Strategies to select examples for Active Learning with CRF

Vincent Claveau Ewa Kijak

IRISA - CNRS - Univ. Rennes 1

TALN 2015

Overview

Annotation of sequences

- many NLP tasks are tackled as annotation problems
- supervised ML techniques are widely used, especially CRF

Problem

- can we reduce the supervision (annotation) cost?

Overview

Annotation of sequences

- many NLP tasks are tackled as annotation problems
- supervised ML techniques are widely used, especially CRF

Problem

- can we reduce the supervision (annotation) cost?

☞ interactive framework: active learning

Overview

Annotation of sequences

- many NLP tasks are tackled as annotation problems
- supervised ML techniques are widely used, especially CRF

Problem

- can we reduce the supervision (annotation) cost?

☞ interactive framework: active learning

☞ which sequence to propose to the expert?

Outline

Outline

Conditional Random Fields

[Lafferty et al., 2001]

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

Conditional Random Fields

[Lafferty et al., 2001]

- feature functions

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

Conditional Random Fields

[Lafferty et al., 2001]

- feature functions
- normalization value

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

Conditional Random Fields

[Lafferty et al., 2001]

- feature functions
- normalization value
- weights of the feature functions

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

Conditional Random Fields

[Lafferty et al., 2001]

- feature functions
- normalization value
- weights of the feature functions
- learning: finds the λ_j that best fit the training data

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

Conditional Random Fields

[Lafferty et al., 2001]

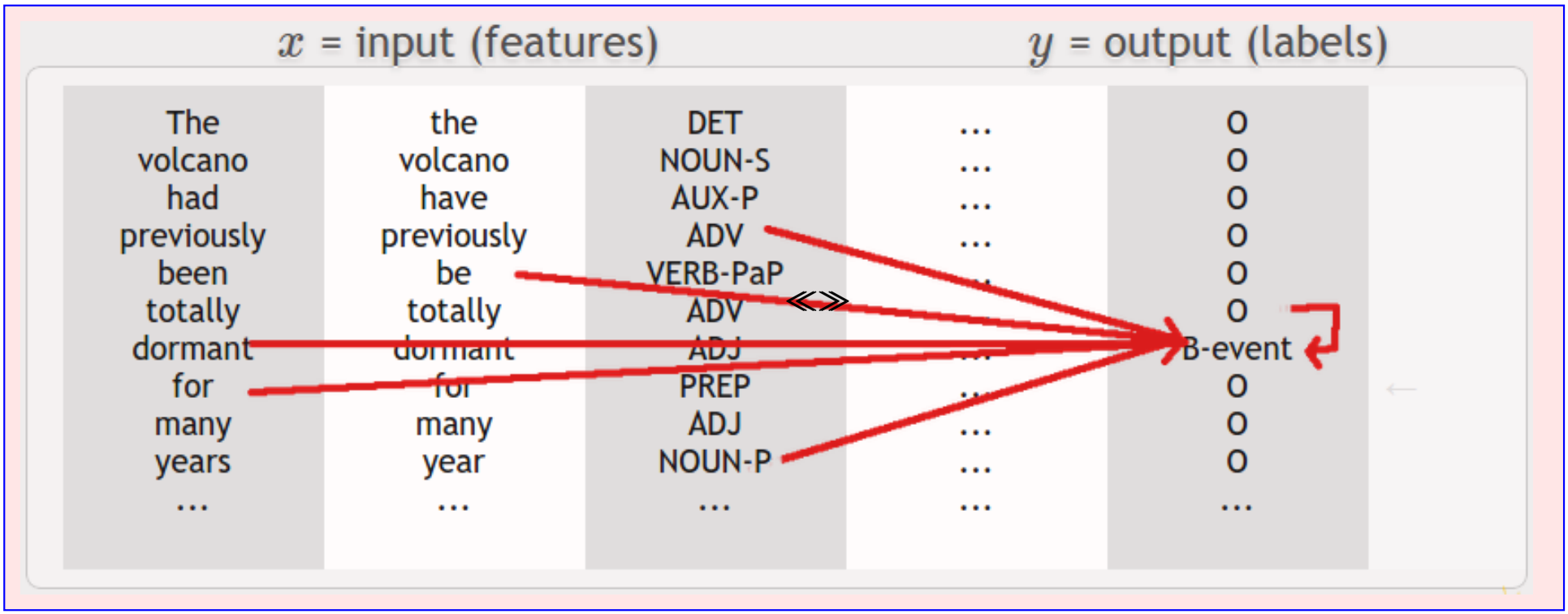
- feature functions
- normalization value
- weights of the feature functions
- learning: finds the λ_j that best fit the training data
- tagging: finds the sequence of tags y^* that maximizes $P(y|x,\lambda)$

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

About feature functions

- user-defined through patterns applied on training data

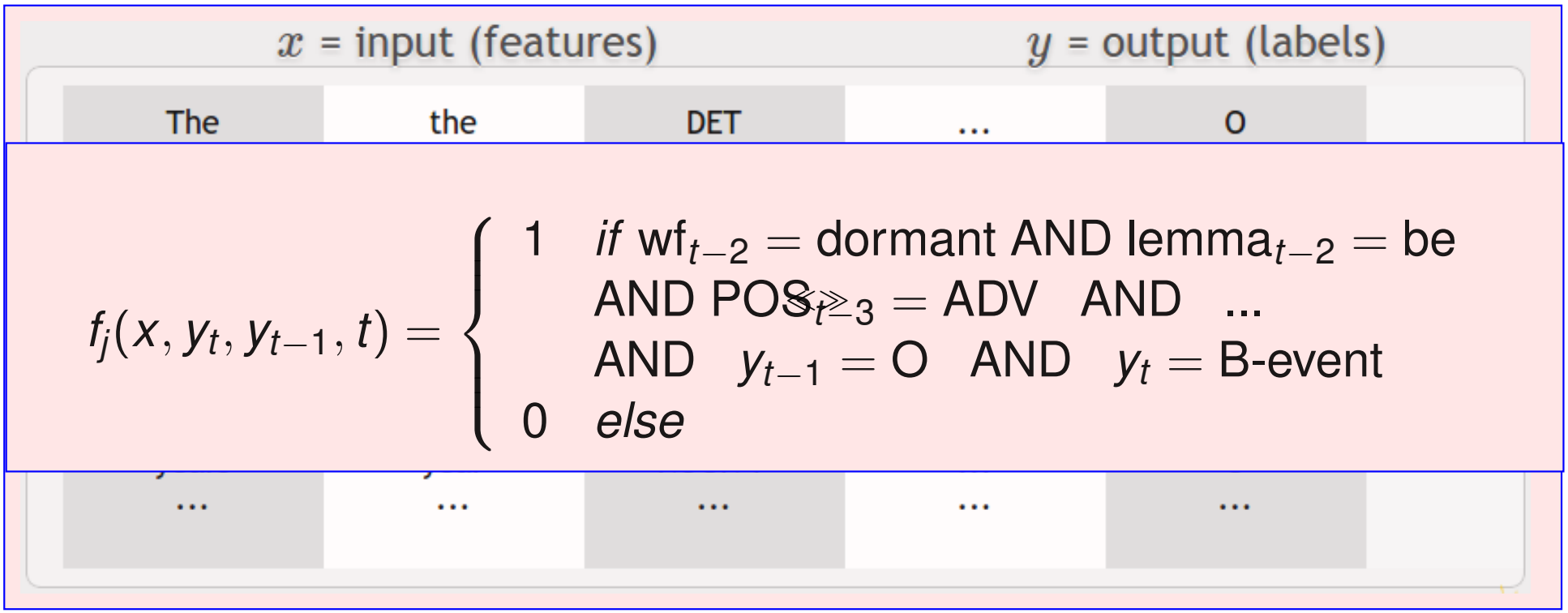
☞ generates huge amounts of such feature functions



About feature functions

- user-defined through patterns applied on training data

☞ generates huge amounts of such feature functions



Outline

Active learning

Principle

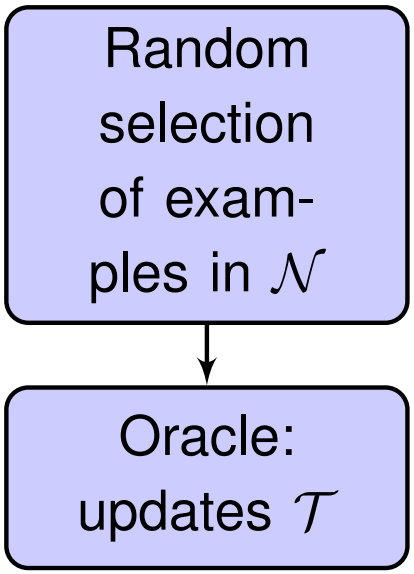
- $\mathcal{N}$ = non annotated data, $\mathcal{T}$ annotated data
- interactive learning/data annotation

Random selection of examples in $\mathcal{N}$

Active learning

Principle

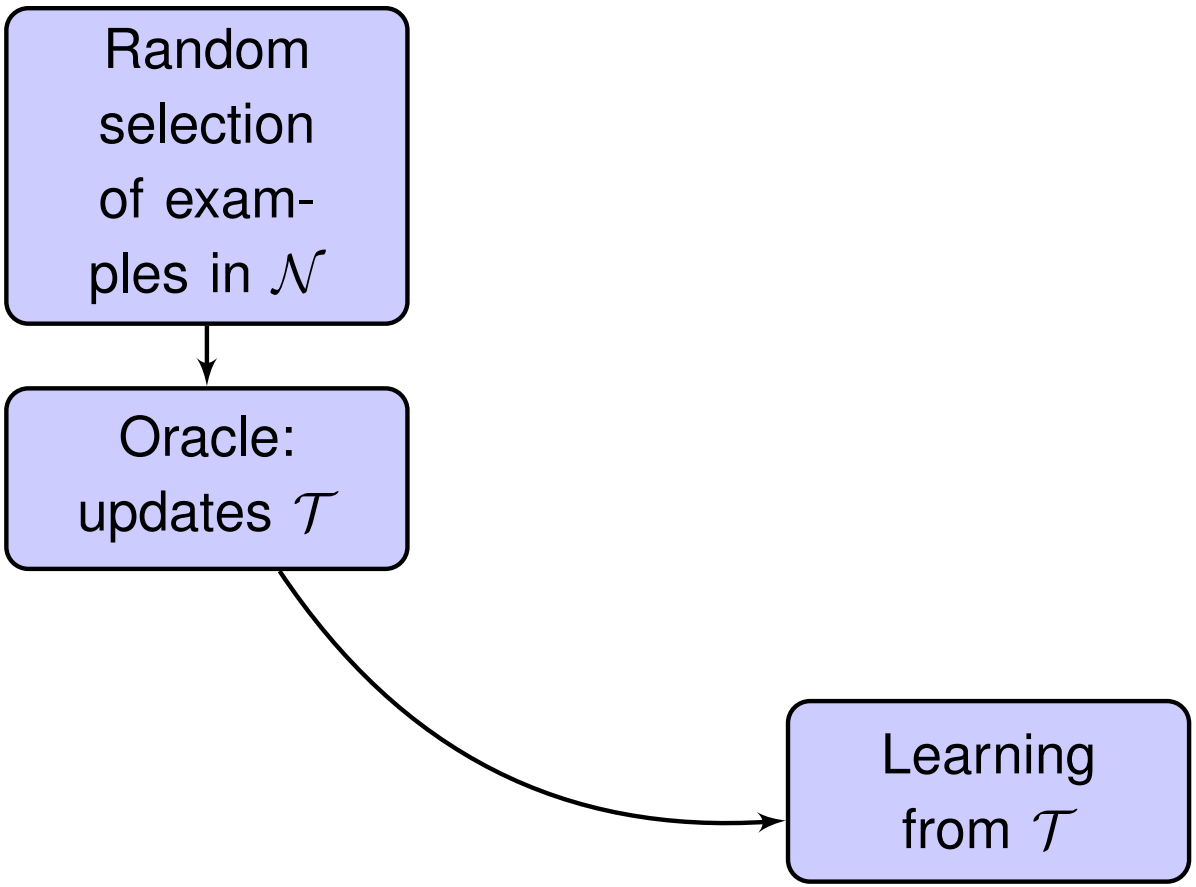
- $\mathcal{N}$ = non annotated data, $\mathcal{T}$ annotated data
- interactive learning/data annotation



Active learning

Principle

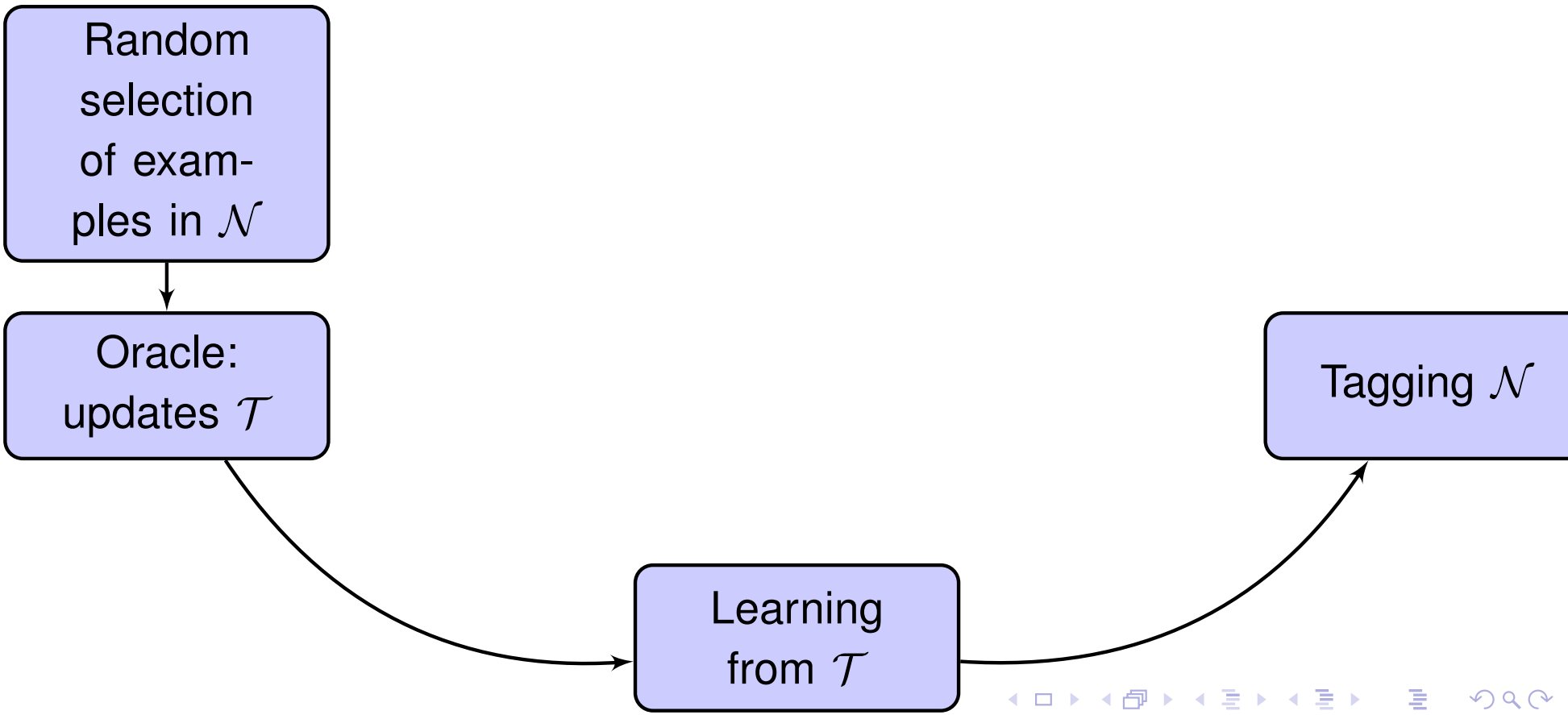
- $\mathcal{N}$ = non annotated data, $\mathcal{T}$ annotated data
- interactive learning/data annotation



