A Weakly Supervised Model for Sentence-Level Semantic Orientation Analysis with Multiple Experts

Lizhen Qu and Rainer Gemulla and Gerhard Weikum
Max Planck Institute for Informatics
Saarbrücken, Germany
{lqu,rgemulla,weikum}@mpi-inf.mpg.de

Abstract

We propose the weakly supervised *Multi-Experts Model* (MEM) for analyzing the semantic orientation of opinions expressed in natural language reviews. In contrast to most prior work, MEM predicts both opinion polarity and opinion strength at the level of individual sentences; such fine-grained analysis helps to understand better why users like or dislike the entity under review. A key challenge in this setting is that it is hard to obtain sentence-level training data for both polarity and strength. For this reason, MEM is weakly supervised: It starts with potentially noisy indicators obtained from coarse-grained training data (i.e., document-level ratings), a small set of diverse base predictors, and, if available, small amounts of fine-grained training data. We integrate these noisy indicators into a unified probabilistic framework using ideas from ensemble learning and graph-based semi-supervised learning. Our experiments indicate that MEM outperforms state-of-the-art methods by a significant margin.

1 Introduction

Opinion mining is concerned with analyzing opinions expressed in natural language text. For example, many internet websites allow their users to provide both natural language reviews and numerical ratings to items of interest (such as products or movies). In this context, opinion mining aims to uncover the relationship between users and (features of) items. Preferences of users to items can be well understood by coarse-grained methods of opinion mining, which focus on analyzing the semantic orientation of documents as a whole. To understand *why* users like or dislike certain items, however, we need to perform more fine-grained analysis of the review text itself.

In this paper, we focus on sentence-level analysis of semantic orientation (SO) in online reviews. The SO consists of *polarity* (positive, negative, or other[1]) and *strength* (degree to which a sentence is positive or negative). Both quantities can be analyzed jointly by mapping them to numerical ratings: Large negative/positive ratings indicate a strong negative/positive orientation. A key challenge in fine-grained rating prediction is that fine-grained training data for both polarity and strength is hard to obtain. We thus focus on a weakly supervised setting in which only coarse-level training data (such as document ratings and subjectivity lexicons) and, optionally, a small amount of fine-grained training data (such as sentence polarities) is available.

A number of lexicon-based approaches for phrase-level rating prediction has been proposed in the literature (Taboada et al., 2011; Qu et al., 2010). These methods utilize a subjectivity lexicon of words along with information about their semantic orientation; they focus on phrases that contain words from the lexicon. A key advantage of sentence-level methods is that they are able to cover all sentences in a review and that phrase identification is avoided. To the best of our knowledge, the problem of rating prediction at the sentence level has not been addressed in the literature. A naive approach would be to simply average phrase-level ratings. Such an approach performs

[1] We assign polarity *other* to text fragments that are off-topic or not directly related to the entity under review.

Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 149–159, Jeju Island, Korea, 12–14 July 2012. ©2012 Association for Computational Linguistics

poorly, however, since (1) phrases are analyzed out of context (e.g., modal verbs or conditional clauses), (2) domain-dependent information about semantic orientation is not captured in the lexicons, (3) only phrases that contain lexicon words are covered. Here (1) and (2) lead to low precision, (3) to low recall.

To address the challenges outlined above, we propose the weakly supervised *Multi-Experts Model* (MEM) for sentence-level rating prediction. MEM starts with a set of potentially noisy indicators of SO including phrase-level predictions, language heuristics, and co-occurrence counts. We refer to these indicators as *base predictors*; they constitute the set of experts used in our model. MEM is designed such that new base predictors can be easily integrated. Since the information provided by the base predictors can be contradicting, we use ideas from ensemble learning (Dietterichl, 2002) to learn the most confident indicators and to exploit domain-dependent information revealed by document ratings. Thus, instead of averaging base predictors, MEM integrates their features along with the available coarse-grained training data into a unified probabilistic model.

The integrated model can be regarded as a Gaussian process (GP) model (Rasmussen, 2004) with a novel *multi-expert prior*. The multi-expert prior decomposes into two component distributions. The first component distribution integrates sentence-local information obtained from the base predictors. It forms a special realization of stacking (Dzeroski and Zenko, 2004) but uses the features from the base predictors instead of the actual predictions. The second component distribution propagates SO information across similar sentences using techniques from graph-based semi-supervised learning (GSSL) (Zhu et al., 2003; Belkin et al., 2006). It aims to improve the predictions on sentences that are not covered well enough by our base predictors. Traditional GSSL algorithms support either discrete labels (classification) or numerical labels (regression); we extend these techniques to support both types of labels simultaneously. We use a novel variant of word sequence kernels (Cancedda et al., 2003) to measure sentence similarity. Our kernel takes the relative positions of words but also their SO and synonymity into account.

Our experiments indicate that MEM significantly outperforms prior work in both sentence-level rating prediction and sentence-level polarity classification.

2 Related Work

There exists a large body of work on analyzing the semantic orientation of natural language text. Our approach is unique in that it is weakly supervised, predicts both polarity and strength, and operates on the sentence level.

Supervised approaches for sentiment analysis focus mainly on opinion mining at the document level (Pang and Lee, 2004; Pang et al., 2002; Pang and Lee, 2005; Goldberg and Zhu, 2006), but have also been applied to sentence-level polarity classification in specific domains (Mao and Lebanon, 2006; Pang and Lee, 2004; McDonald et al., 2007). In these settings, a sufficient amount of training data is available. In contrast, we focus on opinion mining tasks with little or no fine-grained training data.

