Detecting Errors in Corpus Annotation

Detmar Meurers
University of Tübingen

LingLunch, UFR de Linguistique
Université Paris Diderot, Paris 7
March 10, 2010

Introduction

Corpora with "gold standard" annotation are used

- as training and testing material for NLP algorithms/tools
- for searching for linguistically relevant patterns

Such annotation generally results from a semi-automatic markup process, which can include errors through

- automatic processes
- human annotation or post-editing

Effects of Annotation Errors

- Less reliable training of NLP technology
  - van Halteren et al. (2001): a tagger trained on WSJ (Marcus et al. 1993) performs significantly worse than one trained on LOB (Johansson 1986)
- Less reliable evaluation of NLP technology
  - van Halteren (2000): 13.6%–20.5% of cases where WPDV tagger disagrees with BNC-sampler annotation, cause is error in BNC-sampler (0.3% error, Leech 1997). Error rates for other corpora much higher.
  - Padro & Marquez (1998): because of errors in the testing data, cannot tell which of two taggers is better
- Low precision and recall of queries for already rare linguistic phenomena
  - Meurers (2005): low precision of queries for verbal complex patterns since certain finite and non-finite verb forms are not reliably distinguished by German taggers

How to obtain high quality annotation

- Annotate corpus independently several times, then test interannotator agreement (Artstein & Poesio 2009)
  - Interannotator agreement: Can the distinctions made in the annotation scheme can be applied consistently based on the information available in the corpus?
- Define adequate annotation scheme, with explicit documentation and a list of problematic cases to achieve maximal agreement (Voutilainen & Järvinen 1995; Sampson & Babarczy 2003).
  - keep only distinctions which can be reliably and consistently identified and annotated uniquely
  - appendix of difficult cases and how to resolve them crucial

Our research questions

- How about automatic methods for error detection?
  - Detection can feed into repair as second stage of correction (cf. also Oliva 2001; Blaheta 2002).
- What can be done for annotation of language in general?

⇒ Detection of annotation errors through automatic analysis of comparable data recurring in the corpus
  - DECCA project (http://decca.osu.edu; Dickinson 2005)

- Detect errors in common "gold standard" corpora:
  - part-of-speech annotation (Dickinson & Meurers 2003a)
  - syntactic annotation (Dickinson & Meurers 2003b; Boyd, Dickinson & Meurers 2007)
  - discontinuous syntactic annotation (Dickinson & Meurers 2004)
  - dependency annotation (Boyd, Dickinson & Meurers 2008)

  including spoken language corpora (Dickinson & Meurers 2005a).

Variation Detection for POS Annotation
(Dickinson & Meurers 2003a)

- POS tagging reduces the set of lexically possible tags to the correct tag for a specific corpus occurrence.
  - A word occurring multiple times in a corpus can occur with more than one annotation.
- Variation: material occurs multiple times in corpus with different annotations
- Variation can result from
  - genuine ambiguity
  - inconsistent, erroneous tagging
- How can one find such variation and decide whether it's an ambiguity or error?

Classifying variation

- The key to classifying variation lies in the context:
  - The more similar the context of the occurrences, the more likely the variation is an error.
- A simple way of making "similarity of context" concrete is to say it consists of
  - words
  - which immediately surround the variation, and
  - require identity of contexts.

⇒ Extract all *n*-grams containing a token that is annotated differently in another occurrence of the *n*-gram in corpus.
  - variation nucleus: recurring unit with different annotation
  - variation *n*-gram: variation nucleus with identical context

Computing variation *n*-grams

- Example from WSJ: Variation 12-gram with *off*

  (1) *to ward off a hostile takeover attempt by two European shipping concerns*

  - once annotated as a preposition (IN), and
  - once as a particle (RP).
- Note: Such a 12-gram contains two variation 11-grams:

  (2) *to ward off a hostile takeover attempt by two Eur. shipping*
  ward off a hostile takeover attempt by two Eur. shipping concerns

→ Calculate variation *n*-grams based on variation *n*–1-grams to obtain an algorithm efficient enough for large corpora.
  - Essentially an instance of the a priori algorithm (Agrawal & Srikant 1994).

Computing variation *n*-grams

Algorithm

1. Calculate the set of variation unigrams in the corpus and store them.
2. Extend the *n*-grams by one word to either side. For each resulting $(n+1)$-gram
   - check whether it has another instance in the corpus and
   - store it in case there is a variation in the way the occurrences are tagged.
3. Repeat step 2 until we reach an *n* for which no variation *n*-grams are in corpus.

Running this algorithm on the Penn Treebank 3 version of the WSJ, retrieves variation *n*-grams up to length 224.

