NLP evaluation

C. Grivaz, J.-C. Chappelier & M. Rajman

Laboratoire d'Intelligence Artificielle
Faculté I&C

Outline

- Evaluation protocol
- Gold standard
- Inter-annotator agreement
- Evaluation metrics
- Validity of the results

NLP evaluation motivations

- Evaluate the improvement of the technology on a specific task
- Provide gold standards and objective comparison methods
- Develop research and technology in NLP

C. Grivaz, J.-C. Chappelier, M. Rajman

NLP evaluation protocol

from a design by Layerace / Freepik

1. Define a control task
2. Produce a reference (golden truth)
3. Assess the quality of the reference
4. Evaluate NLP system(s) on the reference
5. Compare evaluations (statistical significance)
6. Publish and discuss results

C. Grivaz, J.-C. Chappelier, M. Rajman

Example: *n*-ary classification of linguistic entities

1. Define a control task

Many of the tasks performed by the existing NLP tools can be *generically* modeled as **tagging tasks**, i.e.:

the NLP tool automatically assigns, to each of the linguistic entities (documents, sentences, words, ...) to be processed, a single tag selected out of a finite number of possible tags.

For example:

- a part-of-speech tagger assigns, to each of the words present in a sentence, the grammatical category this word is associated with within this sentence;
- a parser assigns, to each of the sentences present in a corpus, a tag “correct” (resp. “incorrect) depending on whether this sentence can be considered as syntactically correct (resp. incorrect) w.r.t. the grammar used by the parser;
- a language identifier assigns, to each of the documents present in a corpus, a tag identifying the language this document is written in.

Binary vs. *n*-ary classifications

If the number of distinct tags that can be assigned by a classifier is equal to n, the classification is generically referred to as an *n*-ary classification;

More specifically, we have:

- if n = 2 $\longrightarrow$ **binary** classification
- if n = 3 $\longrightarrow$ **ternary** classification

Notice that any *n*-ary classification (using tags $t_1, t_2, ..., t_n$) can be decomposed into a combination of n binary classifications (respectively using the two tags t_i and “not t_i”); however, these n classifications may not be independent!

Examples of binary and *n*-ary classifications

Examples of binary classifications:

- sentiment analysis: negative feeling vs. positive feeling
- relevance analysis: relevant vs. “not relevant”

Examples of n-ary classifications:

- part-of-speech tagging: as many tags as grammatical categories (e.g. Noun, Verb, Adjective, Adverb, Determiner, Pronoun, ...)
- language identification: as many tags as languages to be identified (English, French, Spanish, German, ...)

An illustrative example: an English identifier

Consider a language identifier, i.e. an NLP tool able to automatically associate to any text (or fraction of text) a tag identifying the language it is written in
(e.g. EN for English, FR for French, GE for German, ES for Spanish, etc)

If N languages can be identified, the language identifier corresponds to an N-ary classifier, and ...

...if we keep all EN tags unchanged and transform all the other produced tags into a new tag notEN, we transform the N-ary classifier into a binary classifier (one of the N possible ones) corresponding to an English (text) identifier, i.e. an NLP tool that determines whether a text (or a fraction of text) is written in English or not

NLP evaluation protocol (reminder)

1. Define a control task
2. Produce a reference
3. Assess the quality of the reference
4. Evaluate NLP system(s) on the reference
5. Compare evaluations (statistical significance)
6. Publish and discuss results

Need for a set of correct answers ("Gold standard")

Contrary to some other tasks, there is generally no simple way to know if a NLP system gives correct results

especially when the goal of an NLP task is to mimic something that a human can do

☞ gold standard : set of correct answers for a *sample* of typical inputs for the control task

Evaluation methodology:

the sample of input is then given to the automatic system and its output is compared to the gold standard

Reference = data annotated with expected outputs

In NLP, the **reference** (golden truth) often takes the form of a corpus, in which each of the linguistic entities to be processed is associated with the expected (i.e. “correct”) output, i.e. the output that would be produced by a human expert performing the control task.

We talk of an **annotated corpus**, the *annotations* being the outputs associated with the linguistic entities.

