

Multilingual Revision

Charles Callaway

Istituto per la Ricerca Scientifica e Tecnologica
Istituto Trentino di Cultura, Italy (ITC-IRST)
callaway@irst.itc.it

Abstract

Natural Language Generation has made great strides towards multilingual generation from large-scale knowledge sources. Meanwhile, current research in revision has vastly improved the quality of text that NLG systems produce. However, to-date there has been no attempt at combining revision and multilingual NLG. This paper presents research in multilingual revision, the last major pipelined NLG component to be studied from a multilingual perspective. We describe the linguistic difficulties in achieving multilingual revision, review recent work, and present an implemented framework for multilingual revision rules.

1 Introduction

The quality of initial, unimproved text produced by explanation generation systems has been notoriously poor (Lester and Porter, 1997). Recent work in revision (Dalianis and Hovy, 1993; Robin, 1994; Callaway and Lester, 1997; Harvey and Carberry, 1998; Shaw, 1998) has shown substantial progress towards qualitative and quantitative improvements in the text that explanation generators produce. By necessity, revision systems operate over deep linguistic structures rather than text strings from template generators. Indeed, the increase in variability and flexibility that deep generation systems provide is often touted as a major

advantage over simpler, more easily implemented template generators (Reiter, 1995).

Multilinguality from deep linguistic representations (Paris et al., 1995; Stede, 1996; Bateman and Sharoff, 1998; Scott, 1999; Kruijff et al., 2000) is generally considered to be one of the advantages that deep generation systems possess over templates (although this depends heavily on the definitions of deep and template methods, such as in (Deemter et al., 1999)). By applying multilingual lexica and grammars to a single initial knowledge base, multilingual generators hope to leverage reusable components to produce texts in multiple languages with substantially less work than implementing an equivalent number of monolingual template or deep generators. But to be used effectively and efficiently in a multilingual generation system, these reusable components must be designed from the start for that purpose. Similarly, one of the main motivations for multilingual revision is efficiency: A single formalism for revision rules can greatly increase the amount of resource sharing in a manner analogous to that of grammars.

Just as in monolingual explanation generation, unrevised multilingual text is in general quite undesirable. And while multilingual components have been created for sentence planning (Bateman and Sharoff, 1998), surface realization (Netzer and Elhadad, 1999) and lexical choice (Stede, 1996), no attempt has been made to combine research in revision and multilingual NLG.

This paper presents research in multilingual revision, the last major pipelined NLG component

to be studied from a multilingual perspective. We first present linguistic reasons for the difficulty of discovering “revision rules” in both monolingual and multilingual contexts. Next, we describe recent work in revision and then analyze the elements of revision rules to determine how each element affects the revision process from both a monolingual and multilingual perspective. Finally, we synthesize a revision rule model that performs multilingual revisions under a common formalism in an implemented NLG system.

2 “Unrevising” a Corpus

Text generation systems are notorious for producing correct but low quality text. In contrast, human writers directly produce fluent text even in initial drafts. Because of this, it is very difficult to find a version of a naturally occurring text similar to the types of protosentences produced by systems today. (Meteer, 1990) initially explored this problem by having writers of scientific articles generate paraphrases in order to create a corpus for studying revision rules. The lack of such corpora is a disadvantage to implementers of revision systems because they lack the original source materials with which to create revision rules that would then allow them to achieve results comparable to the original, polished texts.

For example, consider the four text fragments in Figure 1, where (a) represents an excerpt from an original passage in Spanish, (b) represents its “unrevised” version, (c) represents the translation of (a) into English, and (d) is the “unrevision” of (c). While (a) and (c) are easily found in available corpora, the sentence structures in (b) and (d) cannot be found in either existing corpora or authors’ drafts. And yet these are precisely the types of texts needed to discover how simple “protosentences” are combined into larger clause structures.

Thus, one of the tasks that creators of revision systems must do is to collect example corpora from their generation domain and “unrevise” them to determine what types of revisions were performed by the original authors of the domain texts. The unrevised sentences should correspond to the types of protosentences produced by the initial discourse and sentence planners. The result of gathering examples from a set of corpora is a set of

revision rules applicable to that particular domain which can then be used by a revision component to refine and polish the final text produced by a generation system.