Computing variation *n*-grams

Example: WSJ in Penn Treebank 3

- general corpus information:
  - 1,289,201 tokens
  - 51,457 types
  - 23,497 of types appear only once (= 1.8% of tokens)
  - 98.2% of tokens appear more than once
- variation nuclei:
  - 7,033 types
  - 711,994 tokens = 55.2% of all corpus tokens
- variation *n*-grams:
  - longest: 224
  - 2,495 distinct variation nuclei for $6 \leq n \leq 224$
  - 16,319 distinct variation nuclei for $3 \leq n \leq 224$
    - each corpus position counting only in longest *n*-gram

Heuristics for classifying variation

I. The length of the context

Idea: The longer the *n*-gram, the more likely the variation is an error.

Example: In a variation 184-gram, the nucleus lending varies between adjective (JJ) and common noun (NN).

$\overleftarrow{\text{109 identical words}}$ $\underset{\text{JJ/NN}}{\textit{lending}}$ $\overrightarrow{\text{74 identical words}}$

Here, NN is the correct annotation of this *n*-gram.

Note: Heuristics independent of corpus, tagset, or language.

Heuristics for classifying variation

II. Distrust the fringe

Idea: Morphological and syntactic properties are governed locally. The further the variation nucleus is away from the edge of the *n*-gram, the more likely it is an error.

Example: A variation 37-gram with the nucleus *joined* occurring as first word:

(3) a. *John P. Karalis . . .*
b. *John P. Karalis has . . .*

joined the Phoenix , Ariz. , law firm of Brown & Bain . Mr. Karalis , 51 , will specialize in corporate law and international law at the 110-lawyer firm . Before joining Apple in 1986 ,

The context preceding the 37-gram shows:

- In a. the verb must be tagged as past tense (VBD),
- in b. as past participle (VBN).

Why does the non-fringe heuristic work?

- Non-fringe heuristic: one element of recurring context around a recurring nucleus is generally sufficient to determine that a variation in an annotation is erroneous.
- Is this an artifact of the WSJ annotation or is there independent motivation for such a general heuristic?
- Interestingly, recent research on language acquisition by Toby Mintz (USC) has addressed a related question:
  - How do humans discover and learn categories of words?

  His results show that humans seem to make use of such non-fringe patterns (*frames*) to learn categories!

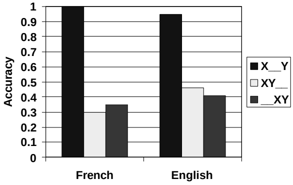
Independent evidence from language acquisition

- Mintz (2002) shows that lexical co-occurrence information of an element surrounded by a frame (i.e., X __ Y) leads to categorization in adults.
- Mintz (2003): frequent frames supply robust category information, consistent across child language corpora.
- Example for a frame from CHILDES (MacWhinney 2000):
  - you put it
  - you want it
  - you see it

  → you ____ it
- Cross-linguistic viability of frame concept confirmed for French (Chemla et al. 2009) and Mandarin (Xiao et al. 2006).

Independent evidence from language acquisition

- Chemla et al. (2009) show that humans categorize words most reliably when surrounded by a frame. The other same size contexts are much worse:



⇒ The non-fringe heuristic used for annotation error detection relies on the basic human cognitive abilities that led to the linguistic categories in the first place.

Results for the WSJ

- Of the 2,495 distinct variation nuclei (types) $6 \leq n \leq 224$:
  - 2,436 are errors (97.64%)
    - Correcting the instances of these variation nuclei by hand yielded 4417 token corrections.
  - 59 are genuine ambiguities
    - 32 were 6-grams, 10 were 7-grams, 4 were 8-grams, . . .
      → relevance of heuristic to prefer long context
    - 57 appeared first/last
      → relevance of heuristic to distrust the fringe
    - 31 are the *first word* of the *n*-gram, varying between two specific tags: past tense verb (VBD) and past participle (VBN).
- Of 7141 distinct non-fringe variation *n*-gram types $3 \leq n \leq 224$, based on sampling we found that
  - 6626 are errors (92.8%) → each at least one correction
  - given 3% estimated POS error rate in the WSJ, the method has a POS error recall of at least 17%

Feedback for revising annotation scheme

For 140 of the 2436 erroneous variation nuclei, the variation was clearly incorrect, but which tag is the correct one is unclear from the guidelines (Santorini 1990).

Example: *Salomon Brothers Inc*

Brothers is tagged

- 27 times as proper noun (NNP)
- 22 as plural proper noun (NNPS).

⇒ Variation *n*-gram error detection helps identify error-prone distinctions, which need to be documented more explicitly or possibly eliminated, e.g.:

- proper vs. common nouns
- certain types of noun-adjective homographs

Related work on POS error detection

- Work with another focus, which could be combined with our consistency-checking approach:
  - Deriving and searching for bigrams of tags which should never be allowed (Květŏn & Oliva 2002). → Inconsistencies are mostly possible bigrams.
  - Sparse Markov transducers used to detect anomalies, i.e., rare local tag patterns (Eskin 2000). → Inconsistencies are mostly recurrent, not rare.
  - Using parsing failures to detect ill-formed annotation serving as parser input (Hirakawa et al. 2000; Müller & Ule 2002). → Language specific resources.
  - Searching and correcting with hand-written rules (Oliva 2001; Blaheta 2002)
- Related to consistency of annotation:
  - Comparing tagger output with gold standard (van Halteren 2000; Abney et al. 1999). Taggers detect consistent behavior in order to replicate it.

