Strategies to select examples for Active Learning with Conditional Random Fields

Vincent Claveau, Ewa Kijak

Strategies to select examples for Active Learning with Conditional Random Fields

Vincent Claveau and Ewa Kijak

IRISA - CNRS - Univ. of Rennes 1
Campus de Beaulieu, Rennes, France
{vincent.claveau;ewa.kijak}@irisa.fr

Abstract. Nowadays, many NLP problems are tackled as supervised machine learning tasks. Consequently, the cost of the expertise needed to annotate the examples is a widespread issue. Active learning offers a framework to that issue, allowing to control the annotation cost while maximizing the classifier performance, but it relies on the key step of choosing which example will be proposed to the expert. In this paper, we examine and propose such selection strategies in the specific case of Conditional Random Fields (CRF) which are largely used in NLP. On the one hand, we propose a simple method to correct a bias of some state-of-the-art selection techniques. On the other hand, we detail an original approach to select the examples, based on the respect of proportions in the datasets. These contributions are validated over a large range of experiments implying several datasets and tasks, including named entity recognition, chunking, phonetization, word sense disambiguation.

Keywords: CRF, conditional random fields, active learning, semi-supervised learning, statistical test of proportion

1 Introduction

Many NLP tasks rely on supervised machine learning. Among the commonly used techniques, Conditional Random Fields (CRF) exhibit excellent performance for tasks related to the sequences annotation (tagging, named entity recognition and information extraction, transliteration...). However, as with all supervised approaches, the cost of the sequence annotation needed to train the models is an important criterion to consider. For simple problems, such as labeling parts-of-speech, some studies show that this cost is relatively low [7], but most of the problems mentioned above rather require a very large number of annotations (see Section 5.2).

To reduce, or at least control, this cost, semi-supervised approaches exploit, in addition to annotated examples, non-annotated examples that are more readily available. Among these approaches, Active Learning allows the expert to annotate additional examples iteratively, thereby controlling the compromise between annotation cost vs. performance of the classifier. Thus, a classifier can be learned or improved at each iteration, and can be used to guide the selection of future examples to annotate. In this article, we are interested in this active learning process, and more specifically in the issue of the selection of examples which are provided to the expert, in the particular case of CRF.

Many methods of selection, either generic to any machine learning algorithm or specific to the CRF (Section 2) have already been developed. In this article, we show that some very conventional methods of the state of the art comprise a bias tending to favor the choice of long examples, that is examples that are expensive to annotate. The first contribution of the paper is to propose a simple technique to remove this bias (Section 3). Another contribution is to propose an original selection technique, relying on the data representations used by the CRF, and based on a criterion balancing the proportions of the attributes in the datasets (Section 4). These different proposals are experimentally evaluated on several datasets and traditional tasks of CRF (Section 5).

2 Context and related work

2.1 Basic notions

Conditional Random Fields [9] are undirected graphical models that represent the probability distribution of annotation y on observations x. They are widely used in NLP thanks to their ability to take into account the sequential aspect and rich descriptions of text sequences. They have been successfully used in many tasks casted as annotation problems, and have become standard tools for information extraction, named entity recognition, tagging, etc. [26, 17, 4, 18, inter alia]. In such cases, x is a sequence of letters or words and y the corresponding sequence of labels. In this context, the conditional probability $P(y|x)$ is defined through a weighted sum of so-called feature functions f_j:

$$P(y|x,\theta) = \frac{1}{Z_\lambda(x)} \exp\left(\sum_j \sum_t \lambda_j f_j(x, y_t, y_{t-1}, t)\right)$$

where $Z_\lambda(x)$ is a normalization factor and θ is the vector of λ_j weights. The feature functions are often binary, returning 1 when a certain combination of labels and observations attributes is satisfied, 0 otherwise. They are applied to each position t of the sequence and the weight λ_j reflects their importance to determine the label. It is important to note that in practice the vector x is not considered as a whole, but only some combinations of attributes on observations around the position t in x are considered. These combinations are user-defined, usually indirectly through a set of patterns $\{\mathrm{Pat}_i\}$. They are applied at each position t of each sequence x ($\mathrm{Pat}_i(x,t)$), and with the information of the labels (y_{t-1} and y_t), they define all the possible feature functions.

