Analyzing Parser Errors

To Improve Parsing Accuracy And To Inform Treebanking Decisions

Samar Husain*, Bhasha Agrawal§

*University of Potsdam, Germany
§IIITDM-Jabalpur, India

January 7, 2012

TLT10,
Heidelberg, Germany

Background

- Treebanks play a crucial role in building tools for various languages

Background

- Treebanks play a crucial role in building tools for various languages
  - Syntactic parsers

Background

- Treebanks play a crucial role in building tools for various languages
  - Syntactic parsers
- Syntactic parsers are needed for further analysis (e.g. semantic processing) (Gildea and Palmer, 2002)

Background

- Treebanks play a crucial role in building tools for various languages
  - Syntactic parsers
- Syntactic parsers are needed for further analysis (e.g. semantic processing) (Gildea and Palmer, 2002)
  - Better parser performance is always welcome!

Background: Factors that affect parser performance

- What affects dependency parser performance?
  - Algorithm
  - Grammar/Learner/Parameterization
  - Training data size
  - Feature space
  - Tagset size
  - Linguistic features
  - ...

Background: Factors that affect parser performance

- What affects dependency parser performance?
  - Algorithm
  - Grammar/Learner/Parameterization
  - Training data size
  - Feature space
  - Tagset size
  - Linguistic features
  - ...
  - Treebank guidelines decisions?

Background: Parser error analysis

- Error analysis feeds into improvements in
  - Algorithm
  - Grammar/Learner/Parameterization
  - Features
  - ...

Background: Parser error analysis

- Error analysis feeds into improvements in
  - Algorithm
  - Grammar/Learner/Parameterization
  - Features
  - ...
  - Treebank guidelines decisions?

Background: Guidelines and Automatic parsers

- Assumptions
  - Guidelines: capture linguistic realities

Background: Guidelines and Automatic parsers

- Assumptions
  - Guidelines: capture linguistic realities
  - Automatic parsers: built to automate these guidelines

Background: Guidelines and Automatic parsers

- Assumptions
  - Guidelines: capture linguistic realities
  - Automatic parsers: built to automate these guidelines
- Not necessarily true

Background: Guidelines and Automatic parsers

- Assumptions
  - Guidelines: capture linguistic realities
  - Automatic parsers: built to automate these guidelines
- Not necessarily true
  - many linguistic phenomena have competing (and equally compelling) analyses

Background: Guidelines and Automatic parsers

- Assumptions
  - Guidelines: capture linguistic realities
  - Automatic parsers: built to automate these guidelines
- Not necessarily true
  - many linguistic phenomena have competing (and equally compelling) analyses
    - eg. coordination

Questions investigated via parser error analysis

1. Can the parsing of certain structures be made easier by using alternative analyses for these structures?

Questions investigated via parser error analysis

1. Can the parsing of certain structures be made easier by using alternative analyses for these structures?
2. Are there certain linguistic cues implicit (or missing) in the current treebank that can be made explicit (or added) in order to make the parsing of complex constructions easier?

Some practical concerns

- How should this process be introduced in the treebanking timeline?

Some practical concerns

- How should this process be introduced in the treebanking time-line?
  - Obviously, not after the entire treebank has been developed

Some practical concerns

- How should this process be introduced in the treebanking timeline?
  - Obviously, not after the entire treebank has been developed
  - On smaller subsets during the initial treebank development phase

Some practical concerns

- How should this process be introduced in the treebanking timeline?
  - Obviously, not after the entire treebank has been developed
  - On smaller subsets during the initial treebank development phase
- Can some of this process be automated?

Some practical concerns

- How should this process be introduced in the treebanking timeline?
  - Obviously, not after the entire treebank has been developed
  - On smaller subsets during the initial treebank development phase
- Can some of this process be automated?
  - Whether alternative structures for a linguistic phenomenon can be obtained from the original analysis deterministically via transformations?