When the annotations are produced by humans (and not by an automated NLP system), we talk of a *manually annotated corpus*.

A reference is therefore a manually annotated corpus produced by humans, who can be considered as experts for performing the control task.

Annotations can be very simple...

For example, in the case of the English text identifier, it could be a simple `EN/notEN` tag associated with each of the texts to be processed:

The cat ate the mouse	EN
My tailor is rich	EN
Sie ist jung	notEN
Luttons ensemble	notEN
El llega tarde	notEN
Come on dude	EN
Come state	notEN

...or quite complicated!

Example (the Penn Discourse Treebank)

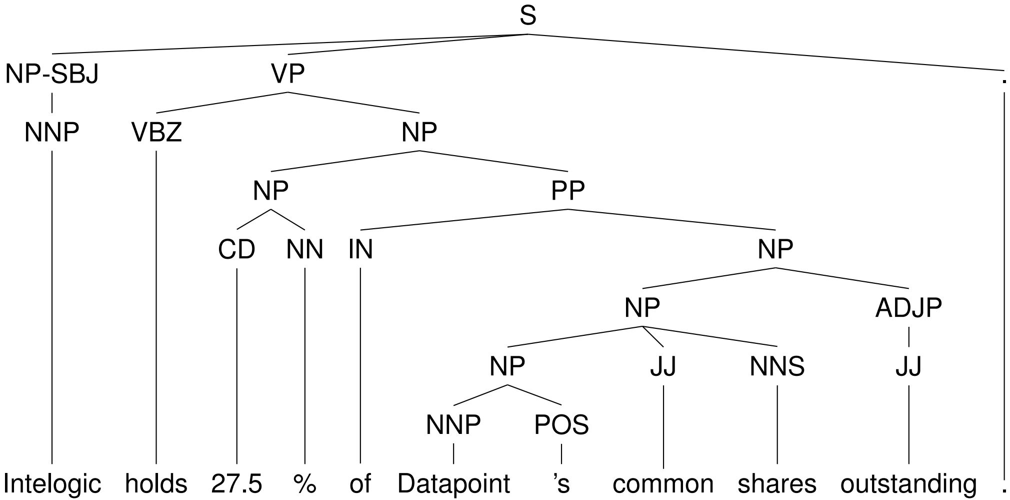
Intelogic holds 27.5% of Datapoint’s common shares outstanding.

```
(S
   (NP-SBJ (NNP Intelogic) )
   (VP (VBZ holds)
     (NP
       (NP (CD 27.5) (NN %) )
       (PP (IN of)
         (NP
           (NP
             (NP (NNP Datapoint) (POS ’s) )
             (JJ common) (NNS shares) )
           (ADJP (JJ outstanding) ))))))
   (. .) )
```

What does it mean?

The former annotation example is a parse tree representing the syntactic structure corresponding to the sentence:

Intelogic holds 27.5% of Datapoint's common shares outstanding.

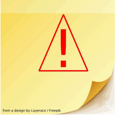


Gold standard impact

- Gold standard creation is extremely expensive
- But globally amortized: if a gold standard exists, the whole field is likely to use it for comparison and evaluation

  Notice however that a systematic reuse of the same gold standard introduces a bias to the evaluated task.

Gold standard creation process



from a design by Layerace / Freepik

- Properly define the task in an annotator manual
- Select the corpus to annotate
- Train annotators:
  - annotation instructions
  - assess annotation quality: inter-annotator agreement (or other appropriate measures)
- Annotate

NLP evaluation protocol (reminder)

1. Define a control task
2. Produce a reference from a large amount of *typical* data (for the task)
3. Assess the quality of the reference
4. Evaluate NLP system(s) on the reference
5. Compare evaluations (statistical significance)
6. Publish and discuss results

Humans do not always agree on NLP tasks

- Despite the annotator manual, divergences always exist
- These divergences highly depend on the subjectivity of the task
- A resource is considered good only if the divergences are low

☞ measure Inter-annotator agreement

Disagreement example: word sense disambiguation

Task: Word Sense Disambiguation (WSD):

label each word of a text (= within context) to its corresponding sense (typically from an ontology)

Example (easy):

> *I can hear bass sounds.*
> *They like grilled bass.* `[fish, named "bar" in French]`

Example (not so easy):
disambiguate usage of `national` with an ontology where `national` means:

> *1) limited to or in the interest of a particular nation*
> *2) concerned with or applicable to or belonging to an entire nation or country*

`[from WordNet 3.1]`

Even relatively objective tasks lead to disagreement: syntax example

Put the block in the box on the table.