Summary for POS error detection

- We discussed a detection methods for POS annotation errors in gold-standard corpora:
  - detect variation within comparable contexts
  - classify such variation as error or ambiguity using general heuristics
- Idea relies on multiple corpus occurrences of a particular word with different annotations
  → particularly useful for hand-corrected, gold-standard corpora
- Evaluation showed the method detects errors in the WSJ with
  - 92.8% precision
  - 17% estimated recall
- Qualitative inspection of the detected variation can provide valuable feedback for annotation scheme (re)design and documentation.

Variation Detection for Syntactic Annotation

(Dickinson & Meurers 2003b, 2004; Boyd, Dickinson & Meurers 2007)

- Let's try to apply variation detection to the syntactic annotation in treebanks!
  - How can two syntactically annotated sentences be compared for this?
- Variation detection is closely related to interannotator agreement testing for multiply annotated corpus.
  - How are multiple annotations of the same sentences compared for testing interannotator agreement?
  - Calder (1997) and Brants & Skut (1998) present algorithm for detecting differences in annotation.
  - algorithm is annotation-driven, asymmetric, and sentence-based

⇒ We are looking for a data-driven, symmetric, string-based approach.

Defining variation nuclei for syntactic annotation

How can we obtain a data-driven definition of a variation nucleus as the unit of data on which the comparison of syntactic annotation can be based?

Problem: No one-to-one mapping between word and label, as with part of speech.

Idea: Decompose variation nucleus detection into series of runs for all relevant string lengths, more specifically

- define one-to-one mapping between string of a given length and the label for that string
- perform runs for strings from length 1 to longest constituent in corpus

Defining variation nuclei for syntactic annotation

How to compare annotation for syntactic variation nuclei

- Only compare categories assigned to the entire nucleus.
- This intentionally ignores the internal structure, which is taken into account when shorter strings are checked.
- To obtain uniform mapping from strings to labels assign special label NIL to non-constituent occurrences of a string.

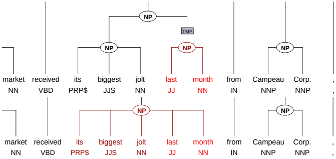
Examples from the WSJ corpus

- Variation between two syntactic category labels:

(4) *maturity* *next Tuesday*

labeled as **NP** twice
PP once

- Variation between constituent and non-constituent:



Computing variation *n*-grams for a treebank

Algorithm

For each constituent length i ($1 \leq i \leq$ |longest-constituent|):

1. Compute the set of nuclei:
   a) Find all constituents of length i: store them with their category label
   b) For each type of string stored as constituent of length i, add NIL for each non-constituent occurrence
2. Compute variation nuclei set as:
   - all nuclei from step 1 with more than one label
3. Generate variation *n*-grams for these variation nuclei, just as defined for part of speech annotation

Results for the WSJ (Penn Treebank 3)

- Total: 6277 distinct, non-fringe variation nuclei
  - distinct: each corpus position is only taken into account for longest variation *n*-gram it occurs in
  - non-fringe: nucleus is surrounded by at least one word of identical context
- We inspecting 100 randomly sampled examples:
  - 71% errors, with 95% confidence interval for point estimate of .71 being (.6211, .7989)

  → between 3898 and 5014 erroneous variation nuclei, each corresponding to at least one token error
- What are the reasons for the misclassified ambiguities?

Misclassified Ambiguities I: Null elements

- 10 of the 29 ambiguous nuclei in sample are null elements varying between two different categories.
- WSJ annotators inserted markers for arguments and adjuncts realized non-locally, or unstated units of measurement (cf. Bies et al. 1995, p. 59).
- **Example:** **EXP** (expletive) annotated as S or SBAR

(5) *. . . **it** [$_S$ ***EXP***] **may be** a wise business investment * [$_S$ to help * keep those jobs and sales taxes within city limits]* .

(6) *. . . **it** [$_{SBAR}$ ***EXP***] **may be** impossible [$_{SBAR}$ for the broker to carry out the order] because . . .*

→ Ambiguity arises where null items occur in place of element non-locally realized.

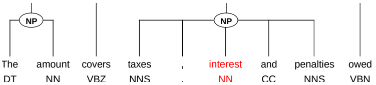
⇒ Eliminating null elements from variation nuclei set raises precision from 71% to 78.9%.

Misclassified Ambiguities II: Coordination

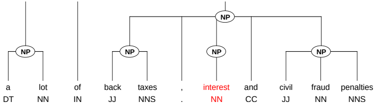
- 6 of the 29 ambiguities deal with coordinate structures.
- Annotation scheme distinguishes simple (i.e., non-modified) and complex coordinate elements.
  - Even if an element is simple, it is annotated like a complex element when conjoined with one.