The learning step for a CRF consists in estimating the weights λ_j from data with known labels. The weights are those that maximize the model log-likelihood on the training (labeled) sequences, for instance with quasi-Newton type algorithms such as L-BFGS [20]. Once learned, applying the CRF model to the new data consists in finding, for a sequence of observations x, the most probable sequence of labels (denoted y^* in the rest of this article), for example with a Viterbi algorithm.

2.2 Semi-supervised learning and active learning

Semi-supervised learning consists in using annotated data (noted $\mathcal{T}$ hereafter) and non-annotated data ($\mathcal{N}$). Its purpose is to reduce the number of annotations and therefore the cost of the annotation, and/or to yield the best classifier performance for a given annotation cost. Different semi-supervised learning approaches have been explored in the context of CRF. Several studies use unlabeled data directly in training the model by modifying the expression of entropy. This change makes the objective function non-concave and therefore requires to adapt the learning process. Another family of approaches consists in adapting the learning and decoding procedures of CRF so that they are able to handle some other knowledge about the sequences rather than completely annotated sequences. For example, this knowledge may be partial annotation of the sequences (labels are known only for a few words [19]). It can also be a priori knowledge on the distribution of labels knowing certain attributes [12]. Although this is not strictly semi-supervised learning, let us mention the work using close techniques exploiting non-annotated data to improve learning on annotated data. For instance, [13] and [6] propose to cluster non-annotated data to build new feature – in this case, word classes – then used to better describe the (labeled) data. In this vein, it is also worth mentioning the work of [2] and those of [23]. They exploit the proximity of an annotated sequence with other sequences to biase the estimation of the CRF parameters. Although the framework of these studies is different from the work presented in this paper, they nonetheless share the idea of exploiting similarity between sequences seen as sets of features.

In this paper, the specific semi-supervised learning framework considered is known as active learning. Its principle is that supervision is carried out by an expert (or oracle) iteratively and interactively [22]. This is often set out in an algorithm whose main steps are as follows:
1) infer a classifier from $\mathcal{T}$;
2) apply the classifier to $\mathcal{N}$;
3) select examples from $\mathcal{N}$;
4) make an expert label these examples and add them to $\mathcal{T}$;
5) go to step 1.
This process is repeated until a stopping criterion is reached (e.g. maximal cost of annotation, minimum classifier performance, or $\mathcal{N}$ is empty).

The crux of these active learning algorithms is step 3, that is the selection of examples to be labeled by the expert. One wants to choose the most beneficial examples for learning, in order to get the best classification performance. This selection problem is often based on the results of the current classifier (Step 2). Much work has been proposed in this regard, particularly in the field of NLP [14] where these labeling problems are common. Regardless of the classifiers used, several families of selection strategies were proposed. The most common one is the uncertainty-based selection: the results from Step 2 are used to select examples for which the current classifier is less confident (see Section 3). A known drawback of this approach is that, at the beginning of the process, when there are few examples annotated, the classifier uncertainty measurements are unreliable. Another very common selection strategy is the selection by committee. Its principle is to learn not one but several classifiers in Step 1, then apply them to $\mathcal{N}$, and finally select examples on which they disagree the most. This approach is often implemented by techniques such as *bagging* and/or *boosting* [1], or by learning different classifiers from different representations of the data [16]. Beside the important computational cost generated by these multiple learning, these techniques also suffer from the same problem as uncertainty-based selection: classifiers are unreliable in the early rounds of iteration when $|\mathcal{T}|$ is small. Another family of selection techniques relies on the expected change in the model caused by adding new examples. The principle here is to select the sample that would impact most the model, assuming that this impact would result in improved performance. The underlying intuition is that the examples chosen in $\mathcal{N}$ will cover cases that are not covered by the examples of $\mathcal{T}$. Practical implementation of this approach heavily depends on the classifier used. [21] proposed several variants of this approach for CRF; only one, named *Information Density*, gave some positive results. It works by selecting the most different sequence in $\mathcal{N}$ with respect to those of $\mathcal{T}$. To assess this difference, the authors represent the sequences by a vector representing the combination of the sequence attributes, as captured by the feature functions. Since the labels of the sequences of $\mathcal{N}$ are unknown, only the features concerning x are considered. The most dissimilar sequence is simply defined as the one having the smallest average cosine with the sequences of $\mathcal{T}$.