The weakly supervised HCRF model (Täckström and McDonald, 2011b; Täckström and McDonald, 2011a) for sentence-level polarity classification is perhaps closest to our work in spirit. Similar to MEM, HCRF uses coarse-grained training data and, when available, a small amount of fine-grained sentence polarities. In contrast to MEM, HCRF does not predict the strength of semantic orientation and ignores the order of words within sentences.

There exists a large number of lexicon-based methods for polarity classification (Ding et al., 2008; Choi and Cardie, 2009; Hu and Liu, 2004; Zhuang et al., 2006; Fu and Wang, 2010; Ku et al., 2008). The lexicon-based methods of (Taboada et al., 2011; Qu et al., 2010) also predict ratings at the phrase level; these methods are used as experts in our model.

MEM leverages ideas from ensemble learning (Dietterichl, 2002; Bishop, 2006) and GSSL methods (Zhu et al., 2003; Zhu and Ghahramani, 2002; Chapelle et al., 2006; Belkin et al., 2006). We extend GSSL with support for multiple, heterogenous labels. This allows us to integrate our base predictors as well as the available training data into a unified model that exploits that strengths of algorithms from both families.

3 Base Predictors

Each of our *base predictors* predicts the polarity or the rating of a single phrase. As indicated above, we do not use these predictions directly in MEM but instead integrate the features of the base predictors (see Sec. 4.4). MEM is designed such that new base predictors can be integrated easily.

Our base predictors use a diverse set of available web and linguistic resources. The hope is that this diversity increases overall prediction performance (Dietterichl, 2002): The *statistical polarity predictor* focuses on local syntactic patterns; it is based on corpus statistics for SO-carrying words and opinion topic words. The *heuristic polarity predictor* uses manually constructed rules to achieve high precision but low recall. Both the *bag-of-opinions rating predictor* and the *SO-CAL rating predictor* are based on lexicons. The BoO predictor uses a lexicon trained from a large generic-domain corpus and is recall-oriented; the SO-CAL predictor uses a different lexicon with manually assigned weights and is precision-oriented.

3.1 Statistical Polarity Predictor

The polarity of an SO-carrying word strongly depends on its target word. For example, consider the phrase "I began this novel with the *greatest* of *hopes* [...]". Here, "greatest" has a positive semantic orientation in all subjectivity lexicons, but the combination "greatest of hopes" often indicates a negative sentiment. We refer to a pair of SO-carrying word ("greatest") and a target word ("hopes") as an *opinion-target pair*. Our statistical polarity predictor learns the polarity of opinions and targets jointly, which increases the robustness of its predictions.

Syntactic dependency relations of the form $A \xrightarrow{R} B$ are a strong indicator for opinion-target pairs (Qiu et al., 2009; Zhuang et al., 2006); e.g., "great"$\xrightarrow{\text{nmod}}$"product". To achieve high precision, we only consider pairs connected by the following predefined set of shortest dependency paths: verb $\xleftarrow{\text{subj}}$ noun, verb $\xleftarrow{\text{obj}}$ noun, adj $\xrightarrow{\text{nmod}}$ noun, adj $\xrightarrow{\text{prd}}$ verb $\xleftarrow{\text{subj}}$ noun. We only retain opinion-target pairs that are sufficiently frequent.

For each extracted pair z, we count how often it co-occurs with each document polarity $y \in \mathcal{Y}$, where $\mathcal{Y} = \{\textit{positive}, \textit{negative}, \textit{other}\}$ denotes the set of polarities. If z occurs in a document but is preceded by a negator, we treat it as a co-occurrence of opposite document polarity. If z occurs in a document with polarity *other*, we count the occurrence with only half weight, i.e., we increase both $\#z$ and $\#(\textit{other}, z)$ by 0.5. These documents are typically a mixture of positive and negative opinions so that we want to reduce their impact. The marginal distribution of polarity label y given that z occurs in a sentence is estimated as $P(y \mid z) = \#(y, z)/\#z$. The predictor is trained using the text and ratings of the reviews in the training data, i.e., without relying on fine-grained annotations.

The statistical polarity predictor can be used to predict sentence-level polarities by averaging the phrase-level predictions. As discussed previously, such an approach is problematic; we use it as a baseline approach in our experimental study. We also employ phrase-level averaging to estimate the variance of base predictors; see Sec. 4.3. Denote by $Z(\mathbf{x})$ the set of opinion-target pairs in sentence $\mathbf{x}$. To predict the sentence polarity $y \in \mathcal{Y}$, we take the Bayesian average of the phrase-level predictors: $P(y \mid Z(\mathbf{x})) = \sum_{z \in Z(\mathbf{x})} P(y \mid z)P(z) = \sum_{z \in Z(\mathbf{x})} P(y, z)$. Thus the most likely polarity is the one with the highest co-occurrence count.

3.2 Heuristic Polarity Predictor

Heuristic patterns can also serve as base predictors. In particular, we found that some authors list positive and negative aspects separately after keywords such as "pros" and "cons". A heuristic that exploits such patterns achieved a high precision ($> 90\%$) but low recall ($< 5\%$) in our experiments.

3.3 Bag-of-Opinions Rating Predictor

We leverage the bag-of-opinion (BoO) model of Qu et al. (2010) as a base predictor for phrase-level ratings. The BoO model was trained from a large generic corpus without fine-grained annotations.

