Evaluation of NLP

Martin Duneld

Department of Computer and Systems Sciences (DSV)

xmartin@dsv.su.se

Why evaluate?

- Otherwise we will not know if what we are developing is any good
- Human languages are very loosely defined
- This makes it very hard to **prove** that something is true (as with mathematics or logic), but we need to show that the system is working as intended/advertised
  - For most NLP systems it is fairly easy to come up with natural language input that the system cannot handle correctly
  - Solution: Test the program against many examples and show that the system handles a certain (acceptable) percentage of them

We can never find the whole population

- We can easily come up with completely new statements
  - ”Colorless green ideas sleep furiously”
    (Noam Chomsky, Syntactic Structures, 1957)

- In some languages we can also easily come up with completely new words
  - *Morphological derivation*
    - “Patient has high fever” ➔ “Patient is fevering”
  - *Compounding*
    - “Barnvagnshjulsekeruträtarlärlingsvikarieassistent”
    - “Perambulator wheel spoke straightener apprentice substitute assistant”

Aspects of evaluation

- General aspects
  - Measure progress
- Commercial aspects
  - Ensure customer satisfaction
  - Sales pitch (edge over competition)
- Scientific aspects
  - “Good Science”
  - Repeatability

What is Good Science?

- Induction
  - Evaluation on data that constitutes a **representative** sample of the total possible population of (target) data

- Falsification (Karl Popper)
  - For a hypothesis to be **falsifiable** it must be possible to make an observation or do an experiment that could prove the hypothesis false
  - Some researchers even mean that **no hypotheses can be verified**, they can only be falsified; all our current knowledge consists of not (yet) falsified hypotheses

Statistics is never a proof!

- Because it is so easy to come up with new forms that our system has never seen before, the results we get from testing on a set of examples are a **not** proof of anything or a measure of how “correct” our method is

- The results are just an indication of how well our method would perform on new, *unseen* data given that the examples we have tested on are **representative** of the full population

Approaches for evaluation

- Intrinsic evaluation
  - Measures the system isolated from how it will later be used
- Extrinsic evaluation
  - Measures the systems efficiency on and how acceptable the systems output is for a specific task
  - Usually requires some form of interaction from “users” (or at least humans)

Stages of development

Stockholm University

- Early stage
  - Intrinsic evaluation on component level
- Mid stage
  - Intrinsic evaluation on system level
- Late stage (close to deployment)
  - Extrinsic evaluation on system level

Manual evaluation

- Human assessors
  - Intrinsic/extrinsic

  \+ Semantic-based assessment

  – Subjective

  – Time consuming

  – Expensive

Semi-automatic evaluation

- Task-based evaluation
  - Extrinsic

  \+ Measures the system’s usability

  – Might entail subjective interpretation of questions and answers

- Keyword association
  - Intrinsic/extrinsic

  \+ No annotation needed

  – Shallow, opens up for qualified guesses

Automatic evaluation

- Sentence recall
  - Intrinsic
  - \+ Cheap and repeatable
  - – Does not differentiate between different but potentially equally good translations, summaries, etc.

- Vocabulary test (word recall)
  - Intrinsic
  - \+ Useful for phrase extraction (e.g. ”key phrase summaries”)
  - – Sensitive to differences in word order and negation (alternative, use *n*-gram recall/ROUGE scores)

Why automatic evaluation?

Stockholm University

- Manual labour is expensive and takes time
- It is practical to be able to evaluate often
  - Does tweaking this **variable** lead to better performance?
  - Variable can here be algorithmic settings, differences in input to algorithm, components in a pipeline etc.

- It is wearisome to evaluate large amounts of data manually
- The human factor
  - Humans tend to get tired and make mistakes

The human factor

- When we use human annotators/assessors it is good practice to present the examples (e.g. summaries, translations, sentences or words) in a **random** order
- The order should be different for each annotator/ assessor
- The task should also be divided into **reasonable sized** sessions

- This to lessen the effect of humans getting tired or bored and start getting sloppy when they perform a repetitive task

Corpora

- A corpus is a set of linguistic data that represents “reality” in a **balanced** and **purposeful** way
  - Sampling strategy

- Raw format vs. annotated data
  - Unprocessed text/speech/video
  - Added linguistic analysis

Ethics

- Informants
  - Must be informed about the data collection (before or after)
  - Must agree to that their data is used
  - Should be anonymous
    - But keep demographic data

- Data should be kept for 10 years
  - Makes the study repeatable/verifiable

Corpora can be...

