

# Aspect Guided Text Categorization with Unobserved Labels

Dan Roth      Yuancheng Tu  
 University of Illinois at Urbana-Champaign  
 {danr, ytu}@illinois.edu

## Abstract

*This paper proposes a novel multiclass classification method and exhibits its advantage in the domain of text categorization with a large label space and, most importantly, when some of the labels were not observed in the training data. The key insight is the introduction of intermediate aspect variables that encode properties of the labels. Aspect variables serve as a joint representation for observed and unobserved labels. This way the classification problem can be viewed as a structure learning problem with natural constraints on assignments to the aspect variables. We solve the problem as a constrained optimization problem over multiple learners and show significant improvement in classifying short sentences into a large label space of categories, including previously unobserved categories.*

## 1. Introduction

Text Categorization is an archetypical example for multiclass classification and has many applications, ranging from spam filtering to E-mail classification to sentiment analysis. Our interest is in the case where the text documents are quite short, typically only a short text snippet of 3-10 words, and the number of possible categories is quite large, at least a few hundreds. In addition, unlike traditional text classification where each label is only one simple phrase such as *shopping*, we consider the case in which each of the categories is more expressive, i.e., represented as a short sentence such as *find the nearest supermarket*.

One case where this situation is important is in the domain of categorizing free text or spoken commands into operating commands to devices, e.g., as a way to control devices in a car. Moreover, in such applications, it is often the case that new *labels* are introduced, with minimal or no training data. The new labels may be variations of existing labels—the system may be familiar with labels such as “turn on the radio” and “turn off the cd player” and is now requested to categorize also into a new label “turn off the GPS”. Traditional text categorization techniques do not do

well when the “document” is very short, and even worse when the number of labels is very large. And, there is no principled solution to the problem of dealing with new, previously unobserved labels, beyond re-training the classifiers. Moreover, category labels are only treated as mutually exclusive flat symbols without any structure.

In this paper we introduce a new method for multiclass classification of text documents that addresses all these issues. Specifically, it handles well the case of a very large number of labels and, most significantly, it can robustly categorize text snippets into previously unobserved labels which traditional methods cannot deal with at all. Our model, *Multi-Aspect Multiclass Classification (MAMuC)*, introduces a set of intermediate *aspect* variables, each representing a property of the data and its associated labels. Rather than training a multiclass classifier to predict a label, we train a structured classifier, that learns to assign values to the aspect variables. This allows us to enforce natural constraints (hard and soft) among aspect variables—e.g, if an aspect takes the value “turn”, it is unlikely that another aspect will take the value “restaurant”, and it is likely that one will take either the value ‘on’ or ‘off’. We define an objective function that scores assignments to the aspect variables; we then predict a category by choosing the one that maximizes the score of the assignment to the aspect variables, subject to the constraints. This view constitutes a reformulation of the original classification problem as a structured learning problem. We consider two training paradigms of the aspect variables. In one, we train individual models for aspect variables, and then predict a category label by combining the aspect models’ predictions and weights via a constrained optimization framework, subject to the constraints. In the other, we train a joint model for the aspect variables, taking into account the constraints during training.

The key novelty in our model is that the variables dictating the decision are not explicit in the problem definition, but rather introduced as an intermediate level in order to exploit hidden structure of the category labels. As we show, our scheme allows us to significantly improve over traditional state-of-the-art multiclass classification. More significantly, one of the key advantages this model provides is

the ability to handle the case of new, previously unobserved labels. While existing multiclass classification cannot deal with unobserved labels, our method can predict *aspects* of the data. Since aspects are chosen to provide a representation of the label space, predicting the aspect allows us to approximate and often predict a label that was never observed in the training data by determining some of its aspects. In our application, where each label is a short sentence, a partial prediction of the label is useful, since it can trigger interaction to clarify its exact value.