In BoO, an opinion consists of three components: an SO-carrying word (e.g., "good"), a set of intensifiers (e.g., "very") and a set of negators (e.g., "not"). Each opinion is scored based on these words (represented as a boolean vector $\mathbf{b}$) and the polarity of the SO-carrying word (represented as $\operatorname{sgn}(r) \in \{-1, 1\}$) as indicated by the MPQA lexicon of Wilson et al. (2005). In particular, the score is computed as $\operatorname{sgn}(r)\boldsymbol{\omega}^{\mathrm{T}}\mathbf{b}$, where $\boldsymbol{\omega}$ is the learned weight vector. The sign function $\operatorname{sgn}(r)$ ensures consistent weight assignment for intensifiers and negators. For example, an intensifier like "very" can obtain a large positive or a large negative weight depending on whether it is used with a positive or negative SO-carrying word, respectively.

3.4 SO-CAL Rating Predictor

The Semantic Orientation Calculator (SO-CAL) of Taboada et al. (2011) also predicts phrase-level ratings via a scoring function similar to the one of BoO. The SO-CAL predictor uses a manually created lexicon, in which each word is classified as either an SO-carrying word (associated with a numerical score), an intensifier (associated with a modifier on the numerical score), or a negator. SO-CAL employs various heuristics to detect irrealis and to correct for the positive bias inherent in most lexicon-based classifiers. Compared to BoO, SO-CAL has lower recall but higher precision.

4 Multi-Experts Model

Our multi-experts model incorporates features from the individual base predictors, coarse-grained labels (i.e., document ratings or polarities), similarities between sentences, and optionally a small amount of sentence polarity labels into an unified probabilistic model. We first give an overview of MEM, and then describe its components in detail.

4.1 Model Overview

Denote by $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ a set of sentences. We associate each sentence $\mathbf{x}_i$ with a set of *initial labels* $\hat{\mathbf{y}}_i$, which are strong indicators of semantic orientation: the coarse-grained rating of the corresponding document, the polarity label of our heuristic polarity predictor, the phrase-level ratings from the SO-CAL predictor, and optionally a manual polarity label. Note that the number of initial labels may vary from sentence to sentence and that initial labels are heterogeneous in that they refer to either polarities or ratings. Let $\hat{\mathbf{Y}} = \{\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_N\}$. Our goal is to predict the unobserved ratings $\mathbf{r} = \{r_1, \ldots, r_N\}$ of each sentence.

Our multi-expert model is a probabilistic model for $\mathbf{X}$, $\hat{\mathbf{Y}}$, and $\mathbf{r}$. In particular, we model the rating vector $\mathbf{r}$ via a multi-expert prior $P_{\mathrm{E}}(\mathbf{r} \mid \mathbf{X}, \boldsymbol{\beta})$ with parameter $\boldsymbol{\beta}$ (Sec. 4.2). P_{E} integrates both features from the base predictors and sentence similarities. We correlate ratings to initial labels via a set of conditional distributions $P_b(\hat{\mathbf{y}}^b \mid \mathbf{r})$, where b denotes the type of initial label (Sec. 4.3). The posterior of $\mathbf{r}$ is then given by

$$P(\mathbf{r} \mid \mathbf{X}, \hat{\mathbf{Y}}, \boldsymbol{\beta}) \propto \prod_b P_b(\hat{\mathbf{y}}^b \mid \mathbf{r}) P_{\mathrm{E}}(\mathbf{r} \mid \mathbf{X}, \boldsymbol{\beta}).$$

Note that the posterior is influenced by both the multi-expert prior and the set of initial labels.

We use MAP inference to obtain the most likely rating of each sentence, i.e., we solve

$$\operatorname*{argmin}_{\mathbf{r}, \boldsymbol{\beta}} -\sum_b \log(P_b(\hat{\mathbf{y}}^b \mid \mathbf{r})) - \log(P_{\mathrm{E}}(\mathbf{r} \mid \mathbf{X}, \boldsymbol{\beta})),$$

where as before $\boldsymbol{\beta}$ denotes the model parameters. We solve the above optimization problem using cyclic coordinate descent (Friedman et al., 2008).

4.2 Multi-Expert Prior

The multi-expert prior $P_{\mathrm{E}}(\mathbf{r} \mid \mathbf{X}, \boldsymbol{\beta})$ consists of two component distributions $\mathcal{N}_1$ and $\mathcal{N}_2$. Distribution $\mathcal{N}_1$ integrates features from the base predictors, $\mathcal{N}_2$ incorporates sentence similarities to propagate information across sentences.

In a slight abuse of notation, denote by $\mathbf{x}_i$ the set of features for the i-th sentence. Vector $\mathbf{x}_i$ contains the features of all the base predictors but also includes bigram features for increased coverage of syntactic patterns; see Sec. 4.4 for details about the feature design. Let $m(\mathbf{x}_i) = \boldsymbol{\beta}^{\mathrm{T}} \mathbf{x}_i$ be a linear predictor for r_i, where $\boldsymbol{\beta}$ is a real weight vector. Assuming Gaussian noise, r_i follows a Gaussian distribution $\mathcal{N}_1(r_i \mid m_i, \sigma^2)$ with mean $m_i = m(\mathbf{x}_i)$ and variance σ^2. Note that predictor m can be regarded as a linear combination of base predictors because both m and each of the base predictors are linear functions. By integrating all features into a single function, the base predictors are trained jointly so that weight vector $\boldsymbol{\beta}$ automatically adapts to domain-dependent properties of the data. This integrated approach significantly outperformed the alternative approach of using a weighted vote of the individual predictions made by the base predictors. We regularize the weight vector $\boldsymbol{\beta}$ using a Laplace prior $P(\boldsymbol{\beta} \mid \alpha)$ with parameter α to encourage sparsity.