- A data set of part-of-speech tagged text

Arrangör	nn.utr.sin.ind.nom
var	vb.prt.akt.kop
Järfälla	pm.gen
naturförening	nn.utr.sin.ind.nom
där	ha
Margareta	pm.nom
är	vb.prs.akt.kop
medlem	nn.utr.sin.ind.nom
.	mad

Corpora can be…

- A data set of parse trees

```
(S
  (NP-SBJ (NNP W.R.) (NNP Grace) )
    (VP (VBZ har)
      (NP
        (NP (CD tre) )
        (PP (IN av)
          (NP
            (NP (NNP Grace) (NNP Energis) )
            (CD sju) (NN styrelseposter) ) ) ) )
(. .) )
```

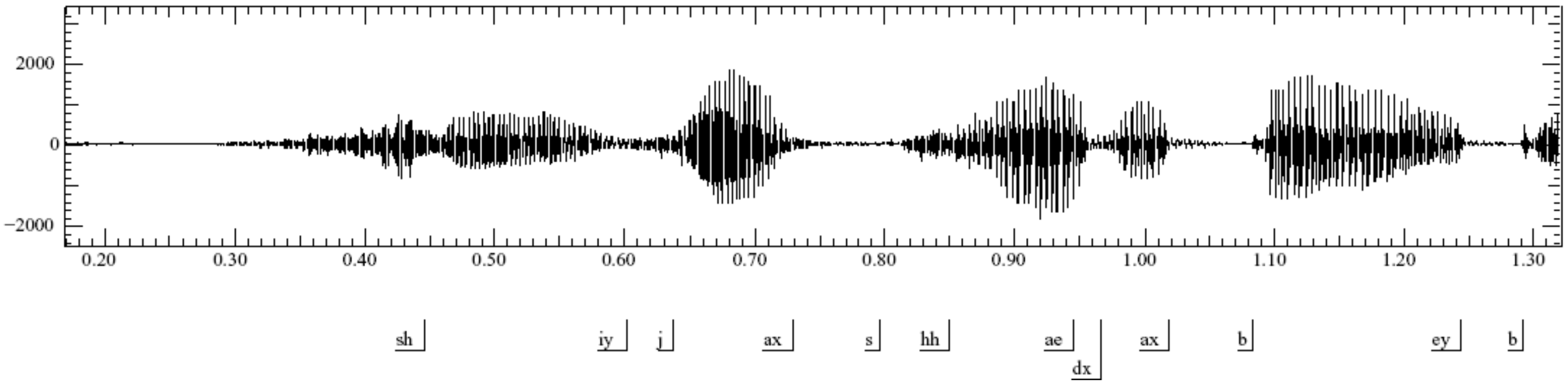
Corpora can be…

- A data set of RST trees (Rethorical Structure Theory)

```
(SATELLITE (SPAN|4||19|) (REL2PAR ELABORATION-
  ADDITIONAL)
(SATELLITE (SPAN|4||7|) (REL2PAR CIRCUMSTANCE)
(NUCLEUS (LEAF|4|) (REL2PAR CONTRAST)
(TEXT _!THE PACKAGE WAS TERMED EXCESSIVE BY
  THE BUSH |ADMINISTRATION,_!|))
(NUCLEUS (SPAN|5||7|) (REL2PAR CONTRAST)
(NUCLEUS (LEAF|5|) (REL2PAR SPAN)
(TEXT _!BUT IT ALSO PROVOKED A STRUGGLE WITH
  INFLUENTIAL CALIFORNIA LAWMAKERS_!))
```

Corpora can be...