In a multilingual context, this corpus analysis must be repeated for each language, because different languages typically have different revision rules (even when the content is the same, as in the translations in Figure 1 (a) and (c), a revision system may end up reorganizing the text in a different manner because different languages and cultures may have different modes of presentation). Thus a system for multilingual revision must be able to handle disparate sets of revision rules as needed, or else resort to having multiple revision components, with the corresponding decrease in efficiency and increase in effort that such an architecture would bring.

A more efficient architectural approach would be to make different multilingual revision decisions for a set of languages by only replacing sets of revision rules. To be effective, the revision module must be orchestrated in such a way that all of the decision-making information can be determined by the revision rules themselves. Thus, a detailed feature analysis is necessary to determine the structure and variability of typical revision rules. In addition, all of the information necessary to choose between different revision rules must be closely tied to the incoming rhetorical structure and protosentences that comprise the sentence plan.

3 Aspects of Revision Operations

Although much research has described the architectural and linguistic aspects of revision, relatively little has been done to describe a *feature-based model* of revision upon which revision decisions can be based. However, most research describing implemented revision systems provides insight into the high-level features that were found necessary for particular projects or domains.

Dalianis’ work (Dalianis and Hovy, 1993) concentrated on the effect that a rule would have on the text in terms of carrying it out procedurally. His revision rules were the result of evaluating human revisions on a set of standardized textual propositions.

Pasé por alto el comentario. Me daba un poco de vergüenza explicar que hacía todo esto porque había conocido una vez a una chica pelirroja. Después le dije que sí, que pensaba aprender teatro. En realidad yo aborrecía a los actores. Eran demasiado extravertidos para mi gusto, y me impresionaban como gente que siempre se estaba saludando y abrazando y eran amigos de todo el mundo. No soporto a la gente que es amiga de todo el mundo, como los animadores de televisión.

(a)

- (1) Pasé por alto el comentario.
- (2) Me daba un poco de vergüenza explicar que hacía todo esto. ("porque")
- (3) Había conocido una vez a una chica pelirroja. ("después")
- (4) Le dije que sí, que pensaba aprender teatro.
- (5) En realidad, yo aborrecía a los actores.
- (6) Eran demasiado extravertidos para mi gusto. ("y")
- (7) Me impresionaban como gente que siempre se estaba saludando ("y")
- (8) Se estaba abrazando ("y")
- (9) Eran amigos de todo el mundo.
- (10) No soporto a la gente ("que")
- (11) La gente es amiga de todo el mundo. ("como")
- (12) Los animadores de televisión ["son amigas..."]

(b)

I skipped over her comment. I was a bit ashamed to explain that I had done all this because I had once met a redheaded girl. Later I told her that yes, I was thinking about studying theater, although in fact, I hated actors. They were too extroverted for my taste, and seemed like people who were always greeting and hugging each other as if they were friends with the whole world. I don't put up with someone who is friends with the whole world, like one of those television hosts.

(c)

- (1) I skipped over her comment.
- (2) I was a bit ashamed to explain I had done all this. ("because")
- (3) I had once met a red-headed girl. ("later")
- (4) I told her that yes, I was thinking about studying theater. ("although")
- (5) In fact, I hated actors.
- (6) They were too extroverted for my taste. ("and")
- (7) They seemed like people who were always greeting each other. ("and")
- (8) They were always hugging each other. ("as if")
- (9) They were friends with the whole world.
- (10) I don't put up with someone. ("who")
- (11) The person is friends with the whole world. ("like")
- (12) One of those television hosts ["is friends with..."]

(d)

Figure 1: "Unrevised" Spanish

- *Aggregation*: Operators which merge two previously separate clauses into one.
- *Ordering*: Operators that reverse the external ordering of clauses (migration) or their internal ordering (linear precedence).
- *Casting*: Operators that alter or enforce the regularity of syntactic structures over multiple clauses.
- *Parsimony*: Operators that prefer fewer overall words in a clause or other numeric quantities such as depth of prepositional phrase or relative clause embeddings.

Shaw's CASPER system of revision operators (Shaw, 1998) focuses on the syntactic dependency notion of hypotactic vs. paratactic operators. CASPER functions in the domain of medical report generation where Shaw noted that clause aggregations could frequently be classified on the basis of

dependency. In both cases, the redundant element is deleted from one of the clauses.