Note that the bigram features in $\mathbf{x}_i$ partially capture sentence similarity. However, such features cannot be extended to longer subsequences such as trigrams due to data sparsity: useful features become as infrequent as noisy terms. Moreover, we would like to capture sentence similarity using gapped (i.e., non-consecutive) subsequences. For example, the sentences "The book is an easy read." and "It is easy to read." are similar but do not share any consecutive bigrams. They do share the subsequence "easy read", however. To capture this similarity, we make use of a novel sentiment-augmented variant of word sequence kernels (Cancedda et al., 2003). Our kernel is used to construct a similarity matrix $\mathbf{W}$ among sentences and the corresponding regularized Laplacian $\widetilde{\mathbf{L}}$. To capture the intuition that similar sentences should have similar ratings, we introduce a Gaussian prior $\mathcal{N}_2(\mathbf{r} \mid 0, \widetilde{\mathbf{L}}^{-1})$ as a component into our multi-expert prior; see Sec. 4.5 for details and a discussion of why this prior encourages similar ratings for similar sentences.

Since the two component distributions feature different expertise, we take their product and obtain the multi-expert prior

$$P_{\mathrm{E}}(\mathbf{r} \mid \mathbf{X}, \boldsymbol{\beta}) \propto \mathcal{N}_1(\mathbf{r} \mid \mathbf{m}, \mathbf{I}\sigma^2)\mathcal{N}_2(\mathbf{r} \mid 0, \widetilde{\mathbf{L}}^{-1})P(\boldsymbol{\beta} \mid \alpha),$$

where $\mathbf{m} = (m_1, \ldots, m_N)$. Note that the normalizing constant of P_{E} can be ignored during MAP inference since it does not depend on $\boldsymbol{\beta}$.

4.3 Incorporating Initial Labels

Recall that the initial labels $\hat{\mathbf{Y}}$ are strong indicators of semantic orientation associated with each sentence; they correspond to either discrete polarity labels or to continuous rating labels. This heterogeneity constitutes the main difficulty for incorporating the initial labels via the conditional distributions $P_b(\hat{\mathbf{y}}^b \mid \mathbf{r})$. We assume independence throughout so that $P_b(\hat{\mathbf{y}}^b \mid \mathbf{r}) = \prod_i P_b(\hat{y}_i^b \mid r_i)$.

Rating Labels For continuous labels, we assume Gaussian noise and set $P_b(\hat{y}_i^b \mid r_i) = \mathcal{N}(\hat{y}_i^b \mid r_i, \eta_i^b)$, where variance η_i^b is a type- and sentence-dependent.

For SO-CAL labels, we simply set $\eta_i^{\text{SO-CAL}} = \eta^{\text{SO-CAL}}$, where $\eta^{\text{SO-CAL}}$ is a hyperparameter. The SO-CAL scores have limited influence in our overall model; we found that more complex designs lead to little improvement. We proceed differently for document ratings. Our experiment suggests that document ratings constitute the most important indicator of the SO of a sentence. Thus sentence ratings should be close to document ratings unless strong evidence to the contrary exists. In other words, we want variance η_i^{Doc} to be small.

When no manually created sentence-level polarity labels are available, we set the value of η_i^{Doc} depending on the polarity class. In particular, we set $\eta_i^{\text{Doc}} = 1$ for both positive and negative documents, and $\eta_i^{\text{Doc}} = 2$ for neutral documents. The reasoning behind this choice is that sentence ratings in neutral documents express higher variance because these documents often contain a mixture of positive and negative sentences.

When a small set of manually created sentence polarity labels is available, we train a classifier that predicts whether the sentence polarity coincides with the document polarity. If so, we set the corresponding variance η_i^{Doc} to a small value; otherwise, we choose a larger value. In particular, we train a logistic regression classifier (Bishop, 2006) using the following binary features: (1) an indicator variable for each document polarity, and (2) an indicator variable for each triple of base predictor, predicted polarity, and document polarity (set to 1 if the polarities match). We then set $\eta_i^{\text{Doc}} = (\tau p_i)^{-1}$, where p_i is the probability of matching polarities obtained from the classifier and τ is a hyperparameter that ensures correct scaling.

Polarity Labels We now describe how to model the correlation between the polarity of a sentence and its rating. An simple and effective approach is to partition the range of ratings into three consecutive partitions, one for each polarity class. We thus considering the polarity classes $\{$*positive*, *other*, *negative*$\}$ as ordered and formulate polarity classification as an ordinal regression problem (Chu and Ghahramani, 2006). We immediately obtain the distribution

$$\begin{aligned}
P_b(\hat{y}_i^b = \text{pos} \mid r_i) &= \Phi\left(\frac{r_i - b^+}{\sqrt{\eta^b}}\right) \\
P_b(\hat{y}_i^b = \text{oth} \mid r_i) &= \Phi\left(\frac{b^+ - r_i}{\sqrt{\eta^b}}\right) - \Phi\left(\frac{b^- - r_i}{\sqrt{\eta^b}}\right) \\
P_b(\hat{y}_i^b = \text{neg} \mid r_i) &= \Phi\left(\frac{b^- - r_i}{\sqrt{\eta^b}}\right),
\end{aligned}$$

where b^+ and b^- are the partition boundaries between *positive*/*other* and *other*/*negative*, respectively,[2] $\Phi(x)$ denotes the cumulative distribution function of the

[2] We set $b^+ = 0.3$ and $b^- = -0.3$ to calibrate to SO-CAL, which treats ratings in $[-0.3, 0, 3]$ as polarity *other*.

