Label Ranking under Ambiguous Supervision for Learning Semantic Correspondences

Antoine Bordes ANTOINE.BORDES@LIP6.FR
Nicolas Usunier NICOLAS.USUNIER@LIP6.FR
LIP6, Université Paris 6. 104, avenue du Pdt Kennedy 75016 Paris, France
Jason Weston JWESTON@GOOGLE.COM
Google. 111 8th Avenue, New York, NY. 10011-5201, USA

Abstract

This paper studies the problem of learning from ambiguous supervision, focusing on the task of learning semantic correspondences. A learning problem is said to be ambiguously supervised when, for a given training input, a set of output candidates is provided with no prior of which one is correct. We propose to tackle this problem by solving a related unambiguous task with a label ranking approach and show how and why this performs well on the original task, via the method of task-transfer. We apply it to learning to match natural language sentences to a structured representation of their meaning and empirically demonstrate that this competes with the state-of-the-art on two benchmarks.

1. Introduction

Annotating training data for supervised learning algorithms is often costly and time-consuming, and depending on the task can even require highly-advanced expertise on the part of the labeler. One opportunity to bypass this requirement is that for many tasks an automatic use of multimodal environments can provide training corpora with little or no human processing. For instance, the time synchronisation of several media can generate annotated corpora: matching movies with corresponding scripts can be used for speech recognition or information retrieval in videos (Cour et al., 2008), matching vision sensors and other sensors can be used to improve robotic vision (Angelova et al., 2007), matching natural language and perceptive events (such as audio commentaries and soccer actions in RoboCup (Chen & Mooney, 2008)) can be used to learn semantics. Indeed, the Internet is abundant with such sources, for example one could think of using the text surrounding pictures in a webpage as image labeling candidates.

Such automatic procedures can build large corpora of ambiguously supervised examples. Indeed, every resulting input instance (picture, video, speech, . . .) is paired with a set of candidate output labels (text caption, subtitle, . . .). The automation of the data collection makes it impossible to directly know which one is correct among them, or even if there exists a correct label. To conceive systems able to efficiently learn from such noisy and ambiguous supervision would be a huge leap forward in machine learning. These methods could then benefit from large training sets obtained with drastically reduced costs.

A domain for which data collection is particularly expensive is semantic parsing (Mooney, 2004). The goal of semantic parsing is to build systems able to understand questions or instructions in natural language in order to bring about a major improvement in human-computer interfacing. Formally, this consists in mapping natural language sentences into structured representations of their meaning which are domain-specific and directly interpretable by a computer. Recent machine learning work (Zettlemoyer & Collins, 2009; Branavan et al., 2009; Ge & Mooney, 2009) exhibits promising progress on this task. Building training data for semantic parsing requires the precise alignment of sentences and formal representations with a costly process that forbids the creation of large-scale corpora. However, for many topics such as finance, music or sports, huge databases paired with corresponding texts are readily available and can be automatically aligned to provide large quantities of ambiguously annotated examples. Unfortunately, this data cannot be used by most semantic parsing methods.

Appearing in *Proceedings of the 27^{th} International Conference on Machine Learning*, Haifa, Israel, 2010.

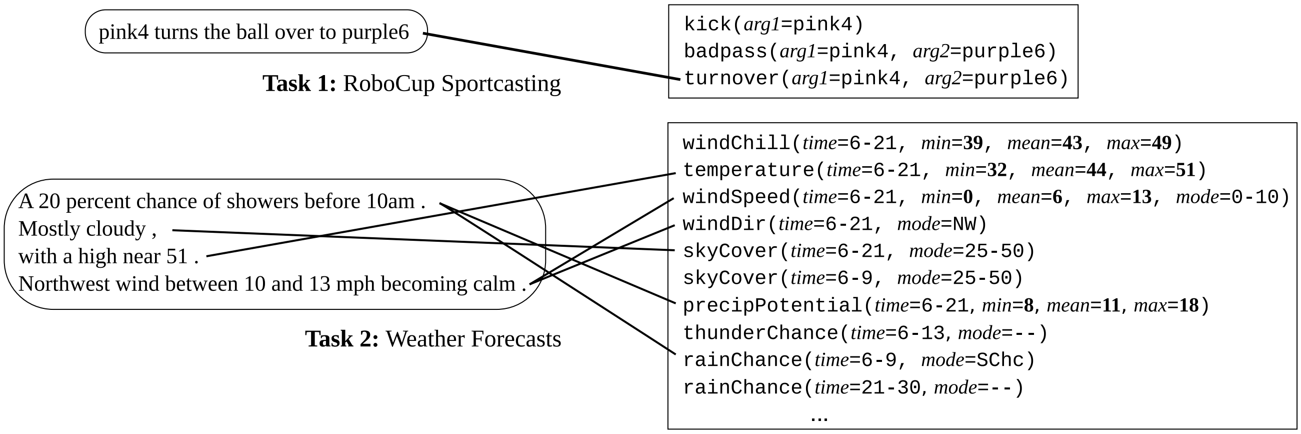
pink4 turns the ball over to purple6
Task 1: RoboCup Sportcasting
kick(arg1=pink4)
badpass(arg1=pink4, arg2=purple6)
turnover(arg1=pink4, arg2=purple6)
A 20 percent chance of showers before 10am .
Mostly cloudy ,
with a high near 51 .
Northwest wind between 10 and 13 mph becoming calm .
Task 2: Weather Forecasts
windChill(time=6-21, min=39, mean=43, max=49)
temperature(time=6-21, min=32, mean=44, max=51)
windSpeed(time=6-21, min=0, mean=6, max=13, mode=0-10)
windDir(time=6-21, mode=NW)
skyCover(time=6-21, mode=25-50)
skyCover(time=6-9, mode=25-50)
precipPotential(time=6-21, min=8, mean=11, max=18)
thunderChance(time=6-13, mode=--)
rainChance(time=6-9, mode=SChc)
rainChance(time=21-30, mode=--)
...