This latter approach is close to those presented in this article: we also make use of sequence representation as sets of attributes, although the criteria we propose is more efficient than [21]'s one (Section 4). Furthermore, the evaluation method used in their study does not properly account for the annotation effort at each iteration: the authors evaluate performance based on the number of labeled sequences, without considering that some can be much longer than others. For our part, a more realistic setting is adopted: the annotation effort is measured in terms of annotated words (or sequence elements), which has implications for selection strategies tested by these authors (next section).

2.3 Experimental context

In the remainder of this article, we will validate our proposals for sequence selection on different tasks for which the CRF are conventionally used. We briefly describe these tasks and data below; for details, the interested reader can refer to the provided references.

We use the dataset of the entity recognition task named the ESTER campaign [8]. It contains 55,000 breath groups from transcripts of radio broadcasts in French; the named entities are annotated into 8 classes (person, place, time...). The CoNLL2002 dataset was proposed for the named entity recognition task in Dutch proposed at CoNLL 2002 [24]. It contains 4 different entity types; 14,000 sequences (sentences) are used in the experiments reported in the following section. The CoNLL2000 dataset contains English newspapers annotated with chunks [25], totalizing about 11,000 sentences and 4 classes (3 types of chunks and a label 'other'). We also experiment with the Sense Disambiguation dataset from Senseval-2 [5]: disambiguation of *hard, line, serve, interest*, each of their senses being represented by a different label in about 16,000 sentences. A somewhat different task is the phonetic transcription of isolated words in English provided by Nettalk dataset. The goal is to transcribe these words in a specific phonetic alphabet. This task is seen as a letter-by-letter annotation task. It has 18,000 words and 52 different labels corresponding to the phonetic alphabet. A preliminary step of data was to align words with their phonetic transcription (and thus to introduce the appropriate symbols 'empty' when needed).

The data are described with usual attributes and patterns for these tasks, with the parts-of-speech, lemmas, capital presence/absence, etc., and the BIO annotation scheme is adopted when necessary (ESTER, CONLL2002, CONLL-2000). Nine tenths is used for training (set $\mathcal{T}$ and $\mathcal{N}$) and the remaining tenth is used for performance evaluation. In most cases, the performance measure used is the word accuracy (rate of correctly labeled words), except for the phonetization task, which is evaluated by the sequence accuracy rate (a word must be completely and correctly phonetized). This evaluation is performed at each iteration and related to the annotation effort ie. the number of words (or symbols) to which the expert added a label.

The CRF implementation used is WAPITI [10], with its default settings unless stated otherwise. It should be noted that tests with other settings (optimization algorithms, normalization ...), not reported in the article, do not change the conclusions presented.

3 Uncertainty-based selection

As we have seen, a common solution for the selection of examples to annotate at each iteration is to propose to the oracle those for which the classifier learned at the previous iteration is less certain. With CRF, this means choosing the sequence x by looking at the probabilities $P(y|x;\theta)$.

3.1 Minimal confidence and sequence entropy

Among the different ways to proceed, [21] shows that two strategies in this family perform well in most cases: (i) the selection with minimal confidence, and (ii) selection from sequence entropy. The first simply consists in choosing in $\mathcal{N}$ the (automatically labeled) sequence whose probability is minimal with the current model: $x = \operatorname{argmin}_{x\in\mathcal{N}} P(y^*|x,\theta)$. The entropy method selects the sequence x with the greatest entropy over all the possible labels y of this sequence:

$$x = \operatorname*{argmax}_{x\in\mathcal{N}} \left(-\sum_{y} P(y|x,\theta) \log P(y|x,\theta) \right)$$

3.2 Length bias

One of the problems of these state-of-the-art approaches is that they tend to choose the longest sequences, as they often have lower probabilities than short sequences. However, the annotation cost is proportional to the sequence length. If one seeks to maximize performance for a minimal cost annotation, it is then potentially an undesirable behavior. To illustrate this, we report in Table 1 correlation between the sequence lengths in the ESTER dataset and their probabilities given by two models respectively trained on 20 and 10,000 randomly chosen sequences.

Size of training set	Pearson r (p-value)	Spearman ρ (p-value)	Kendall τ (p-value)
20 seq.	-0.52 ($< 10^{10}$)	-0.59 ($< 10^{10}$)	-0.44 ($< 10^{10}$)
10,000 seq.	-0.47 ($< 10^{10}$)	-0.56 ($< 10^{10}$)	-0.40 ($< 10^{10}$)

Table 1: Correlation (Pearson r, Spearman ρ, Kendall τ) with their p-value between the sequence lengths and their probabilities according to two models respectively trained on 20 and 10,000 sequences.