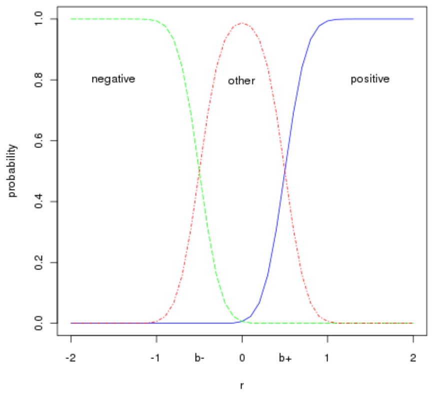

Figure 1: Distribution of polarity given rating.

Gaussian distribution, and variance η^b is a hyperparameter. It is easy to verify that $\sum_{\hat{y}_i^b \in \mathcal{Y}} p(\hat{y}_i^b \mid r_i) = 1$. The resulting distribution is shown in Fig. 1. We can use the same distribution to use MEM for sentence-level polarity classification; in this case, we pick the polarity with the highest probability.

4.4 Incorporating Base Predictors

Base predictors are integrated into MEM via component $\mathcal{N}_1(r_i \mid m_i, \sigma^2)$ of the multi-expert prior (see Sec. 4.2). Recall that m_i is a linear function of the features $\mathbf{x}_i$ of each sentence. In this section, we discuss how $\mathbf{x}_i$ is constructed from the features of the base predictors. New base predictors can be integrated easily by exposing their features to MEM.

Most base predictors operate on the phrase level; our goal is to construct features for the entire sentence. Denote by n_i^b the number of phrases in the i-th sentence covered by base predictor b, and let $\mathbf{o}_{ij}^b$ denote a set of associated features. Features $\mathbf{o}_{ij}^b$ may or may not correspond directly to the features of base predictor b; see the discussion below. A straightforward strategy is to set $\mathbf{x}_i^b = (n_i^b)^{-1} \sum_j \mathbf{o}_{ij}^b$. We proceed slightly differently and average the features associated with phrases of positive prior polarity separately from those of phrases with negative prior polarity (Taboada et al., 2011). We then concatenate the averaged feature vectors, i.e., we set $\mathbf{x}_i^b = (\bar{\mathbf{o}}_{ij}^{b,pos} \;\; \bar{\mathbf{o}}_{ij}^{b,neg})$, where $\bar{\mathbf{o}}_{ij}^{b,p}$ denotes the average of the feature vectors $\mathbf{o}_{ij}^b$ associated with phrases of prior polarity p. This procedure allows us to learn a different weight for each feature depending on its context (e.g., the weight of intensifier "very" may differ for positive and negative phrases). We construct $\mathbf{x}_i$ by concatenating the sentence-level features $\mathbf{x}_i^b$ of each base predictor and a feature vector of bigrams.

To integrate a base predictor, we only need to specify the relevant features and, if applicable, prior phrase polarities. For our choice of base predictors, we use the following features:

SO-CAL predictor. The prior polarity of a SO-CAL phrase is given by the polarity of its SO-carrying word in the SO-CAL lexicon. The feature vector $\mathbf{o}_{ij}^{\text{SO-CAL}}$ consists of the weight of the SO-carrying word from the lexicon as well the set of negator words, irrealis marker words, and intensifier words in the phrase. Moreover, we add the first two words preceding the SO-carrying word as context features (skipping nouns, negators, irrealis markers, and intensifiers, and stopping at clause boundaries). All words are encoded as binary indicator features.

BoO predictor. Similar to SO-CAL, we determine the prior polarity of a phrase based on the BoO dictionary. In contrast to SO-CAL, we directly use the BoO score as a feature because the BoO predictor weights have been trained on a very large corpus and are thus reliable. We also add irrealis marker words in the form of indicator features.

Statistical polarity predictor. Recall that the statistical polarity predictor is based on co-occurrence counts of opinion-topic pairs and document polarities. We treat each opinion-topic pair as a phrase and use the most frequently co-occurring polarity as the phrase's prior polarity. We use the logarithm of the co-occurrence counts with positive, negative, and other polarity as features; this set of features performed better than using the co-occurrence counts or estimated class probabilities directly. We also add the same type of context features as for SO-CAL, but rescale each binary feature by the logarithm of the occurrence count $\#z$ of the opinion-topic pair (i.e., the features take values in $\{0, \log \#z\}$).

4.5 Incorporating Sentence Similarities

The component distribution $\mathcal{N}_2(\mathbf{r} \mid 0, \widetilde{\mathbf{L}}^{-1})$ in the multi-expert prior encourages similar sentences to have similar ratings. The main purpose of $\mathcal{N}_2$ is to propagate information from sentences on which the base predictors perform well to sentences for which base prediction is unreliable or unavailable (e.g., because they do not contain SO-carrying words). To obtain this distribution, we first construct an $N \times N$ sentence similarity matrix $\mathbf{W}$ using a sentiment-augmented word sequence kernel (see below). We then compute the regularized graph Laplacian $\widetilde{\mathbf{L}} = \mathbf{L} + \mathbf{I}/\lambda^2$ based on the unnormalized graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{W}$ (Chapelle et al., 2006), where $\mathbf{D}$ be a diagonal matrix with $d_{ii} = \sum_j w_{ij}$ and hyperparameter λ^2 controls the scale of sentence ratings.