Hindi/Urdu dependency treebank

- The Hindi/Urdu dependency treebank is part of a Multi Representational and Multi-Layered Treebank. (Bhatt et al., 2009)

Hindi/Urdu dependency treebank

- The Hindi/Urdu dependency treebank is part of a Multi Representational and Multi-Layered Treebank. (Bhatt et al., 2009)
    - Dependency annotation
    - Propbanking
    - Phrase structure annotation

Hindi/Urdu dependency treebank

- The Hindi/Urdu dependency treebank is part of a Multi Representational and Multi-Layered Treebank. (Bhatt et al., 2009)
  - Dependency annotation
  - Propbanking
  - Phrase structure annotation
- The dependency framework used in the dependency treebank is inspired by Panini's grammar of Sanskrit. (Bharati et al., 1995), (Begum et al., 2008)

Experimental setup

- All the experiments were conducted on Hindi/Urdu treebank using MaltParser (Nivre et al., 2007b)

Experimental setup

- All the experiments were conducted on Hindi/Urdu treebank using MaltParser (Nivre et al., 2007b)
- We use the dataset released as the part of ICON10 parsing contest (Husain et al., 2010)
  - The training, the development and the test set contained 2972, 543 and 321 sentences respectively

Experimental setup

- All the experiments were conducted on Hindi/Urdu treebank using MaltParser (Nivre et al., 2007b)
- We use the dataset released as the part of ICON10 parsing contest (Husain et al., 2010)
  - The training, the development and the test set contained 2972, 543 and 321 sentences respectively
- The error classification is based on the test data.

Experimental setup

- All the experiments were conducted on Hindi/Urdu treebank using MaltParser (Nivre et al., 2007b)
- We use the dataset released as the part of ICON10 parsing contest (Husain et al., 2010)
  - The training, the development and the test set contained 2972, 543 and 321 sentences respectively
- The error classification is based on the test data.
- Experimental results obtained using 5-fold cross-validation on the complete data-set

Experimental setup

- All the experiments were conducted on Hindi/Urdu treebank using MaltParser (Nivre et al., 2007b)
- We use the dataset released as the part of ICON10 parsing contest (Husain et al., 2010)
  - The training, the development and the test set contained 2972, 543 and 321 sentences respectively
- The error classification is based on the test data.
- Experimental results obtained using 5-fold cross-validation on the complete data-set
- Features and parameters for MaltParser adapted from Ambati et al (2010b). Their setup also gives us our baseline score.

Error Classification

All the wrong attachment instances are classified using

- Edge type and Non-projectivity
- Edge length
- Edge depth

Error Classification

- Edge type: class aims at capturing errors based on the particular linguistic phenomenon that an edge captures

Error Classification

- Edge type: class aims at capturing errors based on the particular linguistic phenomenon that an edge captures
  - Use of dependency labels on the edge

Error Classification

- Edge type: class aims at capturing errors based on the particular linguistic phenomenon that an edge captures
  - Use of dependency labels on the edge
  - These labels were futher classified into coarser classes

Error Classification

- Edge type: class aims at capturing errors based on the particular linguistic phenomenon that an edge captures
  - Use of dependency labels on the edge
  - These labels were futher classified into coarser classes
- Non-projectivity: an arc in a dependency tree is projective if there is a directed path from the head word x to all the words between the two endpoints of the arc (Kuebler et al., 2009)

Error Classification

- Edge length: linear distance between a head and its child

Error Classification

- Edge length: linear distance between a head and its child
- Edge depth: the level in the tree at which an edge is situated wrt root

Error Classification

Edge Type			No. of edges	No. of non-projective edges	No. of edges with correct label
Main (3.25%)			11	0	0
Intra Clausal (83.13%)	Verb Argument Structure (51.18%)	Complement (21%)	71	4	9
		Adjunct (30.18%)	102	10	45
	Non-verbal (23.67%)	Noun-modifier (14.79%)	50	1	3
		Adjective mod. (1.48%)	5	0	0
		Apposition (0.30%)	1	0	0
		Genitive(7.10%)	24	2	6
	Others (8.28%)	Co-ordination (6.80%)	23	1	0
		Complex Predicate (0.59%)	2	0	0
		Others(0.89%)	3	0	1
Inter Clausal (13.61%)	Co-ordination (0.89%)		3	0	0
	Sub-ordination (12.72%)	Conjunction (1.18%)	4	0	0
		Relative Clause (3.55%)	12	11	0
		Clausal Complement (3.85%)	13	6	0
		Apposition (1.48%)	5	4	0
		Verb Modifier (2.66%)	9	3	0

Error Analysis: Intra-clausal Errors

- Verbal complements and adjuncts
- Noun modifications (including apposition and genitive)
- Coordination
- Complex predicates