Active learning

Principle

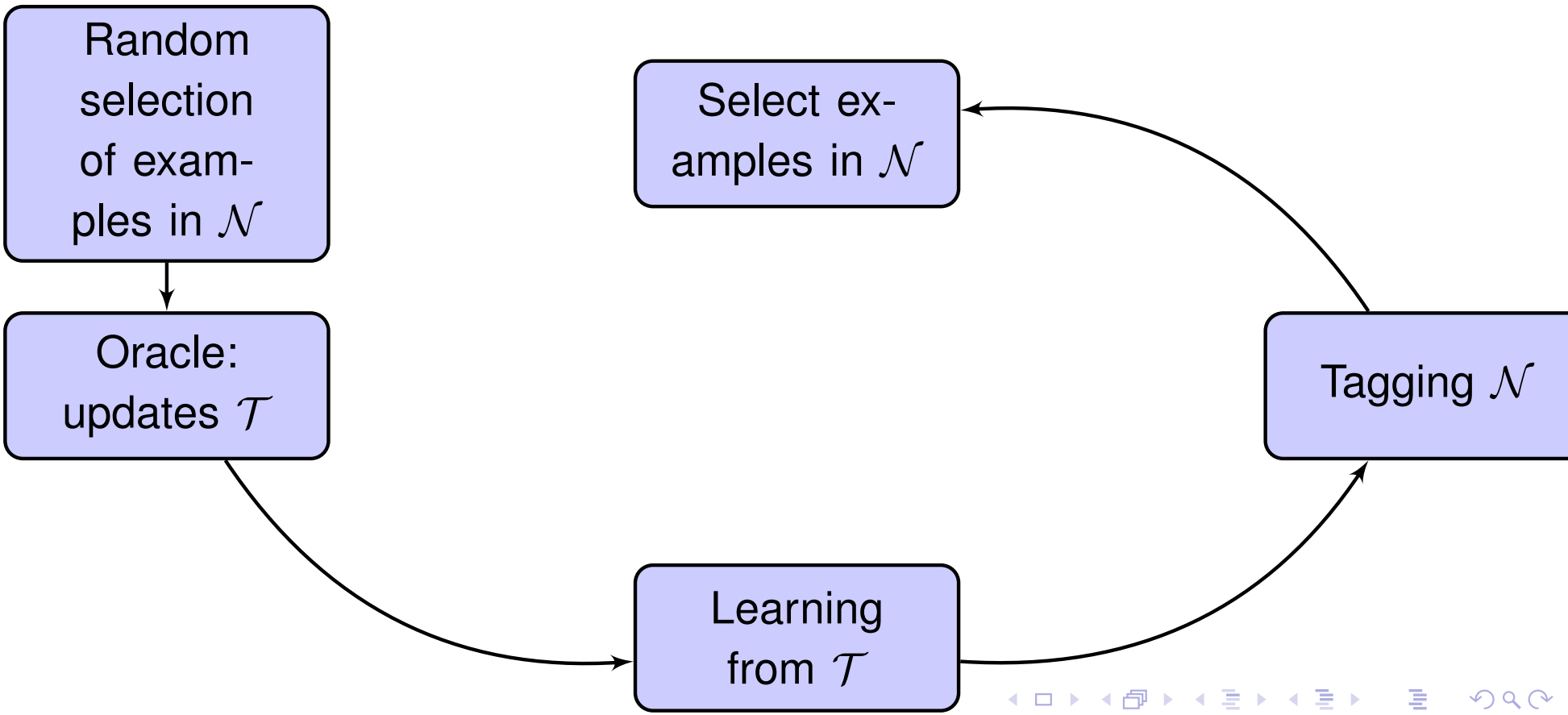
- $\mathcal{N}$ = non annotated data, $\mathcal{T}$ annotated data
- interactive learning/data annotation



Active learning

Principle

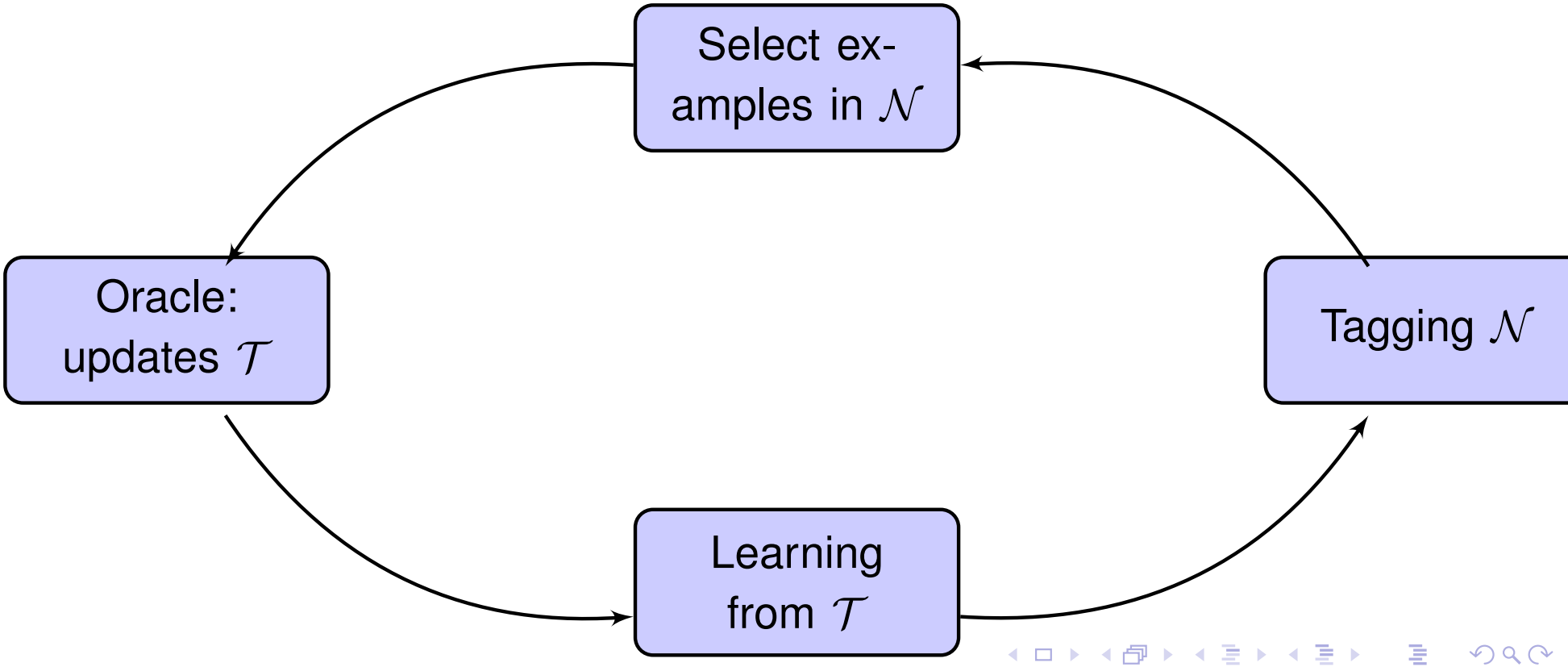
- $\mathcal{N}$ = non annotated data, $\mathcal{T}$ annotated data
- interactive learning/data annotation



Active learning

Principle

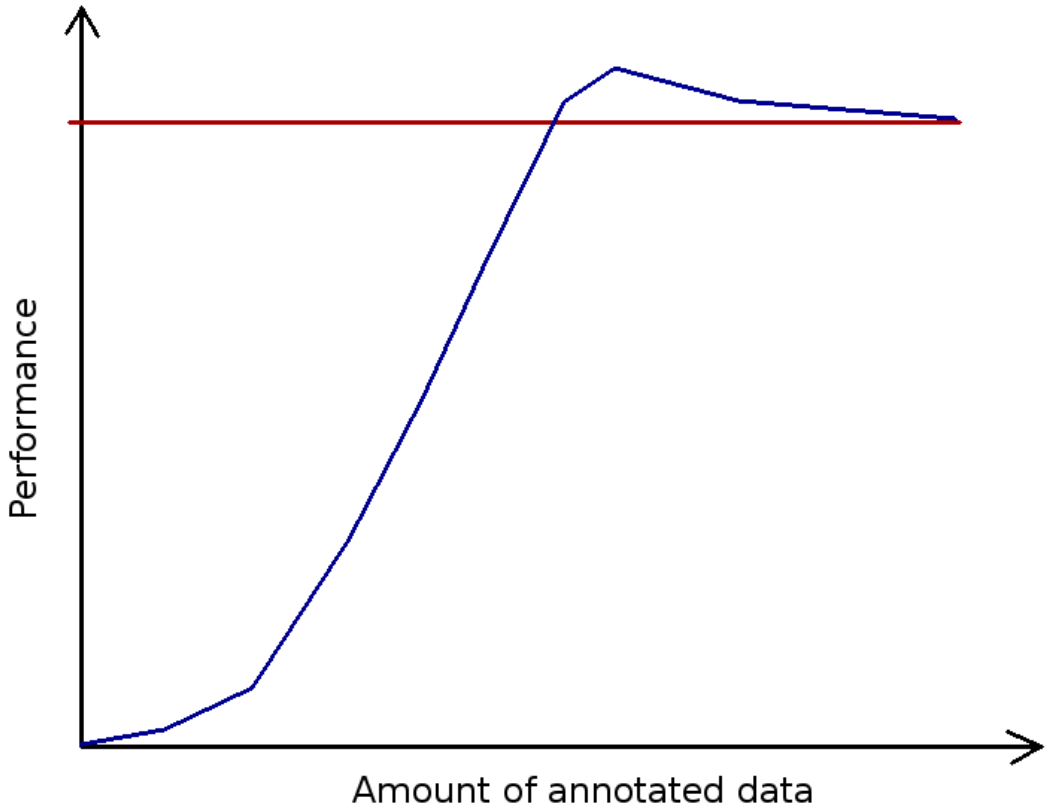
- $\mathcal{N}$ = non annotated data, $\mathcal{T}$ annotated data
- interactive learning/data annotation



Active learning

Goal

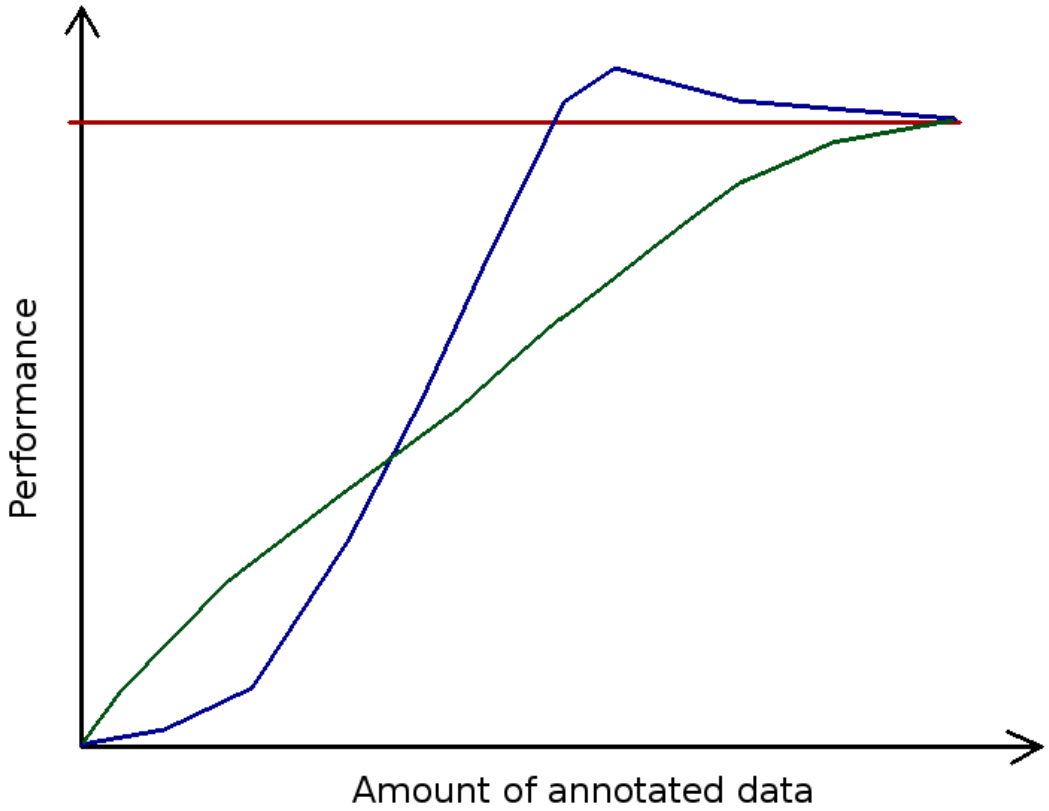
- for a fixed amount of annotation get the best obtainable results
- get similar (or better) results with less annotation



Active learning

Goal

- for a fixed amount of annotation get the best obtainable results
- get similar (or better) results with less annotation



Active learning

Selection strategies

Select by committee

- train several classifiers (bagging/boosting) [Abe & Mamitsuka, 1998; Pierce & Cardie, 2001]
- select the $x \in \mathcal{N}$ for which the classifiers disagree
- pb: classifiers are not reliable when $\mathcal{T}$ is small

Select by uncertainty

- select the $x \in \mathcal{N}$ yielding the lowest $P(y^*|x, \lambda)$
- pb: probabilities are not reliable when $\mathcal{T}$ is small

Active learning

Selection strategies

Select by committee

- train several classifiers (bagging/boosting) [Abe & Mamitsuka, 1998; Pierce & Cardie, 2001]
- select the $x \in \mathcal{N}$ for which the classifiers disagree
- pb: classifiers are not reliable when $\mathcal{T}$ is small

Select by uncertainty

- select the $x \in \mathcal{N}$ yielding the lowest $P(y^*|x, \lambda)$
- pb: probabilities are not reliable when $\mathcal{T}$ is small

☞ show a bias in these selection techniques

Active learning

Selection strategies

Select by model change

- select $x \in \mathcal{N}$ that would impact/change the model the most
- intuition is that x brings new useful information for the task
- implementation chiefly depends on the classifier model
- information density [Settles & Craven (2008)]

Active learning

Selection strategies

Select by model change

- select $x \in \mathcal{N}$ that would impact/change the model the most
- intuition is that x brings new useful information for the task
- implementation chiefly depends on the classifier model
- information density [Settles & Craven (2008)]

☞ new approach based on proportionality between $\mathcal{T}$ and $\mathcal{N}$

Outline

Length normalization

Example

- consider the minimal confidence strategy:

$$\text{next sequence to tag} = \underset{x \in \mathcal{N}}{\text{argmin}}\, P(y^*|x, \theta)$$

Length normalization

Example

- consider the minimal confidence strategy:

$$\text{next sequence to tag} = \underset{x \in \mathcal{N}}{\operatorname{argmin}}\, P(y^*|x,\theta)$$