To gain insight into distribution $\mathcal{N}_2$, observe that

$$\mathcal{N}_2(\mathbf{r} \mid 0, \widetilde{\mathbf{L}}^{-1}) \\ \propto \exp\Big(-\frac{1}{2}\sum_{i,j} w_{ij}(r_i - r_j)^2 - \|\mathbf{r}\|_2^2/\lambda^2\Big).$$

The left term in the exponent forces the ratings of similar sentences to be similar: the larger the sentence similarity w_{ij}, the more penalty is paid for dissimilar ratings. For this reason, $\mathcal{N}_2$ has a smoothing effect. The right term is an L2 regularizer and encourages small ratings; it is controlled by hyperparameter λ^2.

The entries w_{ij} in the sentence similarity matrix determine the degree of smoothing for each pair of sentence ratings. We compute these values by a novel sentiment-augmented word sequence kernel, which extends the well-known word sequence kernel of Cancedda et al. (2003) by (1) BoO weights to strengthen the correlation of sentence similarity and rating similarity and (2) synonym resolution based on WordNet (Miller, 1995).

In general, a word sequence kernel computes a similarity score of two sequences based on their shared subsequences. In more detail, we first define a score function for a pair of shared subsequences, and then sum up these scores to obtain the overall similarity score. Consider for example the two sentences “The book is an easy read.” (s_1) and “It is easy to read.” (s_2) along with the shared subsequence “is easy read” (u). Observe that the words “an” and “to” serve as gaps as they are not part of the subsequence. We represent subsequence u in sentence s via a real-valued projection function $\phi_u(s)$. In our example, $\phi_u(s_1) = \upsilon_{\text{is}}\upsilon^g_{\text{an}}\upsilon_{\text{easy}}\upsilon_{\text{read}}$ and $\phi_u(s_2) = \upsilon_{\text{is}}\upsilon_{\text{easy}}\upsilon^g_{\text{to}}\upsilon_{\text{read}}$. The decay factors $\upsilon_w \in (0, 1]$ for matching words characterize the importance of a word (large values for significant words). On the contrary, decay factors $\upsilon^g_w \in (0, 1]$ for gap words are penalty terms for mismatches (small values for significant words). The score of subsequence u is defined as $\phi_u(s_1)\phi_u(s_2)$. Thus two shared subsequences have high similarity if they share significant words and few gaps. Following Cancedda et al. (2003), we define the similarity between two sequences as

$$k_n(s_i, s_j) = \sum_{u \in \Omega^n} \phi_u(s_i)\phi_u(s_j),$$

where Ω is a finite set of words and n denotes the length of the considered subsequences. This similarity function can be computed efficiently using dynamic programming.

To apply the word sequence kernel, we need to specify the decay factors. A traditional choice is $\upsilon_w = \log(\frac{N}{N_w})/\log(N)$, where N_w is the document frequency of the word w and N is the total number of documents. This IDF decay factor is not well-suited to our setting: Important opinion words such as “great” have a low IDF value due to their high document frequency. To overcome this problem, we incorporate additional weights for SO-carrying words using the BoO lexicon. To do so, we first rescale the BoO weights into $[0, 1]$ using the sigmoid $g(w) = (1 + \exp(-a\omega_w + b))^{-1}$, where ω_w denotes the BoO weight of word w.[3] We then set $\upsilon_w = \min(\log(\frac{N}{N_w})/\log(N) + g(w), 0.9)$. The decay factor for gaps is given by $\upsilon^g_w = 1 - \upsilon_w$. Thus we strongly penalize gaps that consist of infrequent words or opinion words.

To address data sparsity, we incorporate synonyms and hypernyms from WordNet into our kernel. In particular, we represent words found in WordNet by their first two synset names (for verbs, adjectives, nouns) and their direct hypernym (nouns only). Two words are considered the same when their synsets overlap. Thus, for example, “writer” has the same representation as “author”.

To build the similarity matrix $\mathbf{W}$, we construct a k-nearest-neighbor graph for all sentences.[4] We consider subsequences consisting of three words (i.e., $w_{ij} = k_3(s_i, s_j)$); longer subsequences are overly sparse, shorter subsequences are covered by the bigrams features in $\mathcal{N}_1$.

[3] We set $a = 2$ and $b = 1$ in our experiments.

[4] We use $k = 15$ and only consider neighbors with a similarity above 0.001.

5 Experiments

We evaluated both MEM and a number of alternative approaches for both sentence-level polarity classification and sentence-level strength prediction across a number of domains. We found that MEM outperforms state-of-the-art approaches by a significant margin.

5.1 Experimental Setup

We implemented MEM as well as the HCRF classifier of (Täckström and McDonald, 2011a; Täckström and McDonald, 2011b), which is the best-performing estimator of sentence-level polarity in the weakly-supervised setting reported in the literature. We train both methods using (1) only coarse labels (MEM-Coarse, HCRF-Coarse) and (2) additionally a small number of sentence polarities (MEM-Fine, HCRF-Fine[5]). We also implemented a number of baselines for both polarity classification and strength prediction: a document oracle (DocOracle) that simply uses the document label for each sentence, the BoO rating predictor (Base_{BoO}), and the SO-CAL rating predictor ($\text{Base}_{\text{SO-CAL}}$). For polarity classification, we compare our methods also to the statistical polarity predictor ($\text{Base}_{\text{polarity}}$). To judge on the effectiveness of our multi-export prior for combining base predictors, we take the majority vote of all base predictors and document polarity as an additional baseline (Majority-Vote). Similarly, for strength prediction, we take the arithmetic mean of the document rating and the phrase-level predictions of Base_{BoO} and $\text{Base}_{\text{SO-CAL}}$ as a baseline (Mean-Rating). We use the same hyperparameter setting for MEM across all our experiments.

