Enhanced discriminative models with tree kernels and unsupervised training for entity detection

Lina Maria Rojas Barahona, Christophe Cerisara

Enhanced discriminative models with tree kernels and unsupervised training for entity detection

Lina M. Rojas-Barahona
Université de Lorraine-LORIA
Nancy, France
Email: lina.rojas@loria.fr

Christophe Cerisara
CNRS-LORIA
Nancy, France
Email: christophe.cerisara@loria.fr

***Abstract*—This work explores two approaches to improve the discriminative models that are commonly used nowadays for entity detection: tree-kernels and unsupervised training. Feature-rich classifiers have been widely adopted by the Natural Language processing (NLP) community because of their powerful modeling capacity and their support for correlated features, which allow separating the expert task of designing features from the core learning method. The first proposed approach consists in leveraging the fast and efficient linear models with unsupervised training, thanks to a recently proposed approximation of the classifier risk, an appealing method that provably converges towards the minimum risk without any labeled corpus. In the second proposed approach, tree kernels are used with support vector machines to exploit dependency structures for entity detection, which relieve designers from the burden of carefully design rich syntactic features manually. We study both approaches on the same task and corpus and show that they offer interesting alternatives to supervised learning for entity recognition.**

***Index Terms*—Entity recognition, Tree Kernels, Unsupervised Learning.**

I. Introduction

The goal of this work is to detect entities, with a focus on proper nouns, in French text documents. Entity detection is classically realized in the state-of-the-art with a sequence discriminative model, such as a conditional random fields (CRF), which can exploit rich input features typically derived from the words to tag, its surrounding words (linear context) and gazettes, which are list of known entities. This traditionnal approach is highly efficient, but still has to face some important issues, in particular:

- The cost incurred to manually annotate a large enough training corpus;
- The fact that the input features do not exploit the intrinsic linguistic structure of the sentences to tag, despite its fundamental importance for interpreting and relating the surface words together.

We propose next to address both issues, with two original approaches. The first one proposes to use tree kernels within a baseline support vector machine (SVM) to exploit the syntactic structure of parse trees of the input sentence for supervised learning, and the second one explores the use of an unsupervised training algorithm on a discriminative linear models, which opens new research directions to reduce the requirement of prior manual annotation of a large training corpus.

Section II presents the target unsupervised approach, discusses the general issue of how to train discriminative models and some of the solutions proposed in the state-of-the-art, and describes our proposed adaptation of the algorithm for entity detection. Section III presents tree kernels, while Section IV shows how tree kernels can be used to exploit the syntactic structures for entity detection. Section V presents experimental validations of both approaches on the same broadcast news corpus in French. Section VI briefly summarizes some of the related works in the litterature, and Section VII concludes the paper.

II. unsupervised training

A. Context

Unsupervised training of discriminative models poses serious theoretical issues, which prevent such models from being widely adopted in tasks where annotated corpora do not exist. In such cases, generative models are thus often preferred. Nevertheless, discriminative models have various advantages that might be desirable even without supervision, for example their very interesting capacity to handle correlated features and to be commonly equipped with many rich features. Hence, many efforts have been deployed to address this issue, and some unsupervised training algorithms for discriminative models have been proposed in the Natural Language Processing (NLP) community, for instance Unsearn [1], Generalized Expectation [2] or Contrastive Training [3] amongst others.

Our unsupervised approach relies on a novel approximation of the risk of binary linear classifiers proposed in [4]. This approximation relies on only two assumptions: the rank of class marginal is assumed to be known, and the class-conditional linear scores are assumed to follow a Gaussian distribution. Compared to previous applications of unsupervised discriminative training methods to NLP tasks, this approach presents several advantages: first, it is proven to converge towards the true optimal classifier risk; second, it does not require any constraint; third, it exploits a new type of knowledge about class marginal that may help convergence towards a relevant solution for the target task. In this work, we adapt and validate the proposed approach on two new binary NLP tasks: predicate identification and entity recognition.

B. Classifier risk approximation

We first briefly review the approximation of the risk proposed in [4]. A binary (with two target classes: 0 and 1) linear classifier associates a score $f_{\theta^0}(X)$ to the first class 0 for any input $X = (X_1, \cdots, X_{N_f})$ composed of N_f features X_i:

$$f_{\theta^0}(X) = \sum_{i}^{N_f} \theta_i X_i$$

where the parameter $\theta_i \in \mathbb{R}$ represents the weight of the feature indexed by i for class 0. As it is standard in binary classification, we constrain the scores per class to sum to 0:

$$f_{\theta^1}(X) = -f_{\theta^0}(X)$$

In the following, we may use both notations $f_{\theta^0}(X)$ or $f_\theta(X)$ equivalently. X is classified into class 0 iff $f_{\theta^0}(X) \geq 0$, otherwise X is classified into class 1. The objective of training is to minimize the classifier risk:

$$R(\theta) = E_{p(X,Y)}[\mathcal{L}(Y, f_\theta(X))] \quad (1)$$

where Y is the true label of the observation X, and $\mathcal{L}(Y, f_\theta(X))$ is the loss function, such as the hinge loss used in SVMs, or the log-loss used in CRFs. This risk is often approximated by the empirical risk that is computed on a labeled training corpus. In the absence of labeled corpus, an alternative consists in deriving the true risk as follows:

$$R(\theta) = \sum_{y \in \{0,1\}} P(y) \int_{-\infty}^{+\infty} P(f_\theta(X) = \alpha | y) \mathcal{L}(y, \alpha) d\alpha \quad (2)$$