Coordinate structure example

interest in a flat coordinate structure:



interest in a complex coordinate structure:



⇒ Annotation scheme makes a distinction externally motivated

Related work on syntactic error detection

- CCGbank (Hockenmaier & Steedman 2005): derived from Penn Treebank, fixing some errors:
  - e.g.: “Under ADVP, if the adverb has only one child, and it is tagged as NNP, change it to RB.”
- Blaheta (2002): discusses types of errors and some rules to identify them
  - e.g.: “If an IN is occurring somewhere other than under a PP, it is likely to be a mistag.”
- Ule & Simov (2004) search for unexpected rules, using information about a node and its mother
  - Discrepancies between mother and daughter annotation can point to errors

Summary for constituency error detection

- We showed how one can extend the POS-error detection approach to syntactic annotation.
- Illustrated with a case study based on WSJ treebank that the method is successful (71% precision) in detecting inconsistencies in syntactic category annotation.
- Approach supports two aspects of treebank improvement:
  - makes it possible to find and correct erroneous variation in corpus annotation
  - provides feedback for development of empirically adequate standards for syntactic annotation, identifying distinctions difficult to maintain over entire corpus

Increasing recall

(Boyd, Dickinson & Meurers 2007)

The variation *n*-gram method for detecting annotation errors

- Finds recurring data and compares analyses in different corpus instances
- Uses shared context as a heuristic to determine when analyses should be annotated identically

Two ways to increase recall:

- Redefine *variation nuclei*, to extend the set of what counts as recurring data for which annotation is compared.
- Redefine *context* and *heuristics*, to obtain more variation n-grams predicted to be errors.

Using part-of-speech nuclei to increase recall

- Redefine *variation nuclei*: POS instead of words
- Example (WSJ corpus, PennTreebank3 tagset, 45 tags):

(7) a. *Boeing on Friday said 0 it received [$_{NP}$ an/DT order/NN] *ICH* from Martinair Holl*

b. *it received [$_{NP}$ a/DT contract/NN *ICH*] from Timken Co.*

⇒ 59% increase in recall, while maintaining reasonable precision of 68.69% (using annotation-based heuristics)

Limitations of POS nuclei

- Generalizing from word to POS nuclei is not always successful; e.g., POS class not fine grained enough.
- Example: variation trigram “*remains JJ for*”

(8) a. *a virus that *T* [$_{VP}$ remains [$_{ADJP}$ active/JJ] [$_{PP}$ for a few days]]*

b. *remains [$_{ADJP}$ responsible/JJ for the individual policy services department]*

  - Depends upon particular adjective in determining how the *for* phrase attaches
- One could explore refining or lexicalizing some part-of-speech classes to account for such differences.

Using POS contexts for increasing recall

- Use POS tags as more general type of **context** (Dickinson 2005; Dickinson & Meurers 2005b)
  - 68% increase in recall, 53% error detection precision
  - Could be combined with the POS nucleus approach.
- Alternative: Immediate dominance variation method (Dickinson & Meurers 2005c)
  - bottom-up check on RHS of treebank rules (not strings)
  - essentially checks endocentricity
  - example from WSJ:
    - VP → ADVP VBN NP (167 times in WSJ)
    - PP → ADVP VBN NP (twice in WSJ ⇒ errors)

Variation Detection for Dependency Annotation

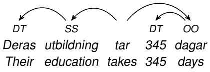
(Boyd, Dickinson & Meurers 2008)

- A range of high-quality dependency treebanks for a variety of different languages are available, e.g.:
  - Prague Dependency Treebank (PDT) of Czech (Hajič et al. 2001)
  - Alpino Dependency Treebank of Dutch (van der Beek et al. 2001)
  - Talbanken05 corpus of Swedish (Nivre et al. 2006)
  - Arboretum treebank for Danish (Bick 2003)
  - Danish Dependency Treebank (Kromann et al. 2004)
- Multi-lingual dependency parsing highlighted by 2006 CoNLL-X Shared Task
- As far as we are aware, little work has been done on automatically detecting errors in dependency treebanks.