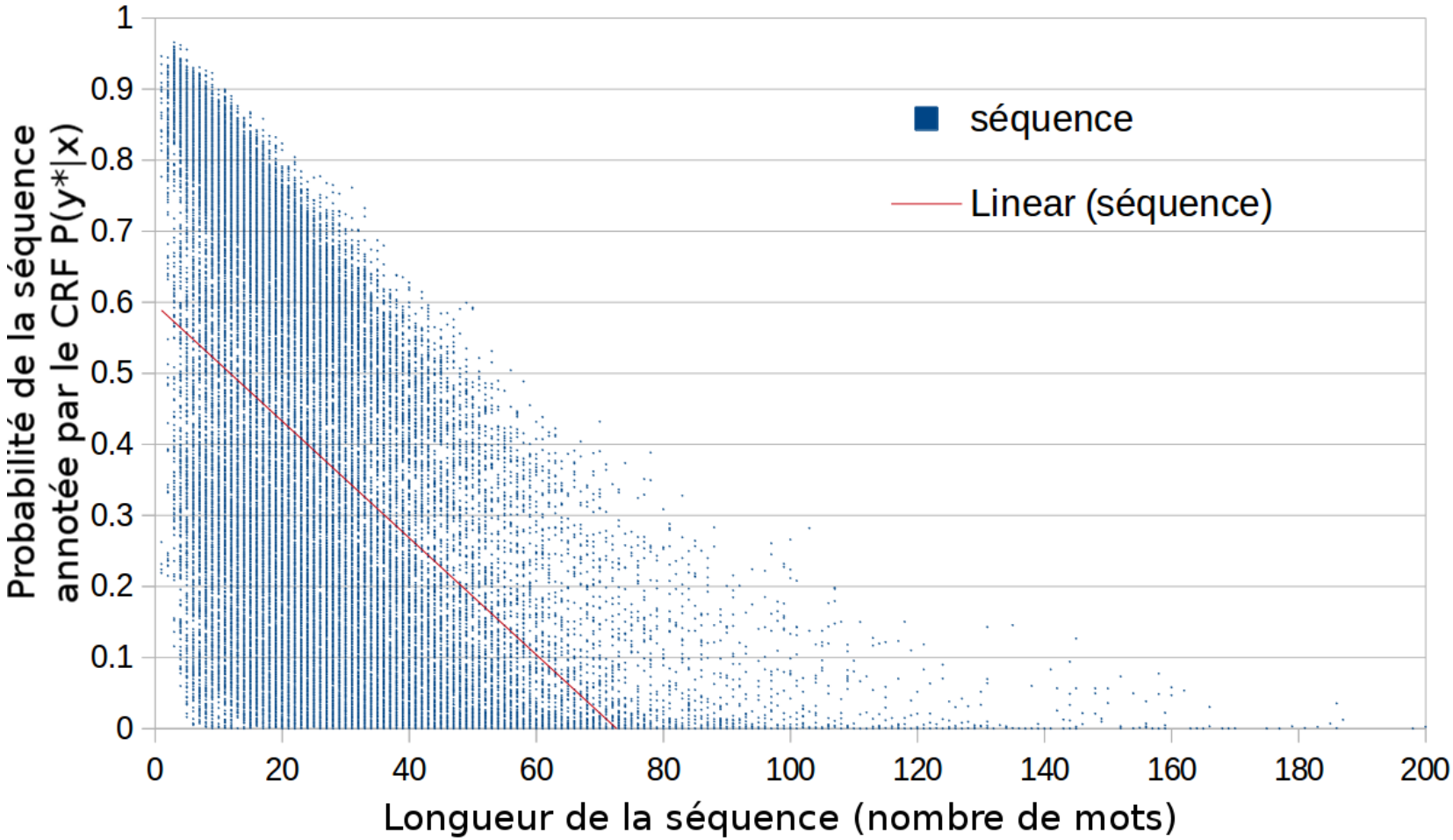
Bias

- tends to select longest sequence first

☞ impact the annotation cost

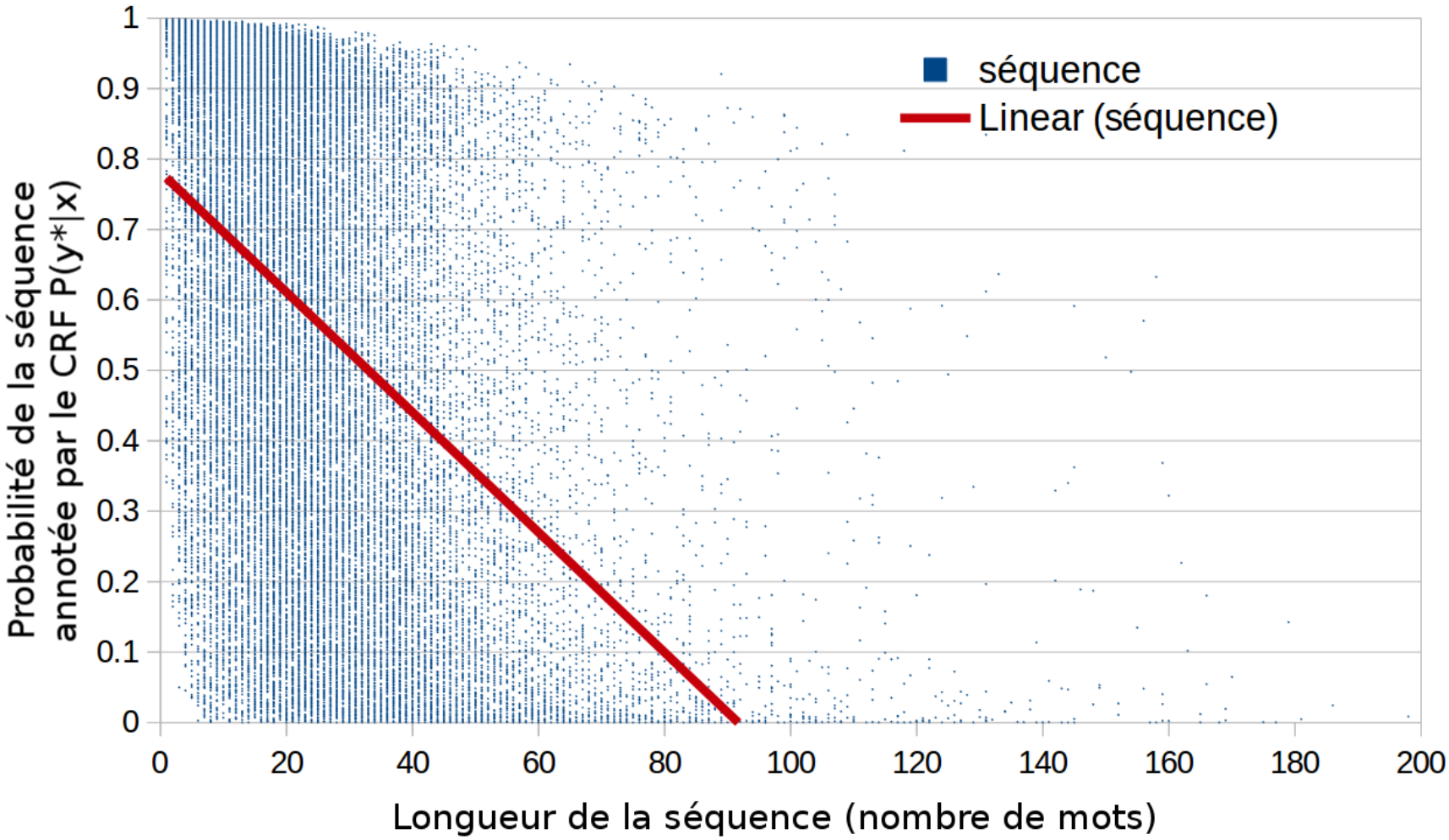
Illustrating the bias of length

Model obtained after 20 iterations



Illustrating the bias of length

Model obtained after 10 000 iterations



Length normalization

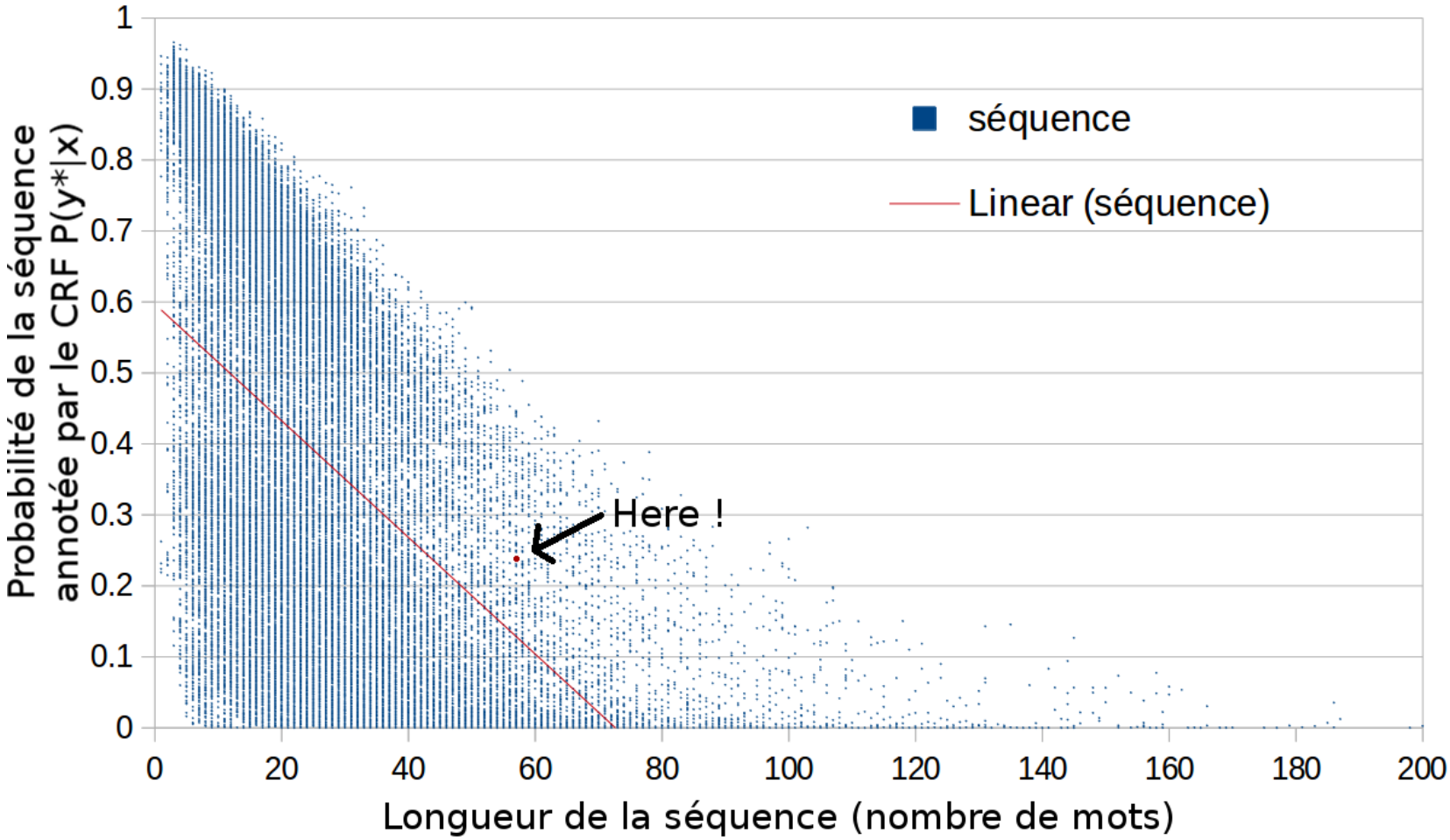
Principle

- for any sequence length, scores should follow a uniform distribution
- ”correct” the distribution as a whole is not feasible

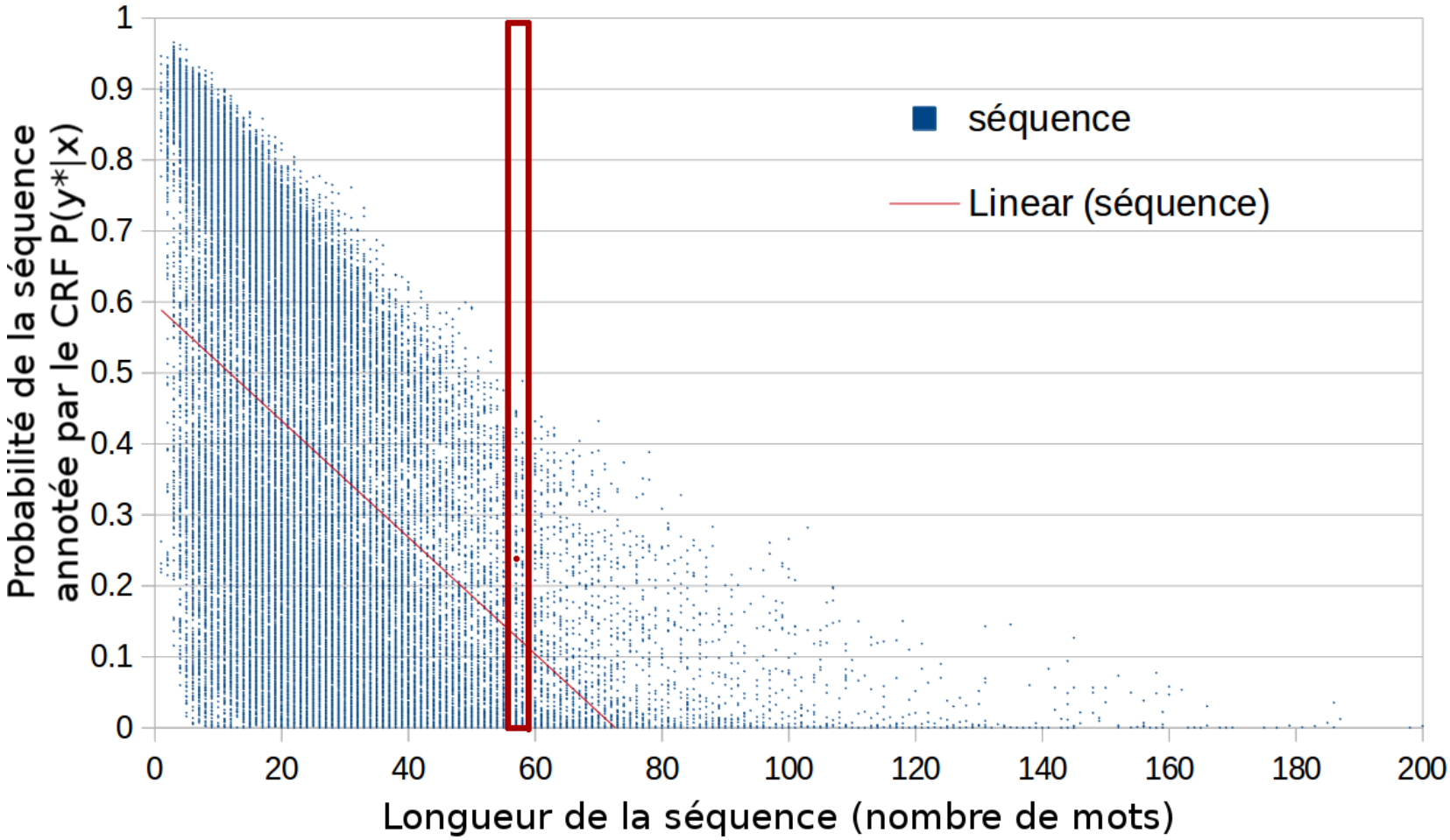
Parzen window

- local density estimation approach
- estimate score locally around the considered sequence
- the score of a sequence is normalized by observing same length sequences

Parzen window



Parzen window



Length normalization

Example

- consider the minimal confidence strategy:

$$\text{next sequence to tag} = \underset{x \in \mathcal{N}}{\text{argmin}}\, P(y^*|x,\theta)$$

- for a sequence of length $|x| = l$, collect scores of every sequence with length $l \pm \epsilon$
- use mean and standard deviations from these scores to center and reduce:

$$\text{next sequence to tag} = \underset{x \in \mathcal{N}}{\arg\min} \left(\frac{P(y^*|x,\theta) - \hat{\mu}_{|x|}}{\hat{\sigma}_{|x|}} \right)$$

Outline

Strategy of proportionality

Goal

- select $x \in \mathcal{N}$ s.t. the resulting model will be closer to a model trained with all examples

Interesting fact

- almost every feature (as expressed by f_j) in the training data are used in the model ($\lambda_j \neq 0$)

