Evaluating the Portability of Revision Rules for Incremental Summary Generation

Jacques Robin
http://www.di.ufpe.br/~jr
jr@di.ufpe.br
Departamento de Informática, Universidade Federal de Pernambuco
Caixa Postal 7851, Cidade Universitária
Recife, PE 50732-970 Brazil

Abstract

This paper presents a quantitative evaluation of the portability to the stock market domain of the revision rule hierarchy used by the system STREAK to incrementally generate newswire sports summaries. The evaluation consists of searching a test corpus of stock market reports for sentence pairs whose (semantic and syntactic) structures respectively match the triggering condition and application result of each revision rule. The results show that at least 59% of all rule classes are fully portable, with at least another 7% partially portable.

1 Introduction

The project STREAK[1] focuses on the specific issues involved in generating short, newswire style, natural language texts that *summarize* vast amount of input tabular data in their historical context. A series of previous publications presented complementary aspects of this project: motivating corpus analysis in (Robin and McKeown, 1993), new revision-based text generation model in (Robin, 1993), system implementation and rule base in (Robin, 1994a) and empirical evaluation of the *robustness* and *scalability* of this new model as compared to the traditional single pass pipeline model in (Robin and McKeown, 1995). The present paper completes this series by describing a second, empirical, corpus-based evaluation, this time quantifying the *portability* to another domain (the stock market) of the revision rule hierarchy acquired in the sports domain and implemented in STREAK. The goal of this paper is twofold: (1) assessing the generality of this particular rule hierarchy and (2) providing a general, semi-automatic methodology for evaluating the portability of semantic and syntactic knowledge structures used for natural language generation. The results reveal that at least 59% of the revision rule hierarchy abstracted from the sports domain could also be used to incrementally generate the complex sentences observed in a corpus of stock market reports.

I start by providing the context of the evaluation with a brief overview of STREAK's revision-based generation model, followed by some details about the empirical acquisition of its revision rules from corpus data. I then present the methodology of this evaluation, followed by a discussion of its quantitative results. Finally, I compare this evaluation with other empirical evaluations in text generation and conclude by discussing future directions.

2 An overview of STREAK

The project STREAK was initially motivated by analyzing a corpus of newswire summaries written by professional sportswriters[2]. This analysis revealed four characteristics of summaries that challenge the capabilities of previous text generators: concise linguistic forms, complex sentences, optional and background facts opportunistically slipped as modifiers of obligatory facts and high paraphrasing power. By greatly increasing the number of content planning and linguistic realization options that the generator must consider, as well as the mutual constraints among them, these characteristics make generating summaries in a single pass impractical.

The example run given in Fig. 1 illustrates how STREAK overcomes these difficulties. It first generates a simple draft sentence that contains only the obligatory facts to include in any game report (location, date, game result and key player statistic). It then applies a series of revision rules[3], each one

[1] Surface Text Reviser Expressing Additional Knowledge.

[2] This 800,000 word corpus covers a whole NBA (National Basketball Association) season.

[3] In Fig. 1, the rule used is indicated above each re-

1. `Initial draft (basic sentence pattern):`
   "Dallas, TX – Charles Barkley *scored* 42 points Sunday as the Phoenix Suns defeated the Dallas Mavericks 123-97."
2. `Adjunctization of Created into Instrument:`
   "Dallas, TX – Charles Barkley **tied a season high with** 42 points Sunday as the Phoenix Suns defeated the Dallas Mavericks 123-97."
3. `Coordinative Conjoin of Clause:`
   "Dallas, TX – Charles Barkley tied a season high with 42 points **and Danny Ainge** added **21** Sunday as the Phoenix Suns defeated the Dallas Mavericks 123-97."
4. `Absorb of Clause in Clause as Result with Agent Control:`
   "Dallas, TX – Charles Barkley tied a season high with 42 points and **Danny Ainge came off the bench** to add 21 Sunday as the Phoenix Suns defeated the Dallas Mavericks 123-97."
5. `Nominalization with Ordinal Adjoin:`
   "Dallas, TX – Charles Barkley tied a season high with 42 points and Danny Ainge came off the bench to add 21 Sunday as the Phoenix Suns **handed** the Dallas Mavericks **their 13th straight home** defeat 123-97."
6. `Adjoin of Classifier to NP:`
   "Dallas, TX – Charles Barkley tied a season high with 42 points and Danny Ainge came off the bench to add 21 Sunday as the Phoenix Suns handed the Dallas Mavericks their **league worst** 13th straight home defeat 123-97."