## 2. Related Work

Multiclass classification is a central problem in machine learning. Existing literature on MCC assumes the independence of the output labels. With the exception of some work on MCC with hierarchies [8], this paper is the first to explore the latent structure of the output labels, and to formulate MCC as a structured learning problem. While there has been a lot of work on structured output learning in the last few years [5, 11, 17, 18, 15], we are unaware of any use of it as a way to resolve MCC problems as we do here. The paradigm that comes closest to ours is that of error correcting output codes [7]. Superficially, our scheme can be viewed as one that first generates an output code, and then assembles it to form a legitimate category using constrained optimization. However, there are several key differences—the key one being that in ECOC the decomposition of the label space is done “syntactically”, without any understanding of the output space. Consequently, the resulting binary classification problems could be very difficult learning problems, and the scheme cannot support prediction on new, previously unobserved labels.

There are several ways to train an objective function for structured output problems. We follow a discriminative approach and train the model both jointly and separately within a constrained optimization framework [3, 13]. It has been shown that, when individual components can be learned reliably, the latter is a better training scheme. Our work substantiates this conclusion in the current context.

## 3. Multi-Aspect Multiclass Classification

Text categorization is an archetypical MCC problem; the goal is to learn a function  $f : X \rightarrow Y$  where  $X$  and  $Y$  are collections of documents and labels. It is generally assumed that variables in  $Y$  do not have any latent structure.

The key contribution of our MAMuC model is the introduction of a collection  $\mathcal{Z} = \{z_{11}, z_{12}, \dots, z_{ij}\}$  of intermediate *aspect* variables. Each aspect variable can be thought of as a *property* of the  $Y$  labels. Aspect variables take values that are interdependent; this allows us to exploit con-

straints among these variables as a way to improve aspect predictions and, consequently, the  $Y$  values.

In our domain (see Sec. 4) the labels are operating commands such as “find nearest Chinese restaurant” or “CD track 3”. We make use of five types of aspects which we call: Topic, Action, Manner, Modifier and Detail; each type can take multiple values. For the aforementioned labels, Topic takes the values *restaurant* and *CD*, resp., and Action takes the values *find* and *null*, resp.

An aspect variable  $z_{ij}$  is a function,  $z_{ij} : X \times Y \rightarrow [0, 1]$  which indicates the probability that the  $i^{th}$  aspect type takes its  $j^{th}$  value. At evaluation time, given a document  $x \in X$ , we predict the label  $y \in Y$  that maximizes the score assigned to the aspect variables  $\mathcal{Z}$ :

$$\hat{y} = \arg \max_y \text{score}(\mathcal{Z}(x, y)).$$

Our formulation follows the one developed in [14, 16, 3]. The score is a linear objective function defined in Eq. 1:

$$\hat{y} = \arg \max_y \sum_{i=1}^m w_i z_{ij}(x, y) - \sum_{C_k \in \mathcal{C}} \rho_k d_{C_k}(\mathcal{Z}(x, y), 1_{C_k}) \quad (1)$$

where:

1.  $z_{ij}$  = probability( $i$ th aspect takes its  $j$ th value).
2.  $C_k : 2^{\mathcal{Z}} \rightarrow \{0, 1\}$  are constraints over possible values of the  $z_{ij}$ s. E.g., if  $C_1(z_{23}, z_{34}, z_{15}) = 0$ , then we do not allow the simultaneous assignment of the 3rd value to the 2nd aspect, the 4th value to the 3rd aspect and the 5th value to the 1st aspect. More concretely, we may not allow the Action aspect take the value ‘turn’, when the Topic aspect takes the value ‘restaurant’. Note that we always have a constraint  $\forall i, j, k : C(z_{ij}, z_{ik}) = 0$ .
3.  $d : \mathcal{Z} \times C_k \rightarrow [0, 1]$  measures the degree to which the the assignment  $z \in \mathcal{Z}$  violates the constraint  $C_k$ .  $1_{C_k}$  denotes the set of all assignments in  $\mathcal{Z}$  that satisfy  $C_k$ ;  $d$  measures the distance from assignment  $z \in \mathcal{Z}$  to  $1_{C_k}$ .

