a sequence is obtained as follows:

$$Y^* = \arg\max_Y \Pr(Y|X;\hat{\Theta}). \tag{5}$$

Equation 5 can be efficiently solved by the Viterbi algorithm (Rabiner, 1990). The marginal probability of states at each position in the sequence can be calculated by a dynamic programming procedure similar to the forward-backward procedure for Hidden Markov Model (HMM) (Rabiner, 1990).

We can compute the forward variables $\alpha_i(y|X)$ by the following two steps.

1. Setting $\alpha_1(y|X)$ equal to the probability of starting with state y.
2. For $i > 1$, use Equation 6.

$$\begin{aligned}\alpha_{i+1}(y|X) &= \sum_{y'} \alpha_i(y'|X) \exp\left(F_{i+1}(y', y, X)\right) \\ &= \sum_{y'} \alpha_i(y'|X) \sum_k \lambda_k f_k(y_i = y', y_{i+1} = y, X) \\ &\quad + \sum_{y'} \alpha_i(y'|X) \sum_l \mu_l g_l(y_{i+1} = y, X).\end{aligned} \tag{6}$$

Then the normalization factors in Equation 1 can be obtained as $Z_X = \sum_y \sum_j \alpha_j(y|X)$. With Z_X expressed, we calculate the marginal probability of each sentence being a *positive* sentence given the entire sentence sequence.

$$\Pr(y_i = \text{“}P\text{”}|X) = \frac{\alpha_i(\text{“}P\text{”}|X) \cdot \beta_i(\text{“}P\text{”}|X)}{Z_X}, \tag{7}$$

where $\beta_i(y|X)$ are the backward values and are similarly defined as $\alpha_i(y|X)$. $\alpha_i(y|X)$'s are values in the forward sequence. $\alpha_i(y|X)$ encodes the probability of being in the current state y given all observations from observation 1 to observation $i-1$. Similarly, $\beta_i(y|X)$ encodes the probability of being in the current state y given observations from $i+1$ to the end of the sequence/sentence. Finally, $\Pr(y_i|X)$ calculated in Equation 7 is the smoothed probability of $\alpha_i(y|X)$ and $\beta_i(y|X)$. In chapter 15 of

(Russell & Norvig, 2003), the authors explain an elaborated example of forward and backward variables and how they should be efficiently calculated.

After constructing the model, choosing the right set of features is critical in getting the most of our CRF model. In this paper, we use features based on two aspects: *semantic* and *syntactic* structure of sentences. Following two subsections are dedicated to describing the two sets of features.

3.5. Semantic Features

Number of Positive/Negative Words. Intuitively, a sentence containing more positive/negative words is more likely to be positive/negative. We use two lists of English sentiment words that contain 1,948 positive words and 4,550 negative words (Wiebe et al., 2005).

Containing Any Positive/Negative Emoticons. With the rapid development of the Internet and Web 2.0, a huge amount of Internet words have become more common. For example, "=)", ": D", and "v_v" represent smiling, laughing, and sadness, respectively. We manually collected 52 positive emoticons and 35 negative emoticons.

Comparative Sentences. People like delivering their opinions by comparison with other similar objects. For example, when a user describes his/her experience with iPhone, it is often done through comparison with Samsung

Android phones (which, say, belong to his/her friends.) A comparative sentence may not be an objective sentence even if it does not have any sentiment words or emoticons. To detect comparative sentences, we use part-of- speech (POS) to tag each word in a sentence. A sentence is comparative if it contains

- comparative adjectives (JJR),
- comparative adverbs (RBR),
- superlative adjectives (JJS),
- superlative adverbs (RBS),
- indicator keywords (126 in total, collected manually, including "compare/comparing to," "in contrast", etc.),
- or some predefined structural patterns ("as <adj./adv.> as", "the same as", "similar to", etc.).