Distribution of feature functions

Preliminary experiment

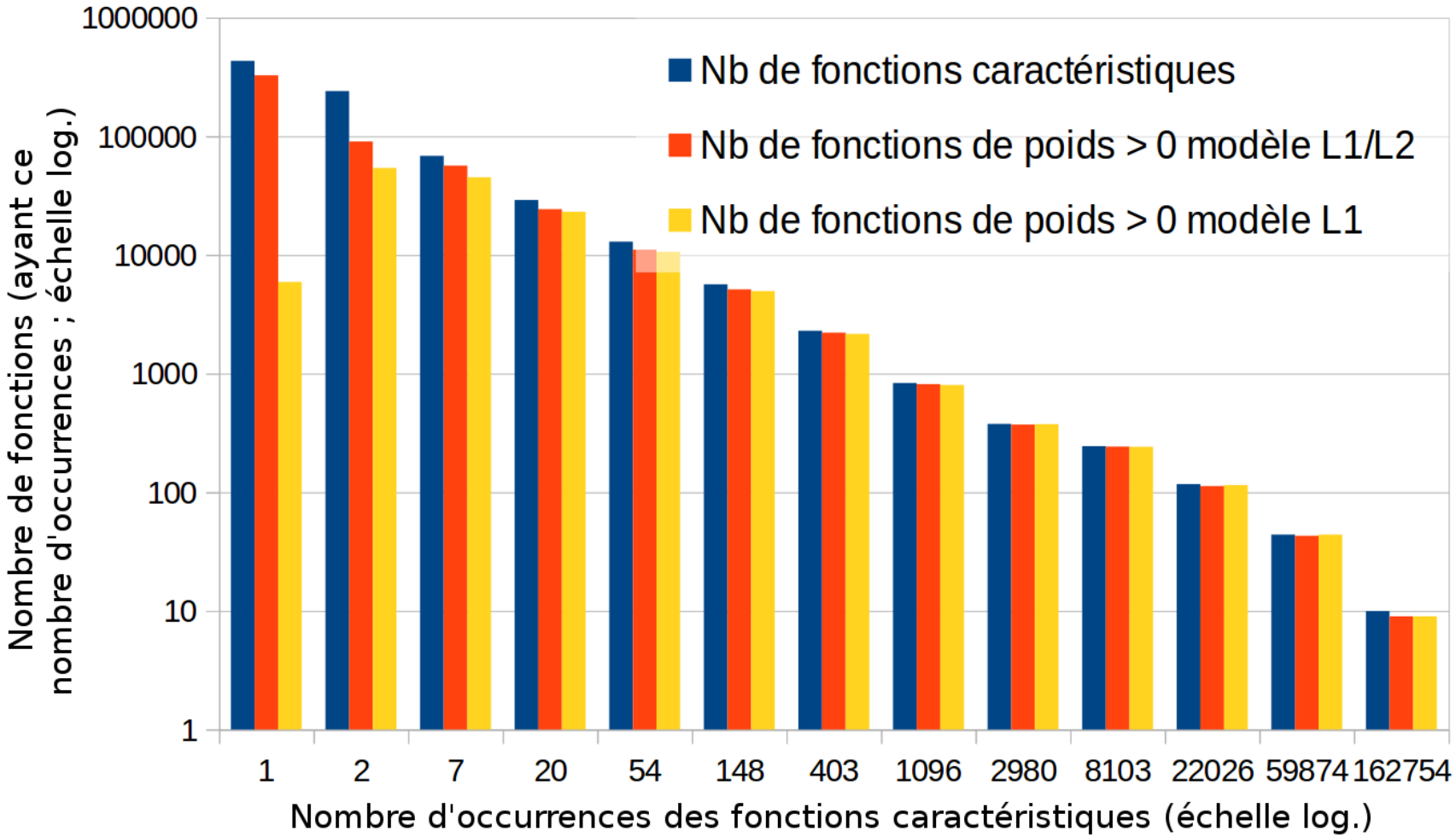
- count how many times a pattern (as encoded by f_j) appears in training data
  - eg., count $\text{wf}_{t-2} = \text{minister}$ AND $\text{capital}_t = \text{YES}$ AND $y_{t-1} = \text{firstname}$ AND $y_t = \text{lastname}$

Distribution of feature functions

Preliminary experiment

- count how many times a pattern (as encoded by f_j) appears in training data
  - eg., count $\text{wf}_{t-2} = \text{minister}$ AND $\text{capital}_t = \text{YES}$ AND $y_{t-1} = \text{firstname}$ AND $y_t = \text{lastname}$
- count how many of these f_j appear in the models ($\lambda_j \neq 0$)

Distribution of feature functions



Distribution of feature functions

Preliminary experiment

- count how many times a pattern (as encoded by f_j) appears in training data
  - eg., count $\text{wf}_{t-2} = \text{minister AND capital}_t = \text{YES}$ AND $y_{t-1} = \text{firstname AND } y_t = \text{lastname}$
- count how many of these f_j appear in the models ($\lambda_j \neq 0$)

Conclusion

- whole dataset appears in the model

Distribution of feature functions

Preliminary experiment

- count how many times a pattern (as encoded by f_j) appears in training data
  - eg., count $\text{wf}_{t-2} = \text{minister}$ AND $\text{capital}_t = \text{YES}$ AND $y_{t-1} = \text{firstname}$ AND $y_t = \text{lastname}$
- count how many of these f_j appear in the models ($\lambda_j \neq 0$)

Conclusion

- whole dataset appears in the model

$\Rightarrow$ if one wants to obtain a similar model, similar data must be provided

  - similar data = same f_j in same proportions

Distribution of feature functions

Preliminary experiment

- count how many times a pattern (as encoded by f_j) appears in training data
  - eg., count $\text{wf}_{t-2} = \text{minister}$ AND $\text{capital}_t = \text{YES}$ AND $y_{t-1} = \text{firstname}$ AND $y_t = \text{lastname}$
- count how many of these f_j appear in the models ($\lambda_j \neq 0$)

Conclusion

- whole dataset appears in the model

$\Rightarrow$ if one wants to obtain a similar model, similar data must be provided

  - similar data = same f_j in same proportions

$\Rightarrow$ make $\mathcal{T}$ similar to whole available data

Distribution of simplified feature functions

Problem

- in practice, f_j cannot be estimated on $\mathcal{N}$ since y_t are unknown
- is previous intuition useful?

Distribution of simplified feature functions

Problem

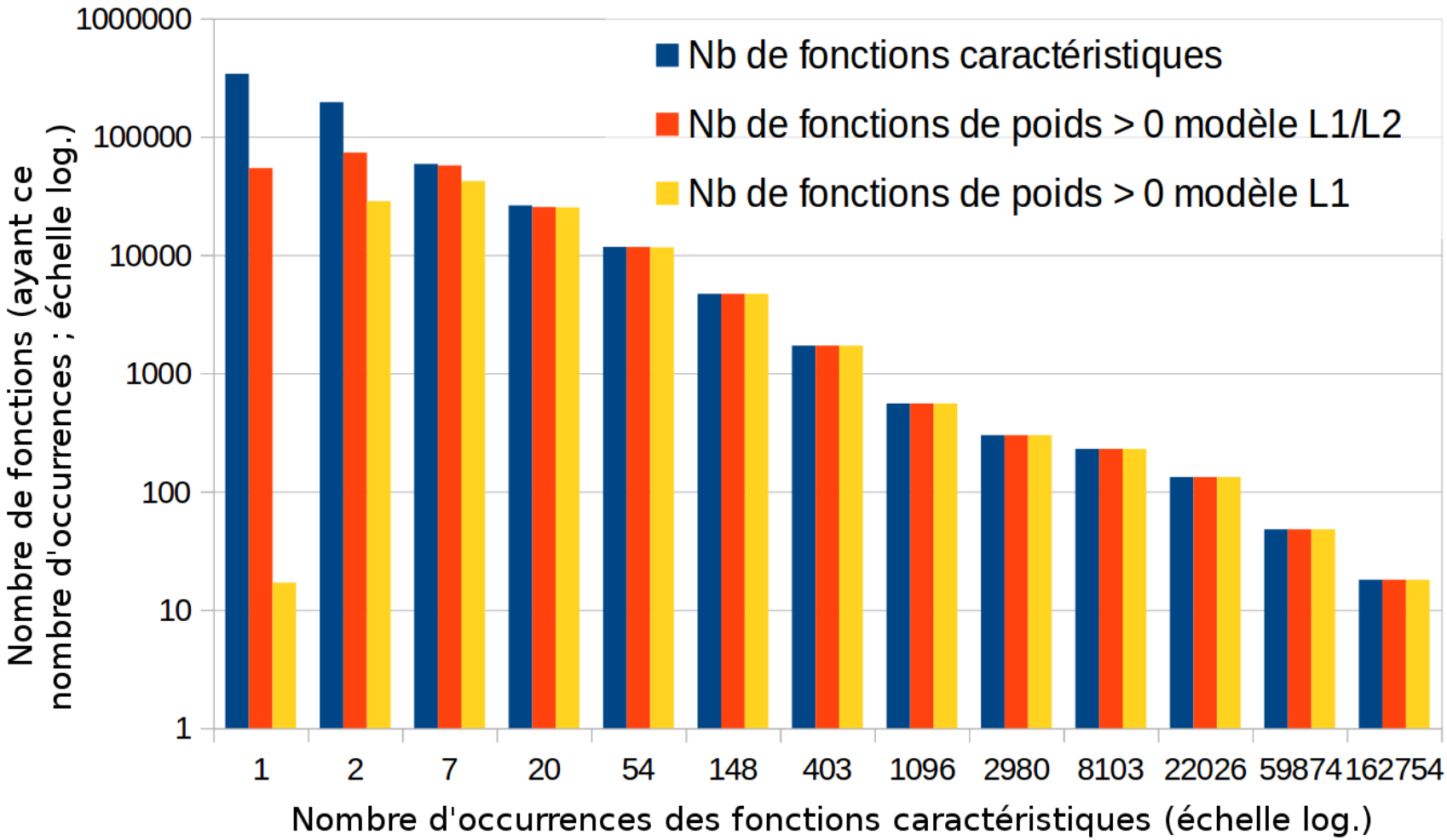
- in practice, f_j cannot be estimated on $\mathcal{N}$ since y_t are unknown
- is previous intuition useful?

Preliminary experiment, bis

- count how many times a pattern without y_{t-1}, y_t
- simplified feature functions f_j^*:

$$f_j^*(x, *, *) = \begin{cases} 1 & \textit{if } \text{wf}_{t-2} = \text{minister AND capital}_t = \text{YES} \\ & \text{AND } y_{t-1} = * \text{ AND } y_t = * \\ 0 & \textit{else} \end{cases}$$

Distribution of simplified feature functions



Representativeness of the training data

Goal

- at each iteration, we want the training set the most similar to the whole dataset
- similar = similar distribution of features, as seen by the feature functions

Example

- consider a feature combination appearing in 10% of the whole dataset
- we want it to appear in 10% of the training set

Test of proportion

Statistical test

- choose the example x that makes the feature proportion of $\mathcal{T} \cup \{x\}$ the most similar to the whole dataset $\mathcal{T} \cup \mathcal{N}$
- for every feature function f_j, we perform a statistical test of proportion between these two sets $\rightarrow z_{j,x}$
- we compute the probability to observe such a proportion difference between the two sets $\rightarrow P(z_{j,x})$

Final score for proportionality

- naive hypothesis: all the feature functions are independent
- thus,

$$\text{next sequence to tag} = \underset{x \in \mathcal{N}}{\text{argmax}} \prod_j P(z_{j,x})$$

Details of the test of proportion

- for two sets $\mathcal{T} \cup \{x\}$ and $\mathcal{T} \cup \mathcal{N}$, denoted 1 and 2:
  - size n_1 and n_2
  - $\hat{p}_1^j = r_1^j / n_1$ is a proportion estimate of a given feature function f_j occurring r_1^j times in set 1
  - $\hat{p}_2^j = r_2^j / n_2$, idem for set 2
  - then we have the following z-score:

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

Outline

Experimental framework

Tasks

- annotation tasks usually handled with CRF

Experimental framework

Tasks

- annotation tasks usually handled with CRF
  - CoNLL2000: chunking, English

Experimental framework

Tasks

- annotation tasks usually handled with CRF
  - CoNLL2000: chunking, English
  - CoNLL2002: named entity recognition, Dutch

Experimental framework

Tasks

- annotation tasks usually handled with CRF
  - CoNLL2000: chunking, English
  - CoNLL2002: named entity recognition, Dutch
  - ESTER: named entity recognition on transcribed texts, French

Experimental framework

Tasks

- annotation tasks usually handled with CRF
  - CoNLL2000: chunking, English
  - CoNLL2002: named entity recognition, Dutch
  - ESTER: named entity recognition on transcribed texts, French
  - Nettalk: grapheme to phoneme

Experimental framework

Tasks

- annotation tasks usually handled with CRF
  - CoNLL2000: chunking, English
  - CoNLL2002: named entity recognition, Dutch
  - ESTER: named entity recognition on transcribed texts, French
  - Nettalk: grapheme to phoneme
  - SensEval-2: word sense disambiguation, English

Experimental framework

Tasks

- annotation tasks usually handled with CRF
  - CoNLL2000: chunking, English
  - CoNLL2002: named entity recognition, Dutch
  - ESTER: named entity recognition on transcribed texts, French
  - Nettalk: grapheme to phoneme
  - SensEval-2: word sense disambiguation, English

Performance evaluation

- independent test set
- precision by words, but for Nettalk: precision on the whole word
- annotation cost = number of words (or letters) annotated

Experimental framework

Technical details

- usual feature patterns and annotation schemes
- CRF implementation: Wapiti [Lavergne et al. 2010]

Experimental framework

Technical details

- usual feature patterns and annotation schemes
- CRF implementation: Wapiti [Lavergne et al. 2010]

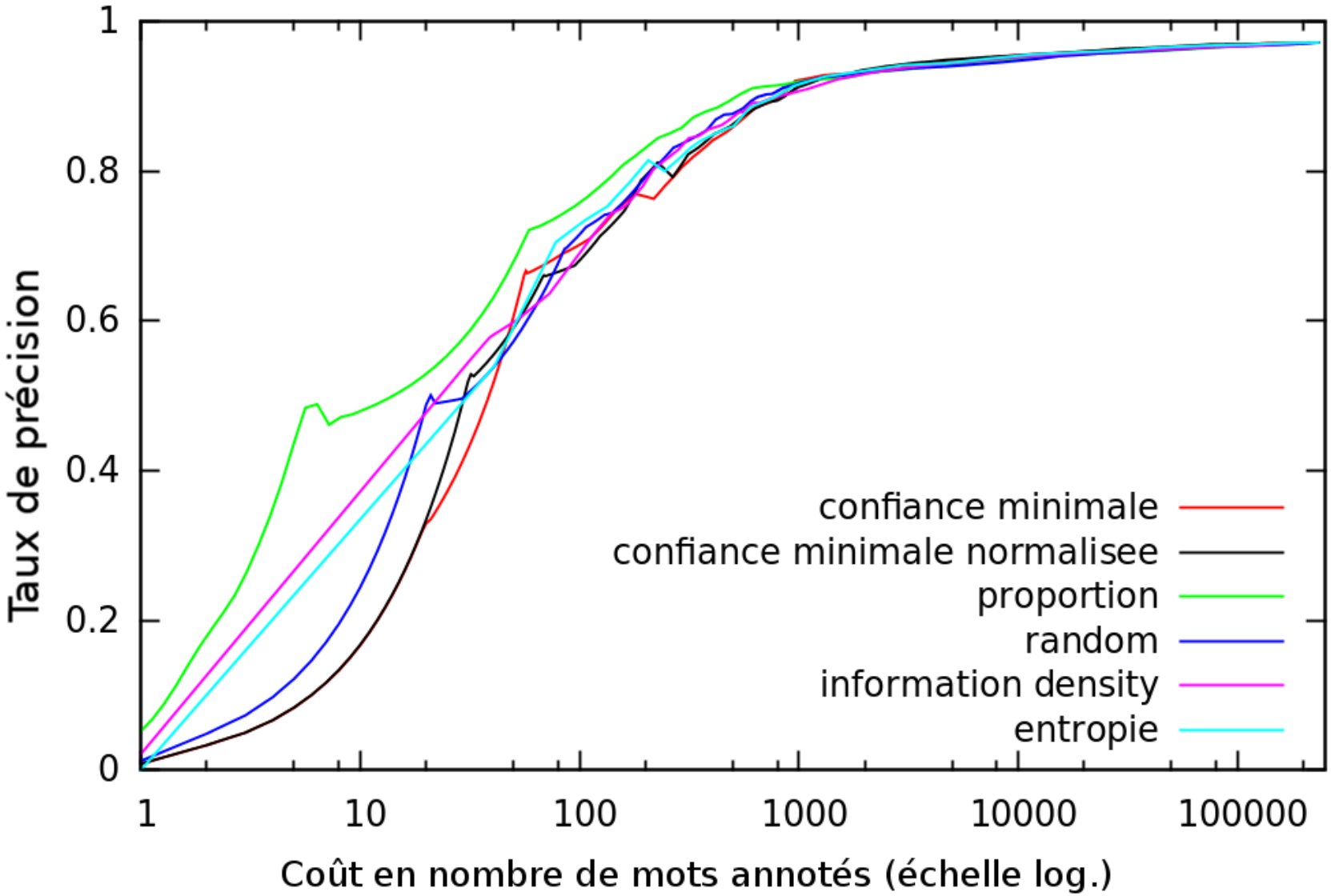
Active learning strategies

- random choice of $x \in \mathcal{N}$
- uncertainty
- entropy
- information density [Seetles & Craven 2008]