Intra-clausal Errors: Verbal complements and adjuncts

- Attachment ambiguities for a child due to intervening non-finite verbs. Frequently leads to long distance dependency

Intra-clausal Errors: Verbal complements and adjuncts

- Attachment ambiguities for a child due to intervening non-finite verbs. Frequently leads to long distance dependency
- Ambiguous post-position of a nominal child node

Intra-clausal Errors: Verbal complements and adjuncts

- Attachment ambiguities for a child due to intervening non-finite verbs. Frequently leads to long distance dependency
- Ambiguous post-position of a nominal child node
- Lack of post-position on the nominal child

Intra-clausal Errors: Verbal complements and adjuncts

- Attachment ambiguities for a child due to intervening non-finite verbs. Frequently leads to long distance dependency
- Ambiguous post-position of a nominal child node
- Lack of post-position on the nominal child
- The non-finite clause gets split due to shared arguments.

Intra-clausal Errors: Coordination

- The number of children of a coordinating conjunction is not fixed (it can be more than 2).

Intra-clausal Errors: Coordination

- The number of children of a coordinating conjunction is not fixed (it can be more than 2).
  - These children can be spread across the entire sentence (leading to long distance dependencies).

Intra-clausal Errors: Coordination

- The number of children of a coordinating conjunction is not fixed (it can be more than 2).
  - These children can be spread across the entire sentence (leading to long distance dependencies).
  - There are no robust cues to identify these children (e.g. commas are not always present)

Intra-clausal Errors: Coordination

- The number of children of a coordinating conjunction is not fixed (it can be more than 2).
  - These children can be spread across the entire sentence (leading to long distance dependencies).
  - There are no robust cues to identify these children (e.g. commas are not always present)
- Children can be sub-trees (e.g. a coordination of multiple non-finite verbs that take their own arguments, genitives, etc.).

Intra-clausal Errors: Coordination

- The number of children of a coordinating conjunction is not fixed (it can be more than 2).
  - These children can be spread across the entire sentence (leading to long distance dependencies).
  - There are no robust cues to identify these children (e.g. commas are not always present)
- Children can be sub-trees (e.g. a coordination of multiple non-finite verbs that take their own arguments, genitives, etc.).
  - This again creates long distance dependencies

Inter-clausal Errors

- The most common source of error in inter-clausal relations is non-projectivity.

Inter-clausal Errors

- The most common source of error in inter-clausal relations is non-projectivity.
- Complex interactions of different conjunctions that can lead to long distance dependencies are another source of error.

Caveats

- Issues with error analysis

Caveats

- Issues with error analysis
  - We tried to neglect errors due to learning problems, data sparsity.

Caveats

- Issues with error analysis
  - We tried to neglect errors due to learning problems, data sparsity.
  - Focused only on those errors that we thought are due to lack of robust features or diffcult to learn structures

Caveats

- Issues with error analysis
  - We tried to neglect errors due to learning problems, data sparsity.
  - Focused only on those errors that we thought are due to lack of robust features or diffcult to learn structures
  - Not a trivial task.

Experiments

- Use of patterns identified in the error analysis
  - to simplify dependency structures
  - to incorporate additional linguistic information and investigate ways to represent/encode this information

Experiment I: Exploring alternative analysis via structural transformations

- Inter-clausal errors: Three most frequent non-projective structures

Experiment I: Exploring alternative analysis via structural transformations

- Inter-clausal errors: Three most frequent non-projective structures
  - Explore the possibility of making these structures projective

Experiment I: Exploring alternative analysis via structural transformations

- Inter-clausal errors: Three most frequent non-projective structures
  - Explore the possibility of making these structures projective
- Intra-clausal coordination errors

Experiment I: Exploring alternative analysis via structural transformations

- Inter-clausal errors: Three most frequent non-projective structures
  - Explore the possibility of making these structures projective
- Intra-clausal coordination errors
  - Explore other alternatives to represent coordination

Experiment I: Exploring alternative analysis via structural transformations

- Inter-clausal errors: Three most frequent non-projective structures
  - Explore the possibility of making these structures projective
- Intra-clausal coordination errors
  - Explore other alternatives to represent coordination
- If the projective counterpart is linguistically sound, one could argue that the guideline decisions for such phenomenons can be revised