This allows us to choose a value  $j$  for each  $z_{ij}$ , taking into account constraints among these variables and consequently choose the category  $\hat{y} \in Y$  as the solution to the integer linear programming in Eq. 1. This objective function is trained discriminatively, and both  $w_i$  and  $\rho_k$  are learned from the training data.

We note that the introduction of the aspect representation can be viewed also as addressing the common sparsity problem in the label space. While it is likely (and happens often in our domain) that a label is represented by a small number of examples, an *aspect* is typically represented by more examples, thus enabling learning more robust aspect functions.

### 3.1. Learning and Inference

We follow a discriminative approach and train the model weights  $w_i$  in two ways, following [13]. In both cases we use an online training paradigm, and make use of the Averaged Perceptron Algorithm [9].

Our *Learning plus Inference* (L + I) training model learns individual  $z_{i*}$  classifiers that are unaware of the constraints. Once we have for each  $z_{i*}$  the distribution of the values it can take, we run the inference in Eq. 1, with the constraints, to assign the labels  $y$ .

The joint training model, *Inference Based Training* (IBT), incorporates the constraints into the training, by running the inference step in Eq. 1 with each evaluation of the classifiers. Model weights are updated when a mistake is encountered after the inference procedure. At decision time, as in L+I, we run the inference in Eq. 1, with the constraints, to assign the labels  $y$ . The details of the algorithms are described in Algorithms 1 and 2, resp.

The constraint penalty weights  $\rho_i$ s in Eq. 1 are learned independently from the weights  $w_i$  [4]. Note that some of the constraints are hard constraints; e.g., those that encode that  $z_{i*}$  takes a single value. For most of the constraints, we set the weights  $\rho_i$  by calculating the corresponding violation probability in the training data:

$$\rho_i = -\log P(C_i \text{ is violated in the training data}) \quad (2)$$

The weight of a constraint that is never violated is set to  $\infty$ ;

---

**Algorithm 1** L + I: Inference refers to the decision making process subject to global constraints, by optimizing Eq. 1. This training algorithm decouples the Learning and Inference. Aspect models  $z_{i*}$  are learned independent of each other, and constraint penalties are also learned independently from the training data.

---

```

1: LEARNING MODELS:
2: for each aspect  $z_{i*}$  do
3:    $z_{i*} = \text{learn}(\text{TrainingData})$ 
4:   note: * ranges over the values of  $i^{\text{th}}$  aspect.
5: end for
6: LEARNING CONSTRAINTS:
7: for each  $C_i$  do
8:   if  $C_i$  is not violated in Training Data then
9:      $\rho_i = \infty$ 
10:  else
11:     $\rho_i = -\log P(C_i \text{ is violated in Training Data})$ 
12:  end if
13: end for
14: INFERENCE at DECISION TIME:
15: Use the distribution over  $z_{i*}$  and solve Eq. 1

```

---

It has been shown that the IBT paradigm is more expressive and should ultimately perform better given sufficiently many training examples [13]. However, it was also shown that when each of the individual classifiers is good, L+I performs better than IBT. In addition, training the individual classifiers independently requires fewer examples and

---

**Algorithm 2** IBT: Aspect models  $z_{i*}$  are learned jointly subject to the constraints. In the on-line algorithm used, model update and global inference (Eq. 1) are interleaved.