- *Hypotactic*: Operators which take two sentences, a base and a modifier, convert the modifier into a dependent clause, and then attach it to the base sentence.
- *Paratactic*: Operators which attach two sentences at the same dependency level, such as with *and* or *or* coordination.

By far the most explicitly organized and classified set of revision operators is described in the work on STREAK (Robin, 1994, Appendix A), a system for writing summary descriptions of basketball games in English. Motivated by (but not implemented using) a Tree Adjoining Grammar approach, Robin makes the following classifications of revision operators:

- *Monotonic*: Operators which leave the base

syntactic structure intact and result only in the attachment of a new phrase or clause to an unmodified existing structure. Examples include **Adjoin**, **Absorb**, **Conjoin**, and **Append** operators.

- *Nonmonotonic*: Operators which break up the base syntactic structure in various ways before attaching a new phrase or clause. Examples include **Recast**, **Adjunctization**, **Nominalization**, **Demotion**, and **Promotion** operators.
- *Side Transformations*: Operators that reduce redundant lexemes left over from previous revision operations. Examples include **Reference Adjustment**, **Argument Control**, **Ellipsis**, **Scope Marking**, **Ordering Adjustment**, and **Lexical Adjustment** operators.

Robin lists 18 distinct types of **adjoin** operators alone, organized according to the syntactic type of the phrase to be adjoined and the syntactic type and position of its attachment location. To illustrate, two of these subclasses of the adjoin operator are shown here:

- *Adjoin Relative Clause to Bottom-Level Nominal*: Attaches a relative clause to an immediately preceding noun phrase as in “to power them to a win over [(the Cavs), (who lost again)]”.
- *Adjoin Relative Clause to Top-Level Nominal*: Attaches a relative clause to a preceding noun phrase which already has postmodifiers as in “to power them to [(a win over the Cavs), (that extended their streak)]”.

However, there are a number of other classifications of revision operators not covered by these three approaches. In addition, these systems are not multilingual in nature and their corpora analyses covered only revisions in the English language. In order to fully understand the revision process both linguistically and computationally, as well as to ensure that this understanding is consistent across languages, it is important to discover and classify as many aspects of revision operators as possible. Some of the additional aspects we have

found in our efforts to build a multilingual revision component are as follows:

- *Rhetorical Type*: Most revision systems assume some underlying theory of discourse structure, such as RST (Mann and Thompson, 1987). These theories define particular rhetorical types such as *greeting* or *persuade* which are used by revision systems to provide additional constraints when selecting revision operators. How these constraints are affected by multilingual text is unknown.
- *Perspective*: While revision systems seem identical in purpose, there are actually great differences in what they expect to accomplish. For example, Dalianis and Hovy’s rules attempt to mimic human revision processes, Robin’s STREAK system attempts to build a single sentence, Shaw’s CASPER system focuses on eliminating redundancy, while other systems try to increase syntactic variety.
- *Syntactic structure*: Most revision systems start from sequences of protosentences and change a subset of those protosentences into different clauses or phrases. However, a complete set of target syntactic structures (especially identifying which of those structures overlap with the syntactic structures of other languages) has not yet been identified.
- *Attachment position*: When protosentences are converted into dependent circumstantial clauses (*e.g.*, when-clauses), a revision operator must choose to place it in front or at the end of a sentence. These operators must take into account whether a previous revision has already occupied one of those slots as well as whether a particular language allows a similar range of syntactic possibilities.
- *Depth of representation*: Robin’s STREAK system explicitly represented multiple levels of syntax, semantics, and pragmatics. More recent systems have shied away from this approach and favored shallower representations to increase efficiency.

- *Scope*: Revision operators can be very local and examine only adjacent protosentences or, at the expense of efficiency, examine protosentences slightly farther away (either to actually use them in aggregating or merely for additional context when making a more local decision).
- *New lexicalizations*: Most sentence planners do not produce many of the discourse-level elements found in polished texts because they are not needed when generating protosentences. For instance, discourse markers (Vander Linden and Martin, 1995; Grote, 1998) are frequently used to show the relationships between individual clauses. If appropriate information were available, it would be possible for revision operators to add discourse markers as they perform clause aggregation.

These factors are important when building a revision system that is scalable in terms of size, genre and language.