Type of Conjunction Words. Conjunction words are typically used to join different parts of a sentence and can have a significant influence on its sentiment. For example, sentences with "but", or "however" are likely to have different sentiment from previous sentence. However, sentences connected via "and", "so that", "before", or "after" usually have similar sentiments. We distinguish three types of conjunction words: subordinating, coordinating, and correlative.

3.6. Syntactic Features

Sentence Position. Intuitively, sentences at the beginning or in the end of a document are likely to be summary sentences. They probably affect sentiments of other sentences in other places. We consider three different values for sentence position: beginning, middle, and end. If the sentence is within first 20% of the sentences or a document, we consider it as a beginning sentence, an end sentence if it is within the last 20%, and middle for all others.

Simple or Compound Sentence. A compound sentence often includes more than one opinion through conjunction words connecting different parts.

Position of Positive/Negative Words. If no positive or negative words occur, this feature value is 0; the value is 1 if they only exist in the first part of a sentence given it is a compound sentence; the value is 2 if they are only in the second part; and the value is -1 if they occur in the first part and other parts. The following example (part of a customer review extracted from Amazon.com) illustrates the importance of this feature.

- "I like the color, but the speed is bad", – negative;
- "I like the color, and the speed is also good" , – more positive;
- "The speed is bad, but I like the color'", – positive.

Position of Negation Words. Negation words could reverse the sentiment. We manually collected 32 negation words, e.g., "not", "never", and "couldn't." The position of the negation word is very sensitive to sentiments. If the negation is very close (sometimes next to) to a sentiment word, it is more likely to reverse the sentiment of that sentiment word. For example,

- "No other camera is better than this one!", – positive (negative for "other camera");
- "This camera is not good!", – negative.

Comparison Subject. The subject of a comparison can change the sentiment of a sentence. For example, a user posted the comment "I like Pepsi more" on the official Coca-cola Facebook page. This sentence would be positive if the subject were Pepsi, but in context, it is not a positive sentence. In this paper, we use the Stanford parser(Levy & Manning, 2004) to get the typed dependencies. Then we could know which subject(typically noun) performs some actions(typically verb) or has some characteristics(typically adjective). In the Stanford typed dependencies manual[1], the label with "nsubj" tells this dependency. For the example above, the subject is "I" instead of "Pepsi" from the following typed dependencies:

- nsubj("like"-2, "I"-1)

[1]http://nlp.stanford.edu/software/dependencies_manual.pdf

- root("ROOT"-0, "like"-2)
- dobj("like"-2, "Pepsi"-3)
- advmod("like"-2, "more"-4)

Similarity to Neighboring Sentences. We define features to represent the similarity between a sentence and its neighboring sentences. In this paper, we only consider the sentences immediately proceeding (sim_to_pre) and following (sim_to_next) the sentence, and we calculate two similarity scores, using cosine similarity and latent semantic indexing (LSI) (Deerwester et al., 1990), a widely used dimension reduction method using singular value decomposition (SVD). Cosine similarity captures word-level similarity, and LSI measures semantic similarity.

4. Experiments and Results

In this section, we describe our experiments on two types of data: online customer reviews on Amazon.com and comments on Facebook pages. Customer reviews are longer and more complex than Facebook comments, but Facebook comments contain more Internet language. The training data is collected and labeled in two different ways: Amazon Mechanical Turk and manual labeling. We also describe our experimental setup and results for our CRF-based technique as well as active learning.

4.1. Data Collection and Preprocessing

We downloaded 300 digital camera reviews and 300 TV reviews from Amazon.com. We used MAXTERMINATOR (Reynar & Ratnaparkhi, 1997) to split the reviews into sentences, yielding 5,156 and 5,036 sentences, respectively. Table 1 shows the data distribution. For each of these reviews, we asked ten different workers from Amazon Mechanical Turk to label the sentences as positive, negative, or objective. After collecting the results, we used majority vote to determine the final label for the sentence. We also randomly selected 500 sentences from each of the camera and TV reviews and checked the labeling accuracy. The average response accuracy for all workers for the camera and TV reviews was 0.66 and 0.62 respectively.

Table 1 should be inserted here.