Figure 1. **Examples of scenarios** for the two tasks studied in this paper. A scenario is composed of a text (left) and a set of records (right). Some of those correspond to interpretations of either, all, or segments of the text (black lines). However these gold alignments are unknown in training. All record structures are identical: a type (`kick`, `windDir`, ...) and several fields (*arg1*, *time*, *mean*, ...) whose values can be integer (in **bold**) or categorical (in `typewriter` font).

In this paper, we tackle the problem of learning semantic correspondences for natural language (Snyder & Barzilay, 2007; Liang et al., 2009). More precisely, this task consists in aligning texts with corresponding database entries in order to provide disambiguated training examples for semantic parsing. To learn it under ambiguous supervision, we propose to solve an associated task and make use of task-transfer. We derive a label ranking approach to a related unambiguous task and demonstrate that a solution to this problem performs well on the original task. In other words, we show that one can bypass the difficulty of ambiguous supervision and still reach good performances on its desired target. We then propose an intuitive framework to directly apply standard ranking algorithms to successfully and quickly learn semantic correspondences even with a poor supervision level. On two concrete problems, RoboCup sportscasting and Weather forecasting, we empirically demonstrate that this new simple approach is competitive with the generative method proposed by Liang et al. (2009), which is the best current approach to the best of our knowledge.

The rest of the paper is organized as follows. Section 2 details the task as well as the datasets we used. In Section 3 we formalize the problem of ranking under ambiguous supervision while in Section 4 we establish the transfer of performance. We explain in Section 5 how to use it for learning semantic matching and describe experimental results in Section 6. Note that an extended version of this paper, containing all proofs and extra-figures, can be downloaded from `http://webia.lip6.fr/~bordes/bordes-icml10-extended.pdf`.

2. Learning Semantic Correspondences

Our interest consists in learning to match a natural language text with a structured representation of its meaning, which is composed of one or several domain-specific database records. All records share the same pattern: a record type followed by a set of fields, which can take either categorical or integer values. However, the kinds of types and fields are task-dependent. We detail them for the specific datasets we consider in this paper in the following subsections.

The training algorithm is given pairs composed by a text and a set of records. Following Liang et al. (2009), we use the term *scenario* to refer to such a pair. The set of records, which we call the *candidate set*, is gathered via a cheap automatic process which introduces noise and ambiguity in the supervision. Hence, the candidate set typically groups together pertinent records regarding the associated text as well as many irrelevant ones. Learning semantic correspondences aims at detecting these relevant records, if any, termed the *gold alignments*, among all the records of the candidate set. Therefore, at test time, one is provided scenarios, i.e. texts paired with candidate sets, as well as gold alignments, which are not given to the training algorithm but are essential for evaluation purposes.

2.1. RoboCup Sportscasting

Our first specific task is to learn to match commentaries with records describing actions of RoboCup soccer games. We used the data collected by Chen & Mooney (2008) corresponding to four RoboCup finals and composed of text commentaries automatically paired with records representing the actions that occurred within 5 seconds of them. The whole dataset groups 1,872 scenarios, and each of them is composed by one sentence and a candidate set containing between 1 and 12 records (with a mean of 2.4). One of those is supposed to correspond to the commentary but it is worth noting that, for more than 15% of the scenarios, the correct record is not in the candidate set: in that case, any prediction is automatically wrong.

In total, there are 9 record types (e.g. `pass`, `kick`, `ballstopped`, ...) and each record can have at most 2 fields (e.g. *arg1*=`purple1`, *arg2*=`pink4`, ...) indicating the player(s) involved in the action. As illustration, Figure 1 (top) depicts a RoboCup scenario which is composed of a commentary (left) and a candidate set of 3 records (right), and contains a gold alignment. In this dataset, all record fields are categorical so that, for example, no a priori association between the field value `pink4` and the word "pink4" is possible. Category names chosen are purely for ease of explanation.