Dependency annotation

Some characteristics

- Dependency annotation
  - captures grammatical relations between words
  - can relate non-adjacent elements
  - may include non-projectivity (dependencies may cross)
- Example from Talbanken05 corpus (Nivre et al. 2006):



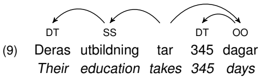
(9) *Deras utbildning tar 345 dagar*
Their education takes 345 days

Corpora used for dependency error detection

- Explore approach on the basis of three diverse dependency annotation schemes for three languages:
  - Talbanken05 corpus of Swedish (Nivre et al. 2006)
    - approx. 320,000 tokens
    - distinguishes 69 dependency relations
  - Prague Dependency Treebank (PDT 2.0) of Czech (Hajič et al. 2003), Analytical layer (surface syntax):
    - 1.5 million tokens (88,000 sentences)
    - distinguishes 28 dependency relations
  - Tiger Dependency Bank (TigerDB) of German (Forst et al. 2004)
    - semi-automatically derived from the Tiger Treebank (Brants et al. 2002), a corpus of German newspaper text taken from the Frankfurter Rundschau.
    - 36,326 tokens (1,868 sentences)
    - distinguishes 53 dependency relations, following English PARC 700 Dependency Bank (King et al. 2003), including sublexical and abstract nodes

Adapting the method to dependency annotation

- What is involved in applying the variation *n*-gram method to dependency annotation?
  - Mapping from a pair of words to their dependency relation label, we have variation nuclei of size 2.
  - We encode the head information into the label:
    - R means the head is on the right
    - L for the left
- Example from Talbanken05 corpus:



(9) Deras utbildning tar 345 dagar
Their education takes 345 days

- utbildning tar: SS-R
- tar dagar: OO-L

Applying the variation *n*-gram method

With the dependency annotated data encoded in this way, there are three different possibilities for errors:

- Errors in labeling: SUBJ vs. OBJ
- Errors in what the head is: OBJ-L vs. OBJ-R
- Errors in dependency identification: OBJ vs. NIL

What needs to be added to the basic picture?

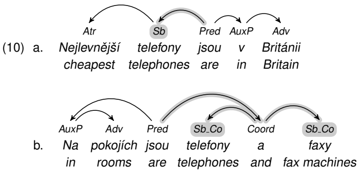
- Take the nature of dependency annotation into account in
  - defining the set of variations that need to be considered
  - determining a notion of context to identify errors → NIL-internal and dependency context
- Dependencies differ from constituency by allowing overlap (in head of dep.) and non-contiguity.
- Note: Variation detection checks each mappings from nucleus to its annotation independently, so no single-head assumption is needed.

Indirect annotation

- Variation *n*-gram approach is strictly data driven
- In mapping from words to dependency labels, each dependency relation label is considered independent of the others.
  - Locality assumption similar to the well-known independence assumption for local trees in PCFGs.
- In some dependency treebanks, no such locality requirement is enforced: some labels are based upon annotation decisions elsewhere in the graph.
- Examples for such *indirect* dependency encoding:
  - prepositions, complementizers, coordination in the PDT (analytical layer).

Indirect annotation

Example: Coordination



(10) a.

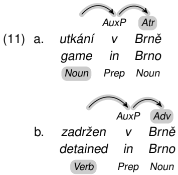
Atr	Sb	Pred	AuxP	Adv
Nejlevnější	*telefony*	*jsou*	*v*	*Británii*
cheapest	*telephones*	*are*	*in*	*Britain*

b.

AuxP	Adv	Pred	Sb_Co	Coord	Sb_Co
Na	*pokojích*	*jsou*	*telefony*	*a*	*faxy*
in	*rooms*	*are*	*telephones*	*and*	*fax machines*

Indirect annotation

Example: Prepositions



(11) a.

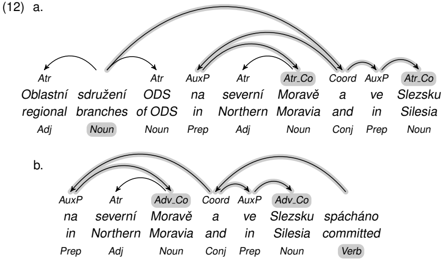
	AuxP	Atr
utkání	*v*	*Brně*
game	*in*	*Brno*
Noun	Prep	Noun

b.

	AuxP	Adv
zadržen	*v*	*Brně*
detained	*in*	*Brno*
Verb	Prep	Noun

Indirect annotation

Example: Indirection can cross significant contexts



(12) a.

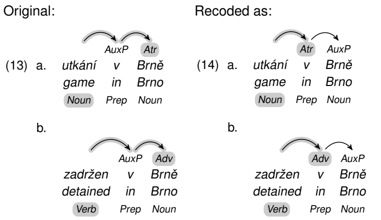
Atr		Atr	AuxP	Atr	Atr_Co	Coord	AuxP	Atr_Co
Oblastní	*sdružení*	*ODS*	*na*	*severní*	*Moravě*	*a*	*ve*	*Slezsku*
regional	*branches*	*of ODS*	*in*	*Northern*	*Moravia*	*and*	*in*	*Silesia*
Adj	Noun	Noun	Prep	Adj	Noun	Conj	Prep	Noun

b.

AuxP	Atr	Adv_Co	Coord	AuxP	Adv_Co	
na	*severní*	*Moravě*	*a*	*ve*	*Slezsku*	*spácháno*
in	*Northern*	*Moravia*	*and*	*in*	*Silesia*	*committed*
Prep	Adj	Noun	Conj	Prep	Noun	Verb

Indirect annotation

Possible recoding of some cases as local to head



Original:

(13) a.