We use next the following hinge loss:

$$\mathcal{L}(y, \alpha) = (1 + \alpha_{1-y} - \alpha_y)_+ \quad (3)$$

where $(x)_+ = \max(0, x)$, and $\alpha_y = f_{\theta^y}(X)$ is the linear score for the correct class y. Similarly, $\alpha_{1-y} = f_{\theta^{1-y}}(X)$ is the linear score for the wrong class.

Given y and α, the loss value in the integral can be computed easily. Two terms in Equation 2 remain: $P(y)$ and $P(f_\theta(X) = \alpha | y)$. The former is the class marginal and is assumed to be known. The latter is the class-conditional distribution of the linear scores, which is assumed to be normally distributed. This implies that $P(f_\theta(X))$ is distributed as a mixture of two Gaussians (GMM):

$$P(f_\theta(X)) = \sum_{y \in \{0,1\}} P(y) \mathcal{N}(f_\theta(X); \mu_y, \sigma_y)$$

where $\mathcal{N}(z; \mu, \sigma)$ is the normal probability density function. The parameters $(\mu_0, \sigma_0, \mu_1, \sigma_1)$ can be estimated from an unlabeled corpus $\mathcal{U}$ using a standard Expectation-Maximization (EM) algorithm for GMM training. Once these parameters are known, it is possible to compute the integral in Eq. 2 and thus an estimate $\hat{R}(\theta)$ of the risk without relying on any labeled corpus. The authors of [4] prove that:

- The Gaussian parameters estimated with EM converge towards their true values;
- $\hat{R}(\theta)$ converges towards the true risk $R(\theta)$;
- The estimated optimum converges towards the true optimal parameters, when the size of the unlabeled corpus $\mathcal{U}$ increases infinitely:

$$\lim_{|\mathcal{U}| \to +\infty} \arg\min_\theta \hat{R}(\theta) = \arg\min_\theta R(\theta)$$

They further prove that this is still true even when the class priors $P(y)$ are not known precisely, but only their relative order (rank) is known. These priors must also be different $P(y = 0) \neq P(y = 1)$.

Given the estimated Gaussian parameters, we use numerical integration to compute Eq. 2. We implemented both Monte Carlo [5] and trapezoidal methods for solving numerically Eq. 2.

In the Monte Carlo integration, the integral is evaluated by sampling T points $(\alpha_t)_T$ according to a hypothesized probability distribution $p(\alpha) = P(f_\theta(X))$ and by computing the sum:

$$I = \frac{1}{n} \sum_{t=1}^{n} \frac{P(f_\theta(X) = \alpha_t | y) \mathcal{L}(y, \alpha_t)}{p(\alpha_t)} \quad (4)$$

Where n is the total number of points (i.e., the number of trials). The simplest integration method uses a uniform distribution $p(\alpha) = \frac{1}{(b-a)}$ and the sum in Equation 4 reduces to Equation 5:

$$I = (b - a) \frac{1}{n} \sum_{t=1}^{n} P(f_\theta(X) = \alpha_t | y) \mathcal{L}(y, \alpha_t) \quad (5)$$

a and b are broadly set so as to capture most if not all possible points in the domain of the integral:

$$a = \min(\mu_{y,0}, \mu_{y,1}) - 6 \max(\sigma_{y,0}, \sigma_{y,1})$$
$$b = \max(\mu_{y,0}, \mu_{y,1}) + 6 \max(\sigma_{y,0}, \sigma_{y,1})$$

As is well known in numerical analysis, the trapezoidal rule for computing the same integral uses the following approximation:

$\int_a^b f(x)dx \approx \frac{h}{2}(f(x_0) + 2f(x_1) + ... + 2f(x_{n-1} + f(x_n))$, where $h = \frac{b-a}{n}$

Our unsupervised training algorithm then implements a coordinated gradient descent, where the gradient of the risk is computed with finite difference.

III. Tree Kernels

Kernel methods explore high-dimensional feature spaces on low-dimensional data, alleviating the burden of meticulously designing and extracting rich features. Then, it is possible to detect nonlinear relations between variables in the data by embedding the data into a kernel-induced feature space. A kernel is a similarity function over pairs of objects. Convolution kernels allow to compute this similarity based on the similarity of object parts. Tree kernels for instance, are convolution kernels that measure this similarity by computing the number of common substructures between two trees T_1 and T_2, exploring in this way rich structured spaces.