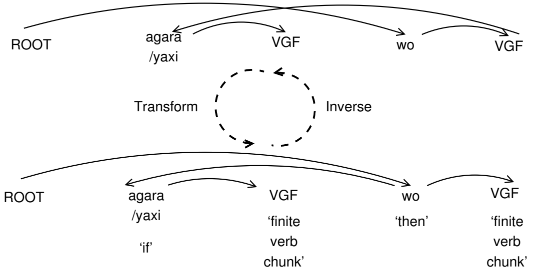
Experiment I: Exploring alternative analysis via structural transformations

- Inter-clausal errors: Three most frequent non-projective structures
  - Explore the possibility of making these structures projective
- Intra-clausal coordination errors
  - Explore other alternatives to represent coordination
- If the projective counterpart is linguistically sound, one could argue that the guideline decisions for such phenomenons can be revised
- Can the original structure be successfully recovered again after transformation?

Experiment I: Exploring alternative analysis via structural transformations

- Transformations lead to structural changes in a dependency tree when relations between nodes are modified leading to a new dependency tree

Structural transformations: Paired Connectives

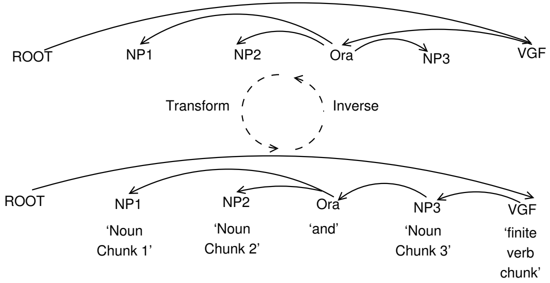


Ex:

'agara'	rAjA	kuCa	'cAhawe'	hEM	'wo'
'if'	'king'	'something'	'demand'	'is'	'then'
usa par	vicAra	'kiya'	jAnA cahie		
'it'	'consider'	'did'	'should'		

' If the king demands something , it should be considered. '

Structural transformations: Intra-clausal Coordination



Ex:	unake	alavA	kaI	'maMwrI'	'Ora'	'vipakRa'	ke
	'them'	'apart from'	'many'	'ministers'	'and'	'opposition'	'of'
	newA	mOjUxa	We				
	'leaders'	'present'	'were'				

' Apart from them, many ministers and leaders of opposition were present. '

Experiment I: Exploring alternative analysis via structural transformations

	Transformation	Transformation Cue Availability	Inverse Transformation	Use of a tool
1	Paired Connective	Yes	Yes	No
2	Relative Clause	Yes	Not 100%	MaxEnt
3	Complement Clause	Yes	Yes	No
4	Intra-clausal Coordination	Yes	Yes	No
5	Complex Predicate argument (with genitive case marker)	Yes	Yes	No

Table: Experiment I: Alternative structures using structure transformation.

Experiment II: Encoding linguistic information

- Analysis of a sentence requires the identification of various linguistic information present in a sentence

Experiment II: Encoding linguistic information

- Analysis of a sentence requires the identification of various linguistic information present in a sentence
- Such linguistic information can correspond to cues that are lexical, morphological, morpho-syntactic, syntactic, etc.

Experiment II: Encoding linguistic information

- Analysis of a sentence requires the identification of various linguistic information present in a sentence
- Such linguistic information can correspond to cues that are lexical, morphological, morpho-syntactic, syntactic, etc.
- But does the treebank provide all the cues necessary for analysis (correctly identifying a relation)?

Experiment II: Encoding linguistic information

- Analysis of a sentence requires the identification of various linguistic information present in a sentence
- Such linguistic information can correspond to cues that are lexical, morphological, morpho-syntactic, syntactic, etc.
- But does the treebank provide all the cues necessary for analysis (correctly identifying a relation)?
- Analyzing a sentence not only involves identification of such information but also exploring ways to encode this information during annotation.

Experiment II: Encoding linguistic information

- Analysis of a sentence requires the identification of various linguistic information present in a sentence
- Such linguistic information can correspond to cues that are lexical, morphological, morpho-syntactic, syntactic, etc.
- But does the treebank provide all the cues necessary for analysis (correctly identifying a relation)?
- Analyzing a sentence not only involves identification of such information but also exploring ways to encode this information during annotation.
- If there are multiple ways to encode some information, is one encoding strategy better than the other?