2.2. Weather Forecasts

The second task concerns learning correspondences between local forecast reports and records representing weather events. These records actually consist in measurements of meteorological indicators such as temperature, wind speed/direction or chance of sleet, automatically extracted from the database of `www.weather.gov`. The dataset, created by Liang et al. (2009), groups 22,146 scenarios collected, each day and night over 3 days, from the local forecasts of 3,753 US cities. Each candidate set contains exactly 36 records.

An example of a scenario is given at the bottom of Figure 1. There are a total of 12 record types (e.g. `temperature`, `windDir`, `thunderChance`, ...) and each of them can have up to 5 fields. Two of them take categorical values: *time* which indicates the time range of the event and *mode* which can describe some of its characteristics (e.g. the direction of the wind). The other three take integer values. Denoted *min*, *mean* and *max*, they provide exact values of some quantifiable indicators like the temperature or wind speed.

Learning for this task is harder than for RoboCup because the candidate sets group more records and each text refers to more than one record (5.8 on average). Indeed, the reports have been split by punctuation into lines, giving an average of 4.6 text lines per scenario, all sharing the same candidate set. During evaluation, gold alignments must be performed at this level and there are approximately 1.2 alignments per line.

3. Ranking and Ambiguous Supervision

The task of label ranking commonly considers a measurable space of observations $\mathcal{X}$, a finite set of labels $\mathcal{Y} = \{1, ..., |\mathcal{Y}|\}$ and a function ϕ that maps any (input, label) pair to a measurable space $\mathcal{M}$. For the particular case of semantic matching, $\mathcal{X}$ is the set of possible sentences, $\mathcal{Y}$ is the set of all possible records (identified with their indices in $\{1, ..., |\mathcal{Y}|\}$) and ϕ is the joint representation of a sentence and a record. A *label scoring function* (LSF in brief) is a real-valued function of the form $h = g \circ \phi$ with $g : \mathcal{M} \to \Re$. The score $h(x, i)$ of the label i for the input x is denoted $h_i(x)$. From now on, we only consider measurable LSFs.

3.1. Ambiguous Supervision

We now describe our setting for label ranking with ambiguous supervision. The data is modeled by triplets of jointly distributed random variables (X, Z, Y) taking values in $\mathcal{X} \times \{0, 1\}^{|\mathcal{Y}|} \times \{0, 1\}^{|\mathcal{Y}|}$. For a given realization (x, z, y) of (X, Z, Y), x is the observation, z is a subset of $\mathcal{Y}$ called the *target set* for x ($z_i = 1$ when i is in the target set), and y is the *candidate set* for x.

A *target task* is defined by a risk functional $\mathcal{R}$ which uses the full knowledge of (X, Z, Y). In the label ranking framework, $\mathcal{R}$ measures the ability of a LSF to give higher scores to the labels of the target set. Learning with ambiguous supervision means that the target sets are unknown at training time: the training data only consists of n realizations $(x^k, y^k)_{k=1}^n$ of (X, Y). For example, in our datasets x is a sentence, z represents its gold alignments (unknown during training) and y its candidate set. The target task consists in ranking the gold alignments above the other candidates.

In this paper, we consider learning with ambiguous supervision as a kind of transfer learning: we intend to find a LSF of low risk as measured by the target task. But since we cannot measure this risk, our approach is based on defining a *proxy risk functional* which only depends on (X, Y), so that the training data helps to find a performing function for it. In the following, we address the issue of how to define such a proxy risk so that what has been learnt on the training data can be transferred to the target task, under some assumptions that are (almost) satisfied by real-life datasets.

3.2. Categories of Supervision

Before describing the specific target tasks of ranking we consider in this paper, we may distinguish several characteristics of the data, which influence the difficulty of their learning. We say that (X, Z, Y) is:

noisy if $\mathbb{P}\big(Z^*(X) \neq Z\big) > 0$, where:

$$\forall x, Z^*(x) = \arg\max_{z \in \{0,1\}^{|\mathcal{Y}|}} \mathbb{P}(\, Z = z |\, X = x),$$

ambiguous if $\mathbb{P}\big(Z \neq Y\big) > 0$,

incomplete if $\mathbb{P}\big(Z \cap Y \neq Z\big) > 0$.

The data exhibits some *noise* when the target set is not deterministic. The supervision is *ambiguous* when the candidate set can be different from the target set. It may be bigger, or even completely uncorrelated (although learning is probably impossible in the latter case). We also qualify an ambiguous supervision as *incomplete* when some target labels are not candidates or when the target set is empty while the candidate set is not. This occurs frequently in the RoboCup dataset as many training sentences have no gold alignment, but rarely in the Weather dataset. The latter is however very noisy: many lines (e.g. “Mostly cloudy”) appear several times with different target sets (e.g. `skycover`(*time*=`6-21`, *mode*=`25-50`) and `skycover`(*time*=`17-30`, *mode*=`25-50`)).

