Transformation-based Interpretation of Implicit Parallel Structures: Reconstructing the meaning of *vice versa* and similar linguistic operators

Helmut Horacek
Fachrichtung Informatik
Universität des Saarlandes
66041 Saarbrücken, Germany
horacek@ags.uni-sb.de

Magdalena Wolska
Fachrichtung Allgemeine Linguistik
Universität des Saarlandes
66041 Saarbrücken, Germany
magda@coli.uni-sb.de

Abstract

Successful participation in dialogue as well as understanding written text requires, among others, interpretation of specifications implicitly conveyed through parallel structures. While those whose reconstruction requires insertion of a missing element, such as gapping and ellipsis, have been addressed to a certain extent by computational approaches, there is virtually no work addressing parallel structures headed by *vice versa*-like operators, whose reconstruction requires transformation. In this paper, we address the meaning reconstruction of such constructs by an informed reasoning process. The applied techniques include building deep semantic representations, application of categories of patterns underlying a formal reconstruction, and using pragmatically-motivated and empirically justified preferences. We present an evaluation of our algorithm conducted on a uniform collection of texts containing the phrases in question.

1 Introduction

Specifications implicitly conveyed through parallel structures are an effective means of human communication. Handling these utterances adequately is, however, problematic for a machine since a formal reconstruction of the representation may be associated with ambiguities, typically requiring some degree of context understanding and domain knowledge in their interpretation. While parallel structures whose reconstruction mainly requires insertion, such as gapping and ellipsis, have been addressed to a certain extent by computational approaches, there is virtually no work addressing parallel structures whose reconstruction requires transformation. Several linguistic operators create specifications of this kind, including: *the other way (a)round*, *vice-versa*, and *analogously*. Consider, for example, the following statement made by a student in an experiment with a simulated tutoring system for proving theorems in elementary set theory (Benzmüller et al., 2003): "If all A are contained in $K(B)$ and this also holds *the other way round*, these must be identical sets" (K stands for set complement). The interpretation of the *the other way round* operator is ambiguous here in that it may operate on immediate dependents ("all $K(B)$ are contained in A") or on the embedded dependents ("all B are contained in $K(A)$") of the verb "contain". The fact that the *Containment* relation is asymmetric and the context of the task – proving that "If $A \subseteq K(B)$, then $B \subseteq K(A)$" holds – suggest that the second interpretation is meant. Assuming this more plausible reading enables a more goal-oriented dialog: the tutorial system can focus on a response to the false conclusion made by the student about the identity of the sets in question, rather than starting a boring clarification subdialog.

The above example and several similar others motivated us to look more systematically at lexical devices that create specifications of this kind. We address the interpretation of such structures by a well-informed reasoning process. Applied techniques include building deep semantic representations, application of patterns underlying formal reconstruction, and using pragmatically-motivated and empirically justified preferences.

The outline of the paper is as follows: We describe phenomena in question. Then we illustrate our natural language analysis techniques. We cate-

Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pages 377–384,
Sydney, July 2006.

gorize underlying interpretation patterns, describe the reconstruction algorithm, and evaluate it.

2 Data Collected From Corpora

In order to learn about cross-linguistic regularities in reconstructing the underlying form of propositions specified by *vice versa* or similar operators, we first looked at several English and German corpora. These included, among others, the Negra, the Frankfurter Rundschau, the Europarl corpora and a corpus of tutorial dialogs on mathematics (Wolska et al., 2004). We also performed several internet searches. We looked at the German phrases *andersrum* and *umgekehrt*, and their English equivalents *vice versa* and *the other way (a)round*. We only considered instances where the parallel structure with a pair of items swapped is not stated explicitly. We excluded cases of the use of *umgekehrt* as a discourse marker, cases in which the transformation needed is of purely lexical nature, such as turning "augment" into "reduce", and instances of *andersrum* as expressing a purely physical change, such as altering the orientation of an object (cf. the Bielefeld corpus[1]).

The classification of *vice versa* utterances presented in Figure 1, reflects the role of the items that must be swapped to build the parallel proposition conveyed implicitly. The examples demonstrate that the task of reconstructing the proposition left implicit in the text may be tricky.