Tree kernels have been widely used for a variety of NLP applications such as relation extraction [6], [7], semantic role labeling [8] as well as parsing [9] and named-entity recognition re-ranking [10]. We follow here the work of [11], [8] on convolution tree kernels. We explored the following tree spaces: (i) **the subset tree** (SST) kernel and (ii) **the partial tree** (PT) kernel (see Figure 1). The former is defined as a tree rooted in any non-terminal node along with all its descendants in which its leaves can be non-terminal symbols and satisfies

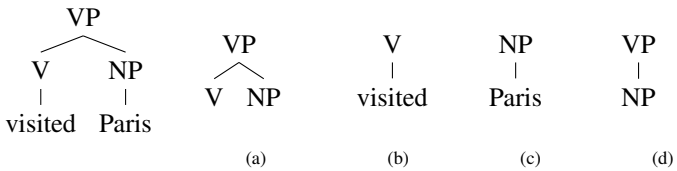


Fig. 1. (a-c) SST subtrees and (d) PT subtree for constituence syntactic trees. Note that in (d) the grammatical rule VP $\rightarrow$ V NP, is broken.

the constraint that grammatical rules cannot be broken. The latter is a more general form of substructure that relax the constraint over the SSTs.

We are interested in studying the impact of using dependency-trees (i.e. a syntactic representation that denotes grammatical relations between words) in tree kernels. Apart from the previously mentioned work on named-entity recognition re-ranking, there is still little work studying the impact of rich syntactic tree structures for the task of entity recognition. Such features would allow the model to take into account the internal syntactic structures of multi-words entities, but also to potentially model the preferred syntactic relations between named entities and their co-occurring words in the sentence. To address these questions, we studied the impact of structured tree-features in supervised models by training and evaluating tree kernel-based models for the binary task of entity detection.

In our experiments we use an optimized SVM implementation of tree-kernels, namely fast-tree kernels [12], in which a compact representation of trees (i.e. a directed acyclic graphs) is used, avoiding processing repeated sub-structures and as a consequence reducing the total amount of computations.

IV. Entity recognition features

The goal of entity recognition is to detect whether any word form in a text refers to *an entity* or not, where an *entity* is defined as a mention of a person name, a place, an organization or a product. We use the ESTER2 corpus [13], which collects broadcast news transcriptions in French that are annotated with named entities. It is worth noting that this corpus contains spontaneous speech, which are characterized by the abundance of irregularities that make it difficult to parse, such as ellipsis, disfluences, false starts, reformulation, hesitations and ungrammaticality (i.e. incomplete syntactical structures) due to pauses and absence of punctuation, as shown in the example presented in Figure 2 (a), in which a comma is missing just before the entity mention.

Vector Features: The following features are used in the unsupervised experiments with a linear classifier. We adapted in these experiments the Stanford supervised linear classifier of the Stanford NLP toolkit [1], in which we added methods to perform risk minimization on an unlabeled corpus. The features used in this context are:

- **Character n-grams** with $n = 4$.
- **Capitalization**: the pattern "Chris2useLC", as defined in Stanford NLP, describing lower, upper case and special characters in words [14].
- **POS tags**: the part of speech tag of every word as given by the Treetagger [15].

The part of speech tags as well as capitalization of words are common important features for entity recognition, while character n-grams constitute a smoother (less sparse) alternative to word forms and are also often used in this context. The label priors $P(y)$ are set so that 90% of the words are not entities and only 10% of the words are entities. The initial weights are obtained after training the linear classifier on 20 manually annotated sentences. The same set of features is used both in the supervised initialization and the unsupervised risk minimization iterations.

Tree Features: The following features are used in the supervised experiments with dependency tree kernels. The input dependency trees have been obtained by automatically parsing the corpus with the MATE Parser [16] trained on the French Tree-Bank [17]. The following features are used for training tree kernels:

- **Top-down tree:** the tree fragment in which the current word is the governor (see Figure 2 (b)).
- **Bottom-up tree:** the tree fragment in which the current word is a dependent (see Figure 2 (c)).

We also consider the following dependency tree variations:

- **Emphasize the current word:** we created another kind of tree by simply introducing a prefix "CW", that stands for current word, in the node of the tree that contains the word in focus.
- **POS-trees:** words in dependency trees are represented by their part of the speech (POS) instead of their word-form.

Therefore, we can combine tree and vector features as well as using either both types of tree-features top-down (TD) and bottom-up (BU) or only one of them.

V. Results

We present in this section the results of our experiments with both the unsupervised and supervised tree-kernel models. We removed from the training set (but not from the tree-structure) all the words that have been annotated with the following part of the speech by the tree tagger: punctuation and determiners.