3.3. Target Tasks

Our work studies the two following target tasks:

Full Ranking This task is defined by the *Full Ranking Risk* $\mathcal{R}^{\text{Full}}$, which measures the ability of a LSF h to rank the target labels above all others:

$$\mathcal{R}^{\text{Full}}(h) = \mathbb{E}\big[\ell^{\text{Full}}(h, X, Z)\big], \text{ with}$$

$$\ell^{\text{Full}}(h, x, z) = (\frac{1}{P} \sum_{i \neq j} \mathtt{I}_{\{(z_i - z_j)\tilde{h}_{ij}(x) < 0\}} \quad (1)$$

for all (x, z), where $P = |\mathcal{Y}|(|\mathcal{Y}| - 1)$ is a normalization factor, $\mathtt{I}_{\{\}}$ is the indicator function and $\tilde{h}_{ij}(x) = sign(h_i(x) - h_j(x))$ with $sign(t) = 2\mathtt{I}_{\{t \geq 0\}} - 1$.

Candidate Set Ranking This task ignores the labels that are not in the candidate set, independently of whether they are target labels or not. The corresponding risk is defined as:

$$\mathcal{R}^{\text{CSet}}(h) = \mathbb{E}\big[\ell^{\text{CSet}}(h, X, Z, Y)\big], \text{ with}$$

$$\ell^{\text{CSet}}(h, x, z, y) = \frac{1}{P} \sum_{i \neq j} \mathtt{I}_{\{y_i = y_j = 1\}} \mathtt{I}_{\{(z_i - z_j)\tilde{h}_{ij}(x) < 0\}} \quad (2)$$

for any (x, z, y).

These risks correspond to standard pairwise ranking risks of label ranking (see e.g. Har-Peled et al., 2002), up to the normalization factor P.[1] The Full Ranking Risk increases linearly with the number of pairs of (non-target, target) labels for which the relative ordering is incorrectly predicted. The Candidate Set Ranking Risk behaves the same way, but restricted to the labels of the candidate set. The difference between these two tasks is clarified by the following lemma:

Lemma 1 *Define, for all x, i,*

- $\eta_i^{\text{Full}}(x) = \mathbb{P}(\, Z_i = 1 |\, X = x)$,
- $\eta_i^{\text{CSet}}(x) = \mathbb{P}(\, Z_i = 1 |\, Y_i = 1, X = x)$,

and denote:
$\mathcal{R}_{bayes}^{\text{Full}} = \mathcal{R}^{\text{Full}}\big(\eta^{\text{Full}}\big)$ *and* $\mathcal{R}_{bayes}^{\text{CSet}} = \mathcal{R}^{\text{CSet}}\big(\eta^{\text{CSet}}\big)$.
Then, for any LSF h, we have: $\mathcal{R}_{bayes}^{\text{Full}} \leq \mathcal{R}^{\text{Full}}(h)$.

Besides, if for any labels i and j with $i \neq j$, $Z_i = 1$ is conditionally independent of Y_j given $Y_i = 1$ and X then:

$$\mathcal{R}_{bayes}^{\text{CSet}} \leq \mathcal{R}^{\text{CSet}}(h) \quad \textit{for any LSF } h.$$

The two target tasks are inherently different as they have different optimal LSF in general. The Full Ranking task is a standard setting, relying on the frequency of a label in the target set given x and independent of the candidate set. Yet, under ambiguous supervision, the candidate set can have a non-trivial influence: correlations can exist between candidate labels, a label can appear very rarely whilst being a target any time it appears, etc. The Candidate Set Ranking task is destined to handle such cases.

The conditional independence assumption used for Candidate Set Ranking is rather natural because it only requires that a candidate label is a target label independent of the appearance of the other labels. Notice that no assumption is made on how the candidates are selected, so they can be quite correlated.

4. Transfer of Performance

We now describe our approach to transfer label ranking with ambiguous supervision to our target tasks. In the absence of prior knowledge on the target task, and more precisely on the distribution of Z, we cannot expect more than learning to rank the labels i according to $\mathbb{P}(Y_i = 1 |\, X = x)$. By Lemma 1 (exchanging Z with Y) we could attempt to minimize the risk defined by $\mathbb{E}\big[\ell^{\text{Full}}(h, X, Y)\big]$. However, minimizing the corresponding empirical risk on the whole training set can be inefficient when $\mathcal{Y}$ is large (as illustration, for Weather, $|\mathcal{X}| \approx 10^5$ lines and $|\mathcal{Y}| \approx 10^6$ records).

4.1. Ranking Risk for Training

In this work, we thus consider the following alternative, which measures the ability of a LSF to rank the candidate labels higher than a *random sample* of $\mathcal{Y}$:

[1] In practice, the risks are normalized in order to be equal to 1 in the worst case. We use the same normalization factor P in all our definitions as it simplifies the notations and does not essentially change the results.