Table 2 should be inserted here.

Table 3 should be inserted here.

We also used Facebook graph API to download 500 comments (from 13 different walls), which we labeled manually. Most of the comments consist of one sentence, with 723 sentences in total. We preprocessed the original data, including word correction (e.g., changing "luv" to "love") and part-of-speech (POS) tagging, which we performed using CRFTagger (Phan, 2006), a Java-based conditional random field POS tagger for English.

4.2. Compared Methods

We compared our proposed method against the following rule-based algorithms and supervised classifiers:

- CSR - Compositional semantic rules (Choi & Cardie, 2008).
- SVM - Support Vector Machine
- LR - Logistic Regression
- HMM - Hidden Markov Models

SVM, LR, HMM, and our CRF method are given the same set of semantic/syntactic features to work with. CSR is a rule-based algorithm.

4.3. Performance Analysis

The first experiment compares the accuracy of our CRF-based model with four other methods on datasets with only semantic features and with all of the features discussed in the two feature sections. Table 2 shows that CRF outperform the other four methods in all cases on the Amazon review dataset. Using our CRF-based method with semantic and syntactic features is 5 – 15% more accurate than the other methods tested. However, as can be seen in Table 3, CSR performs the best on the Facebook comments dataset, while all other methods generated similar results. We believe that this result is due to the length of the Facebook comments, which provide little to no context for our CRF-based method, as well as the

use of emoticons, which convey sentiments directly. All accuracies in Table 2 and Table 3 are calculated based on 10-fold cross validation performance. The average result from 10-fold cross validation is recorded only if the 10 results have a standard deviation less than 0.2 and their pairwise t-test yields p-value less than 0.05.

4.4. Active Learning Framework

Figure 2 should be inserted here.

Since collecting labeled data is expensive, we want to use active learning to build classifiers using less training data for expanding labeled data pool. The basic algorithm is as follows:

1. Use some labeled data to train a model;
2. Apply the trained model to unlabeled data;
3. Pick the data with highest uncertainty and present it to an oracle;
4. Add the newly labeled data into training pool;
5. Iterate steps 1–4 until the accuracy converges.

The fundamental step of active learning procedure is the strategy for choosing what data to present to the oracle (usually a human being). Since our problem is a sequence labeling problem, we propose two different strategies to choose data to be labeled by the oracle. When we apply our trained model to the inference of unlabeled data, we get a probability for each sentence label: $P_d := \{p_1, p_2, \ldots, p_m\}$, where m is the number of

sentences in the document d. In Strategy 1 (S1), we rank the documents based the following score for each document d:

$$\text{score}_{S1}(d) = \frac{1}{m} \sum_{p \in P_d} p. \tag{8}$$

After the documents are ranked, S1 picks the document with the smallest value to present to the oracle.

In Strategy 2 (S2), we rank sentences based on the probability in an ascending order and, for each document, only keep first 50% of the sentences. In other words, the elements in P_d are sorted in ascending order; then we consider $P'_d := \{p_1, p_2, \ldots, p_{\lceil m/2 \rceil}\}$ and rank the documents based the following score for each document d:

$$\text{score}_{S2}(d) = \frac{1}{m} \sum_{p \in P'_d} p. \tag{9}$$

After the documents are ranked, S2 picks the document with the smallest value to present to the oracle.

To evaluate these two strategies, we start from a training size of ten documents and add one document at a time. We compare these strategies against two baseline strategies:

1. Select a document at random to present to the oracle (B1).
2. Select a document based on $\text{score}_{B2}(d) = \min(\{p_1, p_2, \ldots, p_m\})$ (B2).

In our experimental setting, we use customer reviews to test the convergence speed of two different strategies. Figure 2 shows that the second strategy achieves higher accuracy than the first one; however, both strategies achieve an accuracy greater than 65% using a training set of just 50 documents. The higher performance of S2 may be because documents with the smallest average probability may have some sentences with high probability, which do not need to be disambiguated.

