Paraphrase and Ambiguity

Roland Hausser
Universität Erlangen-Nürnberg (em.)
email: rolandhausser662@gmail.com

Abstract

In natural language communication, the speak mode maps cognition-internal content into raw surface data,[1] while the hear mode maps raw surface data into content. The speaker may have a choice between different surfaces for the same content, called *paraphrase*, and the hearer may have to choose between different contents for the same surface, called *ambiguity* (FoCL 11.3).

The restriction of paraphrase to the speak mode and of ambiguity to the hear mode is not reflected in Generative Grammar because its *sign-based substitution-driven* ontology aims at characterizing "well-formedness of expressions," excluding communication (Nativism). For Database Semantics (DBS) with its *agent-based data-driven* ontology, in contrast, the restrictions are fundamental.

keywords: speak mode, hear mode, ambiguity, paraphrase, agent-based data-driven ontology, language surface, raw data, cognitive content

1 Introduction: The Structure of Content

The definition of ambiguity as a surface representing more than one content and of paraphrase as a content with more than one surface requires a definition of content. In natural language, there are three kinds of elementary contents, defined in DBS as follows:

1.1 Basic building blocks of content

a. concepts
are types which match raw data, resulting in tokens.

b. indexicals
receive their interpretation by pointing at the STAR[2] of an utterance.

c. function words[3]
modify concepts or indexical by taking them as arguments.

In the speak mode, the building blocks are connected by the semantic relations of structure, represented graphically as a / line for subject/predicate, a \ line for predicate\object, a | line for modifier|modified, and a − line for conjunct−conjunct. As an example, consider the following semantic relations analysis of Lucy found a big blue square, underlying the speak mode:

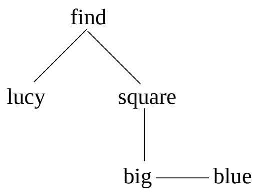
[1] I.e. cognition-external sound waves in speech and pixels in writing.

[2] In DBS, the STAR stands for SPACE, TIME, AGENT (speaker), and RECIPIENT (hearer).

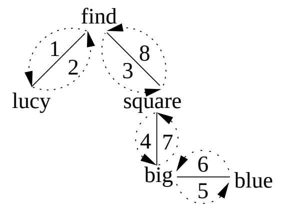
[3] Determiners, conjunctions, and prepositions. In natural language, they may be coded in the morphology (classical Latin) or in the syntax (English).

1.2 GRAPHICAL REPRESENTATION OF A COMPLEX CONTENT

(i) SRG (semantic relations graph)



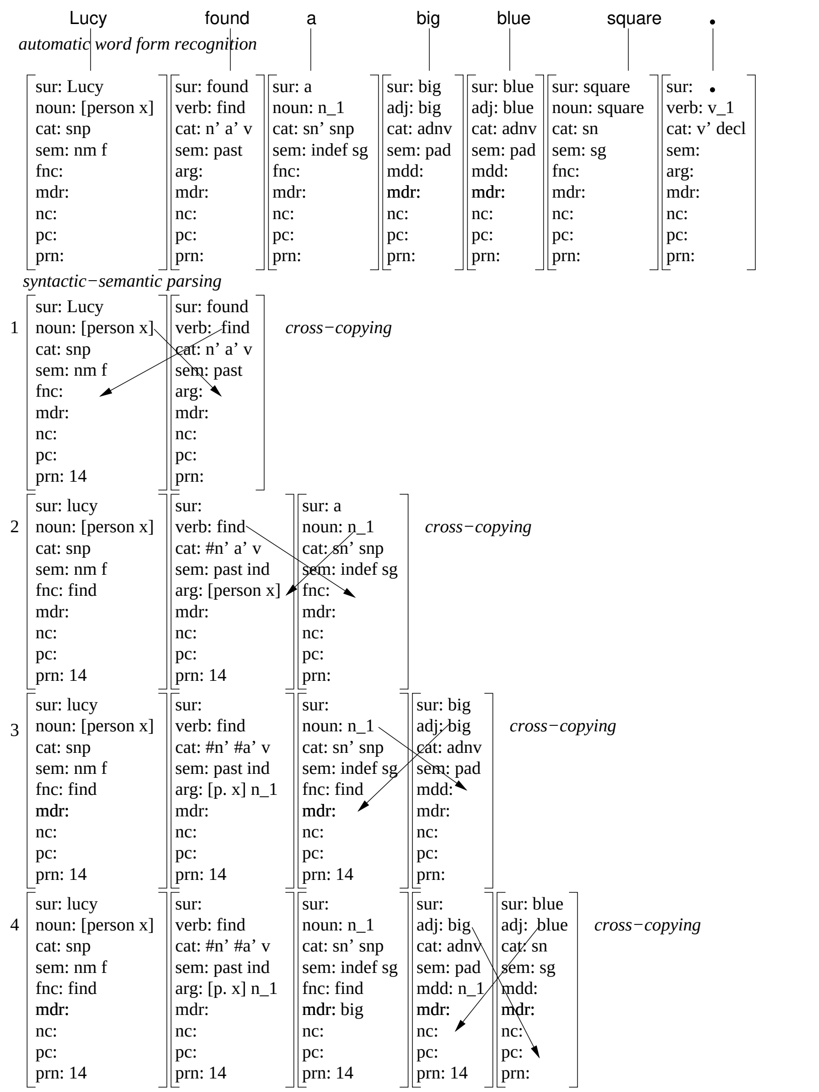
(iii) NAG (numbered arcs graph)