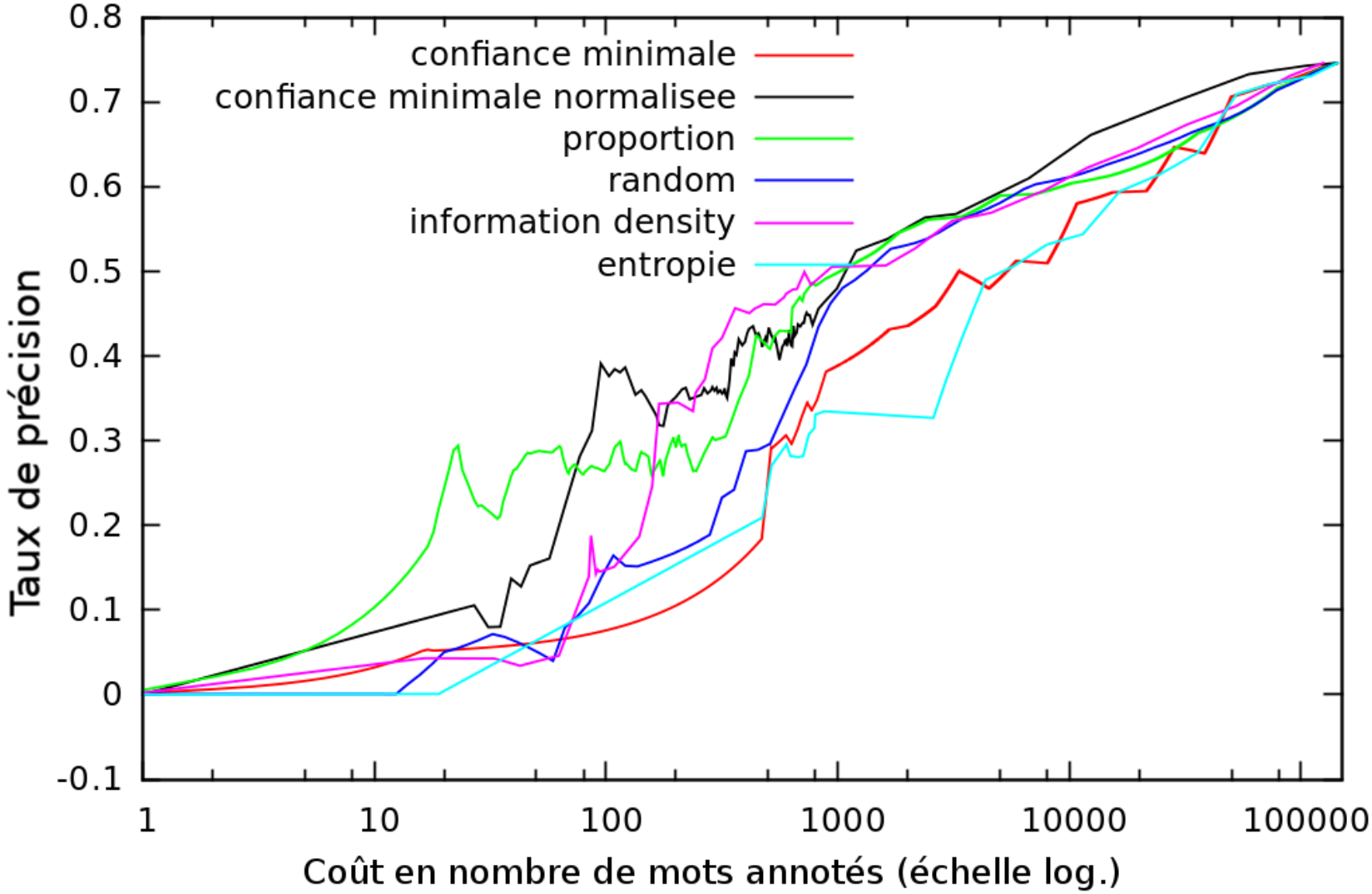
☞ normalized uncertainty

☞ proportionality

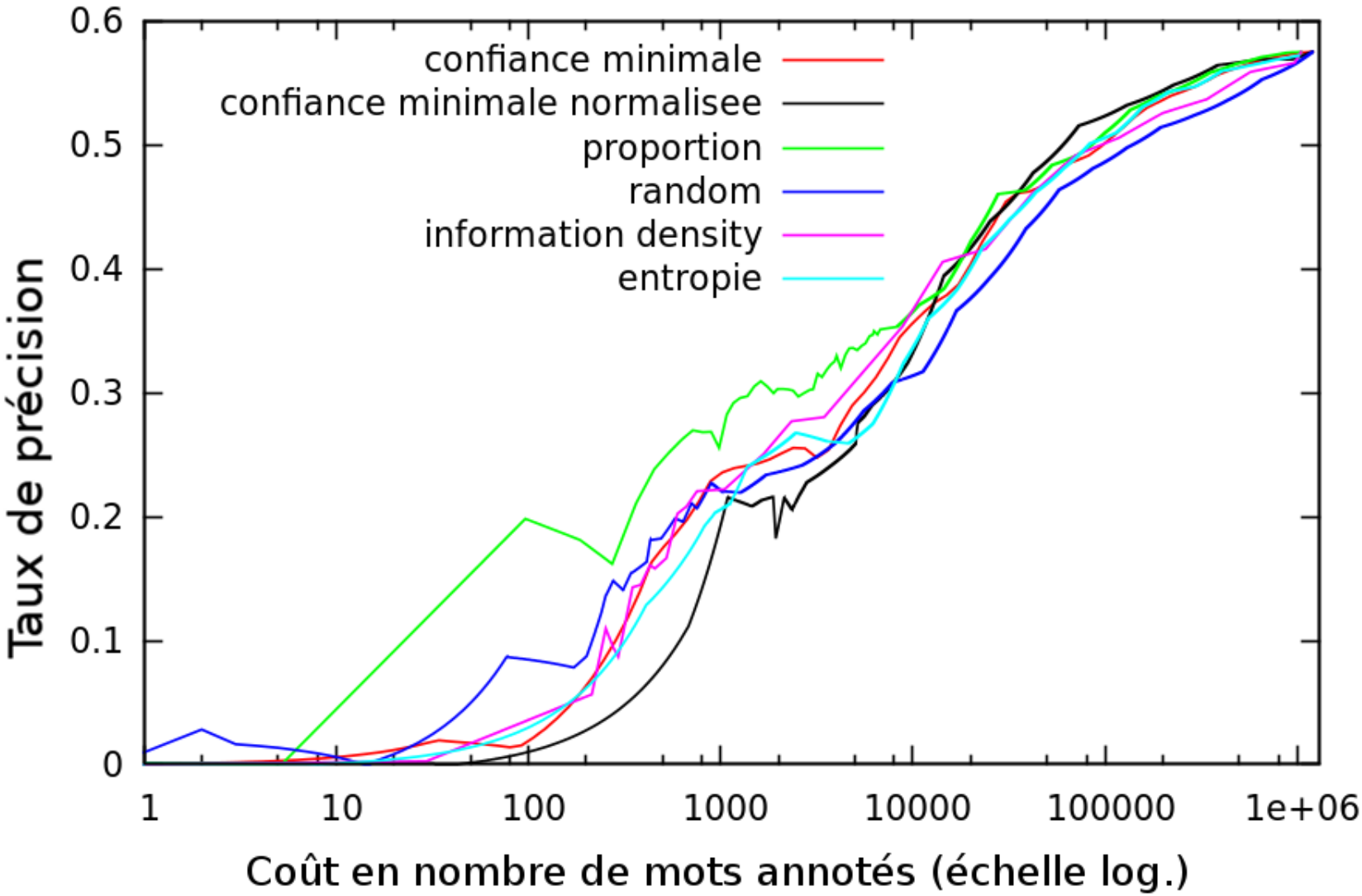
Results on CoNLL 2000 (chunking)



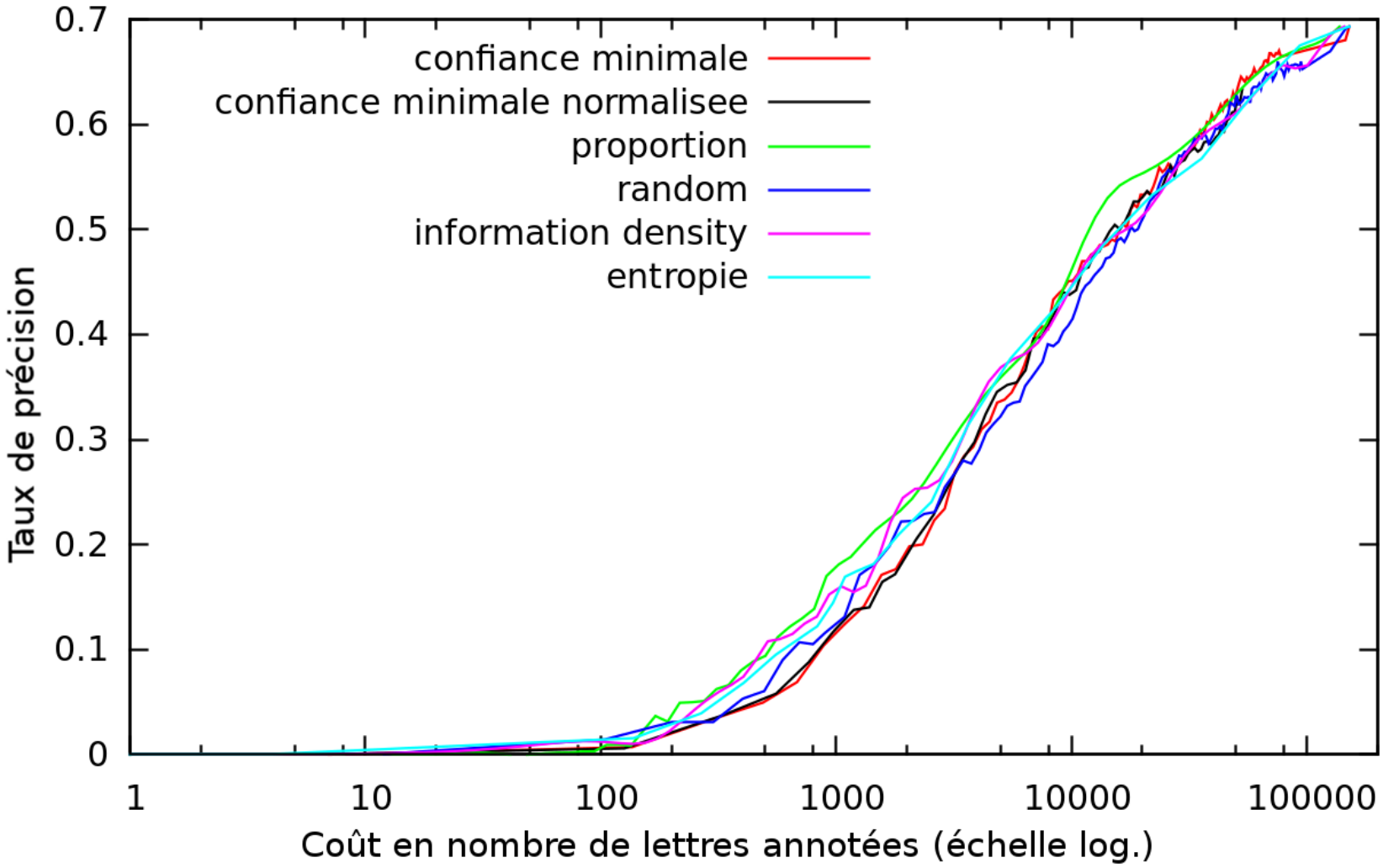
Results on CoNLL 2002 (Dutch NER)



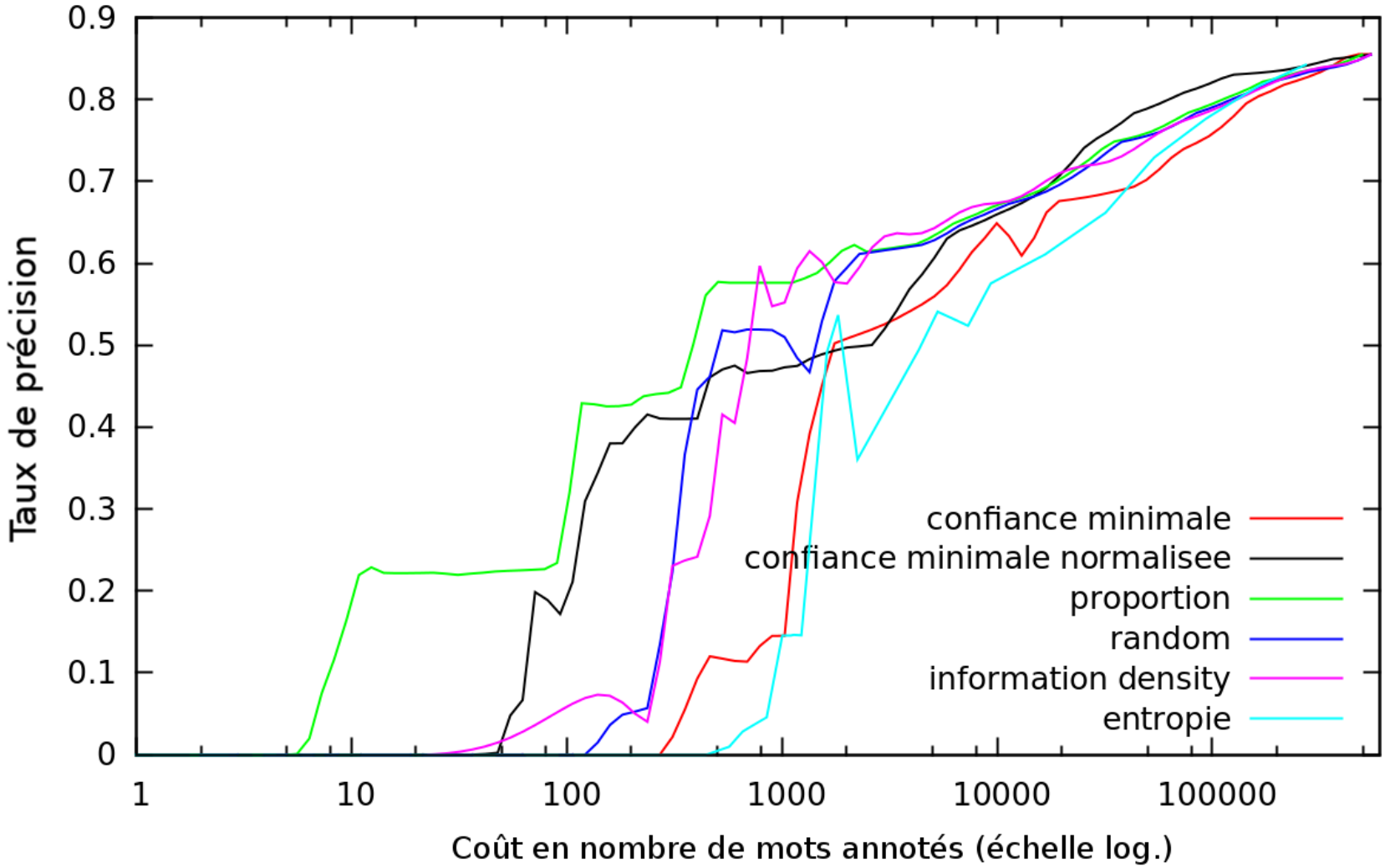
Results on ESTER (French NER)