A. Risk minimization

On the Gaussianity assumption: The proposed approach assumes that the class-conditional linear scores are distributed normally. We invite the interested reader to read [4], where theoretical arguments are given that support the validity of this assumption in various contexts. However, this assumption can not always be taken for granted, and the authors suggest to verify it empirically.

The distributions of $f_\theta(X)$ with the initial and final weights (i.e. the weights obtained after training) on the ESTER2 corpus

[1] http://nlp.stanford.edu/nlp

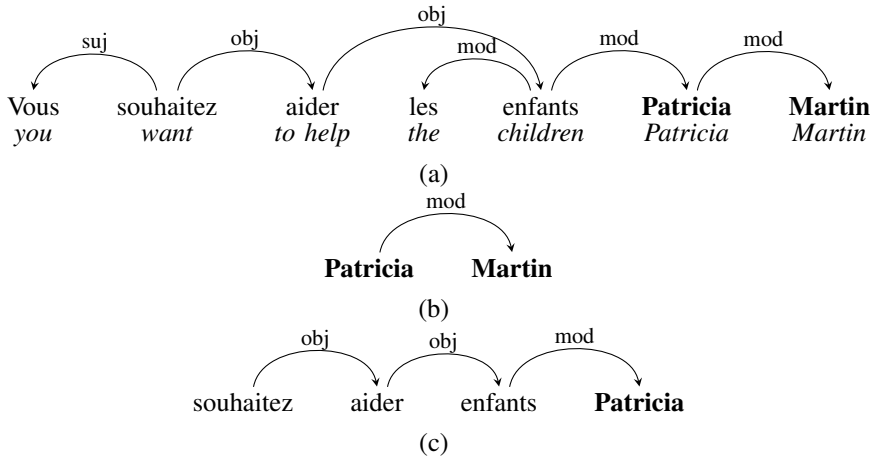
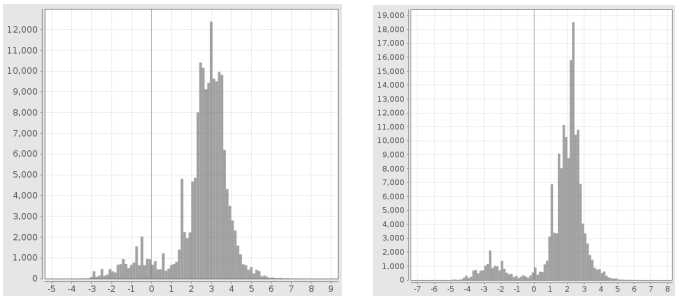


Fig. 2. (a) Dependency tree, (b) top-down and (b) bottom-up tree fragments



(a) (b)

Fig. 3. Distribution of $f_\theta(X)$ on the ESTER2 corpus (unlabeled dataset) (a) using the initial weights trained on 20 sentences; (b) using the weights at the final iteration of the gradient descent algorithm. The largest mode is on the right because the *entity* class is class 1.

Supervised vs Unsupervised			
System	precision	recall	F1
Stanford trained on 20 sent.	89.8%	68%	77.4%
Stanford trained on 520 sent.	90.3%	84.7%	87.5%
Unsupervised trap.	88.7%	**79**%	**83.5**%
Unsupervised MC	88.7%	**79**%	**83.6**%

TABLE I
PERFORMANCE OF THE PROPOSED UNSUPERVISED SYSTEM.

are shown in Figure 3(b) and (c) respectively. These distributions are clearly bi-normal on this corpus, which suggest that this assumption is reasonable in both our NLP tasks.

Experiments with gradient descent: Starting from the initial weights trained on 20 sentences, we now apply the gradient descent algorithm described in [4] that minimizes an estimate of the classifier risk on the full corpus without labels. The next Table reports the entity detection results of the initial linear classifier trained on 20 sentences, but also when trained on the full corpus composed of 520 sentences. This latter results shows the best performances that can be reached when manually labelling a large number of training sentences. The objective of our unsupervised approach is to get as close as possible from these optimal results, but without labels. The metric used in these experiments is the f-measure, which is the harmonic mean of precision and recall.

$$F1 = 2 * \frac{precision * recall}{precision + recall} \quad (6)$$

The unsupervised risk estimation method relies on the continuous integration of the bimodal distribution of the classifier scores on the full corpus, which may be relatively costly to perform especially as this computation is done at every step of the gradient descent.

We have thus made preliminary experiments with two numerical integration approaches: the trapezoidal and Monte Carlo methods [5]. These methods are compared next both in terms of computational cost and approximation quality.

Figure 4 shows the approximation error when using the trapezoidal rule for integration. The x-axis represents the number of parameters of the chosen numerical integration method, i.e., here, the number of trapezoids used. The y-axis represents the squared error between the risk estimated with a nearly infinite precision and the risk estimated with numerical integration and a limited number of parameters. We use the root-square of the approximation error to better view the details, because the trapezoidal and the Monte Carlo methods are known to respectively converge in $O(n^{-2})$ and $O(n^{-\frac{1}{2}})$.