What is the attachment site of *on the table* ?

Measuring inter annotator agreement

- “Inter annotator agreement” (IAA) is considered a measure of the quality of gold standards
- It is also a measure of the subjectivity of a task
- It must be objectively measured and reported

Raw agreement

Simplest measure of agreement:

$$\text{raw agreement} = \frac{\text{nb items agreed}}{\text{total nb of items}}$$

Raw agreement drawback

Raw agreement doesn't take *by-chance agreement* into account

Example

On a binary classification corpus having 70% of non-ambiguous items, two annotators systematically disagree about all ambiguous items:

		A	
		yes	no
B	yes	0	10
	no	20	70

$$\text{raw agreement} = \frac{70}{100}$$

They get a 70% raw agreement despite their complete disagreement!

Dealing with chance agreement

from a design by Layerace / Freepik

Taking chance agreement into account:

- Idea: discount chance agreement

$$\frac{\text{observed\_agreement} - \text{chance\_agreement}}{1 - \text{chance\_agreement}}$$

- Several measures exist,
  which differ in the way they take "chance agreement" into account

Cohen’s kappa

Cohen’s κ (“kappa”) is the most famous inter annotator agreement coefficient for 2 annotators only (generalization: Fleiss’ kappa).

It takes each annotator into account (independently).

Example

		A	
		yes	no
B	yes	0	10
	no	20	70

- Chance of saying yes: A: 0.2, B: 0.1
- Chance of saying no: A: 0.8, B: 0.9
- Chance of saying both yes if independent: $0.2 \times 0.1 = 0.02$
- Chance of saying both no if independent: $0.8 \times 0.9 = 0.72$
- Chance of independent agreement: $0.72 + 0.02 = 0.74$

$$\kappa = \frac{\text{observed\_agreement} - \text{chance\_agreement}}{1 - \text{chance\_agreement}} = \frac{0.7 - 0.74}{1 - 0.74}$$

$$= -0.15$$

Interpretation of Cohen’s kappa

- Positive: better than chance
- Negative: worse than chance (correlated disagreement)
- 1: perfect agreement
- 0 statistical independence
- more than 0.6 is usually considered ok, and more than 0.8 considered good

Practices

- IAA measures are almost always reported, but often only the raw agreement is given
- IAA is often only measured on a sample, sometimes on the whole corpus
- If IAA is computed on a sample, the rest of the corpus is often annotated by one person only
- Only one annotation set is given at the end.
  When several annotations exist, they are processed a posteriori (suppression from the corpus, selection of a tag by vote or some ad hoc decision)

NLP evaluation protocol (reminder)

1. Define a control task
2. Produce a reference from a large amount of *typical* data (for the task)
3. Assess the quality of the reference
4. Evaluate NLP system(s) on the reference
5. Compare evaluations (statistical significance)
6. Publish and discuss results

Importance of separating the data

from a design by Layerace / Freepik

Comparing the program output to a gold standard

Methodological issue: clearly separate the data:

- Separate training (and validation) from testing

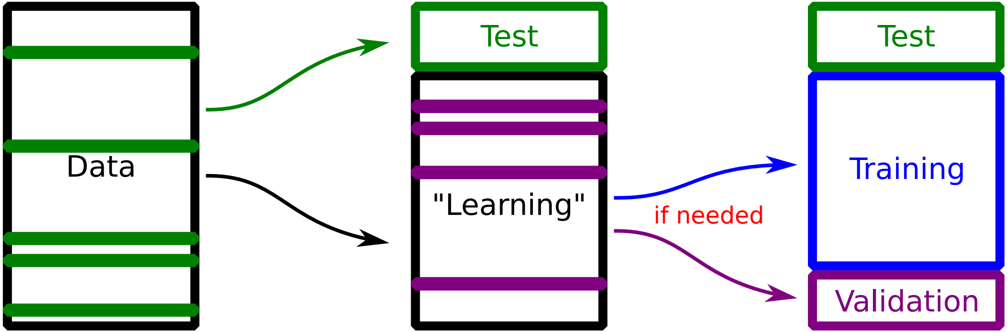
  Do it fully honestly blindly randomly!! ;-)
- Validation set: allows to estimate overfitting or meta-parameters.