- A data set of recorded speech



Well-established corpora

- Pros
  - \+ Well-defined origin and kontext
  - \+ (Often) Well-established evaluation schemes
  - \+ Possibility to compare systems on the same task and data

- Cons
  - – Optimisation on a specific data set
    - Over-fitting
  - – Can establish a “truth” that may not be true (e.g. archaic)

Gold standard

- ”Correct guesses” require that we know what the answer (i.e. correct output) should be
- This “optimal” (or simply desired) result is often called a **gold standard**

- What this gold standard looks like and how you calculate your results differs a lot depending on what the task is
- However, the basic idea is the same - a carefully checked data set that can be used as **ground truth**

Example of a gold standard

- Gold standard for part-of-speech tagging, shallow parsing and IOB parsing (”clause boundering”)

Han	*pn.utr.sin.def.sub*	NPB	*CLB*
är	*vb.prs.akt.kop*	VCB	*CLI*
mest	*ab.suv*	ADVPB|APMINB	*CLI*
road	*jj.pos.utr.sin.ind.nom*	APMINB|APMINI	*CLI*
av	*pp*	PPB	*CLI*
äldre	*jj.kom.utr/neu.sin/plu.ind/def.nom*	APMINB|NPB|PPI	*CLI*
sorter	*nn.utr.plu.ind.nom*	NPI|PPI	*CLI*
.	*mad*	0	*CLO*

Gold standard or gold standards?

- Sometimes several “answers” are (potentially) equally correct!
  - Machine translation
  - Automatic text summarisation

- If possible:
  - List all correct answers (e.g. all tags for ambiguous words)
  - Compare the system output to several correct answers
  - Translate data/task to a simpler – less detailed? – format (example, IOB parsing instead of shallow or full parsing)
  - Solve another problem that is easier to evaluate, and that is related to what we really want to evaluate (synonym tests in *TOEFL*)
  - Evaluate manually!

Common evaluation metrics

- **Precision** = correct guesses / all guesses
- **Recall** = correct guesses / correct answers

- Precision and recall are often mutually dependent
  - Higher recall ➔ lower precision
  - Higher precision ➔ lower recall

- F-score: combines precision and recall into one metric
  - $F_1 = 2*(P*R/(P+R))$

More evaluation terminology

- **True positive**
  - Alarm given at the correct point in the output
- **False negative**
  - No alarm given when one should have been
- **False positive**
  - Alarm given when no alarm should have been given
- **(True negative)**
  - The system is silent on uninteresting data

- Example: For *spelling correction* the above would correspond to detected errors, missed errors, false alarms and correct words without warning

How good is 95%?

- It depends on the problem you are trying to solve!

- Try to determine an expected lower and upper bound for performance (on a specific task)

- A **baseline** shows the performance of a naïve approach (that is, an expected *lower* bound)
  - If we can't beat the baseline it's back to the drawing board

Lower bound

- Baseline
  - Serves as a lower bound for what is acceptable
  - Common to have more than one baseline

- Common baselines
  - Random selection/assignment
  - The most common answer (e.g. the majority class when tagging)
  - Linear selection (e.g. for text summarisation)

- If the system/method being evaluated is fairly advanced the baseline should not be too naïve
  - Use an earlier system/method as an alternative baseline

Upper bound

- Sometimes the upper bound for expected performance is lower than 100%

- Example 1:

  Analysing a sample from a corpus shows that 3% of all answers in an evaluation corpus are incorrect (and randomly distributed)

  – Impossible to learn where random errors occur

Upper bound II

- Example 2:

  In 10% of the cases experts cannot agree on what the correct answer should be

  - Inter-annotator / Inter-assessor agreement (IAA)
  - Low IAA can sometimes be combated by comparison to several sources/answers
  - In other cases we need a more well-defined and precise annotation/assessment task, or that the annotators/assessors discuss and reach a consensus

Is 95.3% better than 94.8%?

- It depends, have you tested on 212 examples or 10 million examples?

- A statistical **significance test** shows us to what degree chance would give us the current difference between the methods *even if they perform comparably well*

- If you evaluate many methods (or the same method repeatedly) on the same data, you need to take this into account
  - Split the data set into train/tune/test subsets

Example of a significance test

- We evaluate a search engine **with and without** the use of stemming
- We have marked 100 documents as either relevant or irrelevant to the test query, and found 30 to be relevant

- *Without stemming* we find **18** of the relevant documents, *with stemming* we find **24** (**9** documents not found before, but miss **3** found without stemming)
  - Does this mean that IR *with stemming* is better?