We can observe that increasing the number of trapezoids also increases the accuracy of numerical integration, and that the approximation error becomes smaller than 10% of the risk value for 20 trapezoids and more.

Figure 5 shows a similar curve (on a different figure to have a better precision on the y-axis) but for Monte Carlo integration, where the x-axis represents the number of Monte Carlo iterations.

Note that both Figures 4 and 5 show the risk approximation error, and not the final impact of this error on the entity recognition task: this is rather shown in Table I.

With regard to complexity, Figure 6 shows the computation time, measured in seconds, required to compute the integrals

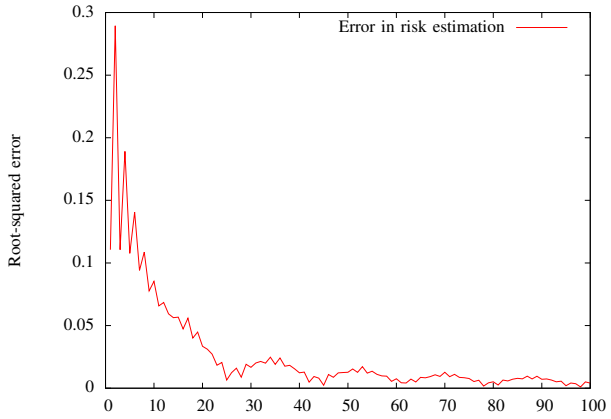


Fig. 4. Approximation error of the trapezoidal rule with regard to the number of trapezoids (i.e., segments) used for numerical integration when computing the unsupervised risk.

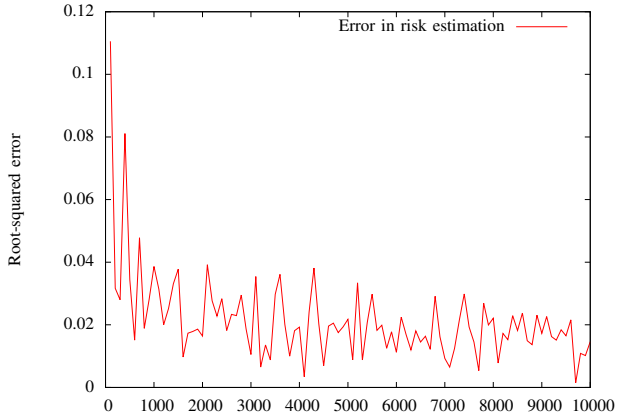


Fig. 5. Approximation error of the Monte Carlo Integration with regard to the number of trials used for approximating the integrals of the risk.

Fig. 6. Computation cost of the trapezoidal rule with regard to the number of trapezoids (i.e., segments) used for numerical integration when computing the unsupervised risk.

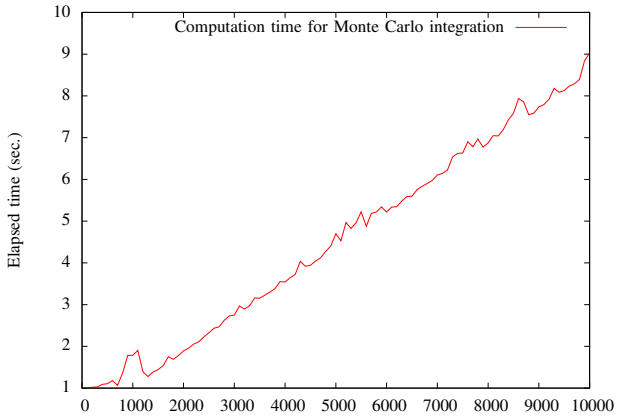


Fig. 7. Computation cost of the Monte Carlo Integration with regard to the number of trials used for approximating the integrals of the risk.

with the trapezoidal rule during risk minimization.

Figure 7 shows a similar curve but with Monte Carlo integration.

The final performance figures are shown in Table I (bottom part). We can observe that the Monte Carlo method takes much more time in our experimental setup, without impacting the final entity detection rate. Indeed, according to our experiments, both the trapezoidal risk and Monte Carlo integration reach the same performances (the differences shown in Table I are not statistically significant) after $2,000$ iterations with an F1-measure of 83.5%.

In the following experiments, we have thus chosen the trapezoidal approach. Figures 8 and 9 respectively show the convergence of the risk estimate with trapezoidal integration and the entity F1-measure as a function of the number of iterations of gradient optimization. Therefore, when evaluated on a test set of $167,249$ words and $10,693$ sentences, both methods outperform the supervised linear classifier trained on 20 sentences.