The first category concerns swapping two case role fillers or *Arguments* of a predicate head. This may be applied to Agent and Patient dependents, as in (1), or to two directional roles as in (2). In the last example in this category, complications arise due to the fact that one of the arguments is missing on the surface and needs to be contextually inserted prior to building the assertions with exchanged directional arguments. Moreover, the swap can also work across clauses as in (3). Complex interrelations may occur when the fillers themselves are composed structures, is in (4), which also makes swapping other pairs of items structurally possible. In this example, the need for exchanging the persons including their mentioned body parts rather than the mere body parts or just the persons requires world knowledge.

The second category comprises swapping applied to *modifiers* of two arguments rather than the arguments themselves. An example is (5); the utterance is ambiguous since, from a purely structural point of view, it could also be categorized as an *Argument* swap, however, given world knowledge, this interpretation is rather infelicitous. Similarly to (3), a contextually-motivated enhancement prior to applying a swapping operation is required in (6); here: a metonymic extension, i.e. expanding the "strings" to "the strings' tones".

The third category comprises occurrences of a "mixed" form of the first two with a modifier substituted for an argument which, in turn, takes the role of the modifier in the reconstructed form. The first example, (7), has already been discussed in the Introduction. The next one, (8), illustrates repeated occurrences of the items to be swapped. Moreover, swapping the items A and B must be propagated to the included formula. The next example, (9), is handled by applying the exchange on the basis of the surface structure: swapping the properties of a triangle for the reconstructed assertion. If a deeper structure of the sentence's meaning is built, this would amount to an implication expressing the fact that a triangle with two sides of equal length is a triangle that has two equal angles. For such a structure, the reconstruction would fall into the next category, exchange of the order of two propositions: here, reversing the implication. In (10), the lexeme "Saxophonist" needs to be expanded into "Saxophone" and "Spieler" ("player"), prior to performing the exchange.

The fourth category involves a swap of entire *Propositions*; in the domain of mathematics, this may pertain to formulas. In (11), swapping applies to the sides of the equation descriptively referred to by the distributivity law. In (12), this applies to the arguments of the set inclusion relation, when the arguments are interpreted as propositions. The last example, (13), requires a structural recasting in order to apply the appropriate swapping operation. When the utterance is rebuilt around the $RESULT$ relation, expressed as an optional case role on the surface, swapping the two propositions – "branching out of languages" and "geographical separation" – yields the desired result.

3 The Interpretation Procedure

In this section, we illustrate our technical contribution. It consists of three parts, each dealt with in a separate subsection: (1) the linguistic/semantic analysis, (2) definitions of rules that support building parallel structures, and (3) the algorithm.

[1] http://www.sfb360.uni-bielefeld.de/

Argument swap	(1)	Technological developments influence the regulatory framework and vice versa.
	(2)	It discusses all modes of transport from the European Union to these third countries and vice versa.
	(3)	Ok – so the affix on the verb is the trigger and the NP is the target. . . . No; the other way round
	(4)	Da traf Völler mit seinem Unterarm auf die Hüfte des für Glasgow Rangers spielenden Ukrainers, oder umgekehrt *Then Völler with his lower arm hit the hip of the Ukrainian playing for Glasgow Rangers, or the other way round*
Modifier swap	(5)	Nowadays, a surgeon in Rome can operate on an ill patient – usually an elderly patient – in Finland or Belgium and vice versa.
	(6)	Der Ton der Klarinette ist wirklich ganz komplementär zu den Seiteninstrumenten und umgekehrt *The clarinet's tone is really very complimentary to strings and vice-versa*
Mixed swap	(7)	Wenn alle A in $K(B)$ enthalten sind und dies auch umgekehrt gilt, muß es sich um zwei identische Mengen handeln *If all A are contained in $K(B)$ and this also holds* vice-versa*, these must be identical sets*
	(8)	Dann ist das Komplement von Menge A in Bezug auf B die Differenz $A/B = K(A)$ und umgekehrt *Then the complement of set A in relation to B is the difference $A/B = K(A)$ and vice-versa*
	(9)	Ein Dreieck mit zwei gleichlangen Seiten hat zwei gleichgroße Winkel und umgekehrt *A triangle with two sites of equal length has two angles of equal size, and vice-versa*
	(10)	. . . Klarinette für Saxophonist und umgekehrt . . . *. . . a clarinet for a saxophonist and vice-versa . . .*
Proposition swap	(11)	Man muß hier das Gesetz der Distributivität von Durchschnitt über Vereinigung umgekehrt anwenden *It is necessary here to apply the law of distributivity of intersection over union in reverse direction*
	(12)	Es gilt: $P(C \cup (A \cap B)) \subseteq P(C) \cup P(A \cap B)$. . . . . Nein, andersrum. *It holds: $P(C \cup (A \cap B)) \subseteq P(C) \cup P(A \cap B)$. . . . . No, the other way round.*
	(13)	Wir wissen, daß sich Sprachen in Folge von geographischer Separierung auseinanderentwickeln, und nicht umgekehrt *We know that languages branch out as a result of geographical separation, not the other way round*