	AuxP	Atr
utkání	*v*	*Brně*
game	*in*	*Brno*
Noun	Prep	Noun

b.

	AuxP	Adv
zadržen	*v*	*Brně*
detained	*in*	*Brno*
Verb	Prep	Noun

Recoded as:

(14) a.

	Atr	AuxP
utkání	*v*	*Brně*
game	*in*	*Brno*
Noun	Prep	Noun

b.

	Adv	AuxP
zadržen	*v*	*Brně*
detained	*in*	*Brno*
Verb	Prep	Noun

Adapting the variation nuclei algorithm

1. Compute the set of nuclei:
   a) Store all dependency pairs with dependency label.
      - The dependency relations annotated in the corpus are handled as nuclei of size two and mapped to their label plus a marker of the head (L/R).
      - The labels of overlapping type-identical nuclei are collapsed into a set of labels.

   b) For each distinct pair of words stored as dependency, search for non-dependency occurrences of words and add the nuclei with label NIL.
      - A trie data structure is used to store all potential nuclei and to guide the search for NIL nuclei.
      - Search is limited to pairs within same sentence.
      - NIL nuclei which overlap with a genuine dependency are not considered.
2. Compute the set of variation nuclei by determining which of the stored nuclei have more than one label.

Dependency annotation error detection results

- Error detection precision:
  - Talbanken 05: 92.9% (with 47% dep. ident. errors)
  - PDT 2.0: 59.7% (with 40% dep. identification errors)
  - TigerDB: 48.1% (with 70% dep. identification errors)
- Qualitative analysis:
  - Talbanken:
    - common problems: determiner (DT), preposition (PA)
    - more errors with adverbials than arguments
  - PDT observations:
    - 49% of false positives due to other indirect annotation scheme decisions (coordination)
    - common problem with AdvAtr vs. AtrAdv, preference for adverbial of predicate vs. attribute of lower node
  - TigerDB:
    - consistent tokenization of multi-word expressions and proper names is a problem, e.g., *Den Haag* (*The Hague*), *zur Zeit* (*at that time*)
    - prepositional argument vs. modifier distinction difficult, e.g., *Bedarf an X* (*demand for X*).
    - false positives due to ambiguous tokens, for which POS disambiguation would help

Summary

- We motivated the need for error detection in annotated corpora, and introduced the variation *n*-gram approach as an automatic error detection method.
- Research on category learning in humans provides independent evidence for the *notion of context* used.
- The method successfully detects errors in
  - part of speech
  - constituency,
  - discontinuous constituency,
  - and dependency annotation
- We showed that the method can provide significant feedback on annotation scheme distinctions which
  - are not sufficiently documented,
  - rely on representational choices not locally motivated,
  - or cannot reliably be made based on the evidence found in the corpus,

References

Abeillé, A. (ed.) (2003). *Treebanks: Building and using syntactically annotated corpora*. Dordrecht: Kluwer Academic Publishers. http://treebank.linguist.jussieu.fr/toc.html.

Abney, S., R. E. Schapire & Y. Singer (1999). Boosting Applied to Tagging and PP Attachment. In P. Fung & J. Zhou (eds.), *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. pp. 38–45.

Agrawal, R. & R. Srikant (1994). Fast Algorithms for Mining Association Rules in Large Databases. In J. B. Bocca, M. Jarke & C. Zaniolo (eds.), *VLDB 1994*. Morgan Kaufmann, pp. 487–499.

Artstein, R. & M. Poesio (2009). Survey Article: Inter-Coder Agreement for Computational Linguistics. *Computational Linguistics* 34(4), 1–42. URL http://www.mitpressjournals.org/doi/abs/10.1162/coli.07-034-R2.

Bick, E. (2003). Arboretum, a Hybrid Treebank for Danish. In *Proceedings of TLT 2003 (2nd Workshop on Treebanks and Linguistic Theory*. Växjö, Sweden, pp. 9–20.

Bies, A., M. Ferguson, K. Katz & R. MacIntyre (1995). *Bracketing Guidelines for Treebank II Style Penn Treebank Project*. University of Pennsylvania. ftp://ftp.cis.upenn.edu/pub/treebank/doc/manual/root.ps.gz.

Blaheta, D. (2002). Handling noisy training and testing data. In *Proceedings of the 7th conference on Empirical Methods in Natural Language Processing*. pp. 111–116. http://www.cs.brown.edu/~dpb/papers/dpb-emnlp02.html.

Boyd, A., M. Dickinson & D. Meurers (2007). Increasing the Recall of Corpus Annotation Error Detection. In *Proceedings of the Sixth Workshop on Treebanks and Linguistic Theories (TLT-07)*. Bergen, Norway. URL http://purl.org/dm/papers/boyd-et-al-07b.html.