In general the proposed model is prone to detect person names that are undetected by the baseline (i.e., the Stanford linear classifier trained on 20 sentences). Table II shows two examples of family names (e.g., Drouelle and Floch-Prigent) that are correctly recognized by our model but ignored by the baseline. Our model also correctly detects entities other than person names, such as the fighter aircraft F16, which are not captured by the initial model. Note also that for the first

-	Baseline (Sup. on 20 sents)		Proposed Model	
Word	Class	Prob.	Class	Prob.
Fabrice	Entity	0.94	Entity	**0.99**
Drouelle	NO	0.53	**Entity**	**0.79**
Floch-Prigent	NO	0.58	**Entity**	**0.69**
Iran	Entity	0.66	Entity	**0.82**
F16	NO	0.73	**Entity**	**0.91**

TABLE II
EXCERPT OF EXAMPLES CORRECTLY CLASSIFIED BY THE UNSUPERVISED APPROACH FOR ENTITY RECOGNITION, IMPROVING THE BASELINE (I.E. THE STANFORD LINEAR CLASSIFIER TRAINED ON 20 SENTENCES). THE LAST COLUMN SHOWS THE OUTPUT PROBABILITY OF THE WINNING CLASS.

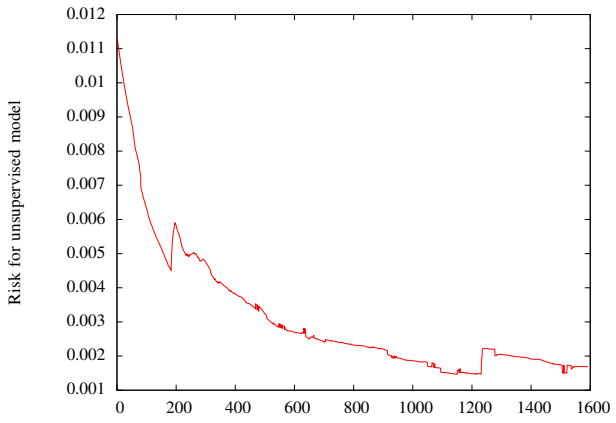


Fig. 8. $\hat{R}(\theta)$ (from Eq. 2) for entity detection, in function of the number of iterations, up to 1600 iterations.

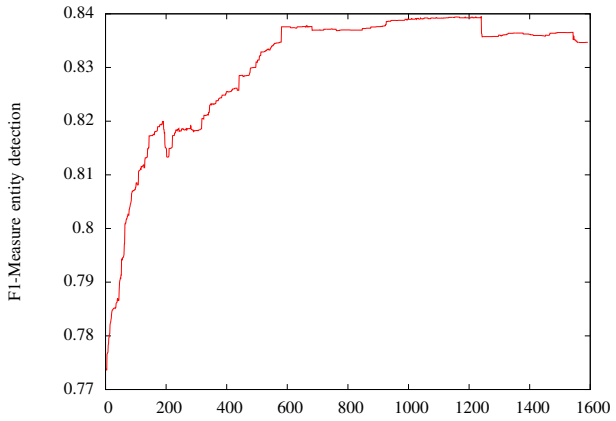


Fig. 9. F1 for entity detection, in function of the number of iterations, up to 1600 iterations

name *Fabrice* and the country *Iran*, the unsupervised model correctly augments their probabilities (where the probabilities correspond to the normalized scores $f_\theta(X)$ given by the model) to belong to the class entity.

B. Experiments with Tree Kernels

We have run experiments on tree kernels using as features top-down (TD), bottom-up (BU) trees as well as using vector features (i.e the same features used in the unsupervised experiments). In our experiments we also used the tree kernel spaces (SST and PT) introduced in Section III. Furthermore, we used either dependency trees or modified dependency trees (as explained in Section IV), in which nodes contain the part of the speech of words instead of the word form.

The baseline is the SVM with linear kernel where only vector features are used for training.

We performed further experiments by introducing or not tree-variations, from which Table III shows a summary. In general PT kernels perform better than SST kernels, in agreement with [11], where they found PT more accurate when using dependency structures. Indeed, STT were mainly thought for constituency trees as they do not have trees with broken grammatical production rules. However, SST tree kernels are more accurate than PT when using POS-trees, suggesting that POS tags behave as non terminals. Although POS tags could help the classifier to capture a more generic tree-structure without having all the word-form variations, word-form dependency trees clearly outperform POS-trees. Bottom-up trees seem to better capture the structural context of entities because entities are more likely to be dependent (leaves) than governors (heads). In fact, bottom up trees increase by (+0.5) and (+0.36) the F1-measure for SST and PT trees respectively. In conclusion, much better results are obtained when combining both top-down and bottom-up trees, especially when using the word in focus or current word (CW) distinction.

VI. Related Work

A number of previous works have already proposed approaches to train discriminative models without or with few labels. Please refer, e.g., to [18], [19] for a general and theoretical view on this topic. For NLP tasks several approaches have also been proposed. Hence, the traditional self- and co-training paradigm can be used to leverage supervised classifiers with unsupervised data [20], [21]. [2] exploit the Generalized Expectation objective function, which penalizes the mismatch between model predictions and linguistic expectation constraints. In contrast, our proposal does not use any manually defined prototype nor linguistic constraint.