The speak mode is driven by a time-linear navigation along the numbered arcs.

The corresponding hear mode derivation is a surface-compositional time-linear concatenation of proplets (nonrecursive feature structures with ordered attributes[4]):

1.3 HEAR MODE CONSTRUCTS CONTENT FROM INPUT SURFACE



5	sur: lucy	sur:	sur:	sur:	sur:	sur: square	*absorption with simultaneous substitution*
	noun: [person x]	verb: find	noun: n_1	adj: big	adj: blue	noun: square	
	cat: snp	cat: #n' #a' v	cat: sn' snp	cat: adnv	cat: adnv	cat: sn	
	sem: nm f	sem: past ind	sem: indef sg	sem: pad	sem: pad	sem: sg	
	fnc: find	arg: [p. x] n_1	fnc: find	mdd: n_1	mdd:	fnc:	
	mdr:	mdr:	mdr: big	mdr:	mdr:	mdr:	
	nc:	nc:	nc:	nc: blue	nc:	nc:	
	pc:	pc:	pc:	pc:	pc: big	pc:	
	prn: 14	prn: 14	prn: 14	prn: 14	prn: 14	prn:	

6	sur: lucy	sur:	sur:	sur:	sur:	sur: •	*absorption*
	noun: [person x]	verb: find	noun: square	adj: big	adj: blue	verb: v_1	
	cat: snp	cat: #n' #a' v	cat: sn' snp	cat: adnv	cat: adnv	cat: v' decl	
	sem: nm f	sem: past ind	sem: indef sg	sem: pad	sem: pad	sem:	
	fnc: find	arg: [p. x] square	fnc: find	mdd: square	mdd:	arg:	
	mdr:	mdr:	mdr: big	mdr:	mdr:	mdr:	
	nc:	nc:	nc:	nc: blue	nc:	nc:	
	pc:	pc:	pc:	pc:	pc: big	pc:	
	prn: 14	prn: 14	prn: 14	prn: 14	prn: 14	prn:	

result

sur: lucy	sur:	sur:	sur:	sur:
noun: [person x]	verb: find	noun: square	adj: big	adj: blue
cat: snp	cat: #n' #a' decl	cat: sn' snp	cat: adnv	cat: adnv
sem: nm f	sem: past ind	sem: indef sg	sem: pad	sem: pad
fnc: find	arg: [p. x] square	fnc: find	mdd: square	mdd:
mdr:	mdr:	mdr: big	mdr:	mdr:
nc:	nc:	nc:	nc: blue	nc:
pc:	pc:	pc:	pc:	pc: big
prn: 14	prn: 14	prn: 14	prn: 14	prn: 14

The hear mode operations are (1) crosscopying (connective $\times$), (2) absorption (connective $\cup$), and (3) suspension (connective $\sim$).

Proplets serve as the computational data structure of DBS. The format encodes unlimited grammatical detail, both in terms of proplet attributes and of their values, yet there is no increase of computational complexity above linear because the processing is without recursion or iteration. Instead DBS relies on a strictly time-linear (i.e. left-associative[5]) derivation order in the speak and the hear mode.

2 Speak Mode Paraphrase: different surfaces for same content

A standard example of paraphrase is the active-passive alternation:

2.1 Paraphrase: Different surfaces for a single content

Mary read a book.
A book was read by Mary.

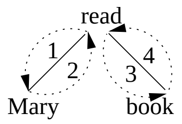
As a speak mode phenomenon, DBS analyzes paraphrase as different traversals of the same content. For example, the common content of the paraphrases 2.1 is represented as the following numbered arcs graph (NAG):

[4] Our definition is the direct opposite to *recursive* feature structures with *unordered* attributes (Carpenter 1992). Popular in Generative Grammar, recursive feature structures with unordered attributes are maximally inefficient for computational pattern matching, but justified by an extraneous notion of generality.