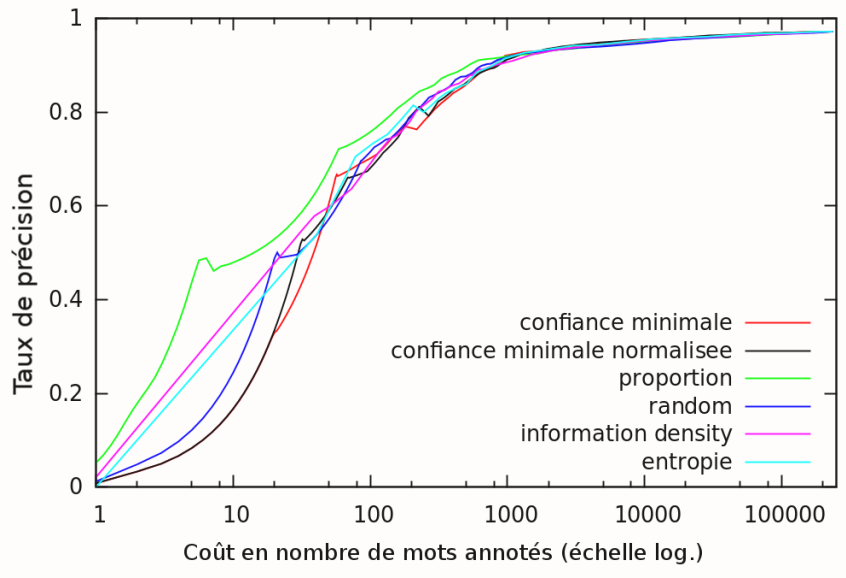
Results on Nettalk (phonetization)

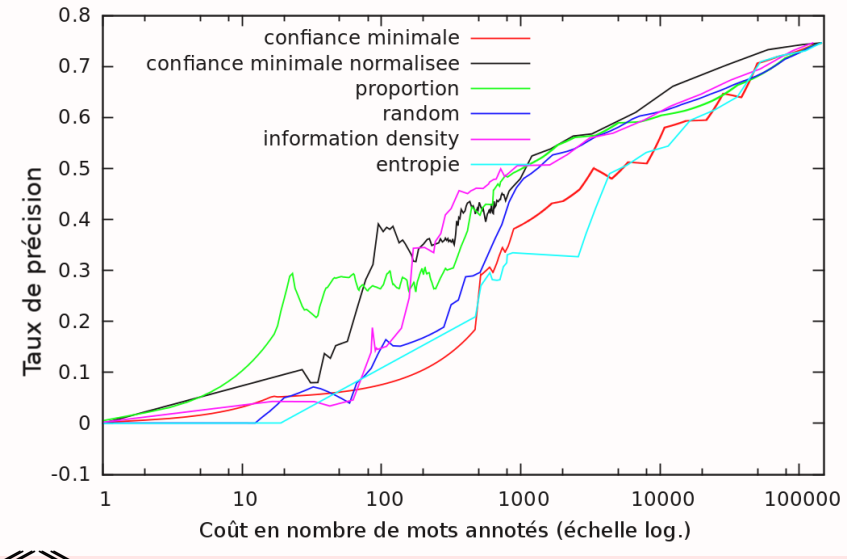


Results on SensEval-2 (WSD)

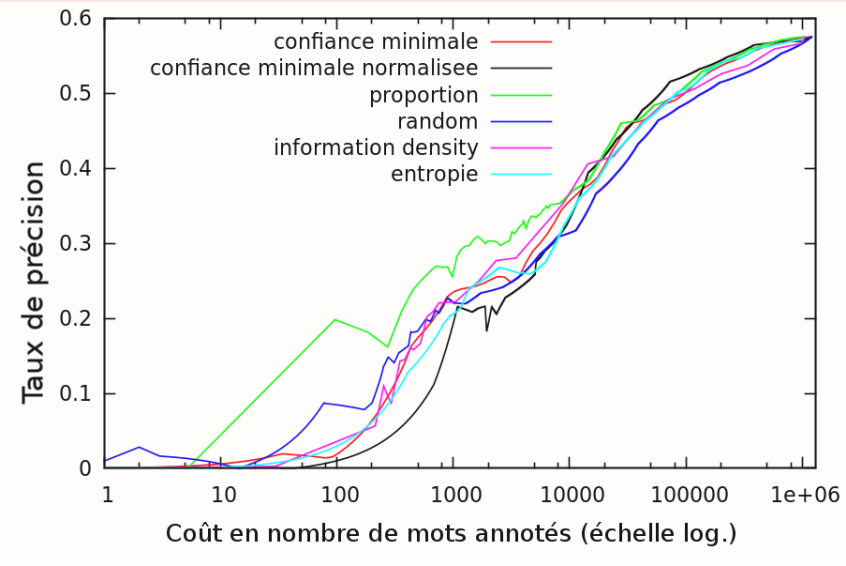


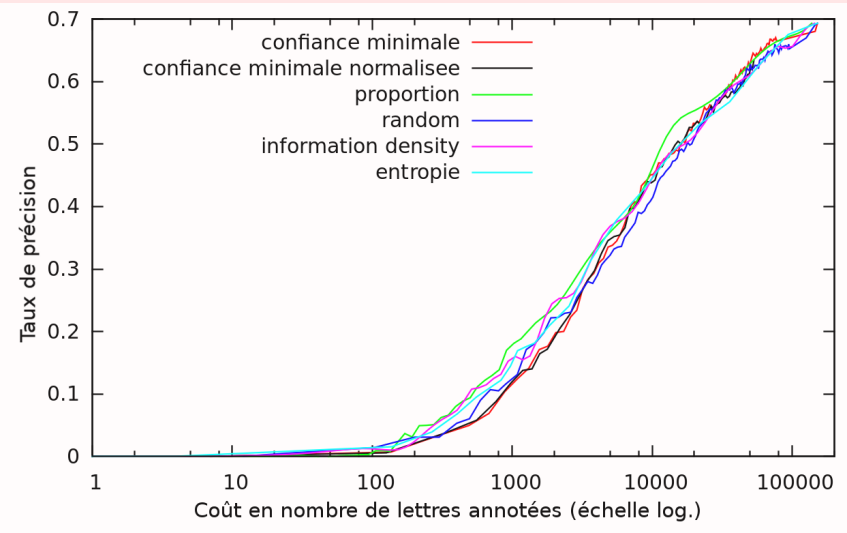
Discussion: convergence?



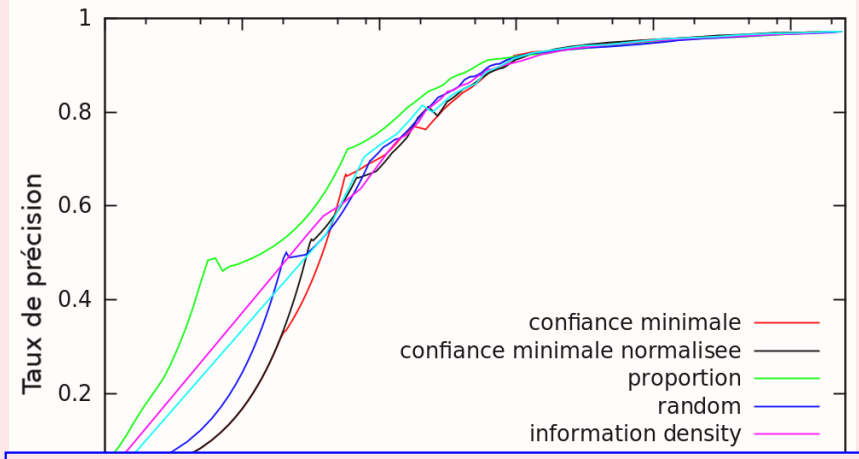


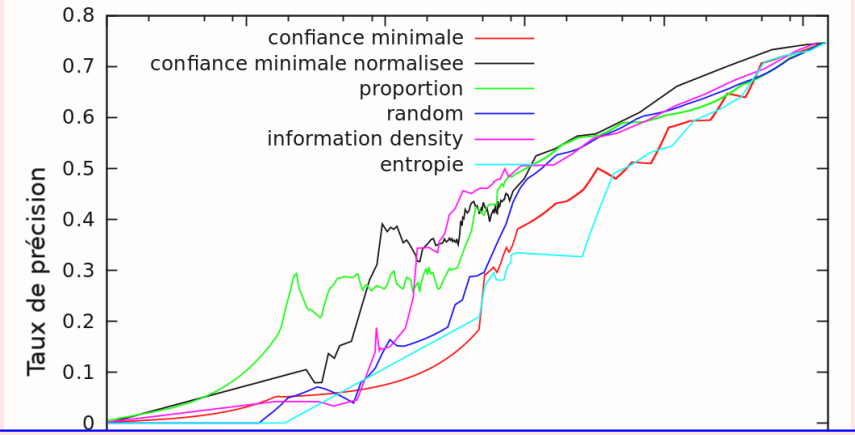
≪≫





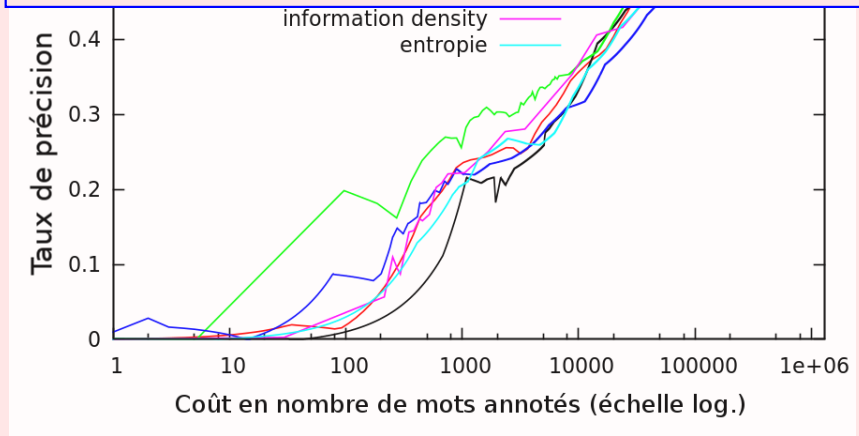
Discussion: convergence?

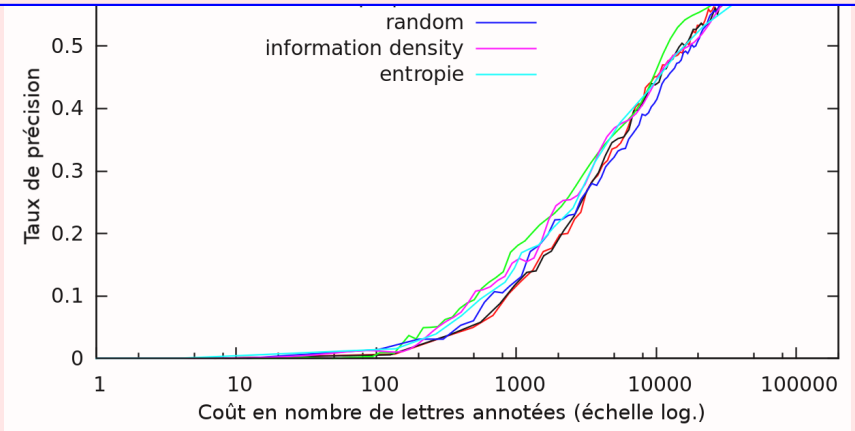




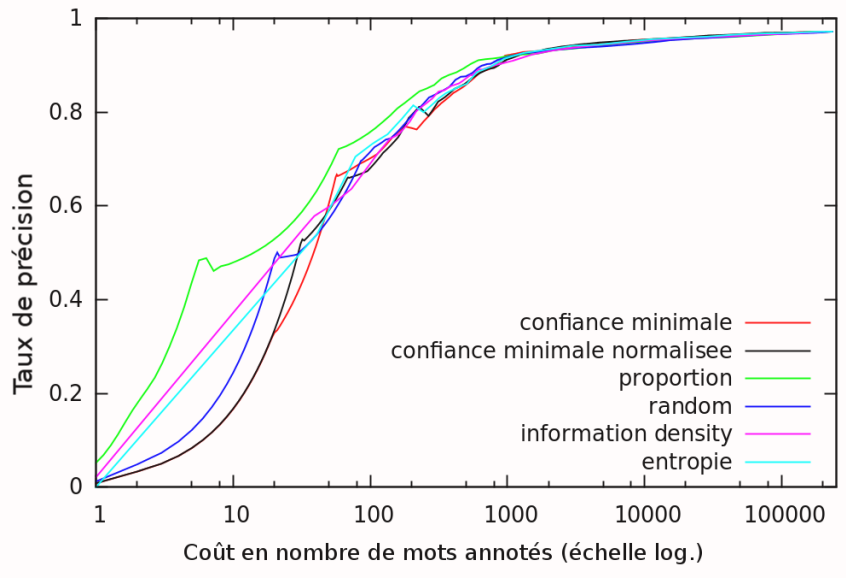
Shapes of the learning curves

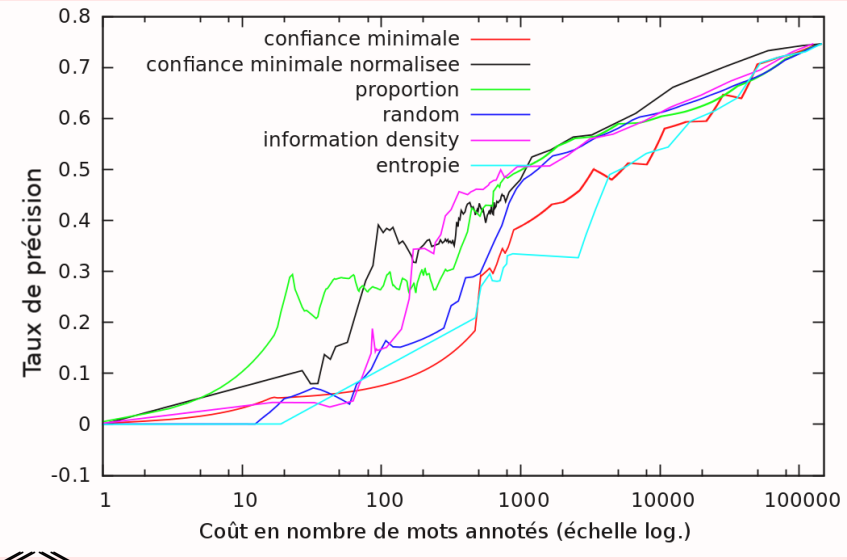
- every dataset (but CoNLL2000) would benefit of more data
- [Garrette & Baldridge, 2013]'s conclusion cannot be generalized