Another interesting approach is *Unsearn* [1], which predicts the latent structure Y and then a corresponding "observation" $\hat{X}$, with a loss function that measures how well $\hat{X}$ predicts X. This method is very powerful and generalizes the EM algorithm, but its performances heavily depend on the quality of the chosen features set for discriminating between the target classes. A related principle is termed "Minimum Imputed Risk" in [22] and applied to machine translation. Our proposed approach also depends on the chosen features, but in a less crucial way thanks to both new assumptions, respectively the known label priors and discrimination of classes based on individual Gaussian distributions of scores. Another interesting generalization of EM used to train log-linear models without labels is *Contrastive Estimation*, where the objective function is modified to locally remove probability mass from implicit negative evidence in the neighborhood of the observations and transfer this mass onto the observed examples [3].

Comparatively, the main advantage of our proposed approach comes from the fact that the algorithm optimizes the standard classifier risk, without any modification nor constraint. The objective function (and related optimal parameters) is thus the same as in classical supervised training.

The authors of [23], [24] state the problem of considering syntactic structures for named entity detection as a joint optimization of the two tasks, parsing and named-entity recognition. Although this is a sophisticated solution that avoid cascade errors, the cost of optimizing joint models is high while the improvement is still modest with respect to performing both tasks in a pipeline. Other works exploit

K. Space	Features	precision	recall	F1
Linear	Vector	89.48%	79.56%	84.23%
SST	TD trees + vector	93.55%	76.60%	**84.23**%
PT	TD trees + vector	93.07%	77.49%	**84.57**%
SST	TD POS-trees + vector	88.72%	75.39%	81.51%
PT	TD POS-trees + vector	82.85%	75.43%	78.97%
SST	TD CW trees + vector	94.09%	76.59%	**84.44**%
PT	TD CW trees + vector	93.56%	77.73%	**84.91**%
SST	BU trees + vector	93.20%	77.67%	**84.73**%
PT	BU trees + vector	93.35%	77.91%	**84.93**%
SST	BU POS-trees + vector	85.41%	71.48%	77.82%
PT	BU POS-trees + vector	85.25%	68.31%	75.85%
SST	BU CW trees + vector	93.19%	77.22%	**84.46**%
PT	BU CW trees + vector	93.07%	78.09%	**84.92**%
SST	TD and BU trees + vector	94.17%	77.71%	**85.15**%
PT	TD and BU trees + vector	94.24%	78.51%	**85.66**%
SST	TD and BU POS-trees + vector	89.95%	76.08%	82.43%
PT	TD and BU POS-trees + vector	85.89%	74.75%	79.93%
SST	TD and BU CW-trees + vector	94.17%	78.18%	**85.43**%
PT	TD and BU CW-trees + vector	94.26%	78.75%	**85.81**%
SST	TD and BU CW-POS-trees + vector	90.70%	74.48%	81.79%
PT	TD and BU CW-POS-trees + vector	86.73%	74.19%	79.97%

TABLE III
PERFORMANCE OF THE TREE-KERNELS.

tree-kernels for named-entity recognition re-ranking [9], [10]. The authors of [25] further use tree-kernels for named entity recognition, however they do not use STT nor PT kernels. They rather introduced a different tree kernel, the sliding tree kernel, but which may not be convolutive as the SST and PT kernels.

VII. CONCLUSION

This work explores two original solutions to improve traditional discriminative classifiers used in the task of entity detection. These solutions address two classical problems of traditional named entity detection systems: the high cost required to manually annotate a large enough training corpus; and the limitations of the input features, which often encode linear word contexts instead of the more linguistically relevant syntactic contexts. The former problem is addressed by adapting a newly proposed unsupervised training algorithm for discriminative linear models. At the contrary to other methods proposed in the litterature to train discriminative models without supervision, this approach optimizes the same classifier risk than the one approximated by a supervised classifier trained on a corpus with labels, hence ultimately leading theoretically to the same optimal solution. We thus demonstrate the applicability of this approach to the entity detection NLP task, and further study the computational complexity of two numerical integration approaches in this context. We also show that the main assumption of the approach, i.e., the gaussianity of the class-conditional distributions of the linear scores, is fulfilled in this task. The latter problem is addressed by considering rich structured input features to a SVM, thanks to an adapted tree-kernel that exploits dependency graphs that are automatically computed on broadcast news sentences. Both approaches are validated on the same French corpus for entity detection and exhibits interesting and encouraging performances, which suggest that there is still room for improvement in the task of entity detection thanks to more linguistically rich features, and to unsupervised training on larger unlabeled corpora.

ACKNOWLEDGMENT

This work was partly supported by the French ANR (*Agence Nationale de la Recherche*) funded project ContNomina.

REFERENCES

[1] H. Daumé III, "Unsupervised search-based structured prediction," in *Proc. of ICML*, Montreal, Canada, 2009.