[5] Aho and Ullman 1977, p.47.

2.2 USING DIFFERENT TRAVERSAL ORDERS IN THE SPEAK MODE

(iii) NAG (numbered arcs graph)



(iv) surface realization active

1	2	3	4
Mary	read	a_book	.
V↙N	N↗V	V↘N	N↖V

surface realization passive

3	4	1	2
a_book	was_read	by_Mary	.
V↘N	N↖V	V↙N	N↗V

Each line in a semantic relations graph has a forward (downward V↙N or V↘N) and a backward (upward, V↗N or V↖N) traversal, indicated by dotted arrows in the NAG (CLaTR 7.4.1). Traversals may be empty, i.e. without a surface realization[6]. Like the hear mode (1.3), the speak mode (2.2) is strictly *surface-compositional* and *time-linear* (methodological principles of DBS).

3 DBS Formalism for the speak mode (language production)

In the active variant of 2.2, the first rule to apply in the sequence of speak mode operations is V↙N:

3.1 Navigating with V↙N from *read* to *Mary* (arc 1)

V↙N (s1)[7]

pattern level	[verb: α arg: β X prn: K]	⇒	[sur: lexnoun($\hat{\beta}$) noun: β fnc: α prn: K]	#-mark β in the arg slot of proplet α
	⇑		⇓	
content level	[sur: verb: read cat: #n′ #a′ decl sem: past arg: Mary book mdr: nc: pc: prn: 3]		[sur: Mary noun: Mary cat: s3 sem: sg fnc: read mdr: nc: pc: prn: 3]	

The operation lexnoun($\hat{\beta}$) in the sur slot of the goal proplet realizes the English surface Mary, based on β matching the initial value of the arg slot in the input proplet.

To realize the predicate and acquire the filler of the object slot, the navigation returns to the V with the speak mode operation N↗V:

[6]For example the traversal of arc 6 and 7 in 1.2.

[7]The 's' indicates a speak mode operation and the '1' refers to the operation number in the DBS speak mode grammar defined in TExer 6.5.1.

3.2 Navigating with N↗V from *Mary* to *read* (arc 2)

N↗V (s2)

pattern level	noun: β fnc: α mdr: Z prn: K	⇒	sur: lexverb($\hat{\alpha}$) verb: α arg: #β X prn: K	#-mark α in the fnc slot of proplet β

Z is NIL, or elementary and #-marked

⇑ ⇓

content level	sur: noun: Mary cat: s3 sem: sg fnc: read mdr: nc: pc: prn: 3	sur: read verb: read cat: #n′ a′ decl sem: past arg: #Mary book mdr: nc: pc: prn: 3

The #-marking of the first arg value in the goal proplet resulted from the instruction of V↙N (3.1). A #-marking instruction applies to a feature, here [arg: #Mary book], and not just to the value. For example, if a value in an arg slot is being #-marked, this does not affect the same value in a mdd slot.

Continuing the navigation from the predicate to the object is based on V↘N:

3.3 Navigating with V↘N from *read* to *book* (arc 3)

V↘N (s3)

pattern level	verb: α arg: #X β Y prn: K	⇒	sur: lexnoun($\hat{\beta}$) noun: β fnc: α prn: K	#-mark β in the arg slot of proplet α

⇑ ⇓

content level	sur: verb: read cat: #n′ #a′ decl sem: past arg: #Mary book mdr: nc: pc: prn: 3	sur: book noun: book cat: sp2 sem: fnc: #read mdr: nc: pc: prn: 3

The arg value #Mary is bound to the variable #X in the input pattern [arg: #X β Y], the arg value book to β, and a possible third argument (in a three place verb) would be bound to Y.

The return from the object to the predicate with N↖V is motivated by the need (i) to realize the punctuation mark (period), and (ii) to get into position for navigating to a possible successor proposition:

3.4 Navigating with N↖V from *book* back to *read* (arc 4)

N↖V (s4)

pattern level	noun: β fnc: α mdr: Z prn: K	⇒	sur: lexverb($\hat{\alpha}$) verb: α arg: #X #β Y prn: K
	Z is #-marked or NIL ⇑		⇓
content level	sur: noun: book cat: sp2 sem: fnc: #read mdr: nc: pc: prn: 3		sur: . verb: read cat: #n′ #a′ decl sem: past arg: #Mary #book mdr: nc: pc: prn: 3

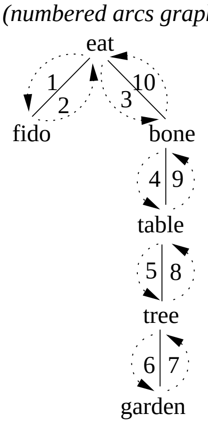
The traversal of the corresponding passive paraphrase in 2.2 uses the same speak mode operations, but in the order V↘N, N↖V, V↙N, N↗V.