  Not to be confused with test set![1]

  ☞ clearly separated from test set (validation set is indeed a kind of training set):
  - Train on the training set
  - Test and adjust meta parameters on validation set
  - Reduce overfitting using the validation set
  - Final testing on the testing set (don't even look at it before!)
- Repeat *all* this several times (to estimate variance)

[1] The more so as so-called "*cross-validation*" is an evaluation method, done on the test set, which has *nothing* to do with the validation set!!

Training, validation and test sets



The confusion matrix

The confusion matrix is not an evaluation metric (i.e. a measure) itself, but it gives complete information about the success and errors from which several evaluation metrics can be derived.

All the evaluation metrics are different kind of summaries of the confusion matrix in one way or another.

The confusion matrix represents, for each reference class, how the system classifies the corresponding items.

Example (ternary classification)

		reference		
		A	B	C
system	A	35	2	10
	B	3	46	1
	C	5	6	12

Example: English identifier (1/2)

Let's consider the English identification example again:

	Reference	System
The cat ate the mouse	EN	EN
My tailor is rich	EN	EN
Sie ist jung	notEN	notEN
Luttons ensemble	notEN	notEN
El llega tarde	notEN	notEN
Come on dude	EN	notEN
Come state	notEN	EN

where the fist column of tags corresponds to the reference tags (produced by human annotators) and the second to the tags produced by a given NLP English text identifier.

Example: English identifier (2/2)

In this case, the corresponding confusion matrix is:

		reference	
		`EN`	`notEN`
system	`EN`	2	1
	`notEN`	1	3

where

- the values on the diagonal correspond to the correct classifications (the `EN-EN` cases are often called the "*true positives*" and the notEN-notEN cases the "*true negatives*")
- the values outside the diagonal correspond to the incorrect classifications (the `EN-notEN` cases are often called the "*false positives*", and the notEN-EN cases, the "*false negatives*")

Evaluation measures

- Standard/Usual (not specific to NLP):
  - Accuracy
  - Precision, Recall (and F-score)
- Dedicated ones

Accuracy

$$\text{accuracy} = \frac{\text{number of correctly classified items}}{\text{total number of items}}$$

$$= \text{(normalized) trace of the confusion matrix}$$

Example (former English identifier)

$$\text{accuracy} = \frac{2+3}{2+1+1+3} = \frac{5}{7} \simeq 71\%$$

- Can be used with any number of classes
- Used for classification tasks where all class have the same importance
- Accuracy does not take the difference between classes into account:
  - asymmetry can result from classes of different importance (e.g. diagnostic)
  - or a class containing much more items than another

A task with asymmetric classes: information retrieval

IR seen as a binary classification task

- a document is *relevant* or *irrelevant* to a query

Example of asymmetry:

- Take a query to which 20 out of 100’000 documents are relevant
- The perfect classifier has the following accuracy

$$\frac{100'000}{100'000} = 100\%$$

- The uninteresting *all documents are irrelevant* classifier gets

$$\frac{99'980}{100'000} = 99.98\%$$

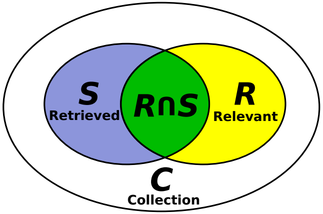
☞ For uneven classes, accuracy may not distinguish excellent from very poor systems

Two types of error for information retrieval and similar tasks

- False positives:
  documents retrieved that should not have been
- False negatives:
  document not retrieved that should have been

A specific confusion matrix:

		reference	
		relevant (R)	irrelevant
system	retrieved (S)	true positives	false positives
	not retrieved	false negatives	true negatives