The length bias can be observed in both cases: in average, the sequence probability given by a CRF model is correlated to its length. This is particularly more pronounced when the model is trained on few sequences, which is precisely characteristic of the first iterations of active learning. Thus, this selection criterion is particularly unsuited at the beginning of active learning. Conversely, a simple normalization of the probabilities by the length of the sequences tends to favor very short sequences which does not provide enough useful information for learning.

3.3 Normalization

Based on the above findings, it seems important to normalize with respect to the sequence length. We propose a local, adaptive method of normalization based on the average probability of sequences for a given length. For this, we propose a method of normalization inspired by the Parzen window estimation method [15, 27]. The underlying idea is that for a fixed sequence length (plus or minus ϵ), the normalized probability scores should be distributed uniformly between 0 and 1. For a sequence x of $\mathcal{N}$ of length l, we estimate the average $\hat{\mu}_l$ and standard deviation $\hat{\sigma}_l$ probabilities on all sequences of $\mathcal{N}$ of length $l \pm \epsilon$, i.e. the set $\{P(y'^*|x') \mid x' \in \mathcal{N}, |x'| = |x| \pm \epsilon\}$. These values are estimated at each iteration, and then used to center and reduce the probabilities used in the previous selection strategies. For example, the selection by minimal confidence is now:

$$x = \underset{x \in \mathcal{N}}{\operatorname{argmin}} \left(\frac{P(y^*|x, \theta) - \hat{\mu}_l}{\hat{\sigma}_l} \right)$$

For each considered close length, it should modify the probability dispersion for sequences of this length, and thus cancel the bias of sequence length previously observed. In practice, in the experiments reported in Section 5, same length sequences are not found using a fixed ϵ but by neighborhood: $\hat{\mu}_l$ is calculated over a fixed number of sequences whose lengths are closest to the one considered. This k-nearest-neighbor approach can better handle cases of *outlier* sequences with very different lengths for which a neighborhood defined with a small ϵ would not cover any other sequence.

4 Representativity of feature functions

The main proposal of this article is to consider that the distribution of attributes, such as captured by the feature functions, can guide the selection of examples to be annotated during an active learning iteration. To support this intuition, we first study how these attributes are distributed in terms of frequency and in terms of use in the models (Subsection 4.1). Based on these considerations, Subsection 4.2 proposes an original method to select sequences to annotate.

4.1 Preliminary study

The feature functions encode the relationship between the description of sequences and labels, as expressed by the patterns $\{\mathrm{Pat}_i\}$. It is interesting to observe their frequencies in the data, in order to see which ones among them are actually used for the prediction. CRF are known to produce large models in the sense that many parts of the data, as seen through the feature functions, are kept in the model [3, 28, for elements of discussion].

In order to study which functions are actually used in the model for the prediction, we first calculate the distribution of the occurrences of all possible feature functions f_j on ESTER data:

$$occ(f_j) = |\{f_j(x^{(m)}, y_{t-1}^{(m)}, y_t^{(m)}, t) = 1 | \forall \text{ example } m, \forall \text{ position } t\}|$$

We then extract from a model trained on the data the feature functions whose weight $|\lambda_j| > 0$. Among the learning settings for CRF, L1 or L2 normalization greatly influences the number of feature functions with non-zero weight. So, a model with a standard L1 and another with a normalization *elastic-net* (mixing equally L1 and L2) are trained on the whole ESTER dataset (full supervision). Figure 1 reports these three distributions.

As expected, we observe that these three distributions are very similar except for the rarest feature functions, especially with the L1 model. Most combinations of attributes/labels from the data therefore appear useful (i.e. their weight $|\lambda_j| > 0$) for predictions in our two models. It means that the CRF models exploit a vast majority of attributes/labels combinations present in the data, in proportion to their frequency in the data: the fact that combinations are very common or rare does not intervene (except for the rarest configurations with L1 model). Thus,

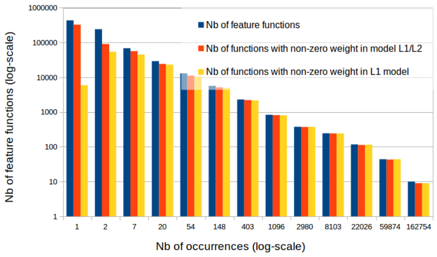

Fig. 1: Distribution of feature functions (number of functions according to their occurrence; log-scale on both axes) and distribution of the functions used in two CRF models; ESTER dataset.