[2] G. Druck, G. Mann, and A. McCallum, "Semi-supervised learning of dependency parsers using generalized expectation criteria," in *Proc. of ACL*, Suntec, Singapore, Aug. 2009, pp. 360–368.

[3] N. A. Smith and J. Eisner, "Unsupervised search-based structured prediction," in *Proc. of ACL*, 2005.

[4] K. Balasubramanian, P. Donmez, and G. Lebanon, "Unsupervised supervised learning II: Margin-based classification without labels," *Journal of Machine Learning Research*, vol. 12, pp. 3119–3145, 2011.

[5] H. Gould and J. Tobochnik, *An introduction to computer simulation methods: applications to physical systems*, ser. Addison-Wesley series in physics. Addison-Wesley, 1988, no. v. 1-2. [Online]. Available: http://books.google.fr/books?id=JfpQAAAAMAAJ

[6] C. M. Cumby and D. Roth, "On kernel methods for relational learning," in *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, 2003, pp. 107–114. [Online]. Available: http://www.aaai.org/Library/ICML/2003/icml03-017.php

[7] A. Culotta and J. Sorensen, "Dependency tree kernels for relation extraction," in *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ser. ACL '04. Stroudsburg, PA, USA: Association for Computational Linguistics, 2004. [Online]. Available: http://dx.doi.org/10.3115/1218955.1219009

[8] A. Moschitti, D. Pighin, and R. Basili, "Tree kernels for semantic role labeling," *Computational Linguistics*, 2008.

[9] M. Collins and N. Duffy, "New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 263–270. [Online]. Available: http://dx.doi.org/10.3115/1073083.1073128

[10] T.-V. T. Nguyen and A. Moschitti, "Structural reranking models for named entity recognition," *Intelligenza Artificiale*, vol. 6, pp. 177–190, December 2012.

[11] A. Moschitti, "Efficient convolution kernels for dependency and constituent syntactic trees," in *In European Conference on Machine Learning (ECML*, 2006.

[12] A. Severyn and A. Moschitti, "Fast support vector machines for convolution tree kernels," *Data Min. Knowl. Discov.*, vol. 25, no. 2, pp. 325–357, Sep. 2012. [Online]. Available: http://dx.doi.org/10.1007/s10618-012-0276-8

[13] S. Galliano, G. Gravier, and L. Chaubard, "The ester 2 evaluation campaign for the rich transcription of french radio broadcasts," in *Proc. of INTERSPEECH*, 2009, pp. 2583–2586.

[14] D. Klein, J. Smarr, H. Nguyen, and C. D. Manning, "Named entity recognition with character-level models," in *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, ser. CONLL '03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 180–183. [Online]. Available: http://dx.doi.org/10.3115/1119176.1119204

[15] H. Schmid, "Improvements in part-of-speech tagging with an application to german," in *Proc. Workshop EACL SIGDAT*, Dublin, 1995.

[16] A. Björkelund, L. Hafdell, and P. Nugues, "Multilingual semantic role labeling," in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, ser. CoNLL '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 43–48. [Online]. Available: http://dl.acm.org/citation.cfm?id=1596409.1596416

[17] M.-H. Candito, B. Crabbé, P. Denis, and F. Guérin, "Analyse syntaxique du français : des constituants aux dé pendances," in *Actes de TALN*, Senlis, 2009.

[18] A. Kapoor, "Learning discriminative models with incomplete data," Ph.D. dissertation, Massachusetts Institute of Technology, Feb. 2006.

[19] A. B. Goldberg, "New directions in semi-supervised learning," Ph.D. dissertation, Univ. of Wisconsin-Madison, 2010.

[20] X. Liu, K. Li, M. Zhou, and Z. Xiong, "Enhancing semantic role labeling for tweets using self-training," in *Proc. AAAI*, 2011, pp. 896–901.

[21] R. S. Z. Kaljahi, "Adapting self-training for semantic role labeling," in *Proc. Student Research Workshop, ACL*, Uppsala, Sweden, Jul. 2010, pp. 91–96.

[22] Z. Li, Z. Wang, J. Eisner, S. Khudanpur, and B. Roark, "Minimum imputed-risk: Unsupervised discriminative training for machine translation," in *Proc. of EMNLP*, 2011, pp. 920–929.

[23] J. R. Finkel and C. D. Manning, "Joint parsing and named entity recognition," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics on ZZZ*. Association for Computational Linguistics, 2009, pp. 326–334, computer science, stanford university.

[24] ——, "Hierarchical joint learning: Improving joint parsing and named entity recognition with non-jointly labeled data," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ser. ACL '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 720–728. [Online]. Available: http://dl.acm.org/citation.cfm?id=1858681.1858755

[25] R. Patra and S. K. Saha, "A kernel-based approach for biomedical named entity recognition," *The Scientific World Journal*, vol. 2013, 2013.