Multiple Aspect Ranking for Opinion Analysis

by

Benjamin Snyder

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2007

Author . . . . . .
Department of Electrical Engineering and Computer Science
May 11, 2007

Certified by . . . . . .
Regina Barzilay
Assistant Professor
Thesis Supervisor

Accepted by . . . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Multiple Aspect Ranking for Opinion Analysis

by

Benjamin Snyder

Submitted to the Department of Electrical Engineering and Computer Science
on May 11, 2007, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

We address the problem of analyzing multiple related opinions in a text. For instance, in a restaurant review such opinions may include food, ambience and service. We formulate this task as a multiple aspect ranking problem, where the goal is to produce a set of numerical scores, one for each aspect. We present an algorithm that jointly learns ranking models for individual aspects by modeling the dependencies between assigned ranks. This algorithm guides the prediction of individual rankers by analyzing meta-relations between opinions, such as agreement and contrast. We provide an online training algorithm for our joint model which trains the individual rankers to operate in our framework. We prove that our agreement-based joint model is more expressive than individual ranking models, yet our training algorithm preserves the convergence guarantees of perceptron rankers. Our empirical results further confirm the strength of the model: the algorithm provides significant improvement over both individual rankers, a state-of-the-art joint ranking model, and ad-hoc methods for incorporating agreement.

Thesis Supervisor: Regina Barzilay
Title: Assistant Professor

Acknowledgments

I gratefully acknowledge my advisor, Regina Barzilay, as my chief collaborator and motivator for this work. Needless to say, her tireless efforts formed this thesis as much as my own hands. I would also like to thank Michael Collins for graciously providing valuable discussion and feedback throughout the last two years.

My friends and colleagues at CSAIL have always kept my spirits buoyed by listening to my stories and ideas. Among them are (in alphabetical order) Harr Chen, Jeremy Fineman, Ali Mohammad, Federico Mora, Tara Sainath, and Natasha Singh-Miller.

Finally, I'd like to dedicate this thesis to my loving family: My parents and grandparents, my four sisters, my brother, my two brothers-in-law, my sister-in-law, and most especially my young nieces and nephew, Ahuva, Leora, Hannah, and Ari.

Bibliographic Note

Portions of this thesis are based on the paper “Multiple Aspect Ranking using the Good Grief Algorithm” with Regina Barzilay [28], which appeared in the Proceedings of NAACL HLT 2007. Note that due to an improved perceptron averaging scheme, the results reported here (for both our models and baselines) are better than those reported in the paper and should be viewed as superseding them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Motivation

Previous work on sentiment categorization makes an implicit assumption that a single score can express the polarity of an opinion text [22, 30, 33]. However, multiple opinions on related matters are often intertwined throughout a text. For example, a restaurant review may express judgment on food quality as well as the service and ambience of the restaurant. Rather than lumping these aspects into a single score, we would like to capture each aspect of the writer's opinion separately, thereby providing a more fine-grained view of opinions in the review.

To this end, we aim to predict a set of numeric ranks that reflects the user's satisfaction for each aspect. In the example above, we would assign a numeric rank from 1-5 for each of: food quality, service, and ambience.

A straightforward approach to this task would be to rank[1] the text independently for each aspect, using standard ranking techniques such as regression or classification. However, this approach fails to exploit meaningful dependencies between users' judgments across different aspects. Knowledge of these dependencies can be crucial in predicting accurate ranks, as a user's opinions on one aspect can influence his or her opinions on others.

[1] In this work, *ranking* refers to the task of assigning an integer from 1 to k to each instance. This task is sometimes referred to as "ordinal regression" [7] and "rating prediction" [21].

1.2 The Good Grief Framework

The framework presented in this work allows an algorithm designer to capture arbitrary label dependencies between related tasks through an explicit *meta-model* which predicts *relations* between the labels of a given input, rather than specific label values. Joined with this meta-model are separate models for each task which predict specific label values. We develop a joint decoding criterion which takes into account the preferences of all component models as well as their measures of confidence in these preferences, measured in terms of prediction margins. Equivalently, we measure the *negative confidence* or ***grief*** of non-preferred predictions for each model. The joint prediction which minimizes the overall *grief* of all models – the task-specific models as well as the meta-model – is then predicted. We refer to this inference method as Good Grief Decoding, and the overall framework as the Good Grief Framework.

We further develop two online training algorithms for jointly training the individual label-prediction models: In the first, the meta-model is trained alongside the label-prediction models in online fashion, using the output of Good Grief Decoding as feedback to update all models. In the second variation, the meta-model is trained ahead of time using any desired batch method, such as SVM optimization, and is then given as input to the joint online training of the label-prediction models.

In this work, we focus exclusively on the case where the underlying label prediction problem is an instance of the *ranking* problem (see footnote 1.1). Ranking itself is a generalization of the binary classification problem, and the extension of our framework to the multiclass classification problem is straightforward. We also focus in the main on one particular meta relation between labels: *the agreement relation*. In the context of opinion analysis, the agreement relation captures whether the user equally likes all aspects of the item or whether he or she expresses different degrees of satisfaction. Since this rhetorical relation can often be determined automatically for a given text [18], it is natural to choose it to improve rank prediction.

Thus, in the course of our experiments, the Good Grief model will usually consist of a ranking model for each aspect as well as an agreement model which predicts whether or not

all rank aspects are equal. The Good Grief decoding algorithm then predicts a set of ranks – one for each aspect – which maximally satisfy the preferences of the individual rankers and the agreement model. For example, if the agreement model predicts consensus but the individual rankers select ranks $\langle 5, 5, 4 \rangle$, then the decoding algorithm chooses whether to "trust" the the third ranker, or alter its prediction and output $\langle 5, 5, 5 \rangle$ to be consistent with the agreement prediction.

1.3 Key Contributions

Our key technical contributions in this work are three-fold: First, our Good Grief method extends the Perceptron framework to allow the modeling of label-dependencies between tasks while preserving its key merits. Second, we demonstrate an increase in expressivity due to our meta-model and provide a mistake bound analysis. Third, we provide extensive experimental results in the task of sentiment analysis to show the practical merits of our method.

1.3.1 Extension of the Perceptron Framework

The Perceptron framework was first proposed by Rosenblatt in 1958 [25]. In this framework, a simple linear model iteratively classifies examples as positive or negative. In response to each incorrect prediction, the model is given the true label and is updated by simply adding or subtracting the input from its feature weights. The key advantages of the Perceptron approach are:

- model simplicity,
- simple and fast training (linear in the number of training examples),
- theoretical guarantees on convergence and generalization [9], and
- simple, exact, and fast decoding.

In addition, the Perceptron method and variants have been shown to be competitive in recent years with more complex methods on many Pattern Recognition and Natural Language Pro-

cessing tasks. The list includes Handwritten Digit Recognition [13], Named Entity Extraction [4], Part-of-Speech Tagging [3], Language Modeling [24], Syntactic Chunking [10], Parsing [5], and Database-Text Alignment [27].

In our work, we build on the practical success of this framework by integrating a meta-model for label dependencies between related tasks. We perform joint decoding in a way that respects the margin-based predictions of all component models. In this way, we essentially *factor out* inter-label dependency predictions, and preserve the key features of the Perceptron framework: speed, simplicity, and accuracy. This factored approach also allows the algorithm designer flexibility in designing the meta-model appropriate for the task at hand.

1.3.2 Theoretical Analysis

We demonstrate that the agreement-based joint model is more expressive than individual ranking models. That is, every training corpus that can be perfectly ranked by individual ranking models for each aspect can also be perfectly ranked with our joint model. In addition, we give a simple example of a training set which cannot be perfectly ranked without agreement-based joint inference, demonstrating the increase in expressive power.

We also provide a general mistake bound analysis for the Good Grief framework which applies to any meta-model. We show that even with the potential increase in expressive power, Good Grief Decoding preserves the finite mistake bound of simple Perceptron training.

1.3.3 Empirical Results

Our experimental results further confirm the strength of the Good Grief model. We apply our joint model to a set of restaurant reviews collected from a consumer website. Associated with each review is a set of five ranks, each on a scale from 1-5, covering food, ambience, service, value, and overall experience. Using the agreement meta-model with Good Grief decoding yields significant improvements over individual ranking models [7], a state-of-the art joint ranking model [1], and multiclass Support Vector Machines [6].

We also perform experiments comparing our model to other decoding methods using an agreement model. One such method first performs agreement classification on each instance and then delegates the instance to a single model (in the case of agreement) or to individually trained ranking models (in the case of disagreement). We found that our model outperforms all other strategies for incorporating an agreement model to which we compared it.

We also compared different methods of training our Good Grief model. The simplest approach is to individually train each ranking model as well as the agreement model, and only apply Good Grief decoding at test time. In fact, even this approach outperforms all baselines. However, larger gains are seen when jointly training all ranking models with a pre-trained perceptron agreement model. The best results with a perceptron agreement model are seen when the meta-model itself is trained jointly with all the ranking models, by using the feedback from Good Grief decoding. Finally, similar results are found when pre-training the agreement model using SVM optimization.

In the last set of experiments, we demonstrate the flexibility of the Good Grief framework by applying two meta-models besides the simple agreement model. Both models perform above all baselines. In addition, one of the models was specifically designed to aid performance on the most difficult-to-rank aspect (*atmosphere*), and in fact on this aspect achieves the best performance of any method.

1.4 Thesis Overview

The remainder of the thesis is organized as follows: In the next chapter we will discuss related work in the areas of Sentiment Analysis, Ordinal Ranking, Multitask Classification, Multifield Information Extraction, and Global Inference using ILP. In chapter 3, we provide a detailed formal description of the Good Grief framework with complete training and decoding algorithms. In chapter 4, we provide a formal analysis of the expressive power of our model, as well as proving a finite mistake bound. In chapter 5, we discuss numerous experiments which show the practical value of our method, and finally we present concluding remarks in chapter 6, along with directions for future research.

Chapter 2

Related Work

Traditionally, categorization of opinion texts has been cast as a binary classification task [22, 30, 33, 11]. More recent work [21, 14] has expanded this analysis to the ranking framework where the goal is to assess review polarity on a multi-point scale. While the ranking approach provides a more fine-grained representation of a *single* opinion, it still operates on the assumption of one opinion per text. Our work generalizes this setting to the problem of analyzing multiple opinions – or multiple aspects of an opinion. Since multiple opinions in a single text are related, it is insufficient to treat them as separate single-aspect ranking tasks. This motivates our exploration of a new method for joint multiple aspect ranking. In this chapter we present background work on ranking and joint ranking. We also survey several lines of NLP research that also deal with the joint prediction of multiple related tasks. These include *(i) Multitask Text Categorization*, where the goal is to classify a document in several related categorization schemes, *(ii) Multifield Information Extraction*, where the goal is to extract multiple fields of a single database entry from raw text, and *(iii) joint entity recognition and relation extraction*, where the goal is to recognize entities in text as well as their relationships to one another.

2.1 Ranking

The ranking, or ordinal regression, problem has been extensively studied in the Machine Learning and Information Retrieval communities. In this section we focus on two online

ranking methods which form the basis of our approach. The first is a model proposed by Crammer and Singer [7]. The task is to predict a rank $y \in \{1, ..., k\}$ for every input $\mathbf{x} \in \mathbb{R}^n$. Their model stores a weight vector $\mathbf{w} \in \mathbb{R}^n$ and a vector of increasing boundaries $b_0 = -\infty \leq b_1 \leq ... \leq b_{k-1} \leq b_k = \infty$ which divide the real line into k segments, one for each possible rank. The model first scores each input with the weight vector: $score(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$. Finally, the model locates $score(\mathbf{x})$ on the real line and returns the appropriate rank as indicated by the boundaries. Formally, the model returns the rank r such that $b_{r-1} \leq score(\mathbf{x}) < b_r$. The model is trained with the Perceptron Ranking algorithm (or "PRank" algorithm). See Figure 2-1 for pseudo-code. This perceptron-style training algorithm responds to incorrect predictions on the training set by updating both the weight vector and the boundary vector: the features of the incorrectly labeled instance are either added or subtracted to the weight vector, to respectively increase or decrease the instance's resulting score. The boundaries between the line-segment corresponding to the predicted rank and the line-segment of the true rank are shifted by a unit value, enlarging the segment of the desired rank and shrinking the segment of the incorrectly predicted rank.