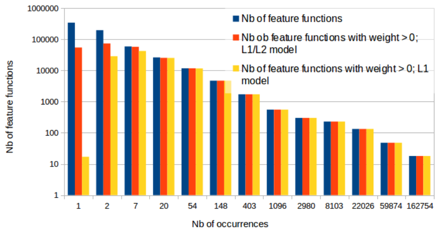

Fig. 2: Distribution of feature functions without label information (number of functions according to their occurrence number; log-scale on both axes) and distribution of the functions used in two CRF models; ESTER dataset.

to build a smaller training set leading to models with similar characteristics, it seems important to offer the maximum variety of combinations accordingly with these proportions, i.e. respecting the distribution of attribute/label combinations of the whole dataset. This result is not specific to the ESTER dataset: the same distributions are observed for every tested dataset (see Sect. 2.3).

In our semi-supervised case, most of the data are not annotated. It is therefore important to check whether these earlier findings are still true without considering the labels. We therefore examine the distribution of feature functions regardless of labels , i.e. only by looking at the attributes concerning x in $\{f_j\}$. These incomplete feature functions (without label information) are notedf_j^*. Formally, we count in the data:
$occ(f_j^*) = |\{f_j(x^{(m)}, y_1, y_2, t) = 1 | \forall \text{ example } m, \forall \text{ position } t, \forall \text{ labels } y_1,\ y_2\}|$.
Figure 2 thus illustrates again the occurrences of feature functions, but regardless of the label. The same trends as before can be observed. These experiments suggest the importance of a varied and representative training set of all combinations of attributes (with no information on the label) defined by the feature functions.

4.2 Test of proportion

We build on the previous observation to propose a new selection strategy. At each iteration of the active learning algorithm, we want the training set which is the most representative of the whole dataset. In other words, we want the sequence distribution, as seen by CRF via feature functions, to be as close as possible to those of $\mathcal{T} \cup \mathcal{N}$. As before, each sequence is seen as the set of feature functions that can be generated from it, not including labels.

To select the sequence x to add to the training set at each iteration (once annotated by the oracle), we need to evaluate how the resulting training set $\mathcal{T} \cup \{x\}$ compares with the whole data at our disposal (annotated or not, ie

$\mathcal{T} \cup \mathcal{N}$). For each feature function, we propose to simply examine whether the proportion of this function observed in the sample $\mathcal{T} \cup \{x\}$ is comparable to that of the sample $\mathcal{T} \cup \mathcal{N}$. These two samples are not independent, but can be considered as such when $|\mathcal{N}| \gg |\mathcal{T}|$, which is ensured in the first iterations of active learning.

More specifically, we perform a statistical test of proportion between the two samples $\mathcal{T} \cup \{x\}$ and $\mathcal{T} \cup \mathcal{N}$, respectively denoted 1 and 2, with size n_1 and n_2. Let $\hat{p}_1^j = r_1^j / n_1$ be the estimator of the proportion of occurrences of a given feature function f_j appearing r_1^j times in sample 1, and $\hat{p}_2^j = r_2^j / n_2$ be the one for sample 2. We can then calculate the z-score:

$$z_{j,x} = \frac{\hat{p}_1^j(f_j) - \hat{p}_2^j(f_j)}{\sqrt{\hat{p}^j \times (1-\hat{p}^j) \times (1/n_1 + 1/n_2)}} \qquad \text{with} \qquad \hat{p}^j = \frac{r_1^j + r_2^j}{n_1 + n_2}$$

The z-score follows a standard normal distribution, allowing us to calculate the probability $P(z_{j,x})$ to observe such a difference in proportion between the two samples. A high probability intuitively means that sample 1 contains a proportion of the feature function f_j comparable to that of sample 2.

It is necessary to combine these probabilities for all feature functions. In order to do so, we make a simplifying assumption by considering that the observations of feature functions are independent. Although this assumption is invalid in most cases, it allows us to propose a simple estimate of the overall probability of the sample x as the product of $P(z_{j,x})$ for every feature function f_j. Finally, the choice of the sequence to add to the training set is the one maximizing this probability: $x^* = \text{argmax}_{x \in \mathcal{N}} \prod_j P(z_{j,x})$

5 Experiments

In this section, we compare experimentally the different selection strategies for active learning previously discussed. The experimental framework is detailed below, and learning curves are presented in Subsection 5.2.