Boyd, A., M. Dickinson & D. Meurers (2008). On Detecting Errors in Dependency Treebanks. *Research on Language and Computation* 6(2), 113–137. URL http://purl.org/dm/papers/boyd-et-al-08.html.

Brants, S., S. Dipper, S. Hansen, W. Lezius & G. Smith (2002). The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*. Sozopol, Bulgaria. www.bultreebank.org/proceedings/paper03.pdf.

Brants, T. & W. Skut (1998). Automation of Treebank Annotation. In *Proceedings of New Methods in Language Processing (NeMLaP-98)*. Syndey. http://www.coli.uni-sb.de/~thorsten/publications/Brants-Skut-NeMLaP98.ps.gz

Calder, J. (1997). On aligning trees. In *Proceedings of the Second Conference of Empirical Methods in Natural Language Processing*. Brown University. http://xxx.lanl.gov/abs/cmp-lg/9707016.

Chemla, E., T. H. Mintz, S. Bernal & A. Christophe (2009). Categorizing words using 'frequent frames': what cross-linguistic analyses reveal about distributional acquisition strategies. *Developmental Science* 12(3). URL http://dx.doi.org/10.1111/j.1467-7687.2009.00825.x.

Dickinson, M. (2005). Error detection and correction in annotated corpora. Ph.D. thesis, The Ohio State University.

Dickinson, M. & W. D. Meurers (2003a). Detecting Errors in Part-of-Speech Annotation. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL-03)*. Budapest, Hungary, pp. 107–114. http://ling.osu.edu/~dm/papers/dickinson-meurers-03.html.

Dickinson, M. & W. D. Meurers (2003b). Detecting Inconsistencies in Treebanks. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT-03)*. Växjö, Sweden, pp. 45–56. http://ling.osu.edu/~dm/papers/dickinson-meurers-tlt03.html.

Dickinson, M. & W. D. Meurers (2004). Error detection with discontinuous constituents. In P. Rodrigues, D. Cavar & J. Herring (eds.), *Proceedings of the First Midwest Computational Linguistics Colloquium*. Bloomington, Indiana.

Dickinson, M. & W. D. Meurers (2005a). Detecting Annotation Errors in Spoken Language Corpora. In *The Special Session on treebanks for spoken language and discourse at NODALIDA-05*. Joensuu, Finland. URL http://ling.osu.edu/~dickinso/papers/dickinson-meurers-nodalida05.html.

Dickinson, M. & W. D. Meurers (2005b). Detecting Errors in Discontinuous Structural Annotation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. pp. 322–329. http://www.aclweb.org/anthology/P/P05/P05-1040.

Dickinson, M. & W. D. Meurers (2005c). Prune Diseased Branches to Get Healthy Trees! How to Find Erroneous Local Trees in a Treebank and Why It Matters. In *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005)*. Barcelona, Spain. URL http://ling.osu.edu/~dm/papers/dickinson-meurers-tlt05.html.

Eskin, E. (2000). Automatic Corpus Correction with Anomaly Detection. In *Proceedings of the First Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-00)*. Seattle, Washington. http://www.cs.columbia.edu/~eeskin/papers/treebank-anomaly-naacl00.ps.

Forst, M., N. Bertomeu, B. Crysmann, F. Fouvry, S. Hansen-Schirra & V. Kordoni (2004). Towards a Dependency-Based Gold Standard for German Parsers.

The TIGER Dependency Bank. In S. Hansen-Schirra, S. Oepen & H. Uszkoreit (eds.), *5th International Workshop on Linguistically Interpreted Corpora (LINC-04) at COLING*. Geneva, Switzerland: COLING, pp. 31–38. URL http://aclweb.org/anthology/W04-1905.

Hajič, J., A. Böhmová, E. Hajičová & B. Vidová-Hladká (2003). The Prague Dependency Treebank: A Three-Level Annotation Scenario. In Abeillé (2003), chap. 7, pp. 103–127. URL http://ufal.mff.cuni.cz/pdt2.0/publications/HajicHajicovaAl2000.pdf. http://treebank.linguist.jussieu.fr/toc.html.

Hajič, J., B. Hladká & P. Pajas (2001). The Prague Dependency Treebank: Annotation Structure and Support. In *IRCS Workshop on Linguistic Databases*.

Hirakawa, H., K. Ono & Y. Yoshimura (2000). Automatic Refinement of a POS Tagger Using a Reliable Parser and Plain Text Corpora. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING)*. Saarbrücken, Germany: ICCL.

Hockenmaier, J. & M. Steedman (2005). *CCGbank: User's Manual*. Tech. Rep. MS-CIS-05-09, Department of Computer Science and Information Science, University of Pennsylvania, Philadelphia.

Johansson, S. (1986). *The Tagged LOB Corpus: Users' Manual*. Norwegian Computing Centre for the Humanities, Bergen.