Precision, Recall and F-score

Deal with unbalanced classes:

- Use two measures instead of one:
  Precision and Recall (to be defined in next slides)

F-score is a summary of the two measures

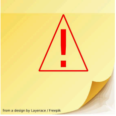
Precision

from a design by Layerace / Freepik

$$\text{precision} = \frac{\text{correctly retrieved documents}}{\text{total number of retrieved documents}} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

- Estimates the likelihood that a retrieved document is indeed relevant to the query
- Ignores false negatives. Take only false positives into account
- Ignores non-retrieved documents. Takes only retrieved documents into account
- Could be biased by retrieving very few documents

Recall (a.k.a. “true positive rate”)



$$\text{recall} = \frac{\text{correctly retrieved documents}}{\text{total number of relevant documents}} = \frac{\text{true positives}}{\text{true positives + false negatives}}$$

- Estimates (one minus) the probability to miss relevant documents
- Ignores false positives. Take only false negatives into account
- Ignores irrelevant documents. Takes only relevant documents into account
- Can be biased by retrieving all documents: gives a perfect score to the system that retrieves all documents

Precision & Recall: example

Spam filtering example:

	System	Reference
email0	OK	OK
email1	OK	Spam
email2	OK	OK
email3	Spam	OK
email4	OK	OK
email5	OK	OK
email6	OK	OK
email7	Spam	Spam
email8	OK	OK
email9	OK	OK
emailA	OK	Spam
emailB	Spam	Spam
emailC	OK	OK
emailD	OK	OK
emailE	OK	OK
emailF	Spam	Spam

Confusion matrix:

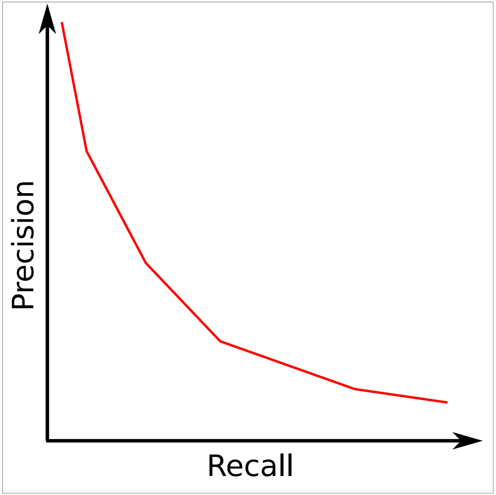
$P =$ $R =$

Note:

- accuracy =
- always-ok system: accuracy= ,
  $R =$,
  P

Precision vs Recall plots

For tasks where recall can be controlled (by controlling the amount of outputs), it’s often more informative to plot precision versus recall



☞ More in the “Information Retrieval” lecture

F-score

- Harmonic mean of precision and recall
- The harmonic mean penalizes large divergence between numbers, contrary to other means

$$\text{F-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

More generally (for given different emphasis to precision and recall):

$$F_\beta = (1 + \beta^2) \times \frac{\text{precision} \times \text{recall}}{(\beta^2 \times \text{precision}) + \text{recall}}$$

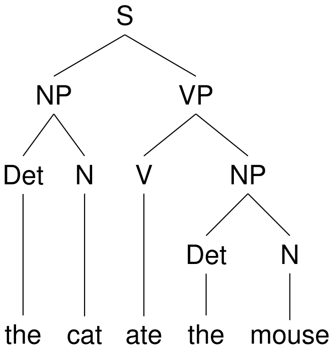
with $\beta \in \mathbb{R}$)
($\beta = 1$ in the first formula above,
$\beta = 0 \quad \longrightarrow$ precision
$\beta \to \infty \quad \longrightarrow$ recall)

Example of a non-classification task evaluated as binary classification: PARSEVAL

- A parser output is a syntactic tree
- But parsers are often evaluated as a binary classification task
- Items: constituents
- Classes: exists/does not exist
- Precision: nb of correctly annotated constituent/constituents in parser's output
- Recall: nb of correctly annotated constituent/constituents in gold standard
- Can be computed taking account of labels or not

Example of parser evaluation (1/4)

Consider the sentence "The cat ate the mouse" associated to the following reference parse tree (i.e. syntactic structure):



"*the cat*" is a constituent (label = `NP`);

"*cat ate the*" is not a constituent;

"*the cat ate*" is not a constituent.

