Annotation: quality testing and automation

LIN 350
Words in a Haystack
Katrin Erk

Quality testing and automation

- How good is a given annotation?
  - Is it correct?
  - Is it consistent?
  - How can you check this for thousands of sentences?
  - The annotation manual may easily be 50 or 100 pages long
- Annotation takes a lot of time
  - SALSA: 20,000 sentences, about 4 years
  - Is there any way we can speed this up?

Overview

- Annotation quality testing:
  - Inter-annotator agreement
  - Intra-annotator agreement
  - Automatic quality testing
  - The kappa measure
- Semi-automatic annotation:
  - Automatic pre-annotation
  - Automatic selection of items to annotate (Active Learning)

Annotation quality testing: the problem

- No annotation is error-free
- Problem of annotation consistency:
  - Same phenomenon annotated the same way today and 6 months ago?
  - Change in annotation guidelines must lead to changes in old annotations
- Are the guidelines clear enough? Will all annotators understand them the same way?
- Simple oversight

Inter-annotator agreement

- Two (or more) annotators annotate the same text
- How often do their analyses agree?
- Time-consuming, since time will be spent re-annotating the same text rather than annotating new text

Inter-annotator agreement

- Salsa:
  - Each lemma is annotated independently by two annotators
  - Adjudication:
    - A third person looks at points where the two annotators disagree
    - Adjudicator chooses one of the two analyses -- or substitutes a totally different one
  - Meta-adjudication:
    - Two adjudicators instead of one
    - Look at disagreements between adjudicators

Intra-annotator agreement

- How consistent is a single annotator?
- Re-annotate a text you have annotated a few months ago, assess disagreement with yourself

Automatically detecting annotation errors

- Turn annotation guidelines into rules
  - WSJ POS tagging manual: “Hyphenated nominal modifiers… should always be tagged as adjectives”
  - POS tags for closed classes: No word that doesn’t belong to the (finite) class may have the tag
- Dickinson and Meurers 2003: same context, different tag: potential error

Automatically detecting annotation errors

- Dickinson and Meurers 2003: error checking for POS tagging
  - variation n-gram: same context words, but one word (variation nucleus) with different tag

    “to ward off a hostile takeover attempt by two European shipping concerns”
  - Long n-gram: probably an error (threshold: n=6)
  - Variation at fringe of n-gram: probably not an error
- Later generalized to syntactic analysis
- In general, not much work on automatic error checking for annotation

How to measure agreement between annotators?

- Simplest measure: percentage of agreement
- But what does it mean? How good is 50% agreement?
  - Just 2 choices, e.g. distinguishing between “celestial body” and “well-known person” sense of “star”: 50% is very bad.
  - 40 choices, e.g. word senses of a high-frequency verb like “go”: 50% not great, but not abysmal either.

Chance agreement

- Imagine two annotators are assigning random tags
- Two tags, both chosen equally often: Annotators will agree 50% of the time

$$P_{agree} = P_{A1}(tag_1) \cdot P_{A2}(tag_1) + P_{A1}(tag_2) \cdot P_{A2}(tag_2) = \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2}$$

- Two tags, one chosen 95% of the time:

$$P_{agree} = P_{A1}(tag_1) \cdot P_{A2}(tag_1) + P_{A1}(tag_2) \cdot P_{A2}(tag_2) = 0.95 \cdot 0.95 + 0.05 \cdot 0.05 = 0.905$$

Estimating chance agreement

- Two annotators, Ann and Bob, N labels
- Probability that
  - Ann chose label 1 AND Bob chose label 1 OR
  - Ann chose label 2 AND Bob chose label 2 OR
  - …
  - Ann chose label N AND Bob chose label N
- For independent probabilities, AND is multiplication, OR is addition

The kappa measure: Correcting for chance agreement

- J. Carletta 1996, Computational Linguistics 22(2)
- Measure from content analysis

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

- P(A): measured agreement
- P(E): estimated chance agreement
- Standard measure today

What is a good kappa value?