The PRank model and algorithm were tested on the `EachMovie` dataset [19], which contains five-point ratings of 1,623 movies contributed by 61,265 people. Many movies are rated by multiple people and many people rate multiple movies. Crammer and Singer train a separate ranking model for each user, with the other users' ratings of the movie in question as features. Thus, a user's ranking model learns to predict his or her preferences based on the preferences of others (by learning how "like-minded" he or she is to each other user). The authors compare this method with two other online learning methods for ranking: the Multiclass Perceptron [8], and Widrow Hoff Online Regression [32]. They found that the PRank model achieved lower ranking loss (the average distance between the true and predicted rank – see chapter 5 for details) than these baselines.

2.1.1 Parameter Sharing

An extension of the PRank framework is provided by Basilico and Hofmann [1] in the context of collaborative filtering. Instead of training a separate model for each user, Basilico

and Hofmann train a joint ranking model which shares a single vector of boundaries across all users. In addition to these shared boundaries, user-specific weight vectors are stored. To compute the score for input $\mathbf{x}$ and user i, the weight vectors for *all* users are employed:

$$score_i(\mathbf{x}) = \mathbf{w}[i] \cdot \mathbf{x} + \sum_j sim(i,j)(\mathbf{w}[j] \cdot \mathbf{x}) \tag{2.1}$$

where $0 \leq sim(i,j) \leq 1$ is the cosine similarity between users i and j, computed on the entire training set. Once the score has been computed, the prediction rule follows that of the PRanking model. The model is trained using the PRank algorithm, with the exception of the new definition for the scoring function.[1] The authors demonstrate how kernels can be used to represent these shared weight vectors as well as to incorporate additional sources of information such as demographic information about users and genre information about the movies. Using this joint representation yields improved performance on the `EachMovie` dataset.

While this model shares parameter values between different ranking problems in an intelligent manner, it fails to explicitly model relations between the rank predictions. In contrast, our framework (besides incorporating parameter sharing in the same manner) explicitly models dependencies between related ranking decisions.

2.2 Multitask Text Categorization

In the general problem of Multitask Classification, several related classification tasks must be performed on each input. Renders et. al. [23] explore the problem of classifying documents in two related category systems. For example, a Xerox hardware customer complaint log contains some written text as well as a tag describing the *type* of problem as well as the *severity* of the problem. Renders et. al. assume that they are given probabilistic classifiers for both categorization tasks, trained independently of one another. They are also given some set of documents which contain tags for both categories from which they can train a

[1] In the notation of Basilico and Hofmann [1], this definition of $score_i(\mathbf{x})$ corresponds to the kernel $K = (K_U^{id} + K_U^{co}) \oplus K_X^{at}$.

joint model. The authors explore two approaches for re-weighting the probabilities of the classifiers based on learned dependencies between tags in the two label-sets. In the first approach, the probability distributions for the tasks are each asymmetrically re-estimated based on the initial output of the two classifiers. In the second approach the probabilities for the two tasks are *jointly* re-estimated.

The main assumption underlying both approaches is that all dependencies on features of the input document are exhausted by the previously trained classifiers, and do not directly affect the re-weighting of probabilities. The re-weighting *only* depends on the initial probabilities of the independent classifiers and general correlation statistics between the two label-sets. A little more formally, let x be an input document, let c_1 and c_2 be the class labels for the two classification categories, and let $\hat{P}(\cdot)$ be the probability estimator obtained from the independently trained classifiers. In both approaches, the authors end up with a re-weighting formula of the following form:

$$P(c_1 = i|x) = \hat{P}(c_1 = i|x) \sum_j \gamma(i,j)\hat{P}(c_2 = j|x) \tag{2.2}$$

The important term here is $\gamma(i,j)$, which is a measure of the *generic compatibility* of the joint labeling decision $c_1 = i, c_2 = j$. The values of γ can simply be the normalized counts of co-occurences of label-pairs, or can be learned in a slightly more complex fashion. In either case, the re-weighting term for a particular pair of class labels is fixed and not sensitive to the particular input being judged.

As we will see in the remaining chapters, the main power of our approach lies in the ability to incorporate a label dependency model which is *sensitive to the features of each input*. In fact, the label dependency model may even use a *richer* set of features than that used by the underlying categorization models. In addition, instead of assuming independently trained component models, we develop a joint training regimen. The Good Grief framework, however, is flexible enough to incorporate generic label-dependencies as well. In Chapter 5, we show experimental results which indicate that full joint learning yields better results than independently trained component models. We also show that using a label-dependency model which is sensitive to input features yields better results than incor-

porating generic label correlations.

2.3 Multifield Information Extraction

The multiple aspect ranking problem is related to some recent work in Multifield Information Extraction, where the goal is to automatically extract related fields of information (analogous to our *aspects*) from raw text.

In 2005, Mann and Yarowsky [17] examined the task of automatically extracting the Birthday, Birthyear, Birthplace, Occupation, and Year-of-Death of a set of famous individuals from biographic web-pages. The authors begin with a biographical training database, and use it to automatically annotate phrases in web documents. These annotated phrases are then used to train a supervised CRF extractor [16] for each field (Birthday, Birthyear etc.). The authors explore various automatic annotation methods, as well as several methods for fusing extracted information from multiple documents.[2] Relevant to our task is the authors' method of "cross-field bootstrapping," in which information about one field is used to influence decisions about others. Their method works as follows: first an extractor is trained for one field (such as Birth*day*). The sentences in the test documents are then annotated with the decisions of this first extractor, and these decisions are used as features for subsequent extractors (for fields such as Birth*year*). The algorithm designer must choose a training order which he or she believes will best allow information from one extraction decision to flow to others.

This method turns out to be effective for the task of biography extraction, probably because writers tend to group very basic biographic information into single summary sentences. Thus knowledge that a biographic sentence mentions a person's birth*day* raises substantially the probability that it will mention his or her birth *year* (as well as birthplace, occupation, and date of death).

As in the work of Renders et. al. discussed in the previous section, this approach allows for *generic* correlations between different extraction tasks to be taken into account. These

[2] As these methods are not relevant to our task – we have a closed set of target labels so have no need for phrase annotation, and exactly one document to consider at a time – we refer the reader to their paper for further details.

correlations are learned as feature weights in the bootstrapped CRF extractors, and these weights either encourage (if positive) or discourage (if negative) multiple-field extractions in a single sentence. In contrast, our approach utilizes a separate label-dependency meta model which can encourage arbitrary relations between multi-aspect label decisions in a manner which is sensitive to the entire range of features of the input. Furthermore, our approach allows for full joint-training of all aspect models as well as the meta model, instead of training each model sequentially using the output of previous models.

A different perspective on Multifield Information Extraction was provided by Wick, Culotta, and McCallum in 2006[31]. They examine the task of collecting contact information from personal web pages into complete database records for each individual. The database fields include: FirstName, LastName, JobTitle, City, State, various phone numbers, Email, and several other categories. The complete record for an individual might not include all fields and may also have a single field repeated with multiple values. Wick et. al. assume they are given text which already has tokens labeled with their true attribute type (e.g. LastName, PhoneNumber, None, etc), and the task is to *partition* the set of field-value pairs into database records, each corresponding to a single individual. The authors propose a method for learning compatibility scores on *sets* of fields and then use agglomerative clustering to produce a complete partition of the field-value pairs. Because the compatibility function is not restricted to pairwise linking decisions, it can examine complex domain-specific features of a proposed record (such as: the number of area codes found in the record). Wick et. al. found that using a global compatibility function with these richer features yielded improvements over simple pairwise compatibility scores.

We view this line of work as complementary to our own. In our work, we assume a fixed set of database fields (aspects) per record (restaurant review) and attempt to jointly extract the field *values* from raw text. Wick et. al.'s paper investigates the reverse problem: the field *values* found in the text are already known, but the *structure* of each record (in terms of the fields it includes) must be predicted. We view this as an important problem in its own right, as often information found in web text is of a fragmentary nature and its scope may not be assumed ahead of time.

2.4 Joint Entity Recognition and Relation Extraction

Roth and Yih [26] tackle the problem of jointly recognizing named entities in text and extracting binary relations expressed in the text between them. For example, in a sentence such as *"Welch was employed by GE"*, we would like to know that Welch is a person, GE is a corporation, and the relation *employer(Welch,GE)* holds between them. The traditional NLP approach would be to first run a named entity recognizer, and then to run a relation extractor on the entities recognized. As Roth and Yih point out, this pipelined approach can lead to problems: If the named entity recognizer predicts that "GE" is a *location*, then the relation extractor can either produce the nonsensical result that the employment relation holds between a *person* and a *location*, or respecting logical type-constraints on the arguments of relations, can predict a relation other than *employment*.

Roth and Yih instead advocate a joint integer linear programming (ILP) approach to this problem. They use previously trained independent classifiers to separately produce probabilities for the named entity decisions and the binary relation decisions. They then seek to maximize the product of probabilities of the two classifiers subject to a prespecified list of logical constraints on the types of relation arguments (e.g., that the *employment* relation must hold between a *person* and an *organization*). This can be formulated as an ILP problem by casting each constraint x as a $\{0, 1\}$ variable which evaluates to 0 when the constraint is satisfied, and to 1 if the constraint is violated. The objective of the linear program is then to minimize the sum of negative log probabilities of the classification decisions, plus the x variables multiplied by an arbitrarily large constant d. Thus, if we represent the two named entity labels respectively as the random variables E_1 and E_2, and we represent the relation between them as R, then the Integer Linear Program (ILP) becomes:

$$\min \left[-\log P(E_1) - \log P(E_2) - \log P(R) + d \left(\sum_i x_i \right) \right]$$

s.t.

$$x_i \in \{0, 1\}, \forall i$$

Since d is set to infinity (or an arbitrarily high number), the solution to this problem is guaranteed to meet all logical constraints, and to otherwise produce the solution which maximizes the product of probabilities of the two classifiers.

Although ILP is in general an NP-hard problem, there are many known heuristics that sometimes produce optimal results quickly. Roth and Yih compared their ILP approach to a baseline pipelined approach and found that existing ILP solvers almost always found the optimal solution quickly. The resulting predictions also proved more accurate than the baseline approach.

This approach obviously has many similarities to the Good Grief framework. In both cases, we seek to maximize the confidence of local prediction models for related tasks while preserving global coherence of the joint prediction. The global coherence in our task, however, is somewhat more subtle than the hard logical constraints imposed by Roth and Yih. We wish to encourage *either* consensus *or* non-consensus to the *degree* that the actual features of the text warrant. In fact, in chapter 5, we experiment with a method which imposes a hard constraint of agreement (ALL AGREE), a model which imposes the decision of the agreement model as a hard constraint (FORCE), as well as a model which promotes a generic agreement bias (GG BIAS). However, none of these methods performs as well as our model, which essentially imposes flexible, *soft* constraints which are sensitive to features of the input. To explain our method in more detail, in the next chapter we provide a formal exposition of the Good Grief framework, as well as training and decoding algorithms.

Input : $(\mathbf{x}^1, y^1), ..., (\mathbf{x}^T, y^T)$

Initialize :

$\left(b_1^1, \ldots, b_{k-1}^1\right) \leftarrow 0$
$b_0^1 \leftarrow -\infty$
$b_k^1 \leftarrow +\infty$
$\mathbf{w}^1 \leftarrow 0$

Loop : For $t = 1, 2, ..., T$:

1. Get a new instance $\mathbf{x}^t \in \mathbb{R}^n$.
2. Predict $\hat{y} = \min_{r \in \{1,..,k\}} \{r : \mathbf{w} \cdot \mathbf{x} - b_r < 0\}$
3. Get a new label y^t.
4. If $(\hat{y}^t = y^t)$ retain model:

$$\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t$$
$$b_r^{t+1} \leftarrow b_r^t, \ \forall r$$

Else update model:

4.a For $r = 1, ..., k-1$:

If $y^t \leq r$: $y_r^t = -1$
else: $y_r^t = 1$

4.b For $r = 1, ..., k-1$:

If $(\hat{y}^t - r)\, y_r^t \leq 0$: $\tau_r^t = y_r^t$
else: $\tau_r^t = 0$

4.c **Update:**

$$\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t + (\textstyle\sum_r \tau_r^t)\mathbf{x}^t$$
$$b_r^{t+1} \leftarrow b_r^t - \tau_r^t, \ \forall r \in 1...k$$

Output : $\mathbf{w}^{T+1}, \mathbf{b}^{T+1}$

Figure 2-1: PRank training algorithm.

Chapter 3

The Algorithm

In this chapter we will formally introduce the Good Grief model for training and decoding with a joint aspect ranking model. The key feature of our idea is the introduction of a meta-model which predicts relations between individual aspects, and in doing so guides the individual rankers towards a globally coherent set of predictions. Although we performed experiments with several meta-models (see chapter 5), for concreteness we will focus our attention here on a meta-model which predicts *agreement* across aspects. We will explicitly note when the discussion applies specifically to agreement models to the exclusion of other meta-models.