A "*constituent*" is defined as any sequence of consecutive words in the sentence that corresponds to the footage (i.e. sequence of leaves) of a subtree in the parse tree associated to the sentence;

in addition, a constituent can be associated to a syntactic label
(the one corresponding to the root of the subtree associated with the constituent)

Example of parser evaluation (2/4)

A sentence of N words thus corresponds to $\frac{N(N+1)}{2}$ possible constituents (not necessarily distinct)

and any parse tree will select a subset of these $\frac{N(N+1)}{2}$ possible constituents.

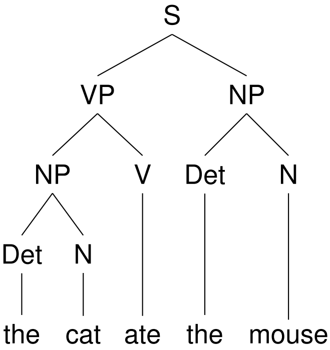
The constituents selected by the reference tree associated to the sentence in the reference can then be interpreted as the "*relevant*" ones with the whole set of possible constituents,

and the constituents selected by the tree associated to the sentence by the parser to evaluate as the "*retrieved*" ones

The Precision and Recall metrics can then be directly used to evaluate the parser

Example of parser evaluation (3/4)

For our former example, assume we have a parser that outputs:



Then we have (not taking into account syntactic labels):

Possible constituents	Reference constituents	System constituents
The	Rel	Ret
cat	Rel	Ret
ate	Rel	Ret
the	Rel	Ret
mouse	Rel	Ret
The cat	Rel	Ret
cat ate	notRel	notRet
ate the	notRel	notRet
the mouse	Rel	Ret
The cat ate	notRel	Ret
cat ate the	notRel	notRet
ate the mouse	Rel	notRet
The cat ate the	notRel	notRet
cat ate the mouse	notRel	notRet
The cat ate the mouse	Rel	Ret

Example of parser evaluation (4/4)

which corresponds to the following confusion matrix:

		reference	
		Ret	notRel
system	Rel	8	1
	notRet	1	5

and the following Precision and Recall scores:

$$P = \frac{8}{8+1} = \frac{8}{9} \simeq 89\%$$

$$R = \frac{8}{8+1} = \frac{8}{9} \simeq 89\%$$

Other NLP measures

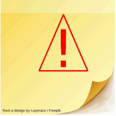
For some specific NLP tasks, ad-hoc measures have been defined:

- **BLEU** (bilingual evaluation understudy) measure:
  n-gram precision-like measure for machine translation
- **METEOR** (Metric for Evaluation of Translation with Explicit ORdering) measure:
  unigram F-score-like measure for machine translation
- **ROUGE** (Recall-Oriented Understudy for Gisting Evaluation) measures:
  n-gram recall-like measures for automated summarization

NLP evaluation protocol (reminder)

1. Define a control task
2. Produce a reference from a large amount of *typical* data (for the task)
3. Assess the quality of the reference
4. Evaluate NLP system(s) on the reference
5. Compare evaluations (statistical significance)
6. Publish and discuss results

Variability of the results



from a design by Layerace / Freepik

Whatever evaluation metric you use, measuring it only once on one single test set is **not** appropriate.

You shall estimate its variability (e.g. variance) as well!

☞ This means having several different test sets...

How to?

One common way is to use so-called “*cross-validation*”.

Cross-validation

- Idea: using several *test*/*learning* sets splittings to get a more accurate estimation of the results

  (Notice: not necessarily any *validation* set here, despite the name!)