- **McNemar's Test:** The **null hypothesis** is that the search engine performs equally well with and without stemming (i.e., there is **no difference** between the methods)

McNemar's test I

- Without stemming: 18 out of 30 relevant documents found
- With stemming: another 9 found, but misses 3 relevant documents found without stemming

	Stemming OK	Stemming FAIL
Inflected words OK	**A: 15** (18-3)	**B: 3**
Inflected words FAIL	**C: 9**	**D: 3** (30-18-9)

- We are interested in **B** and **C**. If B+C is large, calculate $X^2 = ((B-C)^2)/(B+C)$ and look up the Chi-square distribution
- In this case we get $X^2 = 2.0833$, $p(\alpha) = 0.1489$
  - **Not** significant
  - Commonly $p<0.05$ indicates statistical significance

McNemar's test II

We test the search engine on a larger data set and find

- Without stemming: 180 out of 300 relevanta documents
- With stemming: 240 (another 90, but misses 30 that were found without stemming)

	Stemming OK	Stemming FAIL
Inflected words OK		B: 30
Inflected words FAIL	C: 90	

- Now we get X^2 = 29.0083, p(α) = <0.0001
  - **Significant!**

Train/Tune/Test splits

- When developing machine learning models we often split our annotated data into subsets, or slices

- These go by many names, but are often three
  - Training/Tuning/Evaluation
  - Training/Validation/Testing
  - etc.

- Common sizes are 60% of the data for training and 20% for tuning/validating settings for different parameters
- The last 20% is set aside for the very last run and is used only once; for estimating the performance on previously unseen data
  - This slice is sometimes also referred to as **holdout** data

Limited amount of annotated data

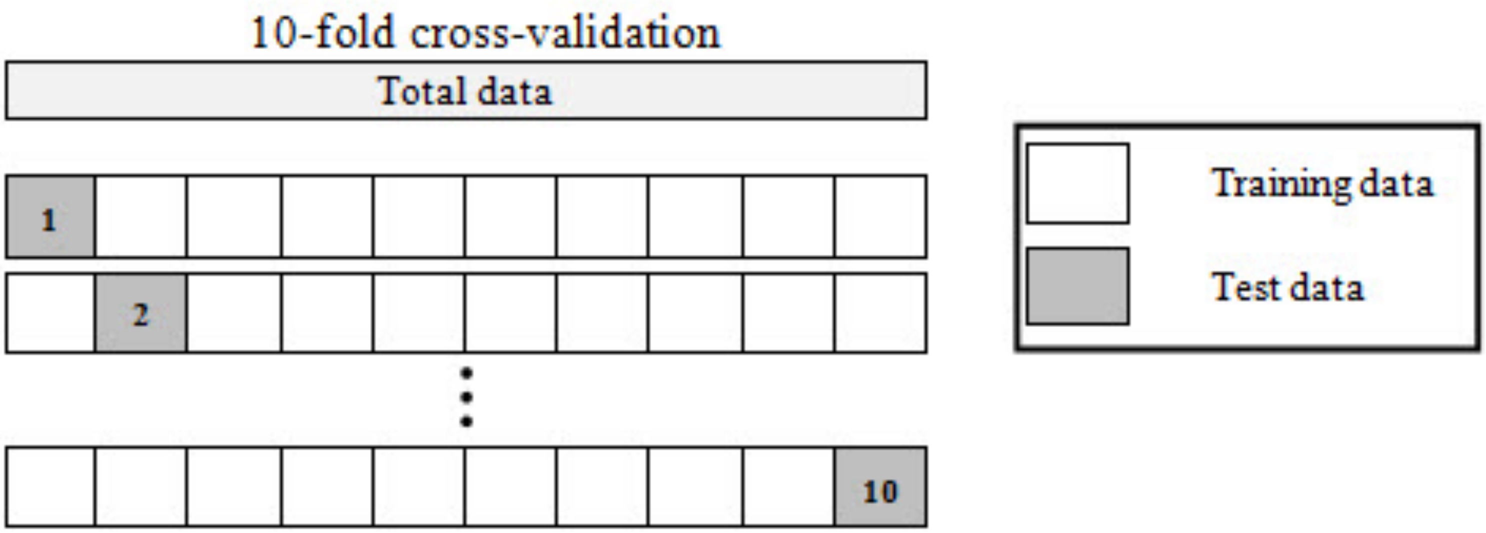
- Limited access to annotated data is often a problem, especially when it comes to machine learning