Proposition 2 *For any LSF h, the* Ambiguous Label Ranking Risk *of h, denoted $\mathcal{R}^{\text{Amb}}(h)$, is defined by:*

$$\mathcal{R}^{\text{Amb}}(h) = \mathbb{E}\big[\ell^{\text{Amb}}\big(h, X, Y^+, Y^-\big)\big]$$

where Y^+ and Y^- are $\{0,1\}$-valued r.v. defined by:

$$Y^+ = Y \;\; and \;\; \mathbb{P}\big(Y^-=1 \big| Y=y, X=x\big) = \mathbb{P}\big(Y^-=1\big) = \frac{s}{|\mathcal{Y}|}$$

and, for any x, y^+, y^-:

$$\ell^{\text{Amb}}\big(h, x, y^+, y^-\big) = \frac{1}{P} \sum_{i,j: i\neq j} \mathtt{I}_{\{y_i^+=1\}} \mathtt{I}_{\{y_j^-=1\}} \mathtt{I}_{\{\tilde{h}_{ij}(x)<0\}}$$

Then, denoting $\eta_i^{\text{Amb}}(x) = \mathbb{P}(Y_i = 1 | X = x)$ we have, for any LSF h:

$$\mathcal{R}^{\text{Amb}}_{bayes} \leq \mathcal{R}^{\text{Amb}}(h) \;\; where \;\; \mathcal{R}^{\text{Amb}}_{bayes} = \mathcal{R}^{\text{Amb}}\big(\eta^{\text{Amb}}\big) \,.$$

In the pointwise loss $\ell^{\text{Amb}}(h, x, y^+, y^-)$, y^+ corresponds to the candidate set for x, and y^-, called the *negative set*, is a random subsample of $\mathcal{Y}$. One way to create it is simply to randomly sample s labels of $\mathcal{Y}$ without replacement, where s is called the *size parameter*. In practice, given the training data $(x^k, y^k)_{k=1}^n$, we create a random subsample $y^{k,-}$ of size s for each k, and apply an existing algorithm for ranking. To be valid, Proposition 2 requires the candidate and the negative sets to be totally independent. Hence, nothing forbids the same label to appear in both sets.

4.2. Coherent Supervision and Task-Transfer

We now address the following issue: to what extent is the performance of the function learnt by minimizing the (empirical) Ambiguous Label Ranking Risk transferred to the target tasks? Such a task-transfer is bound to the following notion of *coherence* between the ambiguous supervision and an arbitrary LSF:

Definition 1 (Coherence) *Denote $\lfloor t \rfloor_+$ the positive part of t. The ambiguous supervision is* coherent *with the LSF ρ if there is $\alpha > 0$ such that, for any x, i, j:*

$$\big\lfloor \eta_i^{\text{Amb}}(x) - \eta_j^{\text{Amb}}(x) \big\rfloor_+ \geq \alpha \big\lfloor \rho_i(x) - \rho_j(x) \big\rfloor_+ .$$

Thus, η^{Amb} is coherent with a LSF when it preserves the relative ordering of labels as well as the relative differences of scores. Our main result is that coherence with one of the Bayes-optimal LSF defined in Lemma 1 implies that the ranking performance on the ambiguous task defined by Proposition 2 is transferred to the corresponding target task:

Theorem 3 (Coherence implies transfer) *If η^{Amb} is coherent with η^{Full}, then there is a constant $\beta^{\text{Full}} > 0$ such that, for any LSF h, we have:*

$$\mathcal{R}^{\text{Amb}}(h) - \mathcal{R}^{\text{Amb}}_{bayes} \geq \beta^{\text{Full}} \big(\mathcal{R}^{\text{Full}}(h) - \mathcal{R}^{\text{Full}}_{bayes}\big).$$

Moreover, under the conditional independence assumption of Lemma 1, if η^{Amb} is coherent with η^{CSet}, then there is $\beta^{\text{CSet}} > 0$ such that for any LSF h:

$$\mathcal{R}^{\text{Amb}}(h) - \mathcal{R}^{\text{Amb}}_{bayes} \geq \beta^{\text{CSet}} \big(\mathcal{R}^{\text{CSet}}(h) - \mathcal{R}^{\text{CSet}}_{bayes}\big).$$

The constants in the theorem increase with the value of α of Definition 1 and decrease with the size parameter s of Proposition 2. Roughly speaking, η^{Amb} is coherent with η^{Full} when the most frequent candidate labels are also the most frequent target labels and, η^{Amb} is coherent with η^{CSet} when the most frequent candidate labels are supposed to be top-ranked in the candidate sets they appear. In any case, when the supervision is coherent with a Bayes-optimal LSF of Lemma 1, there is a strong transfer of performance: an approximately optimal LSF for the ambiguous ranking risk is also approximately optimal for the target task. Let us now discuss this result on our applications.