4 Motivation vs. Action

Analyzing the structure of revision operators in this manner does not imply that any particular architecture is preferred over another. However, much like the STRIPS architecture (Fikes and Nilsson, 1972), our analysis of revision rules in a multilingual environment has shown that every revision operator can be broken down into two parts describing *when* a rule should be fired (motivation, expressed as triggering rules), and if it is fired, *what* are the effects of that rule on the original protosentences (action, expressed as a target syntactic modification).

- *Motivations*: The parts of a revision rule which deal with whether the rule is applicable and which differentiate it with respect to other rules. Aspects which are helpful in deciding applicability include *rhetorical type*, *perspective*, *syntactic structure*, *depth of representation*, *scope*, and if discourse markers are expected to be added, those features of the input which are salient.
- *Actions*: The parts of a revision rule that alter either the internal syntactic structure of

clauses as they are aggregated or the rhetorical relationship(s) between multiple clauses. Aspects which are useful include *monotonicity*, *dependency*, *effect*, *perspective*, *syntactic structure*, and *attachment position*.

This division is apparent despite the language of the text being revised. For example, because languages have syntactic structure and at the lowest level revisions affect that syntactic structure,¹ the decision to alter that structure implies that the revision rule knows which syntactic category it is going to change it to. However, the particular syntactic category might be different given a different language. Given exactly similar circumstances, an English revision rule might prefer to change a protosentence into a prepositional phrase while a Spanish revision rule might prefer a relative clause. In addition, some syntactic options are available in certain languages but not in others (Netzer and Elhadad, 1999).

There are frequent similarities between languages, however. For example, our corpora analyses have shown that coordination with “and” usually occurs in similar situations. Also, the revision rules we have devised for English and Spanish in our domain almost always share identical *motivations* even if they differ in their *actions*. Given the overall structural similarity of revision rules despite their differences in details, the goal of designing a system capable of efficient multilingual revision is then to devise a single architectural component capable of carrying out revision operations by swapping out sets of revision rules rather than creating a separate revision component for each distinct language.

In our experience, different languages have similar sets of motivations for when to apply revision rules and similar sets of actions that carry them out. However, since the *mapping* from the set of initial structures (which drive the analysis of the motivation components of the revision rules) provided by the sentence planner to the set of actions which produce the final structures is language-specific, it is appropriate to apply a functional

¹While revisions do ultimately alter morphological structure, they do so only indirectly. No revision rule to our knowledge either considers or implements decisions based on morphological information.

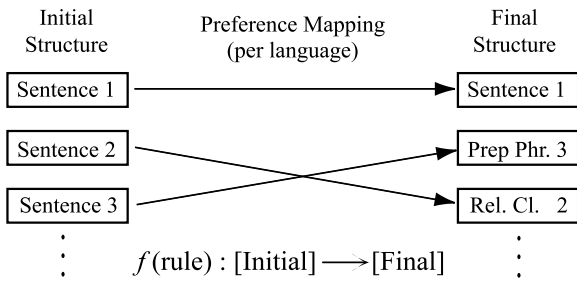


Figure 2: Revision Functions

model. Such a model (Figure 2) provides a separate function for each language which creates a set of revision rules by mapping from a set of motivations to a set of actions.

A multilingual revision architecture then could have a single set of language-neutral motivation criteria, a single set of language-neutral action effects, and a mapping function for each language desired (Figure 3). When the system is directed to switch from generating text for one language to another, the revision component needs only to switch in a new mapping function rather than using an entirely new revision component designed solely for the new language. Another benefit is that once sets of motivations and actions are encoded, it is relatively easy to adjust the effects of the revision module for different genres and styles.

5 Implementation

We started with an existing pipelined, multi-paragraph multilingual NLG system, STORYBOOK (Callaway, 2000; Callaway and Lester, 2002), that takes protosentences and revises them into text. While the major pipelined modules (discourse planner, sentence planner and surface realizer) were already capable of multilingual generation, the revision component, REVISOR (Callaway and Lester, 1997) only worked for English.