---

```

1: LEARNING CONSTRAINTS:
2: for each  $C_i$  do
3:   if  $C_i$  is not violated in Training Data then
4:      $\rho_i = \infty$ 
5:   else
6:      $\rho_i = -\log P(C_i \text{ is violated in Training Data})$ 
7:   end if
8: end for
9: LEARNING MODELS
10: Initialize weights for  $z_{i*}$ 
11: for each example  $(x, y_{\text{gold}})$  in Training Data do
12:   Evaluate  $z_{i*}$ 
13:    $y_{\text{pred}} = \text{solution of Eq. 1}$ 
14:   if  $y_{\text{pred}} \neq y_{\text{gold}}$  then
15:     for each aspect  $z_{ij}$  do
16:       if  $(z_{ij})_{\text{pred}} \neq (z_{ij})_{\text{gold}}$  then
17:          $w(z_{ij}) = w(z_{ij}) + z_{ij}(x, y_{\text{gold}}) - z_{ij}(x, y_{\text{pred}})$ 
18:       end if
19:     end for
20:   end if
21: end for
22: INFERENCE at DECISION TIME:
23: Use the distribution over  $z_{i*}$  and solve Eq. 1

```

---

this paradigm has been used successfully by several authors [12, 6, 1]. Our empirical results, comparing between these two paradigms, (see Sec.4) agree with earlier results, and favor decoupling the training from the inference via L+I.

In both of these learning and inference paradigms we make use of a regularized version of the Averaged Perceptron algorithm [9], implemented within the Sparse Network of Winnow framework [2]. While classical Perceptron comes with generalization bound related to the margin of the data, Averaged Perceptron also comes with a PAC-like generalization bound [9]. This linear learning algorithm is known, both theoretically and experimentally, to be among the best linear learning approaches and is competitive with SVM and Logistic Regression, while being more efficient in training. It also has been shown to produce state-of-the-art results on many natural language applications [12].

Our inference, both at decision time and during training (when running the IBT algorithm) is implemented as an exhaustive grid search algorithm in the space constructed by the top k outputs from each individual classifier. This is facilitated by the fact that, despite the large number of values each of the aspect variables  $z_{i*}$  can take, only one of these,  $z_{ij}$ , can be active in each instance (allowing also for a *null* value). It is easy to observe that when  $w_i$  is the probability that the  $i^{\text{th}}$  aspect takes the value  $j$ , the solution to the optimization problem is the element in  $\mathcal{Z}$  that maximizes the expected number of correct aspects, modulo the constraints. This is especially important when some of the labels have not been observed previously, as discussed in Sec. 4.3.2.

The constraints  $C_i$  used in our application can also be

represented as rules such as “if event  $X = x'$ , then  $A = a$ ” (corresponding to  $C([X = x'] \wedge \neg[A = a]) = 0$ ). For comprehensibility, we present examples this way in Table 1.

With the exception of the constraints that are part of the problem formulation (e.g.,  $\forall i, j, k : C(z_{ij}, z_{ik}) = 0$ ), all the constraints used in our application are learned semi-automatically using the following process. First we generate the  $\mathcal{Z}$  space by mapping from the given labels; then we use a simple association rule mining algorithm on the  $\mathcal{Z}$  space to find interesting constraints. We set minimum thresholds on support and confidence and learned about 45 constraints over the space of 318  $\mathcal{Z}$  tuples.

Constraints	Type	$\rho_i$
$z_a(find) \wedge z_t(restaurant) \rightarrow z_d(nearest)$	hard	$\infty$
$z_m(null) \wedge z_t(store) \rightarrow z_a(find)$	soft	2.207
$z_t(cd) \wedge z_a(play) \wedge z_m(null) \rightarrow z_{mo}(normal)$	hard	$\infty$

**Table 1.** Examples of soft and hard constraints.  $z_a(find)$  means that the value of *Action* aspect is *find*.

## 4. Experiments and Analysis

### 4.1. Data

Our experimental study is done in the domain of categorizing short text snippets provided by passengers in a car, into operating commands to devices such as *radio*, *ac*, *navigation system* in the car. The data set consists of 72,483 labeled examples with 318 short navigation commands as class labels. Table 2 shows several concrete examples from the data set. The data was collected from the OpenMind Indoor Common Sense Project [10].