5.1 Settings

Several selection strategies are experimented: on the one hand, for comparison purposes, we implemented state-of-the-art strategies, namely, selection by minimal confidence, entropy and information density. We also added a simple baseline in which the sequences are selected at random. On the other hand, we tested the normalization process for the minimal confidence selection (cf. Sec. 3.3) and the approach based on proportion (cf. Sec. 4). We do not report results based on selection by committee as they yield lower results than the previous ones in almost every case [21].

All these methods are tested under the same conditions (CRF parameters, patterns...). For initialization, a sequence is randomly chosen to serve as the first example (the same for all selection methods). At each iteration, a single example is selected to be annotated by the oracle and the classifier is re-trained on all annotated data (therefore, this is not an update of the previous CRF model).

5.2 Results

Figures 3 to 7 give the learning curves on our different datasets. The performance of the classifiers learned at each iteration is expressed in function of the cost of accumulated annotation of the set *calT* (i.e. total number of words or symbols seen, according to the task). In the figures, the cost is reported on a logarithmic scale, so one can appreciate the different cases (few annotations vs. many annotations). Several observations stand in. First, these curves have very different appearance from a dataset to another. This is explained by the characteristics of tasks and data, implying that some are more readily feasible with good performance with few annotations (CoNLL2000) or not (CoNLL2002). For all datasets except Nettalk, differences, especially when the annotation cost is small, are sensitive. Regarding Nettalk, it is more difficult to bring out a selection method better than the other. This can certainly be explained by the difficulty of the task and, more precisely, by the huge number of possible labels. Indeed, there are a very large number of possible attributes/labels configurations; therefore, in all cases, it requires an extremely large number of examples to cover all these configurations.

Second, we observe that the three strategies from literature offer an average performance sometimes not far from the *random* strategy. Strategies by minimal confidence and entropy are even sometimes well below *random* (SenseEval-2), obviously penalized by their biases discussed in Section 3. This is important to note; it is often overshadowed by evaluations taking into account the number of sequences, as we have already pointed in the work of [21].

Third, our normalization approach, applied to the minimal confidence strategy, gives satisfying results since it allows to get better or similar results to the non normalized version. It especially performs best when the number of annotation is important (ESTER, CoNLL2002, Senseval-2) even if the logarithmic scale in the figures hides a little this long domination.

Finally, our selection proposal based on proportion tests obtains very good results overall. It behaves generally better than other selection techniques, including *information density*, from which it is conceptually close. It may be noted that our strategy brings a significant gain when dealing with few annotations. This outcome is explained by the fact that the method does not rely on the predictions, unreliable at this stage, of the current classifier. However, this gain is less or even absent compared to other methods when the amount of annotated data becomes very important. This shows the limits of our approach, which does not exploit any information from the classifier, but it also allows to devise joint strategies in which the classification information would also be used when a minimal number of annotations is reached.

6 Conclusive remarks

At a time when most NLP problems are tackled as supervised learning tasks, the cost of annotations by expert is a significant problem. Active learning provides a

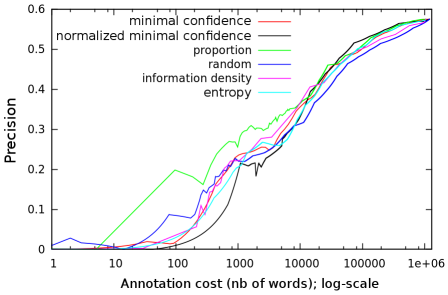

Fig. 3: Learning curve (precision rate vs. annotation cost expressed in words); ESTER dataset; log-scale

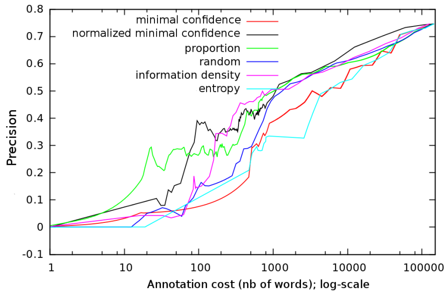

Fig. 4: Learning curve (precision rate vs. annotation cost expressed in words); CoNLL2002 dataset; log-scale