4 Hear Mode Ambiguity: different contents for same surface

Syntactic ambiguity (FoCL Sect. 2.5) as a language-dependent hear mode phenomenon is a single surface for more than one content. A classic example in English is the alternative between the adnominal and adverbial use of a modifier.

4.1 FIRST READING: ADNOMINAL USE OF A MODIFIER

(iii) NAG (numbered arcs graph)



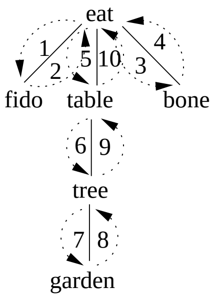
(iv) surface realization

1	2	3	4	5	6	7	8	9	10
Fido	ate	the_bone	on_the_table	under_the_tree	in_the_garden				.
V/N	N/V	V\N	N|N	N|N	N|N	N|N	N|N	N|N	N\V

In this reading, ON THE TABLE IN THE GARDEN UNDER THE TREE modifies BONE as an adnominal.

4.2 SECOND READING: ADVERBIAL USE

(iii) NAG (numbered arcs graph)



(iv) surface realization

1 2 3 4 5 6 7 8 9 10
Fido ate the_bone on_the_table under_the_tree in_the_garden •
V/N N/V V\N N\V N|N N|N N|N N|N N|N N\V

In this reading, ON THE TABLE IN THE GARDEN UNDER THE TREE modifies EAT as an adverbial.

5 Ambiguity is language-dependent

Ambiguities, lexical as well as syntactic, are language-dependent. For example, the following example is syntactically ambiguous in English, but its translation into German has two separate unambiguous readings:

5.1 Flying airplanes can be dangerous.

(a) Fliegende Flugzeuge können gefährlich sein.
(b) Flugzeuge zu fliegen kann gefährlich sein.

The cause of this ambiguity is the absence of a morphological distinction between the adnominal (fliegende) and the infinitival (zu fliegen) use of the English participle flying.

Another syntactic ambiguity in English with an unambiguous counterpart in German is the following:

5.2 They don't know how good meat tastes.

(a) Sie wissen nicht wie gut Fleisch schmeckt.
(b) Sie wissen nicht wie gutes Fleisch schmeckt.

The cause of this ambiguity is the absence of a morphological distinction between the adverbial (gut) and adnominal (gutes) use of English good.

6 Grammatical Analysis of an Ambiguity in DBS

Because the ambiguity of Flying airplanes can be dangerous is syntactic, it is treated in DBS by alternative syntactic-semantic operations, called **PROG×NP** and **ADN×PN**. These two hear mode operations take the same input, defined as follows:

6.1 Lexical analysis of FLYING (NLC A5.1.5) and AIRPLANES:

sur: flying	sur: airplanes
verb: fly	noun: airplane
cat: prog	cat: pn
sem:	sem:
arg:	fnc:
mdr:	mdr:
nc:	nc:
pc:	pc:
prn:	prn:

On one reading, the time-linear concatenation combines flying with a noun phrase as the object-completion of a reduced infinitive:

6.2 Cross-copying FLYING and AIRPLANES in OBJ COMPLETION

PROG×NP

pattern level	verb: α	noun: β	⇒	verb: α	noun: β
	cat: prog	cat: NP		cat: prog	cat: N
	arg:	fnc:		arg: ∅ β	fnc: α
	prn:K	prn:		prn:K	prn: K
	⇑				⇓
content level	sur: flying	sur: airplanes		sur: flying	sur: airplanes
	verb: fly	noun: airplane		verb: fly	noun: airplane
	cat: prog	cat: pn		cat: prog	cat: pn
	sem:	sem:		sem:	sem:
	arg:	fnc:		arg: ∅ airplane	fnc: fly
	mdd:	mdr:		mdd:	mdr:
	nc:	nc:		nc:	nc:
	pc:	pc:		pc:	pc:
	prn: 23	prn:		prn: 23	prn: 23

This hear mode operation connects the proplets *flying* and *airplanes* by cross-copying between the core value [verb: fly] into the continuation slot [fnc:] of *airplanes* and the core value [noun: airplane] into the [arg:] slot of *flying*.