- Repeat k times:
  - split the original data set into n subsets:
  - Repeat n times with a different test (sub)set each time:
    - use $n-1$ subsets for learning and 1 for testing
    - compute evaluation using the using the (left out) test set

- estimate variability of the results

☞ $k \times n$ cross-validation (e.g. 2×5, 1×10): run k times a (different) n-fold cross-validation

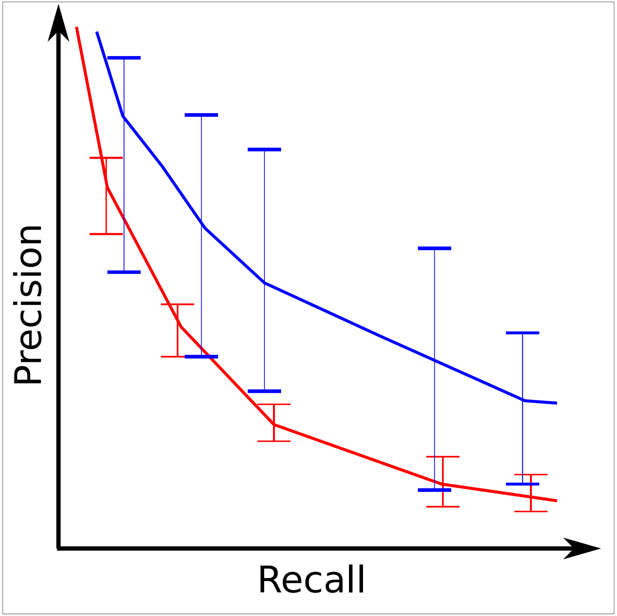
Note: why $k \times n$ rather than $1 \times (k\,n)$?

☞ increases variability; e.g. chance to have two given samples in the same subset is $\simeq k/n$ versus $\simeq 1/(k\,n)$.

("$\simeq 1/X$" is in fact $\frac{N-X}{N-1} \cdot \frac{1}{X}$ where N is the total size of the original corpus)

Statistically significant evaluation

- Having evaluations allow to compute standard deviations of results
- Which allows to compute confidence intervals or even *confidence boxes*



Comparing two systems in a statistically significant way

Simple example: (paired) Student's *t*-test: compare two classifiers on the *same* data of T test subsets
(assuming normal distribution and equal variance;
generalizations: Welch's *t*-test, ANOVA)

Δ_i: difference in performance between the two classifiers on test subset #i

empirical arithmetic mean: $\mu = \frac{1}{T}\sum_{i=1}^{T}\Delta_i$

empirical unbiased standard deviation: $s = \sqrt{\frac{1}{T-1}\sum_{i=1}^{T}(\Delta_i - \mu)^2}$

Then $t = \frac{\mu\sqrt{T}}{s}$ is compared to some threshold value for the desired confidence level.
For instance, at 95%, $|t|$ must be bigger than 1.645 (for $T \gg 1$)
To have a result statistically significant at more than 99%, $|t|$ must be bigger than 2.326

The impact of inter annotator agreement on maximal accuracy

- The best possible result is that of a human
- But diversity exist as long as the IAA is not perfect
- This diversity is not only made of mistakes but of subjectivity as well
- So it would not be realistic for a computer system to go closer to the gold standard than humans do

Evaluation campaigns

- Allow for objective comparison of systems
- Have given rise to a number of hand annotated corpora for specific tasks (e.g. Penn Treebank, many are distributed by the Linguistic Data Consortium (LDC, `http://www.ldc.upenn.edu/`) and the European Language Resources Association (ELRA, `http://www.elra.info/`))
- Evaluation campaigns : specific task, specific evaluation framework, specific time (e.g. conference workshops)
- Example: TREC (information retrieval), ParsEval, SensEval (word sense disambiguation)

Conclusions

- NLP systems need to be evaluated in order to be objectively compared
- Most NLP task can only be evaluated by being compared to solutions done by humans
- Humans do not always agree and some tasks are subjective
- Several measure exist that need to be computed and which significance need to be statistically measured
- To get clean results, test data should never be used in any way for development

References

[1] *Consequences of Variability in Classifier Performance Estimates*, by T. Raeder, T. R. Hoens and N. V. Chawla, in 10th IEEE International Conference on Data Mining (ICDM), pp. 421–430, 2010.

[2] *On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach*, by S. L. Salzberg, in. Data Mining and Knowledge Discovery, 1, pp. 317–327, 1997.