RoboCup Sportscasting In this dataset, most sentences have a single possible target label. Thus, for a sentence x, there is a single $i^*(x)$ such that $\eta^{\text{Full}}_{i^*(x)}(x) > 0$. Note that it can happen that $\eta^{\text{Full}}_{i^*(x)}(x) \neq 1$ because the target set is empty rather often (incomplete supervision). Yet, as soon as $i^*(x)$ is in the candidate set, we are sure it is also in the target set, so $\eta^{\text{CSet}}_{i^*(x)}(x) = 1$, and for any $j \neq i^*(x)$, $\eta^{\text{Full}}_j(x) = \eta^{\text{CSet}}_j(x) = 0$. Hence, by Definition 1, η^{Amb} is coherent with both η^{Full} and η^{CSet}, if there exists $\epsilon > 0$ such that $\eta^{\text{Amb}}_{i^*(x)}(x) > \eta^{\text{Amb}}_j(x) + \epsilon$ for all x and $j \neq i^*(x)$. This assumption only requires the correct label to be the most frequent in the candidate sets, without any other constraint on the remaining labels. It is very likely to be verified in the dataset, so we should be able to learn a LSF performing a transfer to both the Full Ranking and the Candidate Set Ranking tasks.

Weather Forecasting This dataset is noisy, i.e. $\eta^{\text{Full}}_j(x) > 0$ for many labels j, and we have no way to decide whether η^{Amb} can be coherent with η^{Full}. However, given a sentence x *and* a candidate set y, a candidate label i is either correct or incorrect, regardless of the other labels, so $\eta^{\text{CSet}}_i(x) = 0$ or $\eta^{\text{CSet}}_i(x) = 1$. Following the same reasoning as for RoboCup, η^{Amb} is coherent with η^{CSet} as soon as labels i with $\eta^{\text{CSet}}_i(x) = 1$ appear more frequently in the candidate sets of x than those with $\eta^{\text{CSet}}_i(x) = 0$. This is probably true because, in spite of the structure of the candidate sets, records are not totally correlated given x. We might be able to learn the Candidate Set Ranking.

Related Work Jin & Ghahramani (2003) and Hüllermeier & Beringer (2005) tackle the problem of learning from ambiguously labeled examples, but only via empirical evidence. Cour et al. (2009) propose a proxy risk for multiclass classification under ambiguous supervision, and prove a result similar in essence to Theorem 3 for that case. We can notice two fundamental differences with our approach. First, the formulation of their result is strictly weaker, since they do not prove that the Bayes-optimal point of their proxy risk is also Bayes-optimal for their target risk, even under strong assumptions similar to our notion of coherence. Secondly, their framework for ambiguous supervision does not treat the case of incomplete supervision.

5. Practical Ranking Setup

To be more concrete, we now describe how we employed ranking to learn semantic correspondences.

5.1. Learning Model

The training data is a set of pairs $(x^k, y^k)_{k=1}^n$. On RoboCup, x^k is a sentence and y^k the candidate set. On the Weather dataset, x^k is a line, and y^k is the candidate set associated to the line's scenario. On the latter dataset, the candidate sets have many uninformative records[2] which are constantly expressed albeit never correct. As they artificially introduce ambiguity, we discard them from both the training and test sets.

To apply Proposition 2, we define, for each k, $y^{k,+}{=}y^k$; and create the negative set $y^{k,-}$ by sampling a number $s.n_k$ of random records (without replacement) among all the records present in the data, with $n_k = \left|y^k\right|$. The hyperparameter s is actually employed as a multiplicative factor of the size of the candidate set. Given a joint feature function ϕ (discussed below) of (text, record) pairs, we learn a linear LSF with a regularized convex relaxation of the empirical risk corresponding to $\mathcal{R}^{\text{Amb}}$ similar to SVM^{rank} (Joachims, 2006):[3]

$$\min_w \frac{1}{2}||w||^2 + \sum_{k=1}^n \frac{C}{sn_k^2} \sum_{\substack{i:y_i^{k,+}=1\\ j:y_j^{k,-}=1}} \left\lfloor 1 - \langle w, \phi(x^k,i) - \phi(x^k,j)\rangle \right\rfloor_+$$

In the experimental section, we refer to this learning model as the ARank algorithm.

5.2. Feature Representations

We use slightly different feature systems to encode the texts and records of each task in the function ϕ.

Table 1. Datasets and parameters used in the experiments.

	RoboCup	Weather
Scenarios	1,872	2,2146
Records (per scenario)	2.4	36.0
Gold align. (per scenario)	0.8	5.8
Negative set size: s	×50	×1
SVM regularizer: C	10^{-2}	10^{-5}

RoboCup Sportscasting For this dataset, each commentary $x \in \mathcal{X}$ is encoded using a binary vector based on a bag of its words, bigrams and trigrams. Similarly, each record is characterized by a binary vector indicating its type and its different categorical field values. The joint representation $\phi(x, y)$ of a sentence x and a record y is then obtained by performing an outer product of their respective encoding vectors.