On the other reading, a modifier is combined with a plural noun. This constellation may be with and without a determiner, i.e. THE FLYING AIRPLANES vs. FLYING AIRPLANES. For comparison with 6.2, the following operation application shows the latter:

6.3 Cross-copying FLYING and AIRPLANES in PN-MODIFICATION

ADN×PN

pattern level	verb: α cat: prog mdd: prn:K	noun: β cat: PN mdr: prn:	⇒	verb: α cat: prog mdd: β prn:K	noun: β cat: N mdr: α prn: K
	⇑				⇓
content level	sur: flying verb: fly cat: prog sem: arg: mdd: nc: pc: prn: 23	sur: airplanes noun: airplane cat: pn sem: fnc: mdr: nc: pc: prn:		sur: flying verb: fly cat: prog sem: arg: mdd: airplane nc: pc: prn: 23	sur: airplanes noun: airplane cat: pn sem: fnc: mdr: flying nc: pc: prn: 23

The two readings differ in that the cross-copying in 6.2 is from the core features [verb: fly] and [noun: airplane] into the continuation slots fnc and arg (completing the transitive infinitive with an object), but in 6.3 from the core features into the continuation slots mdd and mdr (modifying airplanes with flying).

7 Local vs. global Ambiguities

The ambiguities in Sect. 5 are called [+global][8] in DBS because they hold for complete expressions (sentence, proposition). An example of a [-global] or local ambiguity, in contrast, is the famous 'Gardenpath'[9] sentence by Bever (1970):

7.1 Local Ambiguity

Gardenpath sentence: THE HORSE RACED BY THE BARN (a) .
(b) FELL.

The continuation horse+raced introduces a [-global] ambiguity between (a) *horse raced* (active) and (b) *horse which was raced* (passive), resulting in two parallel derivation strands up to and including barn. Depending on continuing after barn with (a) an interpunctuation or (b) a verb, one of the two [-global] readings is grammatically disambiguated.

[8]The ±global distinction between ambiguities presupposes a time-linear interpretation of natural language, i.e. the computation of *possible continuations*, as in surface-compositional, time-linear DBS, which is agent-based data-driven. This is in contradistinction to sign-based substitution-driven Phrase Structure Grammar, which computes *possible substitutions* to characterize well-formedness without the distinction between the speak and the hear mode, i.e. regardless of communication (Nativism).

[9]So called because the initial interpretation up to barn is misleading, as in 'leading someone down the garden path'. In an era of substitution-driven "Generative Grammar," Bever's example is wide awake and far ahead of its time.

8 Iterating local ambiguities

Local ambiguities may be iterated, as shown by the following examples:

8.1 OBJECT-CLAUSE ITERATION

A. Bob believes Bill.
B. Bob believes that Bill believes Mary.
C. Bob believes that Bill believes that Mary believes Suzy.
D. Bob believes that Bill believes that Mary believes that Suzy believes Tim.

Here, the local ambiguities are between concluding with a full stop and continuing with a subclause. Local ambiguities do not affect the linear time complexity of natural language grammars in DBS.

9 Conclusion

The limitation of paraphrase to the speak mode and of ambiguity to the hear mode is a general phenomenon of natural language. When building a talking robot, the processing of paraphrase must be built only for the speak mode and the processing of ambiguity, local or global, must be built only for the hear mode.

Paraphrase of the speak mode does not affect the computational complexity of natural language communication. The only possible source of a complexity degree above linear would be ambiguity, restricted to the hear mode. However, to affect complexity, ambiguity would have to be (i) iterative/recursive and (ii) at least two readings would have to 'survive' each cycle. As shown by 8.1, (ii) is not the case.

Bibliography

Aho, A.V., and J.D. Ullman (1977) *Principles of Compiler Design*, Reading, MA: Addison-Wesley

Bever, T.G (1970) "The cognitive basis for linguistic structures". In: J.R. Hayes (ed.) *Cognition and the development of language*, pp. 279-362, New York: Wiley

Carpenter, B. (1992) *The Logic of Typed Feature Structures*, Cambridge: CUP

CLaTR = Hausser, R (2011) *Computational Linguistics and Talking Robots*, Springer; preprint 2nd ed. 2017 available online at `lagrammar.net`.

FoCL = Hausser, R. ([1999, 2002] 2014) *Foundations of Computational Linguistics*, Springer

Hausser, R. (2022) "Grammatical Disambiguation: The natural language linear complexity hypothesis," *Language and Information*, to appear

TExer = Hausser, R. (2020) *Twentyfour Exercises in Linguistic Analysis, DBS software design for the Hear and the Speak mode of a Talking Robot* (`lagrammar.net`)