The general goal of our algorithm is to find a rank assignment that is consistent with the predictions of individual rankers and the meta-model. To this end, we develop the Good Grief decoding procedure that minimizes the dissatisfaction (*grief*) of individual components with a joint prediction. See Figure 3-1 for a pictorial overview of the Good Grief framework and an example showing how the agreement meta-model can help guide the individual rankers towards a global solution in line with the review.

In this chapter, we will formally define the grief of each component, and a mechanism for its minimization. We then describe two methods for the joint training of individual rankers that takes into account the Good Grief decoding procedure. Finally, we talk about the feature representation used by our model.

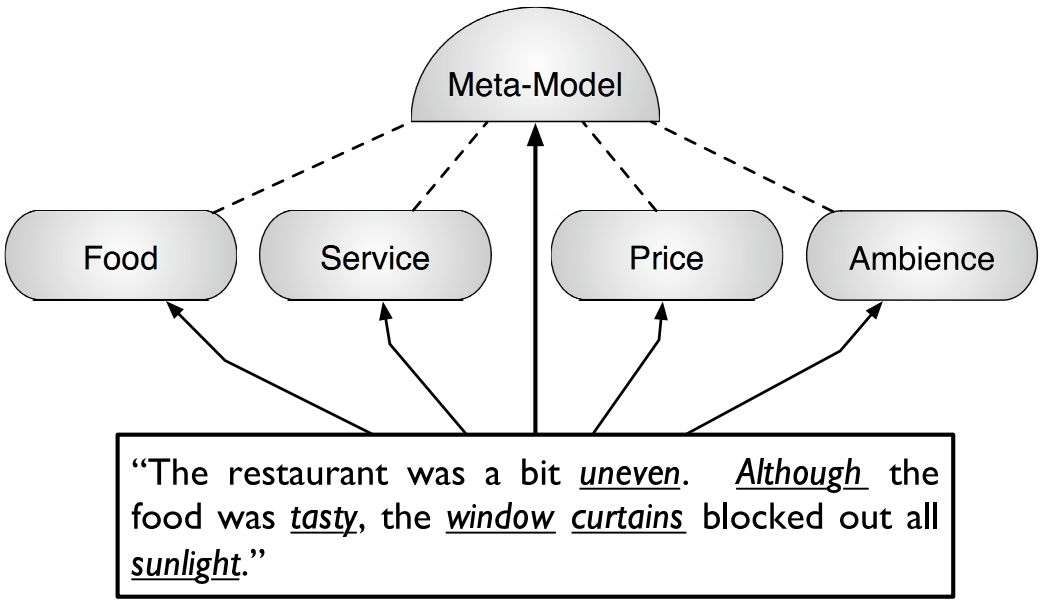

Figure 3-1: Pictorial overview of a Good Grief model, with relevant word-features underlined. In this example, the meta-model will predict *disagreement* with high confidence due to the presence of "uneven" (a content word indicating contrast) and "although" (a discourse word indicating contrast). This prediction should help push the *ambience* model towards a negative prediction.

3.1 Problem Formulation

In an *m-aspect ranking problem*, we are given a training sequence of instance-label pairs $(\mathbf{x}^1, \mathbf{y}^1), ..., (\mathbf{x}^t, \mathbf{y}^t), ...$ Each instance $\mathbf{x}^t$ is a feature vector in $\mathbb{R}^n$ and the label $\mathbf{y}^t$ is a vector of m ranks in $\mathcal{Y}^m$, where $\mathcal{Y} = \{1, .., k\}$ is the set of possible ranks. The i^{th} component of $\mathbf{y}^t$ is the rank for the i^{th} aspect, and will be denoted by $y[i]^t$. The goal is to learn a mapping from instances to rank sets, $H : \mathcal{X} \rightarrow \mathcal{Y}^m$, which minimizes the distance between predicted ranks and true ranks.

3.2 The Model

Our *m-aspect ranking model* contains $m+1$ components: $(\langle \mathbf{w}[1], \mathbf{b}[1] \rangle, ..., \langle \mathbf{w}[m], \mathbf{b}[m] \rangle, \mathbf{a})$. The first m components are individual ranking models, one for each aspect, and the final component is the agreement model, or more generally any meta-model.

3.2.1 Ranking Models

For each aspect $i \in 1...m$, $\mathbf{w}[i] \in \mathbb{R}^n$ is a vector of weights on the input features, and $\mathbf{b}[i] \in \mathbb{R}^{k-1}$ is a vector of boundaries which divide the real line into k intervals, corresponding to the k possible ranks. The default prediction of the aspect ranking model simply uses the ranking rule of the PRank algorithm. This rule predicts the rank r such that $b[i]_{r-1} \leq score_i(\mathbf{x}) < b[i]_r$.[1] The value $score_i(\mathbf{x})$ can be defined simply as the dot product $\mathbf{w}[i] \cdot \mathbf{x}$, or it can take into account the weight vectors for other aspects weighted by a measure of inter-aspect similarity. We adopt the definition given in equation 2.1, replacing the user-specific weight vectors with our aspect-specific weight vectors.

3.2.2 Meta-Model

In general the meta-model can be any model which makes a binary prediction over the set of possible rank-vectors $\mathcal{Y}^m$. More formally, a binary meta-model $\mathbf{a}$ is defined by a partition of the rank-vectors into *positive* and *negative* sets: $\mathbf{a}^+ \cup \mathbf{a}^- = \mathcal{Y}^m$, and a function $score_{\mathbf{a}}(\cdot)$. For an input $\mathbf{x}$, a positive value for $score_{\mathbf{a}}(\mathbf{x})$ indicates a prediction that the associated rank-vector is an element of the positive set $\mathbf{a}^+$, and a negative score indicates a prediction that the rank-vector is in $\mathbf{a}^-$. The absolute value of the score, $|score_{\mathbf{a}}(\mathbf{x})|$, indicates the meta-model's confidence in its prediction.

In the simple case of an agreement model, the meta-model is a vector of weights $\mathbf{a} \in \mathbb{R}^n$. A value of $\mathbf{a} \cdot \mathbf{x} > 0$ predicts that the ranks of all m aspects are equal, and a value of $\mathbf{a} \cdot \mathbf{x} \leq 0$ indicates disagreement. The absolute value $|\mathbf{a} \cdot \mathbf{x}|$ indicates the confidence in the agreement prediction.

Thus, in the terminology of the previous paragraph, the meta-model model defined by

[1] More precisely (taking into account the possibility of ties): $\hat{y}[i] = \min_{r \in \{1,..,k\}}\{r : score_i(\mathbf{x}) - b[i]_r < 0\}$

the agreement model $\mathbf{a}$ is:

$$score_{\mathbf{a}}(\mathbf{x}) = |\mathbf{a} \cdot \mathbf{x}|$$

$$\mathbf{a}^{+} = \{< y[1], \ldots, y[m] > \mid (y[1] = y[2] = \ldots = y[m]) \ \wedge \ (y[i] \in \mathcal{Y}, \forall i)\}$$

$$\mathbf{a}^{-} = \{< y[1], \ldots, y[m] > \mid \neg\,(y[1] = y[2] = \ldots = y[m]) \ \wedge \ (y[i] \in \mathcal{Y}, \forall i)\}$$

3.3 Decoding

The goal of the decoding procedure is to predict a joint rank for the m aspects which satisfies the individual ranking models as well as the global prediction of the meta-model. For a given input $\mathbf{x}$, the individual model for aspect i predicts a default rank $\hat{y}[i]$ based on its feature weight and boundary vectors $\langle \mathbf{w}[i], \mathbf{b}[i] \rangle$. In addition, the agreement model makes a prediction regarding rank consensus based on $\mathbf{a} \cdot \mathbf{x}$. However, the default aspect predictions $\hat{y}[1] \ldots \hat{y}[m]$ may not accord with the agreement model. For example, if $\mathbf{a} \cdot \mathbf{x} > 0$, but $\hat{y}[i] \neq \hat{y}[j]$ for some $i, j \in 1...m$, then the agreement model predicts complete consensus, whereas the individual aspect models do not.

We therefore adopt a joint prediction criterion which simultaneously takes into account *all* model components – individual aspect models as well as the meta-model. For each possible prediction $\mathbf{r} = (r[1], ..., r[m])$ this criterion assesses the level of *grief* associated with the i^{th}-aspect ranking model, $g_i(\mathbf{x}, r[i])$. Similarly, we compute the grief of the meta-model with the joint prediction, $g_{\mathbf{a}}(\mathbf{x}, \mathbf{r})$. Both g_i and $g_{\mathbf{a}}$ are defined formally below, and intuitively indicate the negative-confidence of the models with the specified prediction. The decoder predicts the m ranks which minimize the overall grief:

$$H(\mathbf{x}) = \arg\min_{\mathbf{r} \in \mathcal{Y}^m} \left[g_{\mathbf{a}}(\mathbf{x}, \mathbf{r}) + \sum_{i=1}^{m} g_i(\mathbf{x}, r[i]) \right] \tag{3.1}$$

If the default rank predictions for the aspect models, $\hat{\mathbf{y}} = (\hat{y}[1], ..., \hat{y}[m])$, are in accord with the agreement model (both indicating consensus or both indicating contrast), then the grief of all model components will be zero, and we simply output $\hat{\mathbf{y}}$. On the other hand, if $\hat{\mathbf{y}}$ indicates disagreement but the agreement model predicts consensus, then we have the

option of predicting $\hat{\mathbf{y}}$ and bearing the grief of the agreement model. Alternatively, we can predict some consensus $\mathbf{y}'$ (i.e. with $y'[i] = y'[j], \forall i, j$) and bear the grief of the component ranking models. The decoder H chooses the option with lowest overall grief.[2]

Now we formally define the measures of *grief* used in this criterion.

3.3.1 Aspect Model Grief

We define the grief of the i^{th}-aspect ranking model with respect to a rank r to be the smallest magnitude correction term which places the input's score into the r^{th} segment of the real line:

$$g_i(\mathbf{x}, r) = \min |c|$$

s.t.

$$b[i]_{r-1} \leq score_i(\mathbf{x}) + c < b[i]_r$$

3.3.2 Agreement Model Grief

Similarly, we define the grief of the agreement model with respect to a joint rank $\mathbf{r} = (r[1], \ldots, r[m])$ as the smallest correction needed to bring the agreement score into accord with the agreement relation between the individual ranks $r[1], \ldots, r[m]$:

$$g_{\mathbf{a}}(\mathbf{x}, \mathbf{r}) = \min |c|$$

s.t.

$$\mathbf{a} \cdot \mathbf{x} + c > 0 \ \wedge \ \forall i, j \in 1...m : r[i] = r[j]$$

$$\vee$$

$$\mathbf{a} \cdot \mathbf{x} + c \leq \ \wedge \exists i, j \in 1...m : r[i] \neq r[j]$$

[2]This decoding criterion assumes that the griefs of the component models are comparable. In practice, we take an uncalibrated agreement model $\mathbf{a}'$ and re-weight it with a tuning parameter: $\mathbf{a} = \alpha\mathbf{a}'$. The value of α is estimated using a development set. We assume that the griefs of the ranking models are comparable since they are jointly trained.

More generally, for an arbitrary meta-model $\mathbf{a}$, we define the grief to be:

$$g_{\mathbf{a}}(\mathbf{x}, \mathbf{r}) = \min |c|$$

s.t.

$$score_{\mathbf{a}}(\mathbf{x}) + c > 0 \ \wedge \ \mathbf{r} \in \mathbf{a}^{+}$$

$$\vee$$

$$score_{\mathbf{a}}(\mathbf{x}) + c \leq 0 \ \wedge \ \mathbf{r} \in \mathbf{a}^{-}$$

3.4 Training

Pseudo-code for the two variants of Good Grief training are shown in Figure 3-2 and Figure 3-3. Both of these training algorithms are based on PRanking [7], an online perceptron algorithm. The training is performed by iteratively ranking each training input $\mathbf{x}$ and updating the model. If the predicted rank $\hat{y}$ is equal to the true rank y, the weight and boundaries vectors remain unchanged. On the other hand, if $\hat{y} \neq y$, then the weights and boundaries are updated to improve the prediction for $\mathbf{x}$ (step 4.c in Figures 3-2 and 3-3). See Chapter 5 for pseudo-code in the case of a single ranking model. For further explanation and analysis of the update rule in the case of an individual ranking model, see [7], and see chapter 4 for a theoretical mistake-bound analysis for the first Good Grief variant.