Weather Forecasting We also employ bag of word representations (including bigrams and trigrams) for the sentences, and binary ones for the records. However, some extra-characteristics must be added. First, special care must be taken with the integer valued record fields. Following Liang et al. (2009), we incorporate to ϕ features that express the crucial information of whether a word m matches the value of a record field (e.g. in the example of Figure 1, the number 51 corresponds to a word of the 3^{rd} line and a field of the `temperature` record). We also consider cases where an approximation of the record value might be used in the text, in place of the exact one. So we check for m, m rounded to 5 (up/low), m+/-1 and m+/-2. Then, we add an extra-feature indicating whether the *time* field value of a record corresponds to the majority value of the candidate records of $y^{k,+}$.

5.3. Prediction Strategies

In the next section, we evaluate our model with a ranking measure similar to the Candidate Set Ranking loss. However, we also have to set up a prediction process in order to compare its performance with previous works using a precision/recall measure. For a test scenario, a prediction consists in returning the target set. As our algorithm cannot directly determine the number of records to pick out, we used post-processing heuristics. On the RoboCup dataset, we have to detect when the target set is empty, and for the Weather dataset, we need to predict its size (1 or 2).

In a RoboCup scenario, a single record is correct and if it is not a candidate, then the target set is empty. As we expect to operate the transfer to the Full Rank-

[2] The records with only *time* and *mode*=-- arguments.

[3] One may note that in Proposition 2, the normalization factor and the negative set size are constant, while they may vary. It is a matter of implementation, with no real influence as these values are close to their means. Likewise, the mean of sn_k^2 is not P, but it only changes the C scale.

Table 2. **Top-k ranking accuracies on both datasets.** Results (in %) obtained by 4-fold cross-validations.

Model	RoboCup	Weather
Random Baseline	59.0	9.6
ARank	**91.9**	**75.7**

ing (see Section 4.2), we use it to define the following decision rule: for a test scenario, we rank a random subset of $\mathcal{Y}$ (mixing the n candidates and $s.n$ negative records). We return the top-ranked record if it belongs to the candidate set, and the empty set otherwise.

For the Weather dataset, we must take into account that the same sentence can have different target records depending on its candidate set. Since in Section 4.2 is suggested that we are likely to learn the Candidate Set Ranking, we establish a decision rule based on it. We propose these two simple heuristics. For every scenario, (1) we rank among the candidate set only and always return the top-ranked element, (2) if this record is wind-related (i.e. with the type `windSpeed`, `windDir` or `windChill`), we also return the next wind-related element of the ranked list.

6. Experiments

In this section, we evaluate our ranking formulation, ARank, and compare it to two reference models for the tasks presented in Section 2.

The choice of the hyperparameters C and s has a non-trivial influence on the performances of ARank. However, the way to set them for ambiguously supervised systems is an open issue at the moment, because one cannot conduct any direct evaluation without using gold alignments. As this paper mainly targets to introduce our ranking approach, we leave this question for future work. We did not perform any exhaustive search but tried several reasonable values: 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1} for C and 1, 10, 50 for s. In the following we present the results obtained with the values listed in Table 1 (we kept the same value for all experiments concerning a dataset).

6.1. Ranking Evaluation

We first assess the ability of ARank to discover the hidden ranking within the candidate set. Splitting both datasets in four, our model is evaluated by cross-validation[4] using a ranking metric: the *top-k accuracy.* For a given scenario, if k alignments must be predicted, we measure the proportion of these belonging to the top k elements of the list returned by the algorithm, and average that for all testing scenarios. The results displayed in Table 2, clearly indicate that ARank actually learns to correctly rank the candidates, even for a complex task like weather forecasting for which the random baseline is very low.

Table 3. **$\mathbf{F_1}$ scores on RoboCup** obtained by 4-fold cross-validation as defined in (Chen & Mooney, 2008).

Model	F_1
Random Baseline	48.0
Krisper	67.0
Liang et al. (2009)	75.7
ARank	**83.0**

6.2. Alignment Comparison

In the literature, the standard evaluation metric for semantic matching is a F_1 score based on the number of actual gold alignments detected among the predictions. In this section, we use it to compare the predictions performed by ARank using the strategies defined in Section 5.3 to those of two state-of-the-art methods.

Baselines The first one is Krisper (Kate & Mooney, 2007) which obtained the best results on RoboCup in (Chen & Mooney, 2008). This algorithm works by repeatedly building noisy, unambiguous datasets from the ambiguous one, and training a parser designed for unambiguous supervision only. Recently, Liang et al. (2009) proposed a hierarchical hidden semi-Markov model for learning under ambiguous supervision directly. Their generative approach models the correspondences between text and records using latent variables and is trained with a sophisticated 3-stages process based on EM. They achieve the best current performances on both RoboCup and Weather.

Results In Table 3, we provide cross-validation scores on RoboCup. They express that ARank, despite its simple prediction strategy, attains strong performance behavior and outperforms both baselines, thanks to the good quality of the learnt ranking function. On the Weather data, ARank reaches a F_1 score of 76.4 in cross-validation but there is no cross-validated comparison available in the literature.