- Krippendorff 1980:
  - kappa < 0.67: discard
  - kappa between 0.67 and 0.8 allows tentative conclusions
  - kappa of 0.8 or greater allows definite conclusions
- Also depends on the task

Problems with kappa

- Skew through uneven classes
  - Suppose you have 2 labels, "discourse marker" and "no discourse marker".
  - Label "no discourse marker" will be much more likely
  - So, high chance agreement
  - This penalizes each disagreement btw. annotators more and lowers kappa

Problems with kappa

- Kappa assumes that each item will get one label.
- But what if only some items get labels?
  - Semantic role assignment: not every syntactic constituent bears a role
  - Discourse analysis: not every syntactic constituent is discourse marker or "argument"

Problems with kappa

- Kappa assumes that each item will get one label.
- But what if items can have more than one label?
  - Vagueness and ambiguity in word sense assignment
- Can we measure partial agreement?

Other approaches to ascertaining annotation quality

- OntoNotes: the 90% solution
  - Task: word sense annotation
  - Idea:
    - measure inter-annotator agreement
    - if it is below 90%, re-define the sense labels, then re-annotate
    - repeat if necessary
- What does that mean for the word sense labels they're assigning?

Other approaches to ascertaining annotation quality

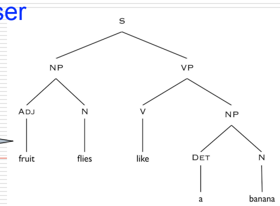
- Annotation as a psycholinguistic experiment
  - Have many people do the same task, at least 20 annotators per item
  - View disagreement between annotators as a graded label, e.g. 60% of annotators assigned label A, 40% assigned label B, then the label is a mixed label, 60%A, 40% B
- But is this valid? What if it's just the annotation manual that is bad and leads to disagreements?

Overview

- Annotation quality testing:
  - Inter-annotator agreement
  - Intra-annotator agreement
  - Automatic quality testing
  - The kappa measure
- Semi-automatic annotation:
  - Automatic pre-annotation
  - Automatic selection of items to annotate (Active Learning)

Automatic annotation

- For word sense annotation: Word Sense Disambiguation system
  - input: a word in context
    - for example "The astronomer married the star"
  - output: a sense label for the *target* word
    - for example: well-known person
- For syntactic annotation: parser
  - input: a sentence
    - for example "Fruit flies like a banana"
  - output: a syntactic tree
    - for example:



Automatic annotation: how can it work?

- Many systems use machine learning:
  - software that learns from examples
  - It looks at some previously annotated samples: training data
  - Then it applies what it has learned to new cases
- "Learning":
  - generalizing over seen training items
  - so the system can treat new cases "the same way" as similar training items
- What does "similar" mean?
  - many ways of defining similarity
  - needed: some sort of formal representation of training and test items

Automatic pre-annotation

- Aim: speeding up annotation
- Problem: automatic annotation more error-prone than manual annotation
- Solution:
  - Data automatically annotated
  - Human annotator checks automatic annotation and corrects errors

Automatic pre-annotation

- POS-tagging:
  Torsten Brants 2000: One human post-editor reduces error rate from 3.3% to 1.2% (German corpus)
- Syntactic annotation in TIGER:
  - Interactive semi-automatic annotation
  - System proposes one constituent
  - Human confirms or corrects
  - System proposes next constituent

Active learning

- Software and human annotator annotate together
- Software figures out the item it is most uncertain about
- Those items it gives to the human to annotate
- The others it does automatically
- Also, it continually learns from what the human annotator does
- Master and Apprentice setting:
  - apprentice (software) does easy tasks it can already do
  - for more complicated tasks, it asks the master (the human)
  - from observing the master, it learns to solve the more difficult cases too

Active learning

- Aim: reduce the amount of data that a human annotator has to label
- Use machine learning
- The more training data a machine learning system has, the better it works
- But often less and well-chosen training data is better than more random training data

Active learning: confidence

- For active learning, machine learning software needs to assess its confidence in labeling a test item:
  - An item from the training set:
    we can be certain (high probability) that we know the correct label
  - An item that is very similar to an item from the training set:
    we can guess that it has the same label as the training item (lower probability)
  - An item that is different from all items from the training set:
    we are uncertain about its label (low probability)

Summary

- Annotation quality checking:
  - Duplicate annotation:
    inter-annotator, intra-annotator agreement
  - Automatic error checking
- Measuring agreement between annotators:
  - Kappa (correcting for chance agreement)
- Automating annotation:
  - Semi-automatic annotation (person checks for errors)
  - Active learning (only selected examples manually annotated, selection done by system)