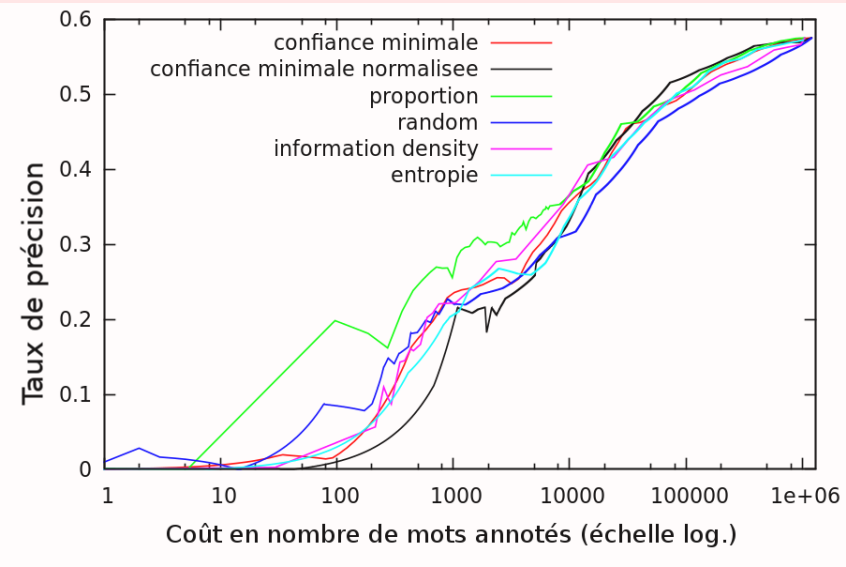


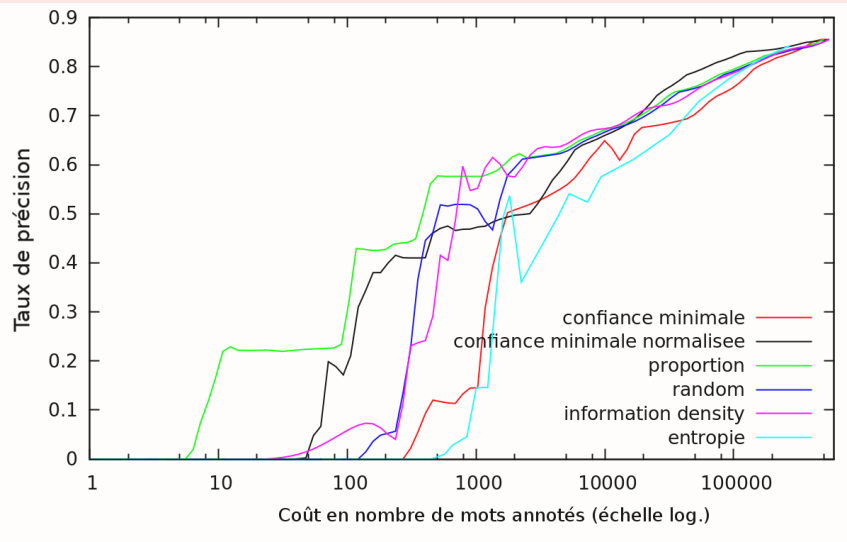
Discussion: state-of-the-art results?



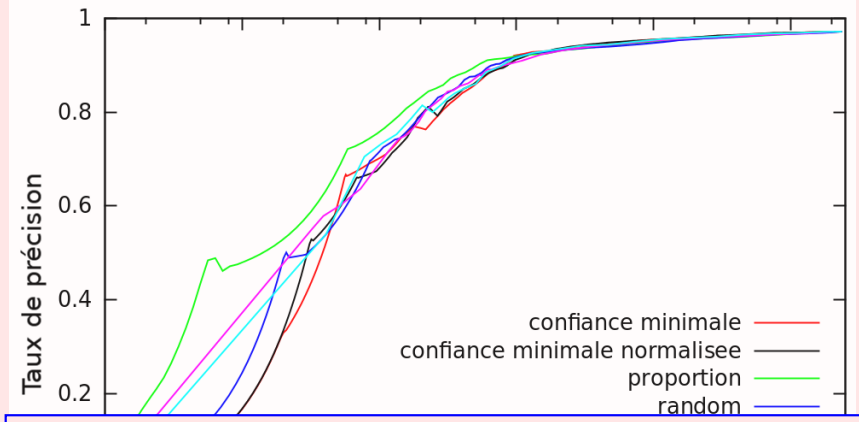


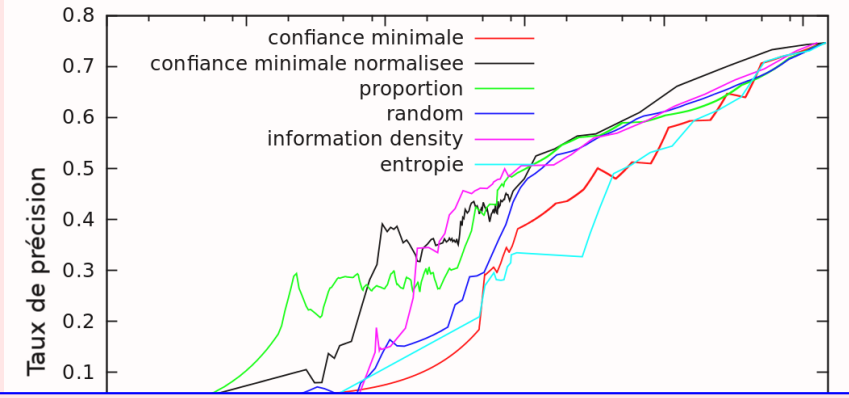
≪≫





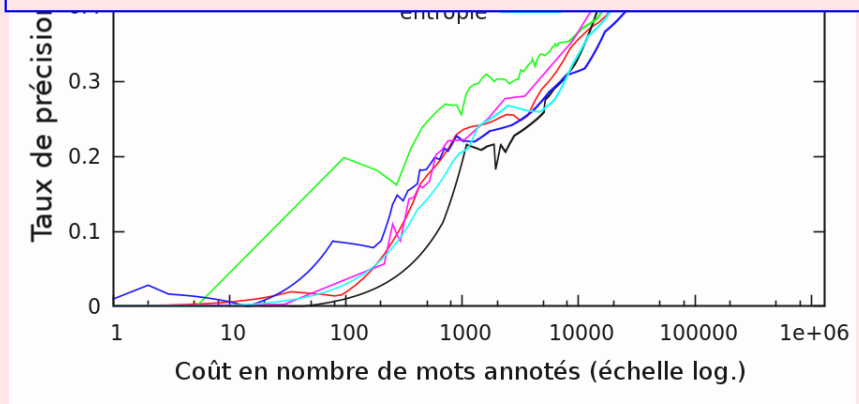
Discussion: state-of-the-art results?

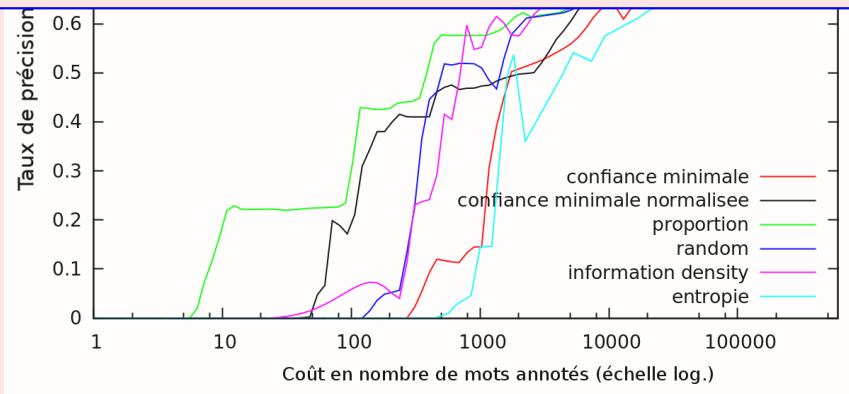




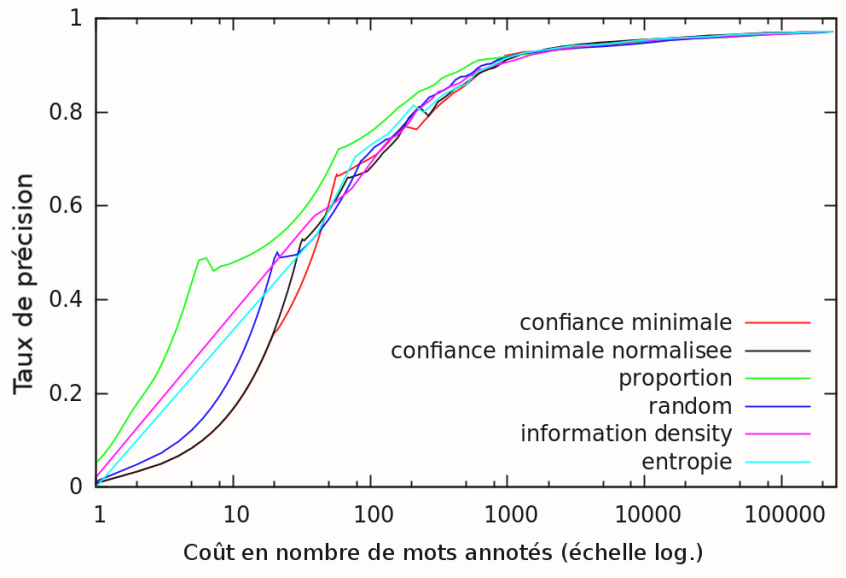
State-of-the-art strategies

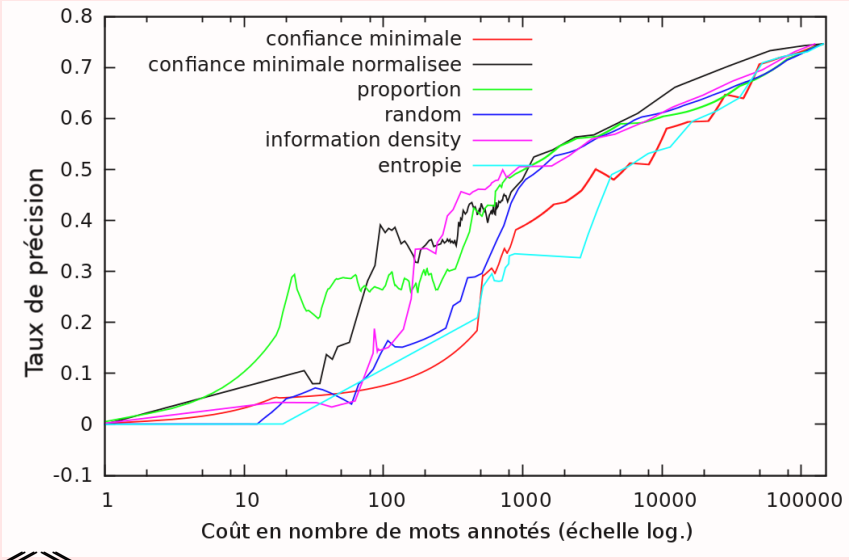
- do not perform well, sometimes worse than random
- hidden fact in the literature [Settles & Craven, 2008] where the length of a sequence does not account for the annotation cost



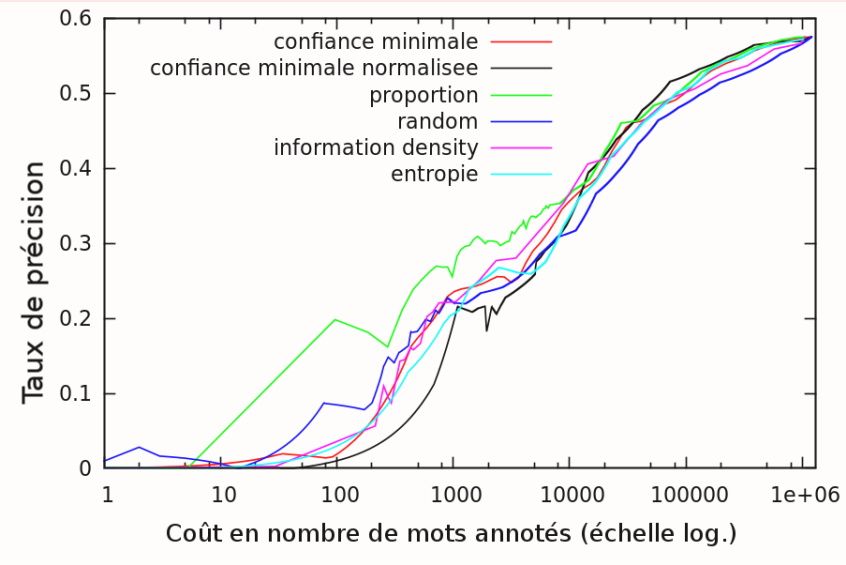


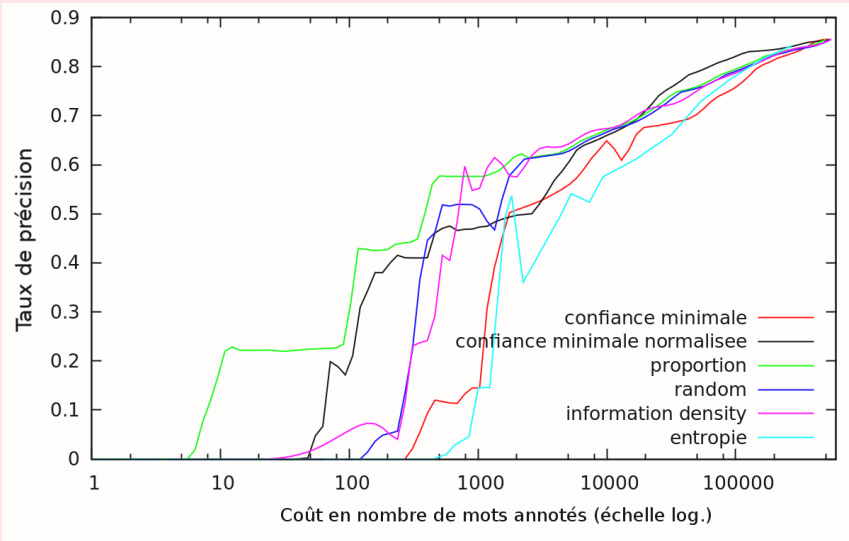
Discussion: normalization effect?