4.5. Applications

Using a CRF-based model to identify sentiment in a document could have many interesting applications. Lots of people express their opinions in a sarcastic style, meaning that the implicit and actual sentiment is completely different from the explicit sentiment. From an individual sentence, it is not easy to recognize these sentences, but it might be possible to detect these kinds of sentences in social media data. By taking advantage of features of social media discussions such as emoticons and sentence context, and identifying sentences where the emoticons or context change the sentiment of the sentence, we may be able to identify sarcasm.

5. Conclusion and future work

In this paper, we investigate the syntactic and semantic features of sentences and apply a CRF-based model to identify sentiment. Due to

CRF's ability to capture context information, it outperforms other supervised and rule-based models for longer documents like online customer review data, but not for Facebook comments, due to their special characteristics. We also demonstrate an active learning strategy that achieves good accuracy with a much smaller training set. We also believe that our approach may be applicable to the problem of detecting sarcastic sentences (e.g., sentences are identified as positive but they contain negative emoticons.). In the future, we would like to extract more interesting and useful features to improve sentiment identification. We could also incorporate a topic model to refine the sentence sentiment with respect to the main topic or topics of the document.

References

Cambira, E., & White, B. (2014). Jumping nlp curves: A review of natural language processing research. *IEEE Computational Intelligence Magazine*, *9*, 2.

Cambria, E., Schuller, B., Xia, Y., & Havasi, C. (2013). New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems*, *28*, 15–21.

Choi, Y., & Cardie, C. (2008). Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*

EMNLP '08 (pp. 793–801). Stroudsburg, PA, USA: Association for Computational Linguistics.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, *41*, 391–407.

Ding, X., Liu, B., & Yu, P. S. (2008). A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 International Conference on Web Search and Data Mining* WSDM '08 (pp. 231–240). New York, NY, USA: ACM.

Esuli, A., & Sebastiani, F. (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06)* (pp. 417–422).

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* KDD '04 (pp. 168–177). New York, NY, USA: ACM.

Hu, X., Tang, J., Gao, H., & Liu, H. (2013). Unsupervised sentiment analysis with emotional signals. In *Proceedings of the 22Nd International Conference on World Wide Web* WWW '13 (pp. 607–618). Republic and

Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee.

Kim, S.-M., & Hovy, E. (2004). Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics* COLING '04. Stroudsburg, PA, USA: Association for Computational Linguistics.

Ku, L.-W., Lee, L.-Y., & Chen, H.-H. (2006). Opinion extraction, summarization and tracking in news and blog corpora. In *Proceedings of AAAI-2006 Spring Symposium on Computational Approaches to Analyzing Weblogs*.

Lafferty, J. D., McCallum, A., & Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning* ICML '01 (pp. 282–289). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Levy, R., & Manning, C. D. (2004). Deep dependencies from context-free statistical parsers: Correcting the surface dependency approximation. In D. Scott, W. Daelemans, & M. A. Walker (Eds.), *ACL* (pp. 327–334). ACL.

Liu, D. C., & Nocedal, J. (1989). On the limited memory bfgs method for large scale optimization. *Math. Program.*, *45*, 503–528.

Liu, J., & Seneff, S. (2009). Review sentiment scoring via a parse-and-paraphrase paradigm. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1* EMNLP '09 (pp. 161–169). Stroudsburg, PA, USA: Association for Computational Linguistics.

McCallum, A. (2003). Efficiently inducing features of conditional random fields. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence* UAI'03 (pp. 403–410). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Mcdonald, R., Hannan, K., Neylon, T., Wells, M., & Reynar, J. (2007). Structured models for fine-to-coarse sentiment analysis. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*.

Mei, Q., Ling, X., Wondra, M., Su, H., & Zhai, C. (2007). Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web* WWW '07 (pp. 171–180). New York, NY, USA: ACM.