Our algorithms depart from PRanking by conjoining the updates for the m ranking models. We achieve this by using Good Grief decoding at each step throughout training. Our decoder $H(\mathbf{x})$ (from equation 3.1) uses *all* the aspect component models as well as the agreement model to determine the predicted rank for each aspect.

3.4.1 Variant 1

First we consider Variant 1 of Good Grief Decoding (Figure 3-2) in more detail. In this version, the meta-model $\mathbf{a}$ is assumed to have been trained ahead of time and is given as input to the Good Grief training algorithm. We then start by **initializing** the boundary

and weight vectors for each aspect ranking model to zero vectors.[3] We then **loop** through the training corpus some number of times. For each instance $\mathbf{x}$, we predict the ranks of all aspects simultaneously (step 2 in Figure 3-2) using the pre-trained agreement model and the current state of the m aspect-specific ranking models. Then, for each aspect we make a separate **update** based on this joint prediction (step 4 in Figure 3-2). Finally, after convergence (or more practically after several runs through the training corpus to avoid over-fitting), the ranking models are outputted and can be used along with the pre-trained agreement to perform Good Grief decoding on unseen test data.

The disadvantage of this variant is that the agreement model is trained without considering the role it will ultimately play in Good Grief decoding. However, this can also free the model designer to utilize more complex batch methods to train the meta-model, such as SVM optimization [2] or boosting [12].

3.4.2 Variant 2

This variant of Good Grief training (Figure 3-3) differs from the first in that the meta-model is trained jointly with the aspect ranking models using perceptron updates. Thus, the meta-model is initialized to the zero vector along with the ranking models, and the Good Grief decoding (step 2) uses the current state of the ranking models as well as the meta-model. After the updates to the ranking models are performed in step 4, an additional step 5 is taken to update the meta-model. Note that instead of using the direct output of the meta-model ($score_{\mathbf{a}}(\mathbf{x})$) to provide feedback, this algorithm instead uses the prediction that results from the *entire* joint model using Good Grief Decoding. Thus, the meta-model is trained to specifically operate within the Good Grief framework.

[3] with the exception of the lowest and highest boundary points, which are set to $-\infty$ and $+\infty$ respectively.

	Average Feature Count	Standard Deviation
Training Corpus	80.47	78.59
Development Corpus	81.69	83.09
Test Corpus	85.56	85.02

Table 3.1: Average number of features found per restaurant review.

3.5 Feature Representation

3.5.1 Ranking Models

Following previous work on sentiment classification [22], we represent each review as a binary vector of lexical features. More specifically, we extract all unigrams and bigrams from the review, discarding those that appear fewer than three times. This process yields about 30,000 total features when applied to the restaurant review corpus (described in more detail in Chapter 5). See Table 3.1 for statistics on how many features on average are active per review. As can be seen, the feature vectors are almost all relatively sparse. Note that we made no effort to perform feature selection, or otherwise filter features for particular aspects. Thus, the presence of (presumably) aspect-specific words such as "expensive" and "tasty" will appear as a features for all rankers, as will (presumably) neutral words such as "restaurant" and "check." We leave it to the training algorithms to assign appropriate weights.

3.5.2 Agreement Model

The agreement model also operates over lexicalized features. The effectiveness of these features for recognition of discourse relations has been previously shown by Marcu and Echihabi [18]. In addition to unigrams and bigrams, we also introduce a feature that measures the maximum contrastive distance between pairs of words in a review. For example, the presence of *"delicious"* and *"dirty"* indicate high contrast, whereas the pair *"expensive"* and *"slow"* indicate low contrast. The contrastive distance for a pair of words is computed by considering the difference in relative weight assigned to the words in individually trained PRanking models.

In the next chapter we turn to a theoretical analysis of the framework proposed here.

We will examine the expressive power of the Good Grief framework with an agreement meta-model. We will also provide a mistake-bound analysis for the first variant of our training algorithm.

Input : $(\mathbf{x}^1, \mathbf{y}^1), ..., (\mathbf{x}^T, \mathbf{y}^T)$, Meta-Model $\mathbf{a}$

Initialize : For aspect $i = 1...m$:

$\left(b[i]_1^1, \ldots, b[i]_{k-1}^1\right) \leftarrow 0$
$b[i]_0^1 \leftarrow -\infty$
$b[i]_k^1 \leftarrow +\infty$
$\mathbf{w}[i]^1 \leftarrow 0$

Loop : For $t = 1, 2, ..., T$:

1. Get a new instance $\mathbf{x}^t \in \mathbb{R}^n$.
2. Predict $\boxed{\hat{\mathbf{y}}^t = H(\mathbf{x}; \mathbf{w}^t, \mathbf{b}^t, \mathbf{a})}$ (Equation 3.1).
3. Get a new label $\mathbf{y}^t$.
4. For aspect $i = 1, ..., m$:

If $(\hat{y}[i]^t = y[i]^t)$ retain model:

$\mathbf{w}[i]^{t+1} \leftarrow \mathbf{w}[i]^t$
$b[i]_r^{t+1} \leftarrow b[i]_r^t, \ \forall r$

Else update model:

4.a For $r = 1, ..., k-1$:
If $y[i]^t \leq r$: $y[i]_r^t = -1$
else: $y[i]_r^t = 1$

4.b For $r = 1, ..., k-1$:
If $(\hat{y}[i]^t - r)\, y[i]_r^t \leq 0$: $\tau[i]_r^t = y[i]_r^t$
else: $\tau[i]_r^t = 0$

4.c **Update:**
$\mathbf{w}[i]^{t+1} \leftarrow \mathbf{w}[i]^t + (\sum_r \tau[i]_r^t)\mathbf{x}^t$
$b[i]_r^{t+1} \leftarrow b[i]_r^t - \tau[i]_r^t, \ \forall r \in 1...k$

Output : $H(\ \cdot\ ; \mathbf{w}^{T+1}, \mathbf{b}^{T+1}, \mathbf{a})$.

Figure 3-2: **Variant 1** of Good Grief Training. This algorithm is based on the PRanking training algorithm. It differs in the joint computation of all aspect predictions $\hat{\mathbf{y}}^t$ based on the Good Grief Criterion (step 2) and the calculation of updates for each aspect based on the joint prediction (step 4). The meta-model $\mathbf{a}$ is assumed to be pre-trained.

Input : $(\mathbf{x}^1, \mathbf{y}^1), ..., (\mathbf{x}^T, \mathbf{y}^T)$

Initialize : For aspect $i = 1...m$:

$\left(b[i]_1^1, \ldots, b[i]_{k-1}^1\right) \leftarrow 0$
$b[i]_0^1 \leftarrow -\infty$
$b[i]_k^1 \leftarrow +\infty$
$\mathbf{w}[i]^1 \leftarrow 0$
$\mathbf{a}^1 \leftarrow 0$

Loop : For $t = 1, 2, ..., T$:

1. Get a new instance $\mathbf{x}^t \in \mathbb{R}^n$.
2. Predict $\boxed{\hat{\mathbf{y}}^t = H(\mathbf{x}; \mathbf{w}^t, \mathbf{b}^t, \mathbf{a}^t)}$ (Equation 3.1).
3. Get a new label $\mathbf{y}^t$.
4. For aspect $i = 1, ..., m$:

   If $(\hat{y}[i]^t = y[i]^t)$ retain model:

   $\mathbf{w}[i]^{t+1} \leftarrow \mathbf{w}[i]^t$
   $b[i]_r^{t+1} \leftarrow b[i]_r^t, \ \forall r$

   Else update model:

   4.a For $r = 1, ..., k-1$:

   If $y[i]^t \leq r$: $y[i]_r^t = -1$
   else: $y[i]_r^t = 1$

   4.b For $r = 1, ..., k-1$:

   If $(\hat{y}[i]^t - r)\, y[i]_r^t \leq 0$: $\tau[i]_r^t = y[i]_r^t$
   else: $\tau[i]_r^t = 0$

   4.c **Update:**

   $\mathbf{w}[i]^{t+1} \leftarrow \mathbf{w}[i]^t + (\sum_r \tau[i]_r^t)\mathbf{x}^t$
   $b[i]_r^{t+1} \leftarrow b[i]_r^t - \tau[i]_r^t, \ \forall r \in 1...k$

5. If $(\mathbf{y} \in \mathbf{a}^+ \wedge \hat{\mathbf{y}} \in \mathbf{a}^+) \vee (\mathbf{y} \in \mathbf{a}^- \wedge \hat{\mathbf{y}} \in \mathbf{a}^-)$ retain meta-model:

   $\mathbf{a}^{t+1} \leftarrow \mathbf{a}^t$

   Else update meta-model:

   If $y^t \in \mathbf{a}^+$: $\mathbf{a}^{t+1} \leftarrow \mathbf{a}^t + \mathbf{x}^t$
   Else: $\mathbf{a}^{t+1} \leftarrow \mathbf{a}^t - \mathbf{x}^t$

Output : $H(\ \cdot\ ; \mathbf{w}^{T+1}, \mathbf{b}^{T+1}, \mathbf{a}^{T+1})$.

Figure 3-3: **Variant 2** of Good Grief Training. In this variant, the meta-model is trained jointly with the component ranking models, using the output from Good Grief decoding (Step 2) to provide feedback for perceptron updates (Step 5).

Chapter 4

Analysis

In this chapter we provide two theoretical analyses of our framework. First we demonstrate that the agreement-based joint model is more expressive than individual ranking models. That is, every training corpus that can be perfectly ranked by individual ranking models for each aspect can also be perfectly ranked with our joint model. In addition, we give a simple example of a training set which cannot be perfectly ranked without agreement-based joint inference, demonstrating the increase in expressive power.

We also provide a general mistake bound analysis for the Good Grief framework which applies to any meta-model. We show that even with the potential increase in expressive power, Good Grief Decoding preserves the finite mistake bound of simple Perceptron training by allowing the meta-model to be "drowned out" when it proves a hindrance during training.

4.1 Expressivity

In this section, we prove that our model is able to perfectly rank a strict superset of the training corpora perfectly rankable by m ranking models individually. We first show that if the independent ranking models can individually rank a training set perfectly, then our model can do so as well. Next, we show that our model is more expressive by providing a simple illustrative example of a training set which can only be perfectly ranked with the inclusion of an agreement model.

First we introduce some notation. For each training instance $(\mathbf{x}^t, \mathbf{y}^t)$, each aspect $i \in 1...m$, and each rank $r \in 1...k$, define an auxiliary variable $y[i]^t_r$ with $y[i]^t_r = -1$ if $y[i]^t \leq r$ and $y[i]^t_r = 1$ if $y[i]^t > r$. In words, $y[i]^t_r$ indicates whether the *true* rank $y[i]^t$ is to the right or left of a *potential* rank r.

Now suppose that a training set $(\mathbf{x}^1, \mathbf{y}^1), ..., (\mathbf{x}^T, \mathbf{y}^T)$ is perfectly rankable for each aspect independently. That is, for each aspect $i \in 1...m$, there exists some ideal model $v[i]^* = (w[i]^*, b[i]^*)$ such that the signed distance from the prediction to the r^{th} boundary: $\mathbf{w}[i]^* \cdot \mathbf{x}^t - b[i]^*_r$ has the same sign as the auxiliary variable $y[i]^t_r$. In other words, the minimum margin over all training instances and ranks, $\gamma = \min_{r,t}\{(\mathbf{w}[i]^* \cdot \mathbf{x}^t - b[i]^*_r) y[i]^t_r\}$, is no less than zero.

Now for the t^{th} training instance, define an agreement auxiliary variable a^t, where $a^t = 1$ when all aspects agree in rank and $a^t = -1$ when at least two aspects disagree in rank. First consider the case where the agreement model $\mathbf{a}$ perfectly classifies all training instances: $(\mathbf{a} \cdot \mathbf{x}^t)a^t > 0, \forall t$. It is clear that Good Grief decoding with the ideal joint model $(\langle \mathbf{w}[1]^*, \mathbf{b}[1]^* \rangle, ..., \langle \mathbf{w}[m]^*, \mathbf{b}[m]^* \rangle, \mathbf{a})$ will produce the same output as the component ranking models run separately (since the grief will always be zero for the default rank predictions). Now consider the case where the training data is not linearly separable with regard to agreement classification. Define the margin of the worst case error to be $\beta = \max_t\{|(\mathbf{a} \cdot \mathbf{x}^t)| : (\mathbf{a} \cdot \mathbf{x}^t)a^t < 0\}$. If $\beta < \gamma$, then again Good Grief decoding will always produce the default results (since the grief of the agreement model will be at most β in cases of error, whereas the grief of the ranking models for any deviation from their default predictions will be at least γ). On the other hand, if $\beta \geq \gamma$, then the agreement model errors could potentially disrupt the perfect ranking. However, we need only re-scale $w^* := w^*(\frac{\beta}{\gamma} + \epsilon)$ and $b^* := b^*(\frac{\beta}{\gamma} + \epsilon)$ to ensure that the grief of the ranking models will always exceed the grief of the agreement model in cases where the latter is in error. Thus whenever independent ranking models can perfectly rank a training set, a joint ranking model with Good Grief decoding can do so as well.