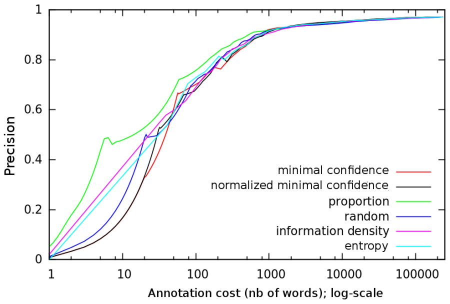

Fig. 5: Learning curve (precision rate vs. annotation cost expressed in words); CoNLL2000 dataset; log-scale

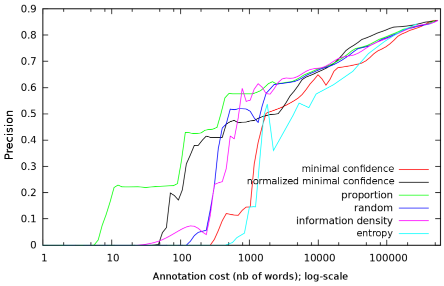

Fig. 6: Learning curve (precision rate vs. annotation cost expressed in words); SensEval-2 dataset; log-scale

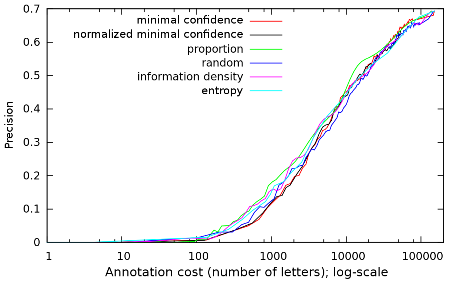

Fig. 7: Learning curve (precision rate in terms of correctly phonetized vs. annotation cost expressed in letters); Nettalk dataset; log-scale

framework to control this cost while maximizing, hopefully, the classifier performance. As we have seen, it is in fact largely dependent on the example selection strategy implemented. In this article, we looked at some of these strategies and we have demonstrated a bias lowering their annotation cost/performance ratio. The normalization that we have proposed can solve this problem in a very simple manner while providing a significant performance gain. And when the annotation costs are limited, our strategy based on an original criterion of proportionality, appears the most advantageous on the several NLP taks examined. Of course, these gains are only appreciable in a real semi-supervised context in which one wants to get the best performance from a few annotated data; when a large amount of data is available, all the strategies tends to give similar results.

Many variations, improvements and research avenues can be explored. Among them, we would try to take into account the dependence between feature functions. In our current proposal, they are considered to be independent for simplification purpose, which is never the case in practice. These dependencies may even be very important because the patterns used to build these feature functions often exploit several times to the same elements (lemma of the current word, PoS the current word ...), and that these elements are themselves in a dependency relationship. This can strongly impact the estimate of the overall proportion probabilities, and ultimately distort the choice of the best example.

Another promising approach is to mix these different selection techniques to combine their benefits. They can obviously be simply merged (vote, product of scores or ranks ...), but it seems more interesting to aim more complex combinations, which could be achieved with *learning to rank* approaches [11].

Finally, in our current framework, the selected sequences are fully annotated. It would be interesting to study the case of partial annotations, under the same constraints to optimize the cost/performance ratio, taking inspiration for example from [19].

Acknowledgments. This work was partly funded by a French government support granted to the CominLabs LabEx managed by the ANR in *Investing for the Future* program under reference ANR-10-LABX-07-01.

References

1. Abe, N., Mamitsuka, H.: Query learning strategies using boosting and bagging. In: Proceedings of the Fifteenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc, Madison, Wisconsin, USA (1998)
2. Ando, R.K., Zhang, T.: A high-performance semi-supervised learning method for text chunking. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. pp. 1–9. ACL '05, Association for Computational Linguistics, Stroudsburg, PA, USA (2005), http://dx.doi.org/10.3115/1219840.1219841
3. Chen, S.: Performance prediction for exponential language models. In: Proc of Annual Conference of the North American Chapter of the Association for Computational Linguistics. pp. 450–458 (June 2009)