We evaluated all methods on Amazon reviews from different domains using the corpus of Ding et al. (2008) and the test set of Täckström and McDonald (2011a). For each domain, we constructed a large balanced dataset by randomly sampling 33,000 reviews from the corpus of Ding et al. (2008). We chose the books, electronics, and music domains for our experiments; the dvd domain was used for development. For sentence polarity classification, we use the test set of Täckström and McDonald (2011a), which contains roughly 60 reviews per domain (20 for each polarity). For strength evaluation, we created a test set of 300 pairs of sentences per domain from the polarity test set. Each pair consisted of two sentences of the same polarity; we manually determined which of the sentences is more positive. We chose this pairwise approach because (1) we wanted the evaluation to be invariant to the scale of the predicted ratings, and (2) it much easier for human annotators to rank a pair of sentences than to rank a large collection of sentences.

We followed Täckström and McDonald (2011b) and used 3-fold cross-validation, where each fold consisted of a set of roughly 20 documents from the test set. In each fold, we merged the test set with the reviews from the corresponding domain. For MEM-Fine and HCRF-Fine, we use the data from the other two folds as fine-grained polarity annotations. For our experiments on polarity classification, we converted the predicted ratings of MEM, Base_{BoO}, and $\text{Base}_{\text{SO-CAL}}$ into polarities by the method described in Sec. 4.3. We compare the performance of each method in terms of accuracy, which is defined as the fraction of correct predictions on the test set (correct label for polarity / correct ranking for strength). All reported numbers are averages over the three folds. In our tables, boldface numbers are statistically significant against all other methods (t-test, p-value 0.05).

5.2 Results for Polarity Classification

Table 1 summarizes the results of our experiments for sentence polarity classification. The base predictors perform poorly across all domains, mainly due to the aforementioned problems associated with averaging phrase-level predictions. In fact, DocOracle performs almost always better than any of the base predictors. However, accurracy increases when we combine base predictors and DocOracle using majority voting, which indicates that ensemble methods work well.

When no fine-grained annotations are available (HCRF-Coarse, MEM-Coarse), both MEM-Coarse and Majority-Vote outperformed HCRF-Coarse, which in turn has been shown to outperform a number of lexicon-based methods as well as classifiers trained on document labels (Täckström and McDonald, 2011a). MEM-Coarse also performs better than Majority-Vote. This is because MEM propagates

[5] We used the best-performing model that fuses HCRF-Coarse and the supervised model (McDonald et al., 2007) by interpolation.

	Book	Electronics	Music	Avg
$\text{Base}_{\text{polarity}}$	43.7	40.3	43.8	42.6
Base_{BoO}	50.9	48.9	52.6	50.8
$\text{Base}_{\text{SO-CAL}}$	44.6	50.2	45.0	46.6
DocOracle	51.9	49.6	59.3	53.6
Majority-Vote	53.7	53.4	58.7	55.2
HCRF-Coarse	52.2	53.4	57.2	54.3
MEM-Coarse	54.4	54.9	**64.5**	57.9
HCRF-Fine	55.9	**61.0**	58.7	58.5
MEM-Fine	**59.7**	59.6	63.8	**61.0**

Table 1: Accuracy of polarity classification per domain and averaged across domains.

	Book		Electronics		Music	
	op	fact	op	fact	op	fact
HCRF-Fine	55.7	55.9	**63.3**	54.6	59.0	57.4
MEM-Fine	**58.9**	**62.4**	60.7	**56.7**	**64.5**	**60.8**

Table 2: Accuracy of polarity classification for sentences with opinion words (op) and without opinion words (fact).

evidence across similar sentences, which is especially useful when no explicit SO-carrying words exist. Also, MEM learns weights of features of base predictors, which leads to a more adaptive integration, and our ordinal regression formulation for polarity prediction allows direct competition among positive and negative evidence for improved accuracy.

When we incorporate a small amount of sentence polarity labels (HCRF-Fine, MEM-Fine), the accuracy of all models greatly improves. HCRF-Fine has been shown to outperform the strongest supervised method on the same dataset (McDonald et al., 2007; Täckström and McDonald, 2011b). MEM-Fine falls short of HCRF-Fine only in the electronics domain but performs better on all other domains. In the book and music domains, where MEM-Fine is particularly effective, many sentences feature complex syntactic structure and SO-carrying words are often used without reference to the quality of the product (but to describe contents, e.g., “a love story” or “a horrible accident”).

Our models perform especially well when they are applied to sentences containing no or few opinion words from lexicons. Table 2 reports the evaluation results for both sentences containing SO-carrying words from either MPQA or SO-CAL lexicons and for sentences containing no such words. The results explain why our model falls short of HCRF-Fine in the electronics domain: reviews of electronic products contain many SO-carrying words, which almost always express opinions. Nevertheless, MEM-Fine handles sentences without explicit SO-carrying words well across all domains; here the propagation of information across sentences helps to learn the SO of facts (such as “short battery life”).

We found that for all methods, most of the errors are caused by misclassifying positive/negative sentences as *other* and vice versa. Moreover, sentences with polarity opposite to the document polarity are hard cases if they do not feature frequent strong patterns. Another difficulty lies in off-topic sentences, which may contain explicit SO-carrying words but are not related to the item under review. This is one of the main reasons for the poor performance of the lexicon-based methods.

Overall, we found that MEM-Fine is the method of choice. Thus our multi-expert model can indeed balance the strength of the individual experts to obtain better estimation accuracy.