Now we give a simple example of a training set which can only be perfectly ranked with the addition of an agreement model. Consider a training set of four instances with two rank aspects:

$$\langle \mathbf{x}^1, \mathbf{y}^1 \rangle = \langle (1,0,1), \quad (2,1) \rangle$$
$$\langle \mathbf{x}^2, \mathbf{y}^2 \rangle = \langle (1,0,0), \quad (2,2) \rangle$$
$$\langle \mathbf{x}^3, \mathbf{y}^3 \rangle = \langle (0,1,1), \quad (1,2) \rangle$$
$$\langle \mathbf{x}^4, \mathbf{y}^4 \rangle = \langle (0,1,0), \quad (1,1) \rangle$$

We can interpret these inputs as feature vectors corresponding to the presence of "good", "bad", and "but not" in the following four sentences:

The food was **good**, **but not** the ambience.

The food was **good**, and so was the ambience.

The food was **bad**, **but not** the ambience.

The food was **bad**, and so was the ambience.

We can further interpret the first rank aspect as the quality of food, and the second as the quality of the ambience, both on a scale of 1-2.

A simple ranking model which only considers the words "good" and "bad" perfectly ranks the food aspect. However, it is easy to see that no single model perfectly ranks the ambience aspect. Consider any model $\langle \mathbf{w}, \mathbf{b} = (b) \rangle$. Note that $\mathbf{w} \cdot \mathbf{x}^1 < b$ and $\mathbf{w} \cdot \mathbf{x}^2 \geq b$ together imply that $w_3 < 0$, whereas $\mathbf{w} \cdot \mathbf{x}^3 \geq b$ and $\mathbf{w} \cdot \mathbf{x}^4 < b$ together imply that $w_3 > 0$. Thus independent ranking models cannot perfectly rank this corpus.

The addition of an agreement model, however, can easily yield a perfect ranking. With $\mathbf{a} = (0, 0, -5)$ (which predicts contrast with the presence of the words "but not") and a ranking model for the ambience aspect such as $\mathbf{w} = (1, -1, 0), \mathbf{b} = (0)$, the Good Grief decoder will produce a perfect rank.

Finally, we note that a similar increase in expressivity can result from expanding the input space to include conjunctions of features (through e.g. a polynomial kernel in the SVM framework). However, we show in chapter 5 that simply using all binary conjunctions of features (the SVM^2 model) actually *degrades* performance. As is often the case, increased model power can lead to over-fitting problems unless carefully crafted and controlled.

4.2 Mistake Bound

Novikoff first proved the convergence of the binary classification perceptron algorithm in 1962 [20]. He showed that if a corpus of radius R is linearly separable with margin γ, then the perceptron algorithm will perform at most $\left(\frac{2R}{\gamma}\right)^2$ updates. The significance of this bound is that the number of updates nowhere depends on the size of the training set itself. Thus, if the perceptron is allowed to iterate over the training corpus indefinitely, only a finite number of mistakes and updates will be made. At some point the perceptron will converge to a solution which perfectly classifies the training set.

Crammer and Singer extend this convergence proof to the case of perceptron ranking [7]. They show that if a corpus with k ranks of radius R is perfectly rankable with margin γ, then the perceptron ranker will incur a ranking loss of at most $\frac{(k-1)(R^2+1)}{\gamma^2}$ during training. Ranking loss is defined as the total distance between true ranks and predicted ranks and *a fortiori* the perceptron ranker will make no more than this number of mistakes and updates.

We spend the remainder of this section providing a convergence proof for our joint Good Grief training algorithm. In particular, we show that if a corpus is perfectly rankable using independent perceptron rankers, then even when using the more expressive Good Grief model during training, convergence is guaranteed. In fact, this proof nowhere assumes anything about the properties or efficacy of the meta-model, other than that it is fixed and finite. In essence we show that the ranking models will eventually "drown out" the meta-model when it proves to be a hindrance. Before we proceed to the proof, we must lay out our notation and definitions.

4.2.1 Definitions and Notation

Given a joint ranking model consisting of a set of ranking models: $\mathbf{v} = (\mathbf{w}[i], \mathbf{b}[i], \ldots, \mathbf{w}[m], \mathbf{b}[m])$, and a meta-model $\mathbf{a}$, we define the *correction term*[1] for aspect i with respect to an input

[1] Similar to definitions of "grief" given in the previous chapter, but without the absolute value taken.

$\mathbf{x} \in \mathbb{R}^n$ and a joint rank $\mathbf{r} \in \{1, \ldots, k\}^m$ to be:

$$c_i(\mathbf{x}, \mathbf{r};\ \mathbf{v}) = \arg\min_c |c|$$

s.t.

$$b[i]_{r[i]-1} \leq \mathbf{w}[i] \cdot \mathbf{x} + c < b[i]_{r[i]}$$

We further define the *correction term* for the meta-model to be:

$$c_{\mathbf{a}}(\mathbf{x}, \mathbf{r};\ \mathbf{a}) = \arg\min_c |c|$$

s.t.

$$\mathbf{a} \cdot \mathbf{x} + c > 0 \wedge \mathbf{r} \in \mathbf{a}^+$$

$$\vee$$

$$\mathbf{a} \cdot \mathbf{x} + c \leq 0 \wedge \mathbf{r} \in \mathbf{a}^-$$

We define the output of a Good Grief model $\mathbf{v}$ to be the joint rank which minimizes the magnitude of these correction terms:

$$H(\mathbf{x};\ \mathbf{v}, \mathbf{a}) = \arg\min_{\mathbf{r}} \left[|c_{\mathbf{a}}(\mathbf{x}, \mathbf{r};\ \mathbf{a})| + \sum_{i=1}^{m} |c_i(\mathbf{x}, \mathbf{r};\ \mathbf{v})| \right] \tag{4.1}$$

Note that this formulation differs slightly from that given in chapter 3, equation 3.1, but is equivalent.

We adopt the following notation for the training of a model. The initial model is denoted by $\mathbf{v}^1 = (\mathbf{w}[1]^1, \mathbf{b}[1]^1, \ldots, \mathbf{w}[m]^1, \mathbf{b}[m]^1) = 0$. The model obtained after all the updates for the t^{th} input will be denoted by $\mathbf{v}^{t+1}$. The model obtained after the update for the i^{th} aspect of the t^{th} input will be denoted by $\mathbf{v}^{t:[i+1]}$. We denote the prediction made for the i^{th} aspect of the t^{th} input during training by

$$\hat{y}[i]^t \triangleq H(\mathbf{x}^t;\ \mathbf{v}^{t:[i]}, \mathbf{a})$$

and define the incurred rank loss as

$$n[i]^t \triangleq |\hat{y}[i]^t - y^t[i]|.$$

The total rank loss for the example is then written as $n^t = \sum_i n[i]^t$. See Figure 3-2 in chapter 3 for the training algorithm that we analyze here. Notice in particular the directional variables $y[i]_r^t \in \{+1, -1\}$, which indicate whether the true rank for instance t and aspect i is equal to or lower than r or whether it is higher than r. Notice also the indicator variables $\tau[i]_r^t \in \{-1, 0, +1\}$ which evaluate to the directional variable $y[i]_r^t$ when r lies between the predicted rank and the true rank, and otherwise evaluates to 0. It is easy to confirm that $n[i]^t = \sum_r |\tau[i]_r^t|$ and thus that $n^t = \sum_{i,r} |\tau[i]_r^t|$.

In this training scenario, we also use an abbreviated notation for the correction terms used during training:

$$c[i]^t \triangleq c_i \left(\mathbf{x}^t, \hat{\mathbf{y}}^t;\ \mathbf{v}^{t:[i]}\right),\ \forall i$$
$$c_a^t \triangleq c_{\mathbf{a}} \left(\mathbf{x}^t, \hat{\mathbf{y}}^t;\ \mathbf{a}\right)$$

Finally, when dealing with some arbitrary model $\mathbf{v}^* = (\mathbf{w}[i]^*, \mathbf{b}[i]^*, \ldots, \mathbf{w}[m]^*, \mathbf{b}[m]^*)$, we adopt the notation:

$$c[i]^{*,t} \triangleq c_i \left(\mathbf{x}^t, H(\mathbf{x}^t; \mathbf{v}^*, \mathbf{a});\ \mathbf{v}^*\right),\ \forall i$$
$$c_a^{*,t} \triangleq c_{\mathbf{a}} \left(\mathbf{x}^t, H(\mathbf{x}^t; \mathbf{v}^*, \mathbf{a});\ \mathbf{a}\right)$$

We no proceed to the theorem and proof.

4.2.2 Mistake Bound Theorem

Mistake Bound. *Let $(\mathbf{x}^1, \mathbf{y}^1), \ldots, (\mathbf{x}^T, \mathbf{y}^T)$ be a training sequence with each $\mathbf{x}^t \in \mathbb{R}^n$ and each $\mathbf{y}^t \in \{1, \ldots, k\}^m$, and let $\mathbf{a} \in \mathbb{R}^n$ be the meta-model. Define the radius of the sequence $R = \max_t \|\mathbf{x}^t\|$. Assume a unit norm joint ranking model*
$\mathbf{v}^* = (\mathbf{w}[i]^*, \mathbf{b}[i]^*, \ldots, \mathbf{w}[m]^*, \mathbf{b}[m]^*)$ which perfectly ranks the training sequence with

margin $\gamma = \min_{r,t,i}\{(\mathbf{w}[i]^* \cdot \mathbf{x}^t - b[i]^*_r)y^t_r\} \geq 0$. *Then the rank loss incurred during Good Grief training* $\sum_{t,i} |\hat{y}[i]^t - y[i]^t|$ *is at most*

$$\frac{(k-1)(R^2+1) + 2kR\|\mathbf{a}\|}{\gamma^2}.$$

Proof. We prove the mistake bound by bounding the rank loss $\sum_t n^t$ from above. To do so, we first bound the norm of the trained model $\|\mathbf{v}^{T+1}\|$ from above *and* below.

Lemma 1. *A lower bound on the norm of the trained model is given by*

$$\|\mathbf{v}^{T+1}\| \geq \gamma \sum_t n^t - \sum_{t,i} n[i]^t \left|c[i]^{*,t}\right|.$$

Proof of Lemma. Consider the example $(\mathbf{x}^t, \mathbf{y}^t)$ received at round t during training. The aspects for this example are jointly ranked by the model $\mathbf{v}^t$ and the models obtained before and after the update for the i^{th} aspect are $\mathbf{v}^{t:[i]}$ and $\mathbf{v}^{t:[i+1]}$ respectively. Now, by the definition of the update rule and then the definition of $\tau[i]^t_r$, we have:

$$\mathbf{v}^* \cdot \mathbf{v}^{t:[i+1]} = \mathbf{v}^* \cdot \mathbf{v}^{t:[i]} + \sum_r \tau[i]^t_r \left(\mathbf{w}[i]^* \cdot \mathbf{x}^t - b[i]^*_r\right) \tag{4.2}$$

$$= \mathbf{v}^* \cdot \mathbf{v}^{t:[i]} + \sum_r \left|\tau[i]^t_r\right| y[i]^t_r \left(\mathbf{w}[i]^* \cdot \mathbf{x}^t - b[i]^*_r\right) \tag{4.3}$$

Now, by hypothesis, the margin of $\mathbf{v}^*$ is at least γ, so

$$y[i]^t_r \left(\mathbf{w}[i]^* \cdot \mathbf{x}^t + c[i]^{*,t} - b[i]^*_r\right) \;\geq\; \gamma \tag{4.4}$$

$$\Rightarrow \quad y[i]^t_r \left(\mathbf{w}[i]^* \cdot \mathbf{x}^t - b[i]^*_r\right) \;\geq\; \gamma - \left(y[i]^t_r \; c[i]^{*,t}\right) \tag{4.5}$$

Plugging this into equation 4.3, we get

$$\mathbf{v}^* \cdot \mathbf{v}^{t:[i+1]} \geq \mathbf{v}^* \cdot \mathbf{v}^{t:[i]} + \sum_r \left|\tau[i]^t_r\right| \left(\gamma - y[i]^t_r \; c[i]^{*,t}\right) \tag{4.6}$$

$$\geq \mathbf{v}^* \cdot \mathbf{v}^{t:[i]} + n[i]^t \left(\gamma - \left|c[i]^{*,t}\right|\right) \tag{4.7}$$