Indeed, in (Liang et al., 2009), in addition to cross-validation, another evaluation scheme is proposed for which models are both trained and tested on all scenarios. For both tasks, we report results in this setting in

[4] We used the split of (Chen & Mooney, 2008) for RoboCup and a random one for Weather.

Table 4. **Alignment results on both datasets.** Following Liang et al. (2009), these results were obtained by training and testing on all scenarios. The table displays F_1 scores as well as [precision/ recall] values.

Model	RoboCup	Weather
Liang et al.	80.5 [77.3/ 84.0]	75.0 [76.3/ 73.8]
ARank	**83.7** [76.6/ 92.3]	**76.6** [78.0/ 75.3]

Table 4. They demonstrate that ARank remains very competitive. We can notice that its performance for both evaluation schemes are somewhat similar, unlike the method of Liang et al. (2009) which loses almost 5% when being cross-validated on RoboCup.

Candidate Set vs Full Ranking In Section 5.3, we explain that for Weather we employ the transfer to the Candidate Set Ranking to predict, implementing a strategy involving only candidates. In fact, if we switch to a prediction strategy based on the more natural Full Ranking and similar to the one used for RoboCup (involving mix of candidate and negative records), F_1 score drops from 76.6 to 72.4, mostly because the recall decreases from 75.3 to 68.8. This means that, for several lines, no record is predicted because a negative one is top-ranked. This is not surprising because the Weather dataset is noisy: many lines (e.g. "mostly cloudy") can refer to different records which can be only discriminated if the candidate set is known. Hence, relying on the standard Full Ranking is not always the most appropriate approach.

7. Conclusion

This paper casts a new light on the task of learning under ambiguous supervision: we demonstrated that solving a derived label ranking problem allows to perform a transfer of performance to the original task. As illustration, we empirically validated the efficiency of this approach by proposing a concrete application for learning semantic correspondences which happens to be very competitive with state-of-the-art methods.

Acknowledgments

Part of this work was supported by the Network of Excellence PASCAL2 and by the French DGA.

References

Angelova, A., Matthies, L., Helmick, D. M., and Perona, P. Dimensionality Reduction Using Automatic Supervision for Vision-Based Terrain Learning. In *Robotics: Science and Systems*. MIT Press, 2007.

Branavan, S.R.K., Chen, H., Zettlemoyer, L., and Barzilay, R. Reinforcement Learning for Mapping Instructions to Actions. In *Proc. of the 47th An. Meeting of the ACL*, 2009.

Chen, D.L. and Mooney, R.J. Learning to Sportscast: A Test of Grounded Language Acquisition. In *Proc. of the 25th Intl Conf. on Mach. Learn.*, 2008.

Cour, T., Jordan, C., Miltsakaki, E., and Taskar, B. Movie/Script: Alignment and Parsing of Video and Text Transcription. In *Proc. of the 10th European Conf. on Computer Vision*, 2008.

Cour, T., Sapp, B., Jordan, C., and Taskar, B. Learning from Ambiguously Labeled Images. In *Proc. of IEEE Conf. on Comp. Vision and Pat. Recog.*, 2009.

Ge, R. and Mooney, R. J. Learning a Compositional Semantic Parser using an Existing Syntactic Parser. In *Proc. of the 47th An. Meeting of the ACL*, 2009.

Har-Peled, S., Roth, D., and Zimak, D. Constraint Classification for Multiclass Classification and Ranking. In *Adv. in Neur. Inf. Proc. Syst.*, 2002.

Hüllermeier, E. and Beringer, J. Learning from Ambiguously Labeled Examples. In *Adv. in Intelligent Data Analysis*, 2005.

Jin, R. and Ghahramani, Z. Learning with Multiple Labels. In *Adv. in Neur. Inf. Proc. Syst.*, 2003.

Joachims, T. Training Linear SVMs in Linear Time. In *Proc. of the 12th ACM Intl Conf. on Knowledge Discovery and Data Mining*, 2006.

Kate, R.J. and Mooney, R.J. Learning Language Semantics from Ambiguous Supervision. In *Proc. of the 22nd AAAI Conf. on Artif. Intel.*, 2007.

Liang, P., Jordan, M. I., and Klein, D. Learning Semantic Correspondences with Less Supervision. In *Proc. of the 47th An. Meeting of the ACL*, 2009.

Mooney, R.J. Learning Semantic Parsers: An Important But Under-Studied Problem. In *Proc. of the 19th AAAI Conf. on Artif. Intel.*, 2004.

Snyder, B. and Barzilay, R. Database-text Alignment via Structured Multilabel Classification. In *In Proc. of the 20th Intl Joint Conf. on Artif. Intel.*, 2007.

Zettlemoyer, L. and Collins, M. Learning Context-Dependent Mappings from Sentences to Logical Form. In *Proceedings of the 47th Annual Meeting of the ACL*, 2009.