Figure 1: Examples of utterances with *vice versa* or similar operators

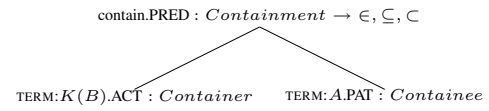

Figure 2: Interpreted representation of the utterance "all A are contained in $K(B)$"

3.1 Linguistic Analysis

The linguistic analysis consists of semantic parsing followed by contextually motivated embedding and enhancements. We assume a deep semantic dependency-based analysis of the source text. The input to our reconstruction algorithm is a relational structure representing a dependency-based deep semantics of the utterance, e.g. in the sense of Prague School sentence meaning, as employed in the Functional Generative Description (FGD) at the tectogrammatical level (Sgall et al., 1986). In FGD, the central frame unit of a clause is the head verb which specifies the *tectogrammatical relations* (TRs) of its dependents (*participants/modifications*). Every valency frame specifies, moreover, which modifications are *obligatory* and which *optional*. For example, the utterance (7) (see Figure 1.) obtains the interpretation presented in Figure 2.[2] which, in the context of an informal verbalization of a step in a naive set theory proof, translates into the following formal statement: "$\forall x.x \in A \Rightarrow x \in K(B)$".

The meaning representations are embedded within discourse context and discourse relations between adjacent utterances are inferred where possible, based on the linguistic indicators (discourse markers). The nodes (heads) and dependency relations of the interpreted dependency structures as well as discourse-level relations serve as input to instantiate the reconstruction patterns. Contextual enhancements (e.g. lexical or metonymic extensions) driven by the reconstruction requirements may be carried out.

Based on analysis of corpora, we have identified combinations of dependency relations that commonly participate in the swapping operation called for by the *vice versa* phrases. Examples of pairs of such relations at sentence level are shown in Figure 3.[3] Similarly, in the discourse context, arguments in, for example, $CAUSE$, $RESULT$, $CONDITION$, $SEQUENCE$ or $LIST$ relations are likely candidates for a swapping operation. During processing, we use the association table as a preference criterion for selecting candidate relations to instantiate patterns. If one of the elements of a candidate pair is an *optional argument* that is not realized in the given sentence, we look at the preceding context to find the first instance of the missing element. Additionally, utterance (10) would call for more complex procedures to identify the required metonymic expansion.

Exchangeable(ACTOR, PATIENT)
Exchangeable(DIRECTION-WHERE-FROM, DIRECTION-WHERE-TO)
Exchangeable(TIME-TILL-WHEN, TIME-FROM-WHEN)
Exchangeable(CAUSE, PRED)
Exchangeable(CONDITION, PRED)

Figure 3: Examples of exchangeable relations

3.2 Interpretation Patterns

In order to accomplish the formal reconstruction task, we define rules that encapsulate specifications for building the implicit parallel text on the basis of the corresponding co-text. The rules consist of a pattern and an action part. Patterns are matched against the output of a parser on a text portion in question, by identifying relevant case roles, and giving access to their fillers. Moreover, the patterns test constraints on compatibility of candidates for swapping operations. The actions apply recasting operations on the items identified by the patterns to build the implicit parallel text.

Within patterns, we perform category membership tests on the representation. Assuming x referring to a semantic representation, $Pred(x)$ is a logical function that checks if x has a $Pred$-feature, i.e., it is an atomic proposition. Similarly, $Conj(x)$ and $Subord(x)$ perform more specific tests for complex propositions: coordination or subordination, respectively. Moreover, $Pred_1(x, x_1)$ accesses the first proposition and binds it to x_1, while $Pred_2(x, x_2)$ does the same for the second one. Within a proposition, arguments and modifiers are accessed by $Case(x, y)$, where y specifies the filler of $Case$ in x, and indices express constraints on identity or distinctiveness of the relations. $Case^+$ is a generalization of $Case$ for iterative embeddings, where individual cases in the chain are not required to be

[2] We present a simplified schematic representation of the tectogrammatical representations. Where necessary, for space reasons, irrelevant parts are omitted.