Applying this inequality repeatedly for each aspect, we get

$$\mathbf{v}^* \cdot \mathbf{v}^{t+1} \geq \mathbf{v}^* \cdot \mathbf{v}^t + n^t\gamma - \sum_i n[i]^t \left|c[i]^{*,t}\right| \tag{4.8}$$

Applying this inequality recursively for all training examples, and by the fact that $\mathbf{v}^1 = 0$, we get

$$\mathbf{v}^* \cdot \mathbf{v}^{T+1} \geq \gamma \sum_t n^t - \sum_{t,i} n[i]^t \left|c[i]^{*,t}\right| \tag{4.9}$$

Finally, by the Cauchy-Schwartz inequality and the fact that $\mathbf{v}^*$ has unit norm we get

$$\|\mathbf{v}^{T+1}\| \geq \gamma \sum_t n^t - \sum_{t,i} n[i]^t \left|c[i]^{*,t}\right| \tag{4.10}$$

□

Lemma 2. *An upper bound on the norm of the trained model is given by*

$$\|\mathbf{v}^{T+1}\|^2 \leq 2\sum_{t,i} n[i]^t \left|c[i]^t\right| + R^2 \sum_t \left(n^t\right)^2 + \sum_t n^t.$$

Proof. Again consider an example $(\mathbf{x}^t, \mathbf{y}^t)$ received at round t during training. The aspects for this example are jointly ranked by the model $\mathbf{v}^t$ and the model obtained after the updates

for all the aspects is $\mathbf{v}^{t+1}$. When we take the square norm of $\mathbf{v}^{t+1}$, we get

$$\begin{aligned}
\left\|\mathbf{v}^{t+1}\right\|^2 &= \left\|\left(\mathbf{w}[1]^{t+1}, \mathbf{b}[1]^{t+1}, \ldots, \mathbf{w}[m]^{t+1}, \mathbf{b}[m]^{t+1}\right)\right\|^2 \\
&= \Big\| \Big(\mathbf{w}[1]^t + \big(\sum_r \tau[1]_r^t\big)\mathbf{x}^t,\ b[1]_1^t - \tau[1]_1^t, \ldots,\ b[1]_{k-1}^t - \tau[1]_{k-1}^t, \\
&\qquad \mathbf{w}[m]^t + \big(\sum_r \tau[m]_r^t\big)\mathbf{x}^t,\ b[m]_1^t - \tau[m]_1^t, \ldots,\ b[m]_{k-1}^t - \tau[m]_{k-1}^t \Big) \Big\|^2 \\
&= \sum_i \left\|\mathbf{w}[i]^t\right\|^2 \ + \ \sum_i \left\|\mathbf{b}[i]^t\right\|^2 \ + \ 2\sum_{i,r} \tau[i]_r^t \left(\mathbf{w}[i]^t \cdot \mathbf{x}^t - b[i]_r^t\right) \\
&\qquad\qquad + \ \big(\sum_{i,r} \tau[i]_r^t\big)^2 \|\mathbf{x}^t\|^2 \ + \ \sum_{i,r} \big(\tau[i]_r^t\big)^2 \\
&\leq \|\mathbf{v}^t\|^2 + 2\sum_{i,r} \tau[i]_r^t \left(\mathbf{w}[i]^t \cdot \mathbf{x}^t - b[i]_r^t\right) + R^2 \left(n^t\right)^2 + n^t
\end{aligned} \tag{4.11}$$

We now use the definition of $\tau[i]_r^t$ to upper bound the second term. We will define the *indicator function* of a predicate p to be $[\![p]\!] = 1$ if p, and $[\![p]\!] = 0$ if $\neg p$. Using this notation along with the definition of $\tau[i]_r^t$, we can rewrite $\sum_{i,r} \tau[i]_r^t \left(\mathbf{w}[i]^t \cdot \mathbf{x}^t - b[i]_r^t\right)$ as:

$$\begin{aligned}
&\sum_{i,r} \left[\!\!\left[y[i]_r^t \left(\mathbf{w}[i]^t \cdot \mathbf{x}^t + c[i]^t - b[i]_r^t\right) \leq 0 \right]\!\!\right] y[i]_r^t \left(\mathbf{w}[i]^t \cdot \mathbf{x}^t - b[i]_r^t\right) \\
&= \sum_{i,r} \left[\!\!\left[y[i]_r^t \left(\mathbf{w}[i]^t \cdot \mathbf{x}^t - b[i]_r^t\right) \leq -y[i]_r^t c[i]^t \right]\!\!\right] y[i]_r^t \left(\mathbf{w}[i]^t \cdot \mathbf{x}^t - b[i]_r^t\right) \\
&\leq \sum_{i,r} \left[\!\!\left[y[i]_r^t \left(\mathbf{w}[i]^t \cdot \mathbf{x}^t - b[i]_r^t\right) \leq \left|c[i]^t\right| \right]\!\!\right] \left|c[i]^t\right| \\
&= \sum_{i,r} n[i]^t \left|c[i]^t\right|
\end{aligned}$$

Plugging this back into 4.11, we get

$$\left\|\mathbf{v}^{t+1}\right\|^2 \leq \|\mathbf{v}^t\|^2 + 2\sum_i n[i]^t \left|c[i]^t\right| + R^2 \left(n^t\right)^2 + n^t.$$

Finally, by applying this inequality inductively along with the fact that $\mathbf{v}^1 = 0$, we get our

lemma:

$$\|\mathbf{v}^{T+1}\|^2 \leq 2\sum_{t,i} n[i]^t \left|c[i]^t\right| + R^2 \sum_t \left(n^t\right)^2 + \sum_t n^t.$$

□

Combining Lemmas 1 and 2, we obtain the following inequality:

$$\begin{aligned}
&\gamma^2\big(\sum_t n^t\big)^2 - 2\gamma\big(\sum_t n^t\big)\big(\sum_{t,i} n[i]^t \left|c[i]^{*,t}\right|\big) + \big(\sum_{t,i} n[i]^t \left|c[i]^{*,t}\right|\big)^2 \\
&\leq 2\sum_{t,i} n[i]^t \left|c[i]^t\right| + R^2 \sum_t \left(n^t\right)^2 + \sum_t n^t.
\end{aligned}$$

Isolating the first term and dividing both sides by $\gamma^2 \sum_t n^t$ we get:

$$\begin{aligned}
\sum_t n^t &\leq \frac{2\sum_{t,i} n[i]^t \left|c[i]^t\right|}{\gamma^2 \sum_t n^t} + \frac{R^2 \sum_t \left(n^t\right)^2 + 1}{\gamma^2 \sum_t n^t} + \frac{\left(\sum_{t,i} n[i]^t \left|c[i]^{*,t}\right|\right)^2}{\gamma^2 \sum_t n^t} - \frac{\sum_{t,i} n[i]^t \left|c[i]^{*,t}\right|}{\gamma} \\
&\leq \frac{2\sum_{t,i} n[i]^t \left|c[i]^t\right|}{\gamma^2 \sum_t n^t} + \frac{(k-1)R^2 + 1}{\gamma^2} + \frac{\left(\sum_{t,i} n[i]^t \left|c[i]^{*,t}\right|\right)^2}{\gamma^2 \sum_t n^t} - \frac{\sum_{t,i} n[i]^t \left|c[i]^{*,t}\right|}{\gamma} \\
&\leq \frac{2R\left\|\mathbf{a}\right\|}{\gamma^2} + \frac{(k-1)R^2 + 1}{\gamma}^2 + \frac{\left(\sum_{t,i} n[i]^t \left|c[i]^{*,t}\right|\right)^2}{\gamma^2 \sum_t n^t} - \frac{\sum_{t,i} n[i]^t \left|c[i]^{*,t}\right|}{\gamma} \\
&= \frac{2R\left\|\mathbf{a}\right\|}{\gamma^2} + \frac{(k-1)R^2 + 1}{\gamma}^2
\end{aligned}$$

The second inequality follow from the fact that the rank loss for a single instance can be at most one less than the number of ranks: $n^t < k - 1$. The third inequality follows from the fact that by the definition of Good Grief decoding (equation 4.1) the default prediction of the component ranking models will only incur a cost through the meta-model. Since the magnitude of the meta-model cost – $c_{\mathbf{a}}(\mathbf{x}, \mathbf{r};\ \mathbf{a})$ – is bounded by $|\mathbf{a} \cdot \mathbf{x}|$, we can infer that $\max_{t,i} |c[i]^t| \leq \max_t |\mathbf{a} \cdot \mathbf{x}^t| \leq R\|\mathbf{a}\|$ (the last inequality using Cauchy-Schwartz). Finally, the last equality follows by hypothesis: Since the model $\mathbf{v}^*$ perfectly ranks the corpus, the correction terms associated with the rankers of $\mathbf{v}^*$ will always be zero. This completes our proof. □

With this theoretical analysis in hand, we can now test the practical merits of our framework. In the next chapter we provide numerous empirical evaluations of our model and its

decoding and training procedures.

Chapter 5

Experiments

In this chapter, we present several sets of experiments to test the practical merits of our approach. We apply our joint model to a set of restaurant reviews collected from a consumer website. Associated with each review is a set of five ranks, each on a scale from 1-5, covering food, ambience, service, value, and overall experience. Using the agreement meta-model with Good Grief decoding yields significant improvements over individual ranking models [7], a state-of-the art joint ranking model [1], and multiclass Support Vector Machines [6].

We also perform experiments comparing our model to other decoding methods using an agreement model. One such method first performs agreement classification on each instance and then delegates the instance to a single model (in the case of agreement) or to individually trained ranking models (in the case of disagreement). We found that our model outperforms all other strategies for incorporating an agreement model to which we compared it.

We also compared different methods of training our Good Grief model. The simplest approach is to individually train each ranking model as well as the agreement model, and only apply Good Grief decoding at test time. In fact, even this approach outperforms all baselines. However, larger gains are seen when jointly training all ranking models with a pre-trained perceptron agreement model. The best results with a perceptron agreement model are seen when the meta-model itself is trained jointly with all the ranking models, by using the feedback from Good Grief decoding. Finally, similar results are found when

	Food	Service	Value	Atmosphere	Experience	**Total**
MAJORITY	0.848	1.056	1.030	1.044	1.028	1.001*
PRANK	0.606	0.676	0.700	0.776	0.618	0.675*
SVM	0.546	0.642	0.664	0.766	0.614	0.646*
SVM^2	0.624	0.736	0.740	0.850	0.658	0.722*
SIM	0.538	0.614	0.656	0.776	0.606	0.638*
GG (SVM)	**0.528**	**0.590**	**0.638**	**0.750**	**0.564**	**0.614**

Table 5.1: Ranking loss on the test set for Good Grief (GG (SVM)) and various baselines. Diacritic (*) indicates statistically significant difference from performance of GG (SVM) using a Fisher sign test ($p < 0.01$).

pre-training the agreement model using SVM optimization.

In the last set of experiments, we demonstrate the flexibility of the Good Grief framework by applying two meta-models besides the simple agreement model. Both models perform above all baselines. In addition, one of the models was specifically designed to aid performance on the most difficult-to-rank aspect (*atmosphere*), and in fact on this aspect achieves the best performance of any method.

5.1 Experimental Set-Up

We evaluate our multi-aspect ranking algorithm on a corpus[1] of restaurant reviews available on the website `http://www.we8there.com`. Reviews from this website have been previously used in other sentiment analysis tasks [15]. Each review is accompanied by a set of five ranks, each on a scale of 1-5, covering food, ambience, service, value, and overall experience. These ranks are provided by consumers who wrote original reviews. Our corpus does not contain incomplete data points since all the reviews available on this website contain both a review text and the values for all the five aspects.

[1]Data and code are available at `http://people.csail.mit.edu/bsnyder/naacl07`

5.1.1 Corpus Statistics

Our corpus contains 4,488 reviews, averaging 115 words. We randomly select 3,488 reviews for training, 500 for development and 500 for testing. The corpus contains 528 among $5^5 = 3025$ possible rank sets. The most frequent rank set $\langle 5, 5, 5, 5, 5 \rangle$ accounts for 30.5% of the training set. However, no other rank set comprises more than 5% of the data. To cover 90% of occurrences in the training set, 227 rank sets are required. Therefore, treating a rank tuple as a single label is not a viable option for this task. We also find that reviews with full agreement across rank aspects are quite common in our corpus, accounting for 38% of the training data. Thus an agreement-based approach is natural and relevant.