Figure 1: Complex sentence generation through incremental revisions in STREAK

opportunistically adding a new fact[4] that either:

- Complements an already included fact (*e.g.*, revision of sentence 2 into 3).
- Justifies its relevance by providing its historical background (*e.g.*, revision of sentence 1 into 2).

Some of these revisions are *non-monotonic*, rewording[5] a draft fact to more concisely accommodate the additional fact (*e.g.*, revision of sentence 1 into 2). Popping additional facts from a priority stack, STREAK stops revising when the summary sentence reaches linguistic complexity limits empirically observed in the corpus (*e.g.*, 50 word long or parse tree of depth 10).

While STREAK generates only single sentences, those complex sentences convey as much information as whole paragraphs made of simple sentences, only far more fluently and concisely. This is illustrated by the 12 sentence paragraph[6] of Fig. 2, which paraphrases sentence 6 of Fig. 1. Because they express facts essentially independently of one another, such multi-sentence paragraphs are much easier to generate than the complex single sentences generated by STREAK.

vised sentence.

[4]Highlighted in bold in Fig. 1.

[5]In Fig. 1, words that get deleted are italicized and words that get modified are underlined.

Charles Barkley scored 42 points. Those 42 points equal his best scoring performance of the season. Danny Ainge is a teammate of Barkley. They play for the Phoenix Suns. Ainge is a reserve player. Yet he scored 21 points. The high scoring performances by Barkley and Ainge helped the Suns defeat the Dallas Mavericks. The Mavericks played on their homecourt in Texas. They had already lost their 12 previous games there. No other team in the league has lost so many games in a row at home. The final score was 123-97. The game was played Sunday.

Figure 2: Paragraph of simple sentences paraphrasing a single complex sentence

3 Acquiring revision rules from corpus data

The rules driving the revision process in STREAK were acquired by reverse engineering[7] about 300 corpus sentences. These sentences were initially classified in terms of:

- The combination of domain concepts they expressed.
- The thematic role and top-level syntactic category used for each of these concepts.

[6]This paragraph was *not* generated by STREAK, it is shown here only for contrastive purposes.

[7]*i.e.*, analyzing how they could be incrementally generated through gradual revisions.

The resulting classes, called *realization patterns*, abstract the mapping from semantic to syntactic structure by factoring out lexical material and syntactic details. Two examples of realization patterns are given in Fig. 3. Realization patterns were then grouped into *surface decrement pairs* consisting of:

- A more complex pattern (called the *target* pattern).
- A simpler pattern (called the *source* pattern) that is structurally the closest to the target pattern among patterns with one less concept[8].

The structural transformations from source to target pattern in each surface decrement pair were then hierarchically classified, resulting in the revision rule hierarchy shown in Fig. 4-10. For example, the surface decrement pair $< R_b^2, R_b^1 >$, shown in Fig. 3, is one of the pairs from which the revision rule **Adjunctization of Range into Instrument**, shown in Fig. 10 was abstracted.

It involves displacing the **Range** argument of the source clause as an **Instrument** adjunct to accommodate a new verb and its argument. This revision rule is a sibling of the rule **Adjunctization of Created into Instrument** used to revise sentence 1 into 2 in STREAK's run shown in Fig. 1 (where the **Created** argument role "42 points" of the verb "to score" in 1 becomes an **Instrument** adjunct in 2).

The bottom level of the revision rule hierarchy specifies the *side revisions* that are orthogonal and sometimes accompany the restructuring revisions discussed up to this point. Side revisions do not make the draft more informative, but instead improve its style, conciseness and unambiguity. For example, when STREAK revises sentence (3) into (4) in the example run of Fig. 1, the **Agent** of the absorbed clause "Danny Ainge added 21 points" becomes *controlled* by the new embedding clause "Danny Ainge came off the bench" to avoid the verbose form:
? "Danny Ainge came off the bench for Danny Ainge to add 21 points".

4 Evaluation methodology

In the spectrum of possible evaluations, the evaluation presented in this paper is characterized as follows:

- Its object is the *revision rule hierarchy* acquired from the sports summary corpus. It thus does not directly evaluate the output of STREAK, but rather the special knowledge structures required by its underlying revision-based model.
- The particular property of this revision rule hierarchy that is evaluated is *cross-domain portability*: how much of it could be re-used to generate summaries in another domain, namely the stock market?
- The basis for this evaluation is *corpus data*[9]. The original sports summary corpus from which the revision rules were acquired is used as the 'training' (or acquisition) corpus and a corpus of stock market reports taken from several newswires is used as the 'test' corpus. This test corpus comprises over 18,000 sentences.
- The evaluation procedure is *quantitative*, measuring percentages of revision rules whose target and source realization patterns are observable in the test corpus. It is also *semi-automated* through the use of the corpus search tool CREP (Duford, 1993) (as explained below).