Experiment II: Encoding linguistic information

- Frequent errors related to verb's argument structure

Experiment II: Encoding linguistic information

- Frequent errors related to verb's argument structure
  - Encoding verb's valency via POS: VM $\rightarrow$ VM-1, VM-2, VM-3
    - No such distinction made in the original annotation

Experiment II: Encoding linguistic information

- Frequent errors related to verb's argument structure
  - Encoding verb's valency via POS: VM $\rightarrow$ VM-1, VM-2, VM-3
    - No such distinction made in the original annotation
  - Encoding subject-verb, object-verb agreement via dependency relations: k1 $\rightarrow$ k1-agr, k2 $\rightarrow$ k2-agr

Experiment II: Encoding linguistic information

- Frequent errors related to verb's argument structure
  - Encoding verb's valency via POS: VM $\rightarrow$ VM-1, VM-2, VM-3
    - No such distinction made in the original annotation
  - Encoding subject-verb, object-verb agreement via dependency relations: k1 $\rightarrow$ k1-agr, k2 $\rightarrow$ k2-agr
    - Agreement not explicitly annotated. But morphological features are present.

Experiment II: Encoding linguistic information

	Addition/ Modification	**Encoded through**	**Cue Availability**	**Inverse**	**Use of a resource**
1	Encoding verb-argument agreement	Dependency label	Yes (morphological)	Yes	No
2	Encoding conjunction valency	POS Tag	Yes (lexical)	Yes	Valency lexicon
3	Encoding verbal valency	POS Tag	Yes (lexical)	Yes	Bilingual Dictioanry, VerbNet

Table: Experiment II: Encoding linguistic information

Effect on parsing accuracy

	Experiments	LAS	UAS	LA	Statistical Significance
	Baseline Accuracy	77.58	88.97	80.48	-
Experiment I	*Paired Connectives	**77.70**	**89.15**	**80.61**	UAS,LAS,LA
	*Corelative and extraposed relative clauses	77.59	89.02	**80.72**	LA
	Clausal Complement	77.47	88.89	80.39	-
	Complex predicate argument (with genitive case marker)	77.50	88.77	80.43	-
	*Intra-clausal coordination	78.21	**89.06**	**81.21**	LAS,LA
Experiment II	Encoding agreement via dependency labels	74.15	88.87	79.98	-
	Encoding conjunction valency through POS	77.60	89.00	80.46	-
	Encoding verb's valency through POS	77.53	88.92	80.42	-

Table: Effect of Experiment I and II on parser accuracy. * shows statistical significance with McNemar's test ($p \leq 0.01$), computed using MaltEval. UAS: Unlabeled Attachment Score, LAS: Labeled Attachment Score, LA: Label Accuracy

Effect on parsing accuracy

- Only structural changes led to improvements in the accuracy

Effect on parsing accuracy

- Only structural changes led to improvements in the accuracy
- Other experiments that made changes in dependency labels, POS tags, etc. to encode valency information and agreement did not lead to any improvements

Effect on parsing accuracy

- Only structural changes led to improvements in the accuracy
- Other experiments that made changes in dependency labels, POS tags, etc. to encode valency information and agreement did not lead to any improvements
  - But we think that the low coverage (only 196 verbs out of 267 were found in the dictionary) of the bilingual dictionary for verbs affected the experiment where we tried to encode verbal valency.

Effect on parsing accuracy

- Only structural changes led to improvements in the accuracy
- Other experiments that made changes in dependency labels, POS tags, etc. to encode valency information and agreement did not lead to any improvements
  - But we think that the low coverage (only 196 verbs out of 267 were found in the dictionary) of the bilingual dictionary for verbs affected the experiment where we tried to encode verbal valency.
  - The lexical information for conjunctions is sufficient to disambiguate the coordination vs. subordination structures correctly; the added valency information seems to be redundant

Final remarks

- Parser error analysis and proposed experiments can play an important role in the evolution of treebanking guidelines decisions and consequently in improved parser performance

Final remarks

- Parser error analysis and proposed experiments can play an important role in the evolution of treebanking guidelines decisions and consequently in improved parser performance
- Does this make treebanks biased to specific parser development?

Final remarks

- Parser error analysis and proposed experiments can play an important role in the evolution of treebanking guidelines decisions and consequently in improved parser performance
- Does this make treebanks biased to specific parser development?
  - As long as the linguistic integrity of the analysis is maintained, this will not be a disadvantage.
- This not only leads to an increase in parser accuracy but also helps create better treebanks