A rank of 5 is the most common rank for all aspects and thus a prediction of all 5's gives a MAJORITY baseline and a natural indication of task difficulty.

5.1.2 Parameter Tuning

We used the development set to determine optimal numbers of training iterations for all models. These numbers were always three or four. After more than four rounds all models experienced some over-fitting. Also, given an initial uncalibrated agreement model $\mathbf{a}'$, we define our agreement model to be $\mathbf{a} = \alpha\mathbf{a}'$ for an appropriate scaling factor α. We tune the value of α on the development set.

5.1.3 Evaluation Measures

We evaluate variants of our algorithm and baselines using *ranking loss* [7, 1]. Ranking loss measures the average distance between the true rank and the predicted rank. Formally, given N test instances $(\mathbf{x}^1, \mathbf{y}^1), ..., (\mathbf{x}^N, \mathbf{y}^N)$ of an m-aspect ranking problem and the corresponding predictions $\hat{\mathbf{y}}^1, ..., \hat{\mathbf{y}}^N$, ranking loss is defined as $\sum_{t,i} \frac{|y[i]^t - \hat{y}[i]^t|}{mN}$. Lower values of this measure correspond to a better performance of the algorithm.

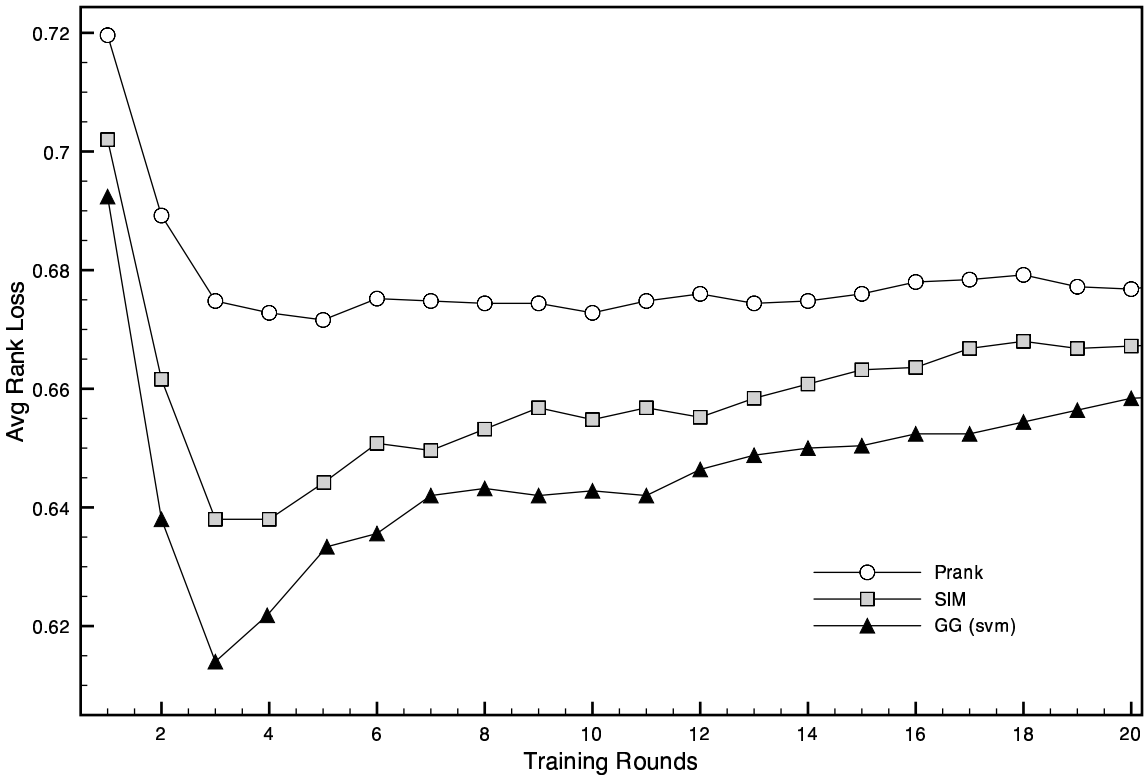

Figure 5-1: Rank loss for our algorithm and baselines as a function of training round.

5.2 Results

5.2.1 Comparison with Baselines

Table 5.1 shows the performance of the Good Grief algorithm (GG (SVM)) along with baselines. We give a brief description of each method and its resulting performance:

- MAJORITY: The majority baseline mentioned in section 5.1 – all aspects are given a rank of 5. On average, the prediction of this baseline is off by about one full rank.

- PRANK: The first competitive baseline learns a separate ranker for each aspect using the PRank algorithm described in section 2. Using this simple learning approach, the average distance between true and predicted ranks is reduced to 0.675.

- SVM: For this baseline, we use the multiclass SVM framework [6] [29] to train a separate classifier for each aspect. The ranks 1-5 are treated as distinct classes and a set of weights is learned for each. While potentially losing some of the generizability of the ranking approaches (by learning separate weights for each aspect), this approach utilizes the well-studied and powerful batch optimization techniques of the Support Vector Machine. This approach yields improvement over PRank learning and results in an average rank loss of 0.646.

- SVM^2: This variant on the SVM baseline utilizes a quadratic kernel. As all our features are binary features indicating the presence or absence of lexical items, this baseline has the effect of operating over a feature space including all pair-wise conjunctions of such lexical items. The effect of this larger feature space is a significant drop in prediction quality, with a rank loss of 0.722.

- SIM: This baseline uses the PRank algorithm to learn separate rankers for each aspect, but shares weights across all rankers using a similarity measure [1]. The method is described in more detail in Chapter 2. This joint model achieves performance gains over the independent PRANK model and reduces the rank loss on average to 0.634.

	Food	Service	Value	Atmosphere	Experience	**Total**
ALL AGREE	0.558	0.634	0.640	0.746	0.586	0.633
DELEGATE	0.550	0.604	0.654	**0.744**	0.592	0.629
FORCE	**0.524**	0.626	0.648	0.768	0.570	0.627
GG (SVM)	0.528	**0.590**	**0.638**	0.750	**0.564**	**0.614**

Table 5.2: Ranking loss on the test set for Good Grief (GG (SVM)) and other agreement-based methods.

- GG (SVM): Good Grief training and decoding with an agreement model trained separately using SVM optimization (**Variant 1** of Good Grief training – see Figure 3-2). This model achieves statistically significant gains (by a Fisher sign-test at $p < 0.05$) over all the baselines. Improvement is seen in all aspects except for *value*, where both GG (SVM) and SIM both achieve rank loss of 0.638.

Figure 5-1 shows the performance of our model along with two of the baselines on the test set as a function of number of training rounds. Although all models tend to experience over-fitting after the third or fourth training iteration, our model maintains a consistent edge over the baselines.

5.2.2 Comparison with other agreement-based methods

In the next set of experiments, the results of which are shown in Table 5.2, we compare the performance of the Good Grief algorithm (GG (SVM)) with other methods which are based on the idea of *agreement*. We find some improvement gains over the baselines with these ad-hoc methods, but none of them match the performance of our model.

- ALL AGREE: This method is essentially a variant of the Good Grief algorithm. The model is forced to predict a consensus rank vector (all 5's, all 4's etc) for *every* instance. In the Good Grief framework this is achieved by setting $score_{\mathbf{a}}(\mathbf{x}) = \infty$ for *all* inputs $\mathbf{x}$. This produces a non-infinite grief only when all aspect ranks agree. Thus, the consensus rank vector with lowest grief will be predicted. This simple method of anchoring the aspect ranks to one another yields performance gains over all the baselines in Table 5.1 with an average rank loss of 0.633.

	Food	Service	Value	Atmosphere	Experience	**Total**
GG BIAS	0.524	0.636	**0.636**	0.764	0.570	0.626
GG DECODE	0.546	0.612	0.658	0.746	0.588	0.630
GG PERCEPT	0.542	0.610	0.648	0.738	0.574	0.622
GG PERCEPT JOINT	**0.490**	0.620	0.678	**0.726**	**0.560**	0.615
GG (SVM)	0.528	**0.590**	0.638	0.750	0.564	**0.614**

Table 5.3: Ranking loss on the test set for variants of Good Grief and various baselines.

- DELEGATE: This method uses a two-step, delegation approach. If the agreement model[2] predicts consensus, then a single ranking model is used to predict a rank for all aspects (and is trained on cases of consensus in the training data). Otherwise, individual rankers trained with PRank are used. As with ALL AGREE, gains are observed over the baselines and an average rank loss of 0.629 is achieved.

- FORCE: Like DELEGATE, this method always predicts a rank vector consistent with the agreement model's prediction (of consensus or non-consensus). However, here this is achieved within the Good Grief framework by setting $score_{\mathbf{a}}(\mathbf{x})$ to ∞ when the agreement model predicts consensus, and $-\infty$ otherwise. However, the griefs of component ranking models are still taken into account when choosing *which* consensus or non-consensus rank vector to predict. Using the Good Grief framework in this way yields a slight performance gain over DELEGATE with average rank loss of 0.627.

5.2.3 Comparison with Good Grief variants

Here we compare variations of the Good Grief algorithm. The results are shown in Table 5.3.

- GG BIAS: In this simplest variation of the Good Grief algorithm, no actual agreement model is utilized. Instead, a single constant bias score is used to encourage agreement across aspects. This is implemented in the Good Grief framework by always setting

[2]trained separately via SVM optimization

	Food	Service	Value	Atmosphere	Experience	**Total**
GG DIVERGE 2	0.528	0.634	0.640	0.766	0.572	0.628
GG FOOD ATMOS	0.544	0.626	0.638	**0.714**	0.602	0.625
GG (SVM)	**0.528**	**0.590**	**0.638**	0.750	**0.564**	**0.614**

Table 5.4: Ranking loss on the test set for agreement-based Good Grief (GG (SVM)) and two Good Grief models with other meta-models.

$score_{\mathbf{a}}(\mathbf{x}) = b$ for some bias score b. This has the effect of pushing borderline cases into agreement. The resulting rank loss is surprisingly low at 0.626.

- GG DECODE: This variant uses PRank training to learn independent ranking models for each aspect and only applies the Good Grief algorithm at test time. An independently trained SVM agreement model is used. Without the benefit of joint training we see a smaller improvement over the baselines and achieve rank loss of 0.630.

- GG PERCEPT: This model uses **Variant 1** (Figure 3-2) of Good Grief training and decoding. The agreement model is pre-trained using the Perceptron algorithm [25]. By training the ranking models to operate in the context of the Good Grief decoder, we achieve gains over GG DECODE as well as GG BIAS, with an average rank loss of 0.622.

- GG PERCEPT JOINT: This model uses **Variant 2** (Figure 3-3 of Good Grief training and decoding. This training variant couples the online training of the agreement model and the ranking models by using the feedback of Good Grief decoding for all model updates. By training the agreement model in tandem with the ranking models, we see improved overall performance, and achieve a rank loss of 0.615. While an improvement over GG PERCEPT, the performance is basically equivalent to that of GG (SVM), which uses training **Variant 1** with a pre-trained SVM agreement model.

5.2.4 Comparison with other meta-models

We performed experiments with two meta-models besides simple agreement. Although neither shows performance gains over the simple agreement-based model, one of them, GG

	Food	Service	Value	Atmosphere	Experience	**Total**
GG (SVM)	0.528	0.590	0.638	0.750	0.564	0.614
DELEGATE ORACLE	0.540	0.596	**0.610**	**0.690**	0.568	0.601
GG ORACLE	**0.510**	**0.578**	0.674	0.694	**0.518**	**0.595**

Table 5.5: Ranking loss on the test set for Good Grief and various oracular models.

FOOD ATMOS, performs best for the hardest aspect: *atmosphere*. Full results are shown in Table 5.4.

- GG DIVERGE 2: A Good Grief variant using a different meta-model: Instead of a simple consensus-based agreement model, here we use a meta-model which predicts whether there is a divergence of at least two rank units between aspect ranks. The meta-model is pre-trained using SVM optimization. This model outperforms all baselines, with an average rank loss of 0.628.

- GG FOOD ATMOS: A Good Grief variant using a different meta-model. This meta-model predicts whether the food aspect (which always gives the best performance) has the same rank as the atmosphere aspect (which always gives the worst performance). Although not performing as well as other Good Grief models for most aspects, this version achieves the highest performance for the Atmosphere aspect.

5.2.5 Comparison with oracles

In this set of experiments, we tested two models which at test time are told by an oracle whether or not each instance has agreement across aspects. The results are shown in Table 5.5. Not surprisingly, both oracle based models outperform all other models, including GG (SVM).