[3] $PRED$ is the immediate predicate head of the corresponding relation.

1a. *Argument* swap within the same clause

$$Pred(x) \wedge Case_1(x, y) \wedge Case_2(x, z) \wedge Type - compatible(y, z) \wedge Exchangeable(Case_1, Case_2) \rightarrow Swap(x, y, z, x_p)$$

1b. *Argument* swap across two clauses

$$Conj(x) \wedge Case_1(x, y) \wedge Case(y, u) \wedge Case_2(x, z) \wedge Case(z, v) \rightarrow Swap(x, u, v, x_p)$$

2. *Modifier* swap

$$Pred(x) \wedge Case_1(x, y) \wedge Case^{+}_{11}(y, u) \wedge Case_2(x, z) \wedge Case^{+}_{21}(z, v) \wedge \neg(Case_1 = Case_2) \wedge Type - compatible(u, v) \rightarrow Swap(x, u, v, x_p)$$

3. *Mixed* swap

$$Pred(x) \wedge Case_1(x, y) \wedge Case_{11}(y, u) \wedge Case_2(x, z) \wedge \neg(Case_1 = Case_2) \wedge Type - compatible(u, z) \rightarrow Swap(x, u, z, x_p)$$

4. *Proposition* swap

$$Subord(x) \wedge Case_1(x, y) \wedge Case_2(x, z) \wedge \neg(Case_1 = Case_2) \rightarrow Swap(x, y, z, x_p)$$

Figure 4: Reconstruction patterns

1. *Lexical expansion*

$$Pred(x) \wedge Case_1(x, y) \wedge Lex - Expand(y, u, Case, v) \wedge Case_2(x, z) \wedge \neg(Case_1 = Case_2) \wedge Type - compatible(v, z) \rightarrow Swap(x, y, Case(u, v), x_p) \wedge Swap(x_p, z, v, x_p)$$

2. *Recast optional case as head of an obligatory*

$$Pred(x) \wedge Case_1(x, u) \wedge Case_2(x, v) \wedge Type(u, tu) \wedge Type(v, tv) \wedge Recastable(tv, Case_2, tu, Case_3) \wedge Case_3(x, w) \wedge Type - compatible(v, w) \wedge \neg(Case_1 = Case_2) \wedge \neg(Case_1 = Case_3) \wedge \neg(Case_2 = Case_3) \rightarrow Swap(x, u, v, x_p) \wedge Add(x_p, Case_3(v, u)) \wedge Remove(x_p, Case_2)$$

3. *Recast an optional case as a discourse relation*

$$Pred(x) \wedge Case(x, y) \wedge Member(Case, Subords) \rightarrow Build(Case(x_p, Case_2(x_p, y) \wedge Case_1(x_p, Remove(x, y))$$

Figure 5: Recasting rules

identical. In addition to access predicates, there are test predicates that express constraints on the identified items. The most basic one is $Type\text{-}compatible(x, y)$, which tests whether the types of x and y are compatible according to an underlying domain ontology. A more specific test is performed by $Exchangeable(Case_1, Case_2)$ to access the associations specified in the previous section. The action part of the patterns is realized by $Swap(x, y, z, x_p)$ which replaces all occurrences of x in z by y and vice-versa, binding the result to x_p. Different uses of this operation result in different instantiations of y and z with respect to the overarching structure x.

There are patterns for each category introduced in Section 2 (see Figure 4). All patterns are tested on a structure x and, if successful, the result is bound to x_p. For *Argument* swap there are two patterns. If the scope of the swap is a single clause (1a), two arguments (case roles) identified as exchangeable are picked. Their fillers must be compatible in types. If the swapping overarches two clauses (1b), the connecting relation must be a conjunction and subject to swapping are arguments in the same relations. For *Modifier* swap (2), type compatible modifiers of distinct arguments are picked. For *Mixed* swap (3), a dependent is picked, as in (1a) and a type-compatible modifier of another argument, as in (2). *Proposition* swap (4) inverts the order of the two clauses.