Our first step was to reorganize the English revision component following the architecture previously described. After analyzing the existing revision rules, we came up with a common set of 54 motivational triggers, 16 syntactic transformation actions, and a mapping between them that simulated the existing revision rule set. We then restructured the rule determination and application mechanisms before verifying that indeed the new

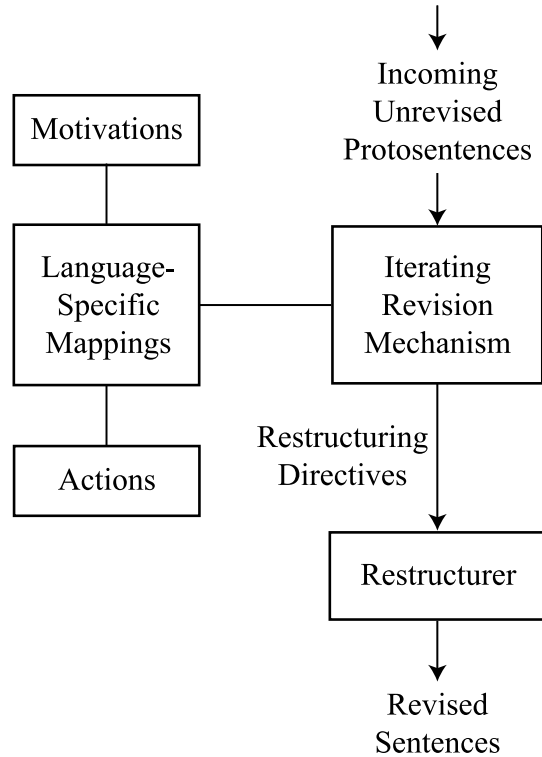


Figure 3: Abstract Revision Architecture

revision component made substantially the same revisions to the text as had the original revision component.

Next, we analyzed our translated Spanish corpus using the “unrevising” strategy described in Section 2. This yielded an additional 7 motivational triggers and 2 syntactic transformation actions for the Spanish corpus that were not accounted for in the original set of English revision rules. Afterwards, we created a mapping function from the appropriate motivational triggers found in both the English and Spanish set to the syntactic actions which we had found in the Spanish corpus.

For example, consider the sentences in Figure 4. REVISOR was initially capable of generating these simple protosentences in both English and Spanish, although initially only the English version worked well with revision, as it was keyed to look for specific information only present in the English output. We rewrote the English revision rules instead to look for more generic tags from the sentence planner indicating a particular protosentence had an intention such as IDENTITY

I don't put up with a person. [+animate-relative-clause = "who"]
The person is friends with the whole world. [+comparison = "like"]
Television hosts are friends with the whole world.

I don't put up with a person who is friends with the whole world,
like television hosts.

No soporto a la gente. [+animate-relative-clause = "que"]
La gente es amiga de todo el mundo. [+comparison = "como"]
Los animadores del televisio'n son amigas de todo el mundo.

No soporto a la gente que es amiga de todo el mundo, como los
animadores del televisio'n.

Figure 4: An example

or DESCRIPTION. We then extracted the syntactic aggregation operations, such as rules for generating relative clauses, from the various revision rules. Next, we wrote the mappings which combined the two (Figure 2), and checked to ensure that the original paragraph was regenerated. Finally, by substituting the appropriate mapping, we were able to also generate the revised Spanish version (Figure 4).

The result was an efficient revision system (execution time measured in tens of milliseconds) that produced the same or very similar revised paragraphs as the original English revision component as well as performing appropriate revisions to the Spanish text. This resulted in a substantial improvement in the amount of time required to create a traditional, standalone revision component for Spanish from scratch.

6 Conclusions

Efficient multilingual revision is possible within a single framework given a detailed analysis not only of a domain and its corpus and the types of revision operations conducted in each language, but by specifying the substructures of revision rules themselves. By isolating the differences in revision rules inherent in particular languages, we can increase the extent of language-neutral architectures and decrease the amount of work required to implement multilinguality in formerly monolingual systems.

A significant amount of the work involved in creating a multilingual revision system lies in conducting corpora analyses. One of the many problems faced by creators of revision systems is that a large amount of text must be examined before one can be confident that a sufficient number of revision rules has been uncovered. And because NLG systems to date are not capable of generating large scale texts, it is extremely difficult to test theories of revision rules. Having modular revision systems that can be easily altered for new languages, styles and genres will improve the quality of texts produced by NLG systems.