- DELEGATE ORACLE: Instead of using a trained agreement model, this oracular variant of delegate is told exactly which cases have consensus across aspects and which do not, and delegates to individual ranking models or a single consensus-case ranking model accordingly. This model outperforms all previously shown models and achieves average rank loss of 0.601.

	Consensus	Non-consensus
PRANK	0.414	0.864
GG (SVM)	0.326	0.823
GG ORACLE	0.281	0.830

Table 5.6: Ranking loss for our model and PRANK computed separately on cases of actual consensus and actual disagreement.

- GG ORACLE: Instead of using a trained agreement model, this oracular variant of the Good Grief model is told exactly which cases have consensus and which do not. The decoding decision is then made which minimizes grief. This is implemented in the Good Grief framework by setting $score_{\mathbf{a}}(\mathbf{x})$ to ∞ in cases of true consensus and $-\infty$ in cases of non-consensus. For most aspects (and overall), this model – which still uses Griefs of the component ranking models – outperforms the DELEGATE ORACLE model. Overall, this model outperforms all other models, with an average rank loss of 0.595.

5.2.6 Analysis of Results

We separately analyze our performance on the 210 test instances where all the target ranks agree and the remaining 290 instances where there is some contrast. As Table 5.6 shows, we outperform the PRANK baseline in both cases. However on the consensus instances we achieve a relative reduction in error of 21.2% compared to only a 4.7% reduction for the other set. In cases of consensus, the agreement model can guide the ranking models by reducing the decision space to five rank sets. In cases of disagreement, however, our model does not provide sufficient constraints as the vast majority of ranking sets remain viable. This explains the performance of GG ORACLE, the variant of our algorithm with perfect knowledge of agreement/disagreement facts. As shown in Table 5.5, GG ORACLE yields substantial improvement over our algorithm, but all of this gain comes from consensus instances (see Table 5.6).

5.2.7 Performance of Agreement Model

We also examine the impact of the agreement model accuracy on our algorithm. The agreement model, when considered on its own, achieves classification accuracy of 67% on the test set, compared to a majority baseline of 58%. However, those instances with high confidence $|\mathbf{a} \cdot \mathbf{x}|$ exhibit substantially higher classification accuracy. Figure 5-2 shows the performance of the agreement model as a function of the confidence value. The 10% of the data with highest confidence values can be classified by the agreement model with 90% accuracy, and the third of the data with highest confidence can be classified at 80% accuracy.

This property explains why the agreement model helps in joint ranking even though its overall accuracy may seem low. Under the Good Grief criterion, the agreement model's prediction will only be enforced when its grief outweighs that of the ranking models. Thus in cases where the prediction confidence ($|\mathbf{a} \cdot \mathbf{x}|$) is relatively low,[3] the agreement model will essentially be ignored.

5.3 Summary

In this chapter we presented several sets of experiments to test the practical merits of our approach. We found that out model outperforms several baselines, including individual ranking models [7], a state-of-the art joint ranking model [1], and multiclass Support Vector Machines [6]. Interestingly, using a quadratic kernel with the multiclass SVM only degraded performance. Thus we see that an increase in expressive power must be cautiously undertaken to avoid the problem of over-fitting.

We also performed experiments comparing our model to other decoding methods using an agreement model. In these other methods, we impose *hard constraints*, either always forcing agreement, or forcing agreement and disagreement according to the dictates of the agreement model. We found that none of these methods proved as effective as the Good Grief framework, which *weighs* the relative confidence of the meta-model against the con-

[3] What counts as "relatively low" will depend on both the value of the tuning parameter α and the confidence of the component ranking models for a particular input $\mathbf{x}$.

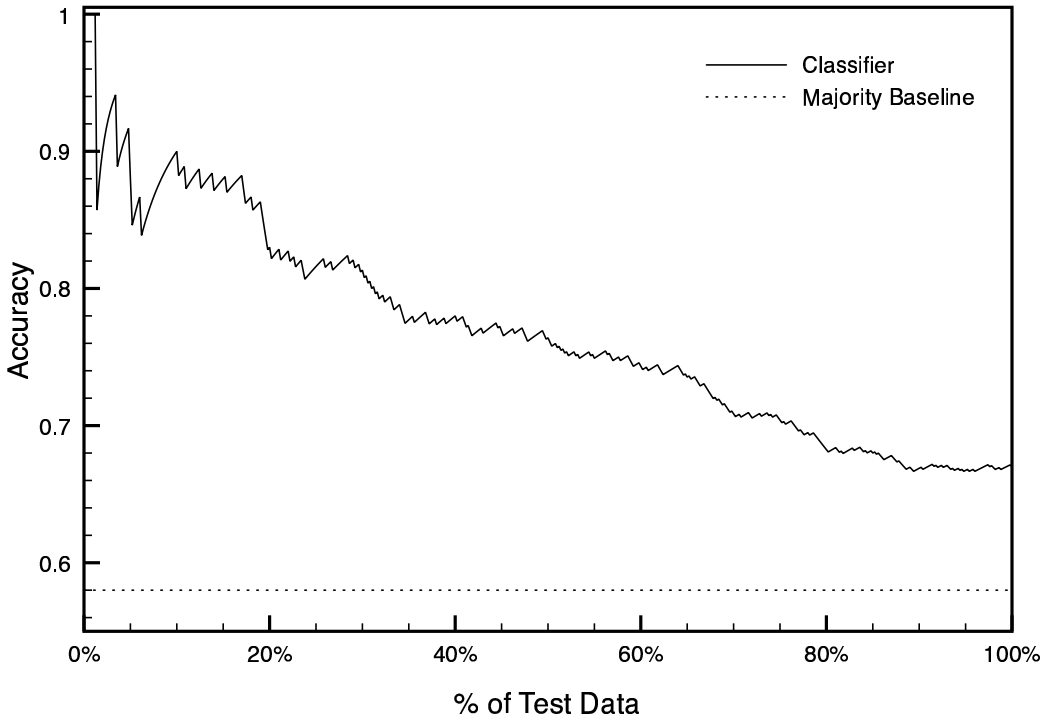

Figure 5-2: Accuracy of the agreement model on subsets of test instances with highest confidence $|\mathbf{a} \cdot \mathbf{x}|$.

fidence of the ranking models when making its prediction. We also compared our method to a simple bias-based method (GG BIAS), which instead of learning an input-sensitive agreement model, simply imposes a soft constraint which always encourages agreement in borderline cases. We found that this method comes up short in comparison to a flexible Good Grief model.

Next we compared different methods of training our Good Grief model. The simplest approach was to individually train each ranking model as well as the agreement model, and only apply Good Grief decoding at test time. In fact, even this approach outperforms all baselines. However, larger gains are seen when jointly training all ranking models with a pre-trained perceptron agreement model. The best results with a perceptron agreement model are seen when the meta-model itself is trained jointly with all the ranking models, by using the feedback from Good Grief decoding. Similar results are found when pre-training the agreement model using SVM optimization.

In summary, the features of our model which seem essential to our performance gains are three-fold:

- joint training using Good Grief decoding as feedback,
- the imposition of *soft* global constraints by weighing the confidence of the meta-model against the confidence of the ranking models, and
- the imposition of *flexible* global constraints by using a trained meta-model which is sensitive to the features of each input.

In the next chapter we conclude our thesis and provide some comments about future research directions.

Chapter 6

Conclusion and Future Work

We considered the problem of analyzing multiple related aspects of user reviews. The algorithm presented jointly learns ranking models for individual aspects by modeling the dependencies between assigned ranks. The strength of our algorithm lies in its ability to guide the prediction of individual rankers using rhetorical relations between aspects such as agreement and contrast. We have demonstrated the expressive power of our framework, while proving that it preserves the convergence guarantees of simpler methods.

We conducted extensive experiments to test the practical benefit of our framework. We found that our method yields significant empirical improvements over individual rankers, a state-of-the-art joint ranking model, and ad-hoc methods for incorporating agreement. Our experiments show that the key benefit of our framework is the incorporation of global coherence predictions through soft and flexible constraints.

In the future, we'd like to explore a broader array of meta-models. Ideally, we'd like to *induce* the structure of the meta-model automatically from a data-set, instead of deciding ahead of time which label relations it should predict. In addition, we'd like to apply our framework to data where the component tasks are not necessarily comparable. For example, sometimes we'd like to perform some mix of extraction, classification, ranking, and regression all on the same input. Finally, we'd like to develop methods which can account for cases where the *number* of tasks to be performed is variable and unknown. For example, in many realistic scenarios we won't know ahead of time which aspects of a restaurant that a reviewer will mention.

Bibliography

[1] Justin Basilico and Thomas Hofmann. Unifying collaborative and content-based filtering. In *Proceedings of the ICML*, pages 65–72, 2004.

[2] Bernhard E. Boser, Isabelle Guyon, and Vladimir Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the COLT*, pages 144–152, 1992.

[3] Michael Collins. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *Proceedings of EMNLP*, pages 1–8, 2002.

[4] Michael Collins. Ranking algorithms for named entity extraction: Boosting and the voted perceptron. In *Proceedings of the ACL*, pages 489–496, 2002.

[5] Michael Collins and Brian Roark. Incremental parsing with the perceptron algorithm. In *Proceedings of the ACL*, pages 111–118, 2004.

[6] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

[7] Koby Crammer and Yoram Singer. Pranking with ranking. In *Proceedings of NIPS*, pages 641–647, 2001.

[8] Koby Crammer and Yoram Singer. Ultraconservative online algorithms for multiclass problems. In *Proceedings of the COLT/EuroCOLT*, pages 99–115, 2001.

[9] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[10] Hal Daumé III and Daniel Marcu. Learning as search optimization: Approximate large margin methods for structured prediction. In *Proceedings of the ICML*, Bonn, Germany, 2005.

[11] Kushal Dave, Steve Lawrence, and David Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of WWW*, pages 519–528, 2003.

[12] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997.

[13] Yoav Freund and Robert E. Schapire. Large margin classification using the perceptron algorithm. In *Proceedings of the COLT*, pages 209–217, 1998.

[14] Andrew B. Goldberg and Xiaojin Zhu. Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. In *Proceedings of the HLT/NAACL workshop on TextGraphs*, pages 45–52, 2006.

[15] Ryuichiro Higashinaka, Rashmi Prasad, and Marilyn Walker. Learning to generate naturalistic utterances using reviews in spoken dialogue systems. In *Proceedings of the COLING/ACL*, pages 265–272, 2006.

[16] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the ICML*, pages 282–289, 2001.

[17] Gideon S. Mann and David Yarowsky. Multi-field information extraction and cross-document fusion. In *Proceedings of the ACL*, 2005.

[18] Daniel Marcu and Abdessamad Echihabi. An unsupervised approach to recognizing discourse relations. In *Proceedings of the ACL*, pages 368–375, 2002.

[19] Paul McJones. Eachmovie collaborative filtering data set. DEC Systems Research Center, 1997. http://www.research.digital.com/SRC/eachmovie.

[20] A. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on Mathematical Theory of Automata*, volume XII, pages 615–622, 1962.

[21] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*, pages 115–124, 2005.

[22] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86, 2002.

[23] Jean-Michel Renders, Éric Gaussier, Cyril Goutte, François Pacull, and Gabriella Csurka. Categorization in multiple category systems. In *Proceedings of the ICML*, pages 745–752, 2006.

[24] Brian Roark, Murat Saraclar, Michael Collins, and Mark Johnson. Discriminative language modeling with conditional random fields and the perceptron algorithm. In *Proceedings of the ACL*, pages 47–54, 2004.

[25] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[26] Dan Roth and Wen tau Yih. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the CoNLL*, pages 1–8. Boston, MA, USA, 2004.

[27] Benjamin Snyder and Regina Barzilay. Database-text alignment via structured multilabel classification. In *Proceedings of the IJCAI*, pages 1713–1718, 2007.

[28] Benjamin Snyder and Regina Barzilay. Multiple aspect ranking using the good grief algorithm. In *Proceedings of HLT-NAACL*, pages 300–307, Rochester, New York, April 2007. Association for Computational Linguistics.

[29] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings the ICML*, 2004.

[30] Peter Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classsification of reviews. In *Proceedings of the ACL*, pages 417–424, 2002.

[31] Michael Wick, Aron Culotta, and Andrew McCallum. Learning field compatibilities to extract database records from unstructured text. In *Proceedings of EMNLP*, pages 603–611, Sydney, Australia, 2006.

[32] Bernard Widrow and Marcian E. Hoff. Adaptive switching circuits. In *1960 IRE WESCON Convention Record*, 1960. Reprinted in *Neurocomputing* (MIT Press, 1988).

[33] Hong Yu and Vasileios Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of EMNLP*, pages 129–136, 2003.