4. Constant, M., Tellier, I., Duchier, D., Dupont, Y., Sigogne, A., Billot, S.: Intégrer des connaissances linguistiques dans un CRF : Application à l'apprentissage d'un segmenteur-étiqueteur du français. In: Traitement Automatique du Langage Naturel (TALN'11). Montpellier, France (2011)
5. Edmonds, P., Cotton, S.: Senseval-2: Overview. In: Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems. pp. 1–5. Association for Computational Linguistics (2001), `http://aclweb.org/anthology/S01-1001`
6. Freitag, D.: Trained named entity recognition using distributional clusters. In: Proceedings of the conference EMNLP (2004)
7. Garrette, D., Baldridge, J.: Learning a part-of-speech tagger from two hours of annotation pp. 138–147 (June 2013), `http://www.cs.utexas.edu/users/ai-lab/?garrette:naacl13`
8. Gravier, G., Bonastre, J.F., Geoffrois, E., Galliano, S., Tait, K.M., Choukri, K.: ESTER, une campagne d'évaluation des systèmes d'indexation automatique. In: Actes des Journées d'Étude sur la Parole, JEP, Atelier ESTER2 (2005)
9. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: International Conference on Machine Learning (ICML) (2001)
10. Lavergne, T., Cappé, O., Yvon, F.: Practical very large scale CRFs. In: Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL). pp. 504–513. Association for Computational Linguistics (July 2010), `http://www.aclweb.org/anthology/P10-1052`
11. Liu, T.Y.: Learning to rank for information retrieval. Foundations and Trends in Information Retrieval 3(3), 225–331 (2009)
12. Mann, G.S., McCallum, A.: Generalized expectation criteria for semi-supervised learning of conditional random fields. In: Proceedings of ACL-08: HLT. pp. 870–878. Colombus, Ohio, USA (2008)
13. Miller, S., Guinness, J., Zamanian, A.: Name tagging with word clusters and discriminative training. In: Procedings of the conference ACL (2004)
14. Olsson, F.: A literature survey of active machine learning in the context of natural language processing. Tech. Rep. Swedish Institute of Computer Science, Swedish Institute of Computer Science (2009)
15. Parzen, E.: On estimation of a probability density function and mode. Ann. Math. Stat. 33, 1065–1076 (1962)
16. Pierce, D., Cardie, C.: Limitations of co-training for natural language learning from large datasets. In: Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP 2001). Pittsburgh, Pennsylvania, USA (2001)
17. Pranjal, A., Delip, R., Balaraman, R.: Part Of speech Tagging and Chunking with HMM and CRF. In: Proceedings of NLP Association of India (NLPAI) Machine Learning Contest (2006)
18. Raymond, C., Fayolle, J.: Reconnaissance robuste d'entités nommées sur de la parole transcrite automatiquement. In: Actes de la conférence Traitement Automatique des Langues Naturelles. Montréal, Canada (2010)
19. Salakhutdinov, R., Roweis, S., Ghahramani, Z.: Optimization with EM and Expectation-Conjugate-Gradient. In: Proceedings of the conference ICML (2003)
20. Schraudolph, N.N., Yu, J., Günter, S.: A stochastic quasi-Newton method for online convex optimization. In: Proceedings of 11th International Conference on Artificial Intelligence and Statistics. Workshop and Conference Proceedings, vol. 2, pp. 436–443. San Juan, Puerto Rico (2007)

21. Settles, B., Craven, M.: An analysis of active learning strategies for sequence labeling tasks. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1069–1078. ACL Press (2008)
22. Settles, B.: Active learning literature survey. Computer sciences technical report 1648, University of Wisconsin–Madison (2010)
23. Smith, N., Eisner, J.: Contrastive estimation: Training log-linear models on unlabeled data. In: Proceedings of ACL (2005)
24. Tjong Kim Sang, E.F.: Introduction to the conll-2002 shared task: Language-independent named entity recognition. In: Proceedings of CoNLL-2002. pp. 155–158. Taipei, Taiwan (2002)
25. Tjong Kim Sang, E.F., Buchholz, S.: Introduction to the conll-2000 shared task: Chunking. In: Cardie, C., Daelemans, W., Nedellec, C., Tjong Kim Sang, E. (eds.) Proceedings of CoNLL-2000 and LLL-2000. pp. 127–132. Lisbon, Portugal (2000)
26. Wang, T., Li, J., Diao, Q., Wei Hu, Y.Z., Dulong, C.: Semantic event detection using conditional random fields. In: IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW '06) (2006)
27. Wasserman, L.: All of Statistics: A Concise Course in Statistical Inference. Springer Texts in Statistics (2005)
28. Zhou, H., Hastie, T.: Regularization and variable selection via the elastic net pp. 301–320 (2005)