In addition to the the pattern matching tests, the *Argument* and the *Proposition* swap operations undergo a feasibility test if knowledge is available about symmetry or asymmetry of the relation (the $Pred$ feature) whose cases are subject to the swapping operation: if such a relation is known as asymmetric, the result is considered implausible due to semantic reasons, if it is symmetric, due to pragmatic reasons since the converse proposition conveys no new information; in both cases such a swapping operation is not carried out.

To extend the functionality of the patterns, we defined a set of recasting rules (Figure 5) invoked to reorganize the semantic representation prior to testing applicability of a suitable reconstruction rule. In contrast to inserting incomplete information contextually and expanding metonymic relations the recasting operations are intended purely to accommodate semantic representations for this purpose. We have defined three recasting rules (numbered accordingly in Figure 5):

1. *Lexical recasting*

   The semantics of some lexemes conflates the meaning of two related items. If one of them is potentially subject to swapping, it is not accessible for the operation without possibly af-

Build-Parallel-Structure (x)

1. Determine scopes for applying swap operations
$Structures \leftarrow \epsilon$
if $Pred(x)$ **then** $Scopes \leftarrow \{x\}$ **else**
  if $Subord(x) \vee Conj(x) \wedge Case_2(x, z)$
    then $Scopes \leftarrow \{z, x\}$
endif endif

2. Match patterns and build swapped structures

forall $Scope_1$ in $Scopes$ **do**
  $Structures \leftarrow Structures \cup$
  $< X - swap(Scope_1) >$
  $< X - swap(Y - recast(Scope_1)) >$
end forall
return $Sort(Apply - priorities(Structures))$

Figure 6: Reconstruction algorithm

fecting the other so closely related to it. The representation of such lexemes is expanded, provided there is a sister case with a filler that is type compatible.

2. *Case recasting*

   The dependency among items may not be reflected by the dependencies in the linguistic structure. Specifically, a dependent item may appear as a sister case in overarching case frame. The purpose of this operation is to build a uniform representation, by removing the dependent case role filler and inserting it as a modifier of the item it is dependent on.

3. *Proposition recasting*

   Apart from expressing a discourse relation by a connective, a proposition filling a subordinate relation may also be expressed as a case role (argument). Again, uniformity is obtained through lifting the argument (case filler) and expressing the discourse relation as a multiple clause construct.

Additional predicates are used to implement recasting operations. For example, the predicate $Lex{-}Expand(y, u, Case, v)$ re-expresses the semantics of y by u, accompanied by a $Case$ role filled by v. $Type(x, y)$ associates the type y with x. The type information is used to access $Recastable(t_1, C_1, t_2, C_2)$ table to verify whether case C_1 with a t_1-type filler can also be expressed as case C_2 with type t_2. $Build(x)$ creates a new structure x. $Remove(x, y)$ is realized as a function, deleting occurrences of y in x, and $Add(x, y)$ expands x by an argument y.

3.3 The Structure Building Algorithm

In this section, we describe how we build implicitly conveyed parallel structures based on the definitions of swapping operations with optional incorporation of recasting operations if needed. The procedure consists of two main parts (see Figure 6). In the first part, the scope for applying the swapping rules defined in Figure 4 is determined, and in the second part, the results obtained by executing the rules are collected. Due to practical reasons, we introduce simplifications concerning the scope of *vice-versa* in the current formulation of the procedure. While the effect of this operator may range over entire paragraphs in some involved texts, we only consider single sentences with at most two coordinated clauses or one subordinated clause. We feel that this restriction is not severe for uses in application-oriented systems.

The procedure *Build-Parallel-Structure* takes the last input sentence x, examines its clause structure, and binds potential scopes to variable $Scopes$. For composed sentences, the entire sentence (x) as well as the second clause ($Case_2(x, z)$) is a potential scope for building parallel structures.