5.3 Results for Strength Prediction

Table 3 shows the accuracy results for strength prediction. Here our models outperformed all baselines by a large margin. Although document ratings are strong indicators in the polarity classification task, they lead to worse performance than lexicon-based methods. The main reason for this drop in accuracy is that the document oracle assigns the same rating to all sentences within a review. Thus DocOracle cannot rank sentences from the same review, which is a severe limitation. This shortage can be partly compensated by averaging the base predictions and document rating (Mean-Rating). Note that it is non-trivial to apply existing ensemble methods for the weights of individual base predictors because of the absence of the sentence ratings as training labels. In contrast, our MEM models use indirect supervision to adaptively assign weights to the features from base predictors. Similar to polarity classification, a small amount of sentence polarity labels often improved the performance of MEM.

	Book	Electronics	Music	Avg
Base$_{\text{BoO}}$	58.3	51.6	53.5	54.5
Base$_{\text{SO-CAL}}$	60.6	57.1	47.6	55.1
DocOracle	45.1	36.2	41.4	40.9
Mean-Rating	70.3	57.0	60.8	62.7
MEM-Coarse	68.7	60.5	**69.5**	66.2
MEM-Fine	**72.4**	**63.3**	67.2	**67.6**

Table 3: Accuracy of strength prediction.

6 Conclusion

We proposed the Multi-Experts Model for analyzing both opinion polarity and opinion strength at the sentence level. MEM is weakly supervised; it can run without any fine-grained annotations but is also able to leverage such annotations when available. MEM is driven by a novel multi-expert prior, which integrates a number of diverse base predictors and propagates information across sentences using a sentiment-augmented word sequence kernel. Our experiments indicate that MEM achieves better overall accuracy than alternative methods.

References

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. 2006. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *The Journal of Machine Learning Research*, 7:2399–2434.

Christopher M. Bishop. 2006. *Pattern recognition and machine learning*, volume 4. Springer New York.

Nicola Cancedda, Éric Gaussier, Cyril Goutte, and Jean-Michel Renders. 2003. Word-sequence kernels. *Journal of Machine Learning Research*, 3:1059–1082.

Oliver Chapelle, Bernhard Schölkopf, and Alexander Zien. 2006. *Semi-Supervised Learning*. MIT Press.

Yejin Choi and Claire Cardie. 2009. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, volume 2, pages 590–598.

Wei Chu and Zoubin Ghahramani. 2006. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6(1):1019.

Thomas G. Dietterichl. 2002. Ensemble learning. *The Handbook of Brain Theory and Neural Networks*, pages 405–408.

Xiaowen Ding, Bing Liu, and Philip S. Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the International Conference on Web Search and Data Mining*, pages 231–240.

Saso Dzeroski and Bernard Zenko. 2004. Is combining classifiers with stacking better than selecting the best one? *Machine Learning*, 54(3):255–273.

Jerome H. Friedman, Trevor Hastie, and Rob Tibshirani. 2008. Regularization paths for generalized linear models via coordinate descent. Technical report.

Guohong Fu and Xin Wang. 2010. Chinese sentence-level sentiment classification based on fuzzy sets. In *Proceedings of the International Conference on Computational Linguistics*, pages 312–319. Association for Computational Linguistics.

Andrew B. Goldberg and Xiaojun Zhu. 2006. Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. In *HLT-NAACL 2006 Workshop on Textgraphs: Graph-based Algorithms for Natural Language Processing*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 168–177.

Lun-Wei Ku, I-Chien Liu, Chia-Ying Lee, Kuan hua Chen, and Hsin-Hsi Chen. 2008. Sentence-level opinion analysis by copeopi in ntcir-7. In *Proceedings of NTCIR-7 Workshop Meeting*.

Yi Mao and Guy Lebanon. 2006. Isotonic Conditional Random Fields and Local Sentiment Flow. *Advances in Neural Information Processing Systems*, pages 961–968.

Ryan T. McDonald, Kerry Hannan, Tyler Neylon, Mike Wells, and Jeffrey C. Reynar. 2007. Structured models for fine-to-coarse sentiment analysis. In *Proceedings of the Annual Meeting on Association for Computational Linguistics*, volume 45, page 432.

George A. Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the Annual Meeting on Association for Computational Linguistics*, pages 271–278.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 124–131.

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.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2009. Expanding Domain Sentiment Lexicon through Double Propagation. In *International Joint Conference on Artificial Intelligence*, pages 1199–1204.

Lizhen Qu, Georgiana Ifrim, and Gerhard Weikum. 2010. The bag-of-opinions method for review rating prediction from sparse text patterns. In *Proceedings of the International Conference on Computational Linguistics*, pages 913–921.

Carl Edward Rasmussen. 2004. *Gaussian processes in machine learning*. Springer.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly D. Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2):267–307.

Oscar Täckström and Ryan T. McDonald. 2011a. Discovering Fine-Grained Sentiment with Latent Variable Structured Prediction Models. In *Proceedings of the European Conference on Information Retrieval*, pages 368–374.

Oscar Täckström and Ryan T. McDonald. 2011b. Semi-supervised latent variable models for sentence-level sentiment analysis. *In Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 569–574.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing*, pages 347–354.

Xiaojin Zhu and Zoubin Ghahramani. 2002. Learning from labeled and unlabeled data with label propagation. Technical report.

Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. 2003. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the International Conference on Machine Learning*, pages 912–919.

Li Zhuang, Feng Jing, and Xiaoyan Zhu. 2006. Movie review mining and summarization. In *Proceedings of the ACM international conference on Information and knowledge management*, pages 43–50.