Labels	Texts
find nearest restaurant	locate next diner where is the closest restaurant show me where I can eat nearby
display map	show map setting point location route display
radio seek down	change the radio station run down through the stations tune down

**Table 2.** Concrete examples from the data set. Notice that the label carries similar meaning to the text.

### 4.2. Aspects Variables

We define a collection of 5 types of aspect variables: Topic, Action, Manner, Modifier and Detail, each of which can be thought of as a property of a text snippet and its corresponding label – the car command. If an aspect is not

present in the data, the value *null* is assigned. Examples of data, labels and corresponding aspects are listed in Table 3. Each of the five aspects can take multiple values. Using the notation in Sec. 3,  $z_{ij}$ , indicates a variable with the aspect type set to  $i \in \{1, 2, 3, 4, 5\}$  and the value of the  $i^{th}$  type aspect is set to  $j$ .

### 4.3. Evaluation and Results

We evaluate our text categorization scheme by reporting the results of our scheme compared to the standard MCC protocol. We evaluate in two settings. In one, all labels are assumed to be seen in the training data. We therefore randomly select 80% of the examples for each label as training, 10% as development and the rest as test data. In the second setting, we evaluate the ability of our method to predict labels that are not part of the training data. In this case, we eliminate examples that correspond to 10% of the labels from the training data. These examples are then presented during testing and we measure the performance of our model on them.

Note that classifiers trained using MCC cannot make any predictions on these examples; however, since most aspects of these labels are observed in training, we expect that our MAMuC model will do reasonably well on them.

We evaluated the system’s performance using two measures. The first is the standard *accuracy* computed as the percentage of correctly labeled examples. The second metric is motivated by the specific application, where even a partially labeled instance is useful; for example, it could trigger an interaction with the passenger in order to get a clarification. This measure, a **Weighted Aspect based Metric (WAM)**, is a weighted Hamming distance computed at the aspect level, between the predicted and the correct aspect value; it assigns a weighted per-aspect score to partially correct predictions. We formally define it in equation 3.

Formally, let  $n$  be the number of test examples,  $m$  be the number of aspects and  $\omega_k$  be the weight assigned to the  $k$ -th aspect. We denote by  $a_k^i$  the  $k$ -th aspect of the  $i$ -th example and by  $g_k^i$  be the true value of the  $k$ -th aspect of this example. Then,  $1_{[a_k^i: g_k^i]}$  is an indicator variable that takes the value 1 when  $a_k^i$  equals  $g_k^i$ , and 0 otherwise. The WAM measure is then written as follows:

$$\text{WAM} = \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^m \omega_k 1_{[a_k^i: g_k^i]} \quad \sum_{k=1}^m \omega_k = 1 \quad (3)$$

#### 4.3.1 Standard Multiclass Classification Setting

The results of the setting in which all labels in test have been observed in training are summarized in Tables 4,5 and 6.

The baseline result in Table 4, achieved with a regularized MCC Averaged Perceptron, is the best result achieved

Original Label(318)	Input Text	Topic (98)	Action (27)	Manner (40)	Modifier (49)	Detail (14)
climate control defrost off	stop defrosting	climate control	defrost	off	null	null
display current location	show location	location	display	null	current	null
find nearest american restaurant	american food	restaurant	find	null	american	nearest
volume up	turn it up louder	volume	null	up	null	null
d v d title chapter	disc chapter title	d v d	null	null	title	chapter

**Table 3.** Example of input text, their associated gold standard labels and the corresponding values of the five aspects. There are 318 different labels, and the values in parenthesis above the aspect columns indicate the number of possible values each aspect can take.