- We want much data for **training**
  - Better results
- We want much data for **evaluation**
  - More reliable results

- If possible, create your own (synthetic) data!
  - Missplel (Ericson, 2003)

K-fold Cross-Validation I
Example, *k*=10

1. Split the data set into 10 equally sized subsets
2. Set aside 10% data for evaluation, train on 90%
3. Set aside next subset, train on the other 9
4. … and again, in total 10 times



K-fold Cross-Validation II

- Calculate the mean of the 10 (*k*) evaluation runs and report as the result

- Variants:
  - Stratified *k*-fold cross-validation
  - Leave-*p*-out cross-validation

- For extra validity, you can still set aside holdout data that is not used in the cross-validation
  - The cross-validation is in that case used only for training and tuning

Concrete examples I

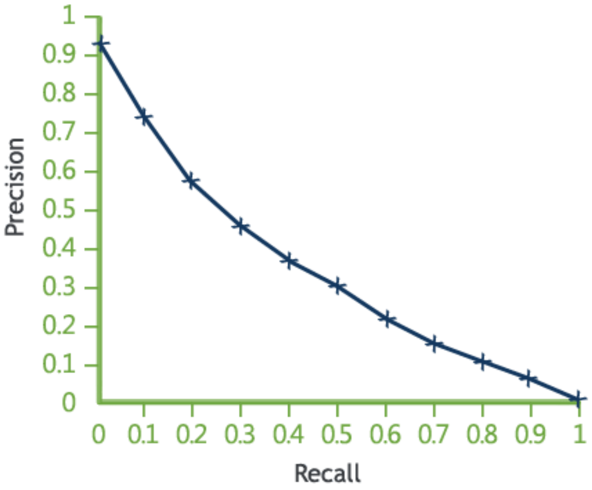
- Tagging
  - Force the tagger to assign precisely one tag per token (e.g. words) – calculate the precision
- Parsing: what happens when the parser is almost correct?
  - Cross-brackets: [A [B C]] instead of [[A B] C]
  - Partial trees (some parsers fail here)
  - How many sentences got full parse trees?
- Spell checking
  - Recall and precision for alarms
  - How far down the list of suggestions is the correct answer?

Concrete examples II

- Grammar checking
  - How many false alarms (precision)?
  - How many errors are detected (recall)?
  - How many of these get a correct diagnosis?
- Text summarisation
  - How many *n*-grams overlap with the gold standard?
  - ROUGE scores
- Machine translation
  - How many *n*-grams overlap with the gold standard?
  - BLEU scores

Concrete examples III

- Synonyms
  - How many questions on the TOEFL test can the system answer correctly?
- Information retrieval
  - What is the precision at *x* number of hits, or at *x*% recall? Mean precision from different intervalls
  - Precision/recall graphs



Concrete examples IV

- Text categorisation
  - How many documents were assigned the correct category?
- Clustering
  - How clean are the clusters?
  - Entropy, similarity etc.
  - **Important!** Clustering should *always* also be evaluated on a specific task (i.e. task-based evaluation)

Statistics is not everything!

- So far we have mostly looked at how to calculate different metrics and how to interpret these

- However, statistics is never a substitute for actually looking at our system's output and compare it qualitatively to the reference standard (the gold standard)
  - Error analysis!

- Quantitative and qualitative evaluation tell us different things, and complement each other
  - Statistics shows us **tendencies** over large amounts of data
  - Qualitative analysis gives us **detailed knowledge**, but is often carried out on a randomly selected small subset of the same data

Conferences and campaigns

- TREC – Text REtrieval Conferences
  - Information Retrieval/Extraction and TDT
- CLEF – Cross-Language Evaluation Forum
  - Information Retrieval on texts in European languages
- DUC – Document Understanding Conference
  - Automatic Text Summarisation
- SENSEVAL
  - Word Sense Disambiguation
- ATIS – Air Travel Information System
  - DARPA Spoken Language Systems

… mfl.

Questions?