7 Acknowledgements

This work was supported by the PEACH and TICCA projects, funded by the Autonomous Province of Trento.

References

- J. Bateman and S. Sharoff. 1998. Multilingual grammars and multilingual lexicons for multilingual text generation. In *ECAI Workshop on Multilinguality in the Lexicon-II*, pages 1–8, Brighton, UK.
- Charles Callaway. 2000. *Narrative Prose Generation*. Ph.D. thesis, North Carolina State University, Raleigh, NC.
- Charles B. Callaway and James C. Lester. 1997. Dynamically improving explanations: A revision-based approach to explanation generation. In *Proceedings of the Fifteenth International Joint Conference*

- on *Artificial Intelligence*, pages 952–58, Nagoya, Japan.
- Charles B. Callaway and James C. Lester. 2002. Narrative prose generation. *Artificial Intelligence*, 139(2):213–252, August.
- Hercules Dalianis and Eduard Hovy. 1993. Aggregation in natural language generation. In *Proceedings of the Fourth European Workshop on Natural Language Generation*, Pisa, Italy.
- Kees van Deemter, Emiel Krahmer, and Mariët Theune. 1999. Plan-based vs. template-based nlg: A false opposition? In *Proceedings of the KI-99 Workshop Between Templates and Free Choice in Natural Language Generation*, Bonn, Germany.
- R. E. Fikes and N. J. Nilsson. 1972. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3–4):189–208.
- Brigitte Grote. 1998. Representing temporal discourse markers for generation purposes. In *Proceedings of the Discourse Relations and Discourse Markers Workshop*, pages 22–28, Montréal, Canada.
- Terrence Harvey and Sandra Carberry. 1998. Integrating text plans for conciseness and coherence. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, pages 512–518, August.
- Geert-Jan Kruijff, Elke Teich, John Bateman, Ivana Kruijff-Korbayová, Hana Skoumalová, Serge Sharoff, Lena Sokolova, Tony Hartley, Kamenka Staykova, and Jiří Hana. 2000. Multilinguality in a text generation system for 3 Slavic languages. In *COLING–2000: Proceedings of the 18th International Conference on Computational Linguistics*, Saarbruecken, Germany.
- James C. Lester and Bruce W. Porter. 1997. Developing and empirically evaluating robust explanation generators: The KNIGHT experiments. *Computational Linguistics*, 23(1):65–101.
- William C. Mann and Sandra A. Thompson. 1987. Rhetorical structure theory: A theory of text organization. Technical Report ISI/RS-87-190, USC/Information Sciences Institute, Marina del Rey, CA, June.
- Marie W. Meteer. 1990. *The Generation Gap: The Problem of Expressibility in Text Planning*. Ph.D. thesis, University of MA, February.
- Yael Dahan Netzer and Michael Elhadad. 1999. Bilingual Hebrew-English generation of possessives and partitives: Raising the input abstraction level. In *Proceedings of the 37th Meeting of the Association for Computational Linguistics*, pages 144–151, College Park, Maryland, June.
- Cécile L. Paris, Keith Vander Linden, Markus Fischer, Anthony Hartley, Lyn Pemberton, Richard Power, and Donia Scott. 1995. A support tool for writing multilingual instructions. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1398–1404, Montréal, Canada.
- Ehud Reiter. 1995. NLG vs. templates. In *Proceedings of the Fifth European Workshop on Natural Language Generation*, Leiden, The Netherlands.
- Jacques Robin. 1994. *Revision-Based Generation of Natural Language Summaries Providing Historical Background*. Ph.D. thesis, Columbia University, December.
- Donia R. Scott. 1999. The multilingual generation game: Authoring fluent texts in unfamiliar languages. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, Stockholm, Sweden.
- James Shaw. 1998. Clause aggregation using linguistic knowledge. In *Proceedings of the 9th International Workshop on Natural Language Generation*, pages 138–147, Niagara-on-the-Lake, Canada.
- Manfred Stede. 1996. *Lexical Semantics and Knowledge Representation in Multilingual Sentence Generation*. Ph.D. thesis, University of Toronto, Toronto, Ontario.
- Keith Vander Linden and James H. Martin. 1995. Expressing rhetorical relations in instructional text: A case study of the purpose relation. *Computational Linguistics*, 21(1):29–57.