Narayanan, R., Liu, B., & Choudhary, A. (2009). Sentiment analysis of conditional sentences. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1* EMNLP '09 (pp. 180–189). Stroudsburg, PA, USA: Association for Computational Linguistics.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, *2*, 1–135.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10* EMNLP '02 (pp. 79–86). Stroudsburg, PA, USA: Association for Computational Linguistics.

Peng, F., & McCallum, A. (2006). Information extraction from research papers using conditional random fields. *Inf. Process. Manage.*, *42*, 963–979.

Phan, X.-H. (2006). Crftagger: Crf english pos tagger, .

Popescu, A.-M., & Etzioni, O. (2005). Extracting product features and opinions from reviews. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing* HLT '05 (pp. 339–346). Stroudsburg, PA, USA: Association for Computational Linguistics.

Rabiner, L. R. (1990). Readings in speech recognition. chapter A tutorial on hidden Markov models and selected applications in speech recognition. (pp. 267–296). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Reynar, J. C., & Ratnaparkhi, A. (1997). A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the fifth conference on Applied natural language processing* ANLC '97 (pp. 16–19). Stroudsburg, PA, USA: Association for Computational Linguistics.

Riloff, E., & Wiebe, J. (2003). Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing* EMNLP '03 (pp. 105–112). Stroudsburg, PA, USA: Association for Computational Linguistics.

Russell, S. J., & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. (2nd ed.). Pearson Education.

Sha, F., & Pereira, F. (2003). Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1* NAACL '03 (pp. 134–141). Stroudsburg, PA, USA: Association for Computational Linguistics.

Stoyanov, V., & Cardie, C. (2008). Topic identification for fine-grained opinion analysis. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1* COLING '08 (pp. 817–824). Stroudsburg, PA, USA: Association for Computational Linguistics.

Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the*

40th Annual Meeting on Association for Computational Linguistics ACL '02 (pp. 417–424). Stroudsburg, PA, USA: Association for Computational Linguistics.

Wiebe, J., & Wilson, T. (2002). Learning to disambiguate potentially subjective expressions. In *proceedings of the 6th conference on Natural language learning - Volume 20* COLING-02 (pp. 1–7). Stroudsburg, PA, USA: Association for Computational Linguistics.

Wiebe, J., Wilson, T., & Cardie, C. (2005). Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, *39*, 165–210.

Wiebe, J. M., Bruce, R. F., & O'Hara, T. P. (1999). Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics* ACL '99 (pp. 246–253). Stroudsburg, PA, USA: Association for Computational Linguistics.

Wilson, T., Wiebe, J., & Hwa, R. (2004). Just how mad are you? finding strong and weak opinion clauses. In *Proceedings of the 19th national conference on Artifical intelligence* AAAI'04 (pp. 761–767). AAAI Press.

Xie, Y., Chen, Z., Cheng, Y., Zhang, K., Honbo, D., Agrawal, A., & Choudhary, A. (2013). Muses: a multilingual sentiment elicitation system for social media data. *IEEE Intelligent Systems*, *99*, 1.

Table 1: The distribution of sentiment labels of customer reviews on Amazon.com and comments on Facebook.

Source	**Documents**	**Sentences**	**Positive**	**Negative**	**Neutral**
Digital Camera	300	5,156	2,524	1,185	1,447
TV	300	5,036	2,364	1,252	1,420
Facebook	500	723	313	157	253

Yu, H., & Hatzivassiloglou, V. (2003). Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 conference on Empirical methods in natural language processing* EMNLP '03 (pp. 129–136). Stroudsburg, PA, USA: Association for Computational Linguistics.

Zhang, K., Xie, Y., Cheng, Y., Honbo, D., Downey, D., Agrawal, A., Liao, W.-k., & Choudhary, A. (2012). Sentiment identification by incorporating syntax, semantics and context information. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval* SIGIR '12 (pp. 1143–1144). New York, NY, USA: ACM.

Zhang, L., & Liu, B. (2011). Identifying noun product features that imply opinions. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2* HLT '11 (pp. 575–580). Stroudsburg, PA, USA: Association for Computational Linguistics.