In the second part of the procedure, each swapping pattern is tested for the two potential scopes, and results are accumulated in $Structures$. The call $< X - swap(Scope_1) >$, with X being either $Case$, $Argument$, $Mixed$, or $Prop$ expresses building a set of all possible instantiations of the pattern specified when applied to $Scope_1$. Some of these operations are additionally invoked with alternative parameters which are accommodated by a recasting operation fitting to the pattern used, that call being $< X - swap(Y - recast(Scope_1)) >$, where Y is $Case$, Lex, or $Prop$. Finally, if multiple readings are generated, they are ranked according to the following prioritized criteria:

1. The nearest scope is preferred;
2. Operations swapping "duals", such as left-right, are given priority;
3. Candidate phrases are matched against the corpus; items with higher bigram frequencies are preferred.

Linguistic analysis, structure reconstruction patterns, recasting rules, and the algorithms operating on top of these structures are formulated in a domain-independent way, also taking care that the tasks involved are clearly separated. Hence, it is up to a concrete application to elaborate lexical

semantic definitions required (e.g. for a saxophonist to capture example (10) in Figure 1) to define the tables $Exchangeable$ and $Recastable$, and to enhance preference criteria.

4 Evaluation

We conducted an evaluation of the parallel structure building algorithm on a sample of sentences from Europarl (Koehn, 2002), a parallel corpus of professionally translated proceedings of the European Parliament aligned at the document and sentence level. At this point, we were able to conduct only manual evaluation. This is mainly due to the fact that we did not have access to a wide-coverage semantic dependency parser for English and German.[4] In this section, we present our corpus sample and the evaluation results.

Evaluation sample To build the evaluation sample, we used sentence- and word-tokenized English German part of Europarl. Using regular expressions, we extracted sentences with the following patterns: (i) for English, phrases *the other way a*round* or *vice versa* (ii) for German: (ii-1) the word *umgekehrt* preceded by a sequence of *und* ("and"), *oder* ("or"), *sondern* ("but"), *aber* ("but") or comma, optional one or two tokens and optional *nicht* ("not"), (ii-2) the word *umgekehrt* preceded by a sequence *gilt* ("holds") and one or two optional tokens, (ii-3): the word *anders(he)*rum*. We obtained 137 sentences.

Next, given the present limitation of our algorithm (see Section 3.3), we manually excluded those whose interpretation involved the preceding sentence or paragraph,[5] as well as those in which the interpretation was explicitly spelled out. There were 27 such instances. Our final evaluation sample consisted of 110 sentences: 82 sentences in English–German pairs and 28 German-only.[6]

[4] In the future, we are planning an automated evaluation in which as input to the implemented algorithm we would pass manually built dependency structures.

[5] For example, sentences such as: "Mr President , concerning Amendment No 25 , I think the text needs to be looked at because in the original it is the other way round to how it appears in the English text ."

[6] The reason for this split is that the English equivalents of the German sentences containing the word *umgekehrt* may contain phrases other than *the other way round* or *vice versa*. Depending on context, phrases such as *conversely*, *in* or *the reverse*, *the opposite*, *on the contrary* may be used. Here, we targeted only *the other way round* and *vice versa* phrases. If the German translation contained the word *umgekehrt*, and the English source one of the alternatives to our target, in the evaluation we included only the German sentence.

Category	No. of instances
Arg	64
Modifier	5
Arg/Mod	3
Mixed	6
Arg/Mixed	2
Prop	1
Arg/Prop	1
Lex	18
Other	10
Total	110

Table 1: Distribution of patterns

Distribution of categories We manually categorized the structures in our sample and marked the elements of the dependency structures that participate in the transformation. Table 1. presents the distribution of structure categories. We explicitly included counts for alternative interpretations. For example *Arg/Mod* means that either the *Argument* or *Modifier* transformation can be applied with the same effect, as in the sentence "External policy has become internal policy, and vice versa": either the words "external" and "internal" may be swapped (*Modifier*), or the whole NPs "external policy" and "internal policy" (*Argument*). *Lex* means that none of the patterns was applicable and a lexical paraphrase (such as use of an antonym) needed to be performed in order to reconstruct the underlying semantics (i.e. no parallel structure was involved). *Other* means that there was a parallel structure involved, however, none of our patterns covered the intended transformation.

Evaluation results The evaluation results are presented in Tables 2. and 3. Table 2. shows an overview of the results. The interpretation of the result categories is as follows:

Correct: the algorithm returned the intended reading as a unique interpretation (this includes correct identification of "lexical paraphrases" (the *Lex* category in Table 1.);

Ambig.: multiple results were returned with the intended reading among them;

Wrong: the algorithm returned a wrong result (if multiple results, then the intended one was not included);

Failed: the algorithm failed to recognize a parallel structure where one existed because no known pattern matched.