Basic principle As explained in section 3, a revision rule is associated with a list of surface decrement pairs, each one consisting of:

- A *source* pattern whose content and linguistic form match the triggering conditions of the rule (*e.g.*, R_b^1 in Fig. 3 for the rule **Adjunctization of Range into Instrument**).
- A *target* pattern whose content and linguistic form can be derived from the source pattern by applying the rule (*e.g.*, R_b^2 in Fig. 3 for the rule **Adjunctization of Range into Instrument**).

This list of decrement pairs can thus be used as the *signature* of the revision rule to detect its usage in the test corpus. The needed evidence is the simultaneous presence of two test corpus sentences[10], each one respectively matching the source and target patterns of at least one element in this list. Requiring occurrence of the *source* pattern in the test corpus is necessary for the computation of conservative portability estimates: while it may seem that one target pattern alone is enough evidence, without the presence of the corresponding source pattern, one cannot rule out the possibility that, in the test domain, this target pattern is either a basic pattern or derived from another source pattern using another revision rule.

[8] *i.e.*, the source pattern expresses the same concept combination than the target pattern minus one concept.

[9] Only the corpus analysis was performed for both domains. The implementation was *not* actually ported to the stock market domain.

[10] In general, *not* from the same report.

Realization pattern R_b^2:

- Expresses the concept pair:
  <game-result(winner,loser,score), **streak(winner,aspect,result-type,length)**>.
- Is a *target* pattern of the revision rule `Adjunctization of Range into Instrument`.

winner	aspect		type	streak	length			score	game-result	loser
agent	action		affected/located		location			instrument		
proper	verb		NP		PP			PP		
		det	classifier	noun		prep		NP		
							det	number	noun	PP
Utah	**extended**	**its**	**win**	**streak**	**to 6 games**	**with**	a	99-84	triumph	over Denver
Boston	**stretching**	**its**	**winning**	**spree**	**to 9 outings**	**with**	a	118-94	rout	of Utah

Realization pattern R_b^1:

- Expresses the single concept <game-result(winner,loser,score)>.
- Is a *source* pattern of the revision rule `Adjunctization of Range into Instrument`.
- Is a surface decrement of pattern R_b^2 above.

winner			score	game-result	loser
agent	action		range		
proper	support-verb		NP		
		det	number	noun	PP
Chicago	claimed	a	128-94	victory	over New Jersey
Orlando	recorded	a	101-95	triumph	against New York

Figure 3: Realization pattern examples

Partially automating the evaluation The software tool CREP[11] was developed to partially automate detection of realization patterns in a text corpus. The basic idea behind CREP is to approximate a realization pattern by a regular expression whose terminals are words or parts-of-speech tags (POS-tags). CREP will then automatically retrieve the corpus sentences matching those expressions. For example, the CREP expression C_1 below approximates the realization pattern R_b^1 shown in Fig. 3:
(C_1) `TEAM 0= (claimed|recorded)@VBD 1- SCORE 0= (victory|triumph)@NN 0= (over|against)@IN 0= TEAM`

In the expression above, '`VBD`', '`NN`' and '`IN`' are the POS-tags for past verb, singular noun and preposition (respectively), and the sub-expressions '`TEAM`' and '`SCORE`' (whose recursive definitions are not shown here) match the team names and possible final scores (respectively) in the NBA. The CREP operators '`N=`' and '`N-`' (N being an arbitrary integer) respectively specify exact and minimal distance of N words, and '`|`' encodes disjunction.

[11] CREP was implemented (on top of FLEX, GNUS' version of LEX) and to a large extent also designed by Duford. It uses Ken Church's POS tagger.

Because a realization pattern abstracts away from lexical items to capture the mapping from concepts to syntactic structure, approximating such a pattern by a regular expression of words and POS-tags involves encoding each concept of the pattern by the disjunction of its alternative lexicalizations. In a given domain, there are therefore two sources of inaccuracy for such an approximation:

- Lexical ambiguity resulting in false positives by over-generalization.
- Incomplete vocabulary resulting in false negatives by over-specialization[12].