Algorithm	Accuracy (%)	WAM (%)
Baseline	67.84	86.14
Aspects (No Inference)	64.70	88.94
MAMuC (IBT)	61.76	84.16
<b>MAMuC (L+I)</b>	<b>71.18</b>	<b>89.65</b>
<b>Error Reduction (%)</b>	<b>10.39</b>	<b>25.32</b>

**Table 4.** Comparing several approaches in the standard text categorization setting. The Baseline is learned via a MCC Averaged Perceptron [9]. IBT is the joint training model and L+I is the decoupled model. Our best model result (MAMuC with L+I training) is only marginally lower than human performance.

Rank	MAMuC Accuracy(%)
Top 1	71.18
Top 2	80.74
Top 3	84.08
Top 4	86.49
Top 5	88.17

**Table 5.** Performance of top-k accuracy ( $k \in \{1, 2, 3, 4, 5\}$ ) of the MAMuC model.

on this data set, when compared to other common MCC algorithms. The MAMuC result shows a significant improvement of about 3.5% in accuracy and WAM, which is more than 10% (25%) error reduction in accuracy (WAM, resp.). The improvement achieved by incorporating the inference step, as indicated in the *MAMuC (L+I)* row, shows the effectiveness of the global constraints in our model<sup>1</sup>. Among our two training paradigms, L+I is shown to perform much better than IBT. Given these results we performed an analysis of the individual aspect classifiers, shown in Table 6. Each individual classifier is shown to have a pretty high baseline accuracy. And, these results are shown to improve significantly when we consider each aspect’s accuracy once global constraints were taken into account. This analysis agrees with earlier studies on the tradeoff between L+I and IBT training paradigms [13], which argues that L+I is likely to perform better when the individual models are relatively easy to learn. We note that we have experimented with other

<sup>1</sup>Human performance on this data set is around 75%. Three annotators labeled a sample of 100 examples from this data set. The Fleiss Kappa agreement for this experiment is 0.764, which means substantial agreement among annotators.

Aspects	MAMuC	baseline	error reduction (%)
Topic	<b>86.14</b>	81.55	24.88
Action	<b>88.31</b>	82.72	32.35
Manner	<b>89.98</b>	87.35	20.79
Modifier	<b>91.15</b>	89.51	15.64
Detail	<b>92.68</b>	89.59	29.68

**Table 6.** Improvement of individual aspect classifiers  $z_{i*}$  with inference over a MCC baseline.

structured learning algorithms, such as structured SVM [18] but these produced even weaker results, probably since the structure supported there is not expressive enough for our setting.

Finally, we show results on the top k ranking of the model prediction; table 5 shows that the model’s accuracy is more than 88% in its top five predictions.

### 4.3.2 Predicting Previously Unobserved Labels

The main results in this setting are summarized in Tables 7 and 8. In this experimental setting, sentences in the test data have labels that were not observed in training. This explains the baseline 0 performance in Table 7 – MCC algorithms cannot handle this case. However, in MAMuC, we can predict the aspects of the text snippet provided in test, and evaluate how much it can (partially) predict the new label.

Table 7 summarizes the results of the MAMuC model when we insist that our model predicts all aspects, and then use these to construct a new label (using the inference procedure).

Algorithm	Accuracy (%)	WAM (%)
Baseline	<b>0.00</b>	58.43
Aspect(No Inference)	21.39	69.86
<b>MAMuC</b>	<b>28.16</b>	<b>70.27</b>
Error Reduction (%)	28.16	28.48

**Table 7.** MAMuC’s ability to predict unobserved labels. Traditional classification methods cannot make any prediction on these examples. However MAMuC can achieve partial success by predicting the intermediate aspect representation.

In fact, the intermediate aspect representation can be used in order to support *partial* label prediction. We do this by considering the top  $n$  most confidently predicted aspects ( $n \leq 5$ ). Table 8 shows the accuracy of a partial prediction consisting of the aspect prediction for the  $n$  most confidently predicted aspects. For example, if our model only makes predictions on its top 2 aspects, its accuracy on unseen labels is 82.67%. This evaluation shows that our model is very good at predicting aspects of unobserved labels, even if it cannot “name” the label. This can be used, for example, in an interactive setting – where partial labels are reliably proposed by the model, and the new label is then derived via interaction. In addition, this model can be used to enrich the label space with new labels.