Table 3. shows within-category results. Here, Correct result for *Other* means that the algorithm correctly identified 8 cases to which no current pattern applied. The two Wrong results for *Other*

Result	No. of instances
Correct	75
Ambig.	21
Wrong	4
Failed	10
Total	110

Table 2: Evaluation results

Category	Correct	Ambig.	Wrong	Failed	Total
Arg	46	17	0	1	64
Mod	3	2	0	0	5
Arg/Mod	3	–	0	0	3
Mixed	4	2	0	0	6
Arg/Mixed	2	–	0	0	2
Prop	1	0	0	0	1
Arg/Prop	0	–	0	1	1
Lex	16	0	2	0	18
Other	8	0	2	0	10

Table 3: Within-category results

mean that a pattern was identified, however, this pattern was not the intended one. In two cases (false-negatives), the algorithm failed to identify a pattern even though it fell into one of the known categories (*Argument* and *Prop*).

Discussion The most frequently occurring pattern in our sample is *Argument*. This is often a plausible reading. However, in 3 of the 4 false-positives (Wrong results), the resolved incorrect structure was *Arg*. If we were to take *Arg* as baseline, aside from missing the other categories (altogether 12 instances), we would obtain the final result of 63 Correct (as opposed to 96; after collapsing the Correct and Ambig. categories) and 15 (as opposed to 4) Wrong results.

Let us take a closer look at the false-negative cases and the missed patterns. Two missed known categories involved multiple arguments of the main head: a modal modifier (modal verb) and an additive particles ("also") in one case, and in the other, rephrasing after transformation. To improve performance on cases such as the former, we could incorporate an exclusion list of dependents that the transformation should disregard.

Among the patterns currently unknown to the algorithm, we found four types (one instance of each in the sample) that we can anticipate as frequently recurring: aim and recipient constructs involving a head and its Aim- and Beneficiary-dependent respectively, a temporal-sequence in which the order of the sequence elements is reversed, and a comparative structure with swapped relata. The remaining 6 structures require a more involved procedure: either the target dependent is deeply embedded or paraphrasing and/or morphological transformation of the lexemes is required.

5 Conclusions and Future Research

In this paper, we presented techniques of formal reconstruction of parallel structures implicitly specified by *vice versa* or similar operators. We addressed the problem by a domain-independent analysis method that uses deep semantics and contextually enhanced representations, exploits recasting rules to accommodate linguistic variations into uniform expressions, and makes use of patterns to match parallel structure categories.

Although we dedicated a lot of effort to building a principled method, the success is limited with respect to the generality of the problem: in some cases, the scope of reconstruction overarches entire paragraphs and deciding about the form requires considerable inferencing (cf. collection at http://www.chiasmus.com/). For our purposes, we are interested in expanding our method to other kinds of implicit structures in the tutorial context, for example, interpretations of references to analogies, in the case of which structure accommodation and swapping related items should also be prominent parts.

References

C. Benzmüller, A. Fiedler, M. Gabsdil, H. Horacek, I. Kruijff-Korbayová, M. Pinkal, J. Siekmann, D. Tsovaltzi, B.Q. Vo, and M. Wolska. 2003. A Wizard-of-Oz experiment for tutorial dialogues in mathematics. In *Supplementary Proceedings of the 11th Conference on Artifi cial Intelligence in Education (AIED-03); Vol. VIII. Workshop on Advanced Technologies for Mathematics Education*, pages 471–481, Sydney, Australia.

P. Koehn. 2002. Europarl: A multilingual corpus for evaluation of machine translation, Draft, Unpublished.

P. Sgall, E. Hajičová, and J. Panevová. 1986. *The meaning of the sentence in its semantic and pragmatic aspects*. Reidel Publishing Company, Dordrecht, The Netherlands.

M. Wolska, B.Q. Vo, D. Tsovaltzi, I. Kruijff-Korbayová, E. Karagjosova, H. Horacek, M. Gabsdil, A. Fiedler, and C. Benzmüller. 2004. An annotated corpus of tutorial dialogs on mathematical theorem proving. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC-04)*, pages 1007–1010, Lisbon, Potugal.