Lexical ambiguities can be alleviated by writing more context-sensitive expressions. The vocabulary can be acquired through additional exploratory CREP runs with expressions containing wild-cards for some concept slots. Although automated corpus search using CREP expressions considerably speeds-up corpus analysis, manual intervention remains

[12] This is the case for example of C_1 above, which is a simplification of the actual expression that was used to search occurrences of R_b^1 in the test corpus (*e.g.*, C_1 is missing "win" and "rout" as alternatives for "victory").

necessary to filter out incorrect matches resulting from imperfect approximations.

Cross-domain discrepancies Basic similarities between the finance and sports domains form the basis for the portability of the revision rules. In both domains, the core facts reported are statistics compiled within a standard temporal unit (in sports, one ballgame; in finance, one stock market session) together with streaks[13] and records compiled across several such units. This correspondence is, however, imperfect. Consequently, before they can track down usage of a revision rule in the test domain, the CREP expressions approximating the signature of the rule in the acquisition domain must be *adjusted* for cross-domain discrepancies to prevent false negatives. Two major types of adjustments are necessary: lexical and thematic.

Lexical adjustments handle cases of partial mismatch between the respective vocabularies used to lexicalize matching conceptual structures in each domain. (*e.g.*,, the verb *"to rebound from"* expresses the interruption of a streak in the stock market domain, while in the basketball domain *"to break"* or *"to snap"* are preferred since *"to rebound"* is used to express a different concept).

Thematic adjustments handle cases of partial differences between corresponding conceptual structures in the acquisition and test domains. For example, while in sports `game-result` involves antagonistic teams, its financial domain counterpart `session-result` concerns only a single indicator. Consequently, the sub-expression for the `loser` role in the example CREP expression C_1 shown before, and which approximates realization pattern R_b^1 for `game-result` (shown in Fig. 3), needs to become optional in order to also approximate patterns for `session-result`. This is done using the CREP operator ? as shown below:

(C_1*'*): `TEAM 0= (claimed|recorded)@VBD 1-SCORE 0= (victory|triumph)@NN 0= ((over|against)@IN 0= TEAM)?`

Note that it is the CREP expressions used to automatically retrieve test corpus sentence pairs attesting usage of a revision rule that require this type of adjustment and *not* the revision rule itself[14]. For example, the `Adjoin of Frequency PP to Clause` revision rule attaches a streak to a `session-result` clause *without* `loser` role *in exactly the same way* than it attaches a streak to a `game-result` *with* `loser` role. This is illustrated by the two corpus sentences below:

P_2^a: "The Chicago Bulls beat the Phoenix Suns 99 91 **for their 3rd straight win**"

P_2^t: "The Amex Market Value Index inched up 0.16 to 481.94 **for its sixth straight advance**"

Detailed evaluation procedure The overall procedure to test portability of a revision rule consists of considering the surface decrement pairs in the rule signature in order, and repeating the following steps:

1. Write a CREP expression for the acquisition *target* pattern.
2. Iteratively delete, replace or generalize sub-expressions in the CREP expression - to gloss over thematic and lexical discrepancies between the acquisition and test domains, and prevent false negatives - until it matches some test corpus sentence(s).
3. Post-edit the file containing these matched sentences. If it contains only false positives of the sought target pattern, go back to step 2. Otherwise, proceed to step 4.
4. Repeat step (1-3) with the *source* pattern of the pair under consideration. If a valid match can also be found for this source pattern, stop: the revision rule is portable. Otherwise, start over from step 1 with the next surface decrement pair in the revision rule signature. If there is no next pair left, stop: the revision rule is considered non-portable.

Steps (2,3) constitute a general, generate-and-test procedure to detect realization patterns usage in a corpus[15]. Changing one CREP sub-expression may result in going from too specific an expression with no valid match to either: (1) a well-adjusted expression with a valid match, (2) still too specific an expression with no valid match, or (3) already too general an expression with too many matches to be manually post-edited.

It is in fact always possible to write more context-sensitive expressions, to manually edit larger no-match files, or even to consider larger test corpora in the hope of finding a match. At some point however, one has to estimate, guided by the results of previous runs, that the likelihood of finding a match is too

[13] *i.e.*, series of events with similar outcome.

[14] Some revision rules do require adjustment, but of another type (*cf.* Sect. 5).

[15] And since most generators rely on knowledge structures equivalent to realization patterns, this procedure can probably be adapted to semi-automatically evaluate the portability of virtually any corpus-based generator.

small to justify the cost of further attempts. This is why the last line in the algorithm reads "considered non-portable" as opposed to "non-portable". The algorithm guarantees the validity of positive (*i.e.*, portable) results only. Therefore, the figures presented in the next section constitute in fact a *lower-bound* estimate of the actual revision rule portability.

5 Evaluation results

The