King, T. H., R. Crouch, S. Riezler, M. Dalrymple & R. M. Kaplan (2003). The PARC 700 Dependency Bank. In *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora, held at the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL-03)*. Budapest. URL http://www2.parc.com/isl/groups/nltt/fsbank/.

Kromann, M. T., L. Mikkelsen & S. K. Lynge (2004). Danish Dependency Treebank: Annotation Guide. http://www.id.cbs.dk/~mtk/treebank/guideT.html.

Květŏn, P. & K. Oliva (2002). Achieving an Almost Correct PoS-Tagged Corpus. In P. Sojka, I. Kopeček & K. Pala (eds.), *Text, Speech and Dialogue 5th International Conference, TSD 2002, Brno, Czech Republic, September 9-12, 2002*. Heidelberg: Springer, no. 2448 in Lecture Notes in Artificial Intelligence (LNAI), pp. 19–26.

Leech, G. (1997). *A Brief Users' Guide to the Grammatical Tagging of the British National Corpus*. UCREL, Lancaster University, Lancaster. http://www.hcu.ox.ac.uk/BNC/what/gramtag.html.

MacWhinney, B. (2000). *The CHILDES project: Tools for analyzing talk*. Mahwah, NJ: Lawrence Erlbaum Associates, third edition ed.

Marcus, M., B. Santorini & M. A. Marcinkiewicz (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* 19(2), 313–330. ftp://ftp.cis.upenn.edu/pub/treebank/doc/cl93.ps.gz.

Meurers, W. D. (2005). On the use of electronic corpora for theoretical linguistics. Case studies from the syntax of German. *Lingua* 115(11), 1619–1639. http://ling.osu.edu/~dm/papers/meurers-03.html.

Mintz, T. H. (2002). Category induction from distributional cues in an artificial language. *Memory & Cognition* 30, 678–686.

Mintz, T. H. (2003). Frequent frames as a cue for grammatical categories in child directed speech. *Cognition* 90, 91–117.

Müller, F. H. & T. Ule (2002). Annotating topological fields and chunks – and revising POS tags at the same time. In *Proceedings of COLING*. http://ling.osu.edu/~dm/02/spring/795K/mueller-ule.ps.

Nivre, J., J. Nilsson & J. Hall (2006). Talbanken05: A Swedish Treebank with Phrase Structure and Dependency Annotation. In *Proceedings of the fifth international conference on Language Resources and Evaluation (LREC2006)*. Genoa, Italy.

Oliva, K. (2001). The Possibilities of Automatic Detection/Correction of Errors in Tagged Corpora: A Pilot Study on a German Corpus. In V. Matoušek, P. Mautner, R. Mouček & K. Taušer (eds.), *Text, Speech and Dialogue. 4th International Conference, TSD 2001, Zelezna Ruda, Czech Republic, September 11-13, 2001, Proceedings*. Springer, vol. 2166 of *Lecture Notes in Computer Science*, pp. 39–46.

Padro, L. & L. Marquez (1998). On the Evaluation and Comparison of Taggers: the Effect of Noise in Testing Corpora. In *COLING-ACL*. pp. 997–1002. URL citeseer.ist.psu.edu/padro98evaluation.html.

Sampson, G. & A. Babarczy (2003). Limits to annotation precision. In *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora (LINC-03)*. pp. 61–68. http://www.grsampson.net/Alta.html.

Santorini, B. (1990). Part-Of-Speech Tagging Guidelines for the Penn Treebank Project (3rd Revision, 2nd printing). Ms., UPenn.

Ule, T. & K. Simov (2004). Unexpected Productions May Well be Errors. In *Proceedings of Fourth International Conference on Language Resources and Evaluation (LREC 2004)*. Lisbon, Portugal. http://www.sfs.uni-tuebingen.de/~ule/Paper/us04lrec.pdf.

van der Beek, L., G. Bouma, R. Malouf & G. van Noord (2001). The Alpino Dependency Treebank. In *Computational Linguistics in the Netherlands (CLIN) 2001*, Amsterdam: Rodopi.

van Halteren, H. (2000). The Detection of Inconsistency in Manually Tagged Text. In A. Abeillé, T. Brants & H. Uszkoreit (eds.), *Proceedings of the Second Workshop on Linguistically Interpreted Corpora (LINC-00)*. Luxembourg.

van Halteren, H., W. Daelemans & J. Zavrel (2001). Improving Accuracy in Word Class Tagging through the Combination of Machine Learning Systems. *Computational Linguistics* 27(2), 199–229.

Voutilainen, A. & T. Järvinen (1995). Specifying a shallow grammatical representation for parsing purposes. In *Proceedings of the 7th Conference of the EACL*. Dublin, Ireland. http://www.aclweb.org/anthology/E95-1029.

Xiao, L., X. Cai & T. Lee (2006). The development of the verb category and verb argument structures in Mandarin-speaking children before two years of age. Paper presented at The Seventh Tokyo Conference on Psycholinguistics. Keio University.