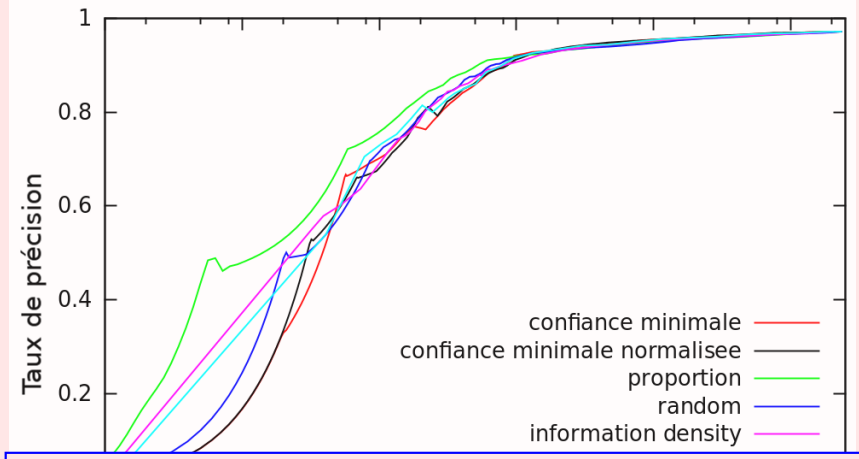


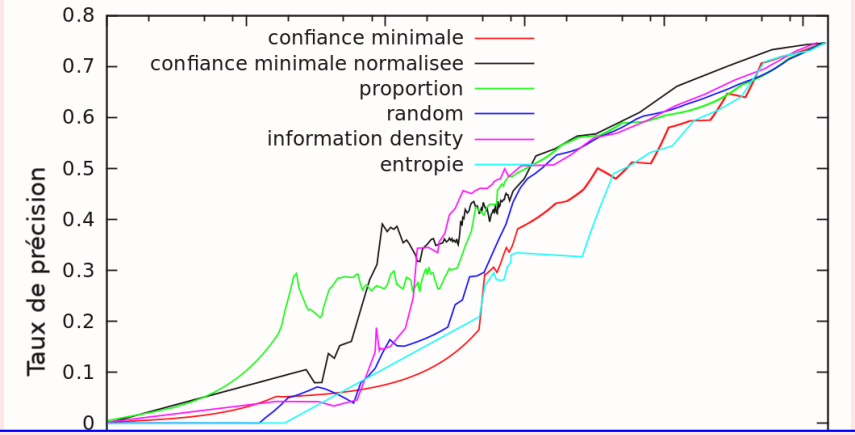
≪≫





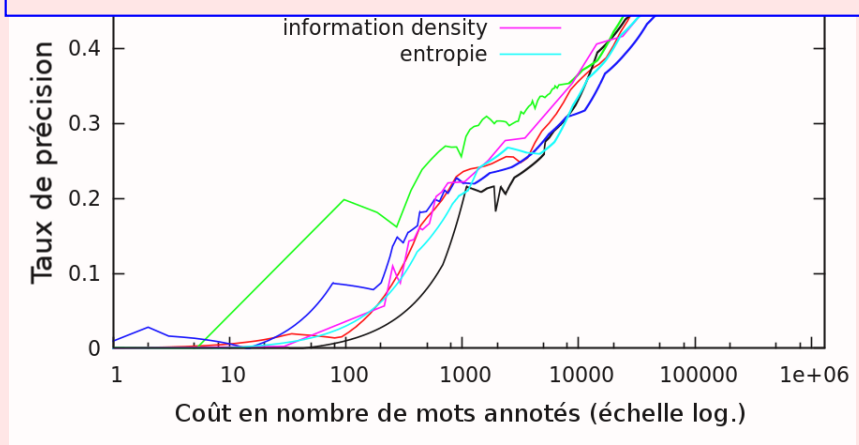
Discussion: normalization effect?

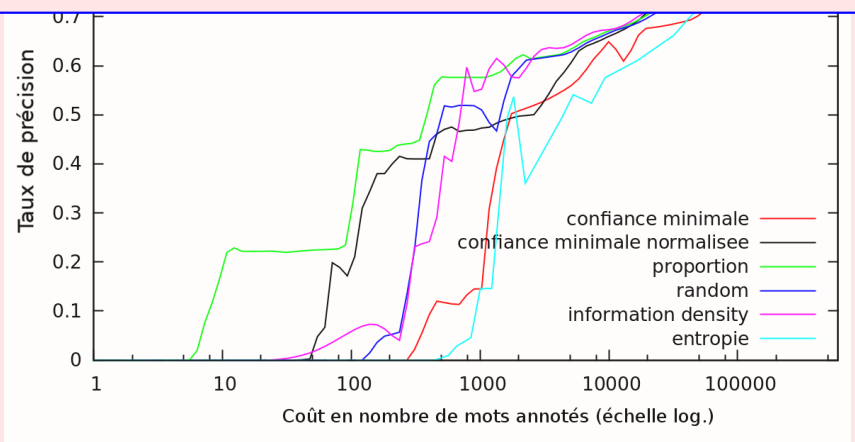




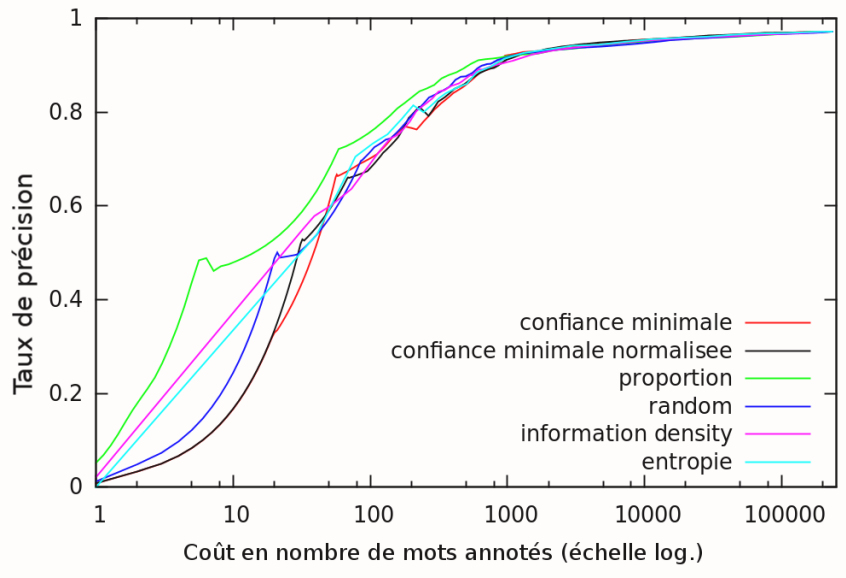
Effect of normalization

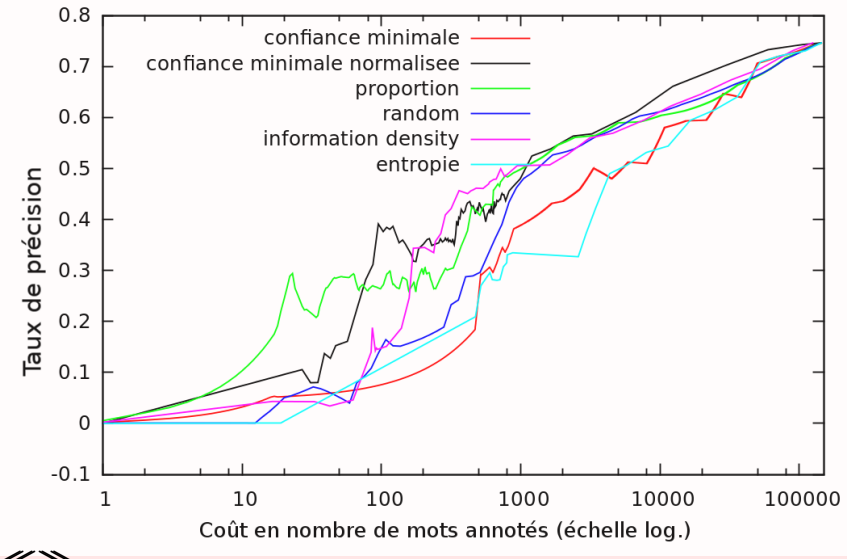
- improve results in every case but ESTER
- best strategy when $\mathcal{T}$ is big enough



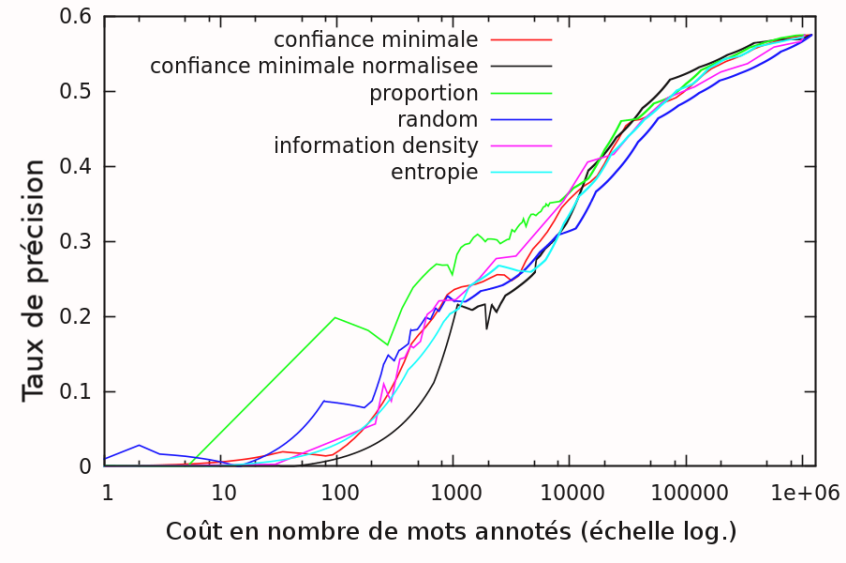


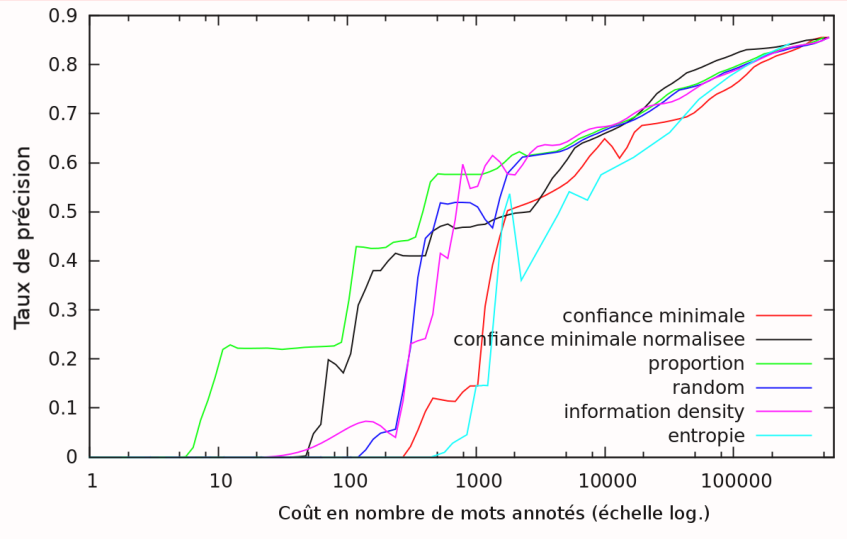
Discussion: proportionality strategy?



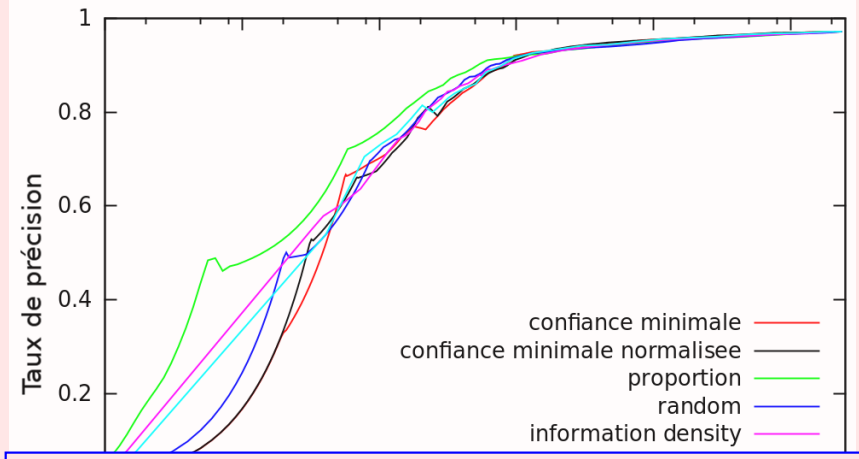


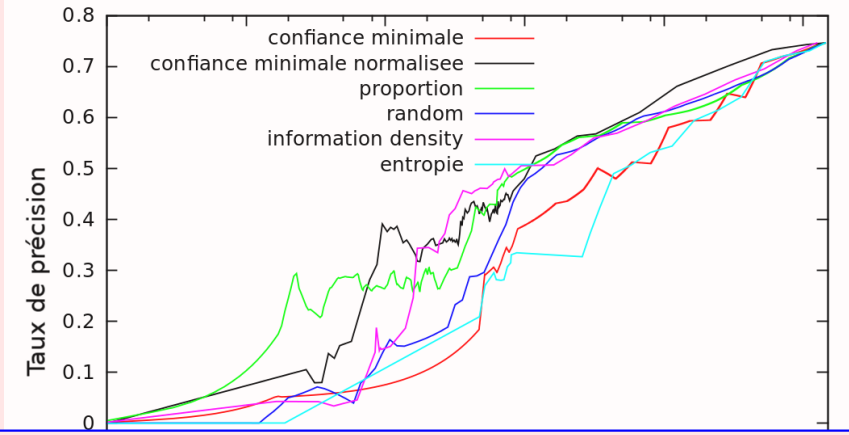
≪≫





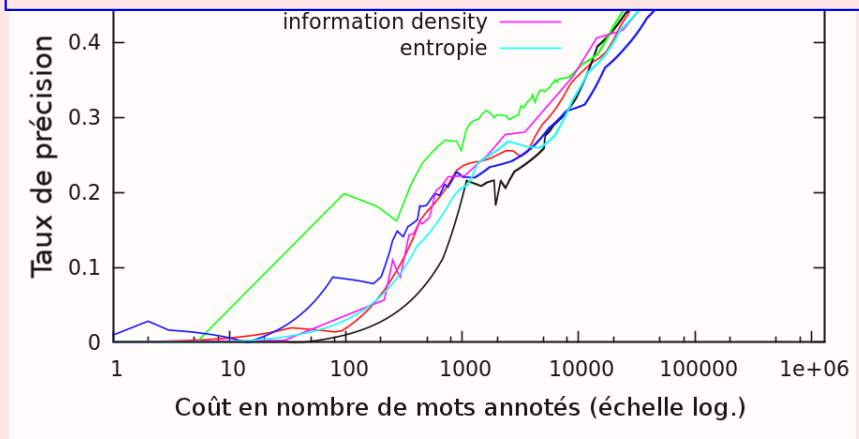
Discussion: proportionality strategy?

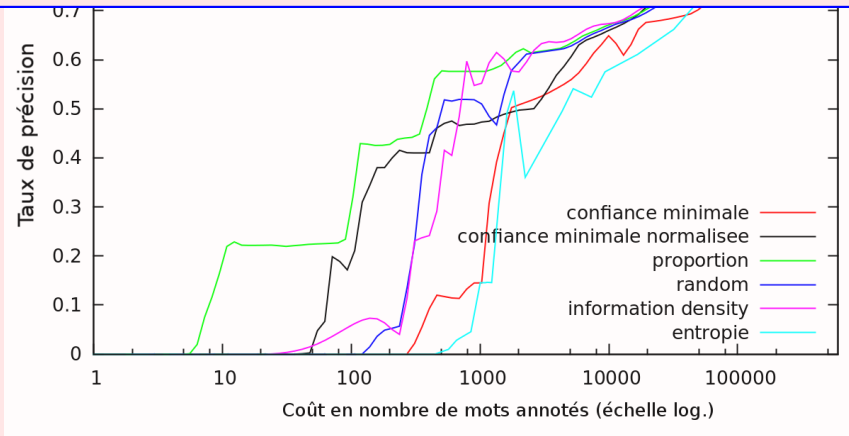




Strategy based on proportionality

- performs well in most tasks ≪≫
- huge gains when $\mathcal{T}$ is small, less interesting after that





Outline

Conclusion

About state-of-the-art

- existing strategies not always evaluated with realistic annotation cost
- when cost is length dependent, most existing strategies are not better than random
- length normalization improves results for uncertainty based strategies

Conclusion

About state-of-the-art

- existing strategies not always evaluated with realistic annotation cost
- when cost is length dependent, most existing strategies are not better than random
- length normalization improves results for uncertainty based strategies

Proportionality

- same family than information density [Setteles & Craven 2008]
- good results on the first iterations, where CRF results are not yet reliable

Foreseen work

Normalization

- apply to other uncertainty based strategies (entropy...)
- consider features instead of length to normalize

Foreseen work

Normalization

- apply to other uncertainty based strategies (entropy...)
- consider features instead of length to normalize

Combining strategies

- use proportionality first and then normalized uncertainty
- learn to combine strategies = learn to rank $x \in \mathcal{N}$