Partial Prediction	Accuracy(%) (no inference)
Top 1 Aspect	92.61
Top 2 Aspect	82.67
Top 3 Aspect	65.17
Top 4 Aspect	43.43
All 5 Aspect	21.39

**Table 8.** Top  $n$  partial prediction evaluation for MAMuC in the case of previously unobserved labels. The accuracy of the partial prediction is measured before the inference procedure.

## 5. Conclusion

We propose a new approach to Multiclass Classification of text documents that exploits the structure of the label space as a way to improve categorization of short text snippets. Our approach works by introducing a set of intermediate *aspect* variables that capture properties of the text.

The key advantage of this view is that *aspects* can be constrained to support the prediction of better labels. Since values taken by aspect variables are constrained in a natural way, we predict labels that correspond to an assignment of values to aspects that satisfies all constraints.

Even more significantly, the aspects provide a shared representation for the label space, one that is shared also by previously unobserved labels. Therefore, once we can predict aspects reliably, we can say something sensible about aspects of the unobserved label too and, at least, predict it partially. As we have shown, this yields significant improvements on our problem and has great potential to correctly predict unobserved labels which traditional multiclass classification methods cannot get at all.

## Acknowledgments

The authors would like to thank Ming-Wei Chang, Michael Connor for insightful discussions of the algorithm

and Rakesh Gupta for discussions of the problem at the preliminary stages of this research, and for providing us with the data set. This research was supported partly by a grant from Honda Research and by MIAS, a DHS-IDS Center for Multimodal Information Access and Synthesis at UIUC.

## References

- [1] R. Barzilay and M. Lapata. Aggregation via Set Partitioning for Natural Language Generation. In *NAACL*, 2006.
- [2] A. Carlson, C. Cumby, J. Rosen, and D. Roth. The SNoW learning architecture. Technical report, 1999.
- [3] M. Chang, L. Ratinov, and D. Roth. Guiding semi-supervision with constraint-driven learning. In *ACL*, 2007.
- [4] M. Chang, L. Ratinov, and D. Roth. Constraints as prior knowledge. In *ICML Workshop on Prior Knowledge for Text and Language Processing*, pages 32–39, July 2008.
- [5] M. Collins. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *EMNLP*, 2002.
- [6] P. Denis and J. Baldridge. Joint determination of anaphoricity and coreference resolution using integer programming. In *NAACL*, 2007.
- [7] T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.
- [8] R. El-Yaniv and N. Etzion-Rosenberg. Hierarchical multiclass decompositions with application to authorship determination. Technical report, 2004.
- [9] Y. Freund and R. E. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296, 1999.
- [10] R. Gupta and M. Kochenderfer. Common sense data acquisition for indoor mobile robots. In *AAAI*, 2004.
- [11] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.
- [12] V. Punyakanok, D. Roth, and W. Yih. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2), 2008.
- [13] V. Punyakanok, D. Roth, W. Yih, and D. Zimak. Learning and inference over constrained output. In *IJCAI*, 2005.
- [14] D. Roth and W. Yih. A linear programming formulation for global inference in natural language tasks. In *CoNLL*, 2004.
- [15] D. Roth and W. Yih. Integer linear programming inference for conditional random fields. In *Proc. of the International Conference on Machine Learning (ICML)*, pages 737–744, 2005.
- [16] D. Roth and W. Yih. Global inference for entity and relation identification via a linear programming formulation. In L. Getoor and B. Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2007.
- [17] K. Toutanova. Competitive generative models with structure learning for nlp classification tasks. In *EMNLP*, 2006.
- [18] I. Tsochanaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector learning for interdependent and structured output spaces. In *ICML*, 2004.