Table 2: Customer review accuracy results of CRF-based model comparing with other methods (Compositional Semantic Rule - CSR, Support Vector Machine - SVM, Logistic Regression - LR, and Hidden Markov Model - HMM). The models were tested using datasets with semantic features only (SO) and with semantic and syntactic features (SS).

Data	**CSR**	**SVM**	**LR**	**HMM**	**CRF**
Camera (SO)	0.57	0.633	0.615	0.631	0.654
Camera (SS)	-	0.640	0.648	0.651	**0.72**
TV (SO)	0.54	0.612	0.60	0.629	0.630
TV (SS)	-	0.622	0.619	0.633	**0.665**
Overall (SO)	0.55	0.622	0.610	0.627	0.634
Overall (SS)	-	0.632	0.637	0.640	**0.693**

Table 3: Facebook comments data accuracy results of CRF-based model comparing with other methods (Compositional Semantic Rule - CSR, Support Vector Machine - SVM, Logistic Regression - LR, and Hidden Markov Model - HMM). The models were tested using datasets with semantic features only (SO) and with semantic and syntactic features (SS).

Data	**CSR**	**SVM**	**LR**	**HMM**	**CRF**
FB (SO)	**0.72**	0.60	0.610	0.607	0.612
FB (SS)	-	0.60	0.612	0.61	0.614

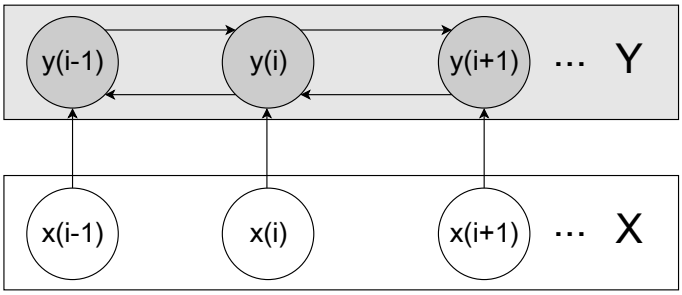

Figure 1: Graphical representation of a simple CRF model. x_i's can be understood as sentences in a document and y_i's are the sentiment labels for each x_i. Directed edges represent variable dependencies. Unlike Hidden Markov Models, y_i and y_{i-1} influence each other in CRF setup. In addition, CRF is a conditional probabilistic model, meaning that all y_i can be conditioned on X.

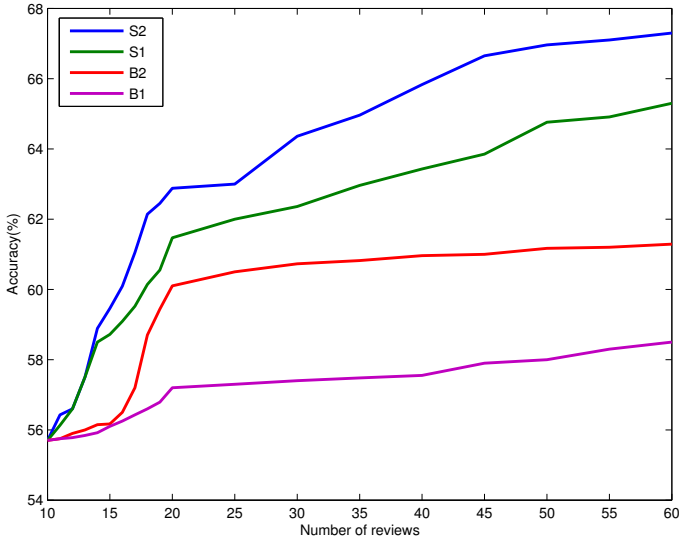

Figure 2: Convergence of the active learning technique applied to our datasets. X-axis describes the number of labeled samples chosen by the active learner. Y-axis shows how the model accuracy climbs as the active learner is given more labeled samples. With a good selecting strategy (S2), the model can converge to almost *full capacity* using only dozens of labeled samples. By *full capacity*, we mean the model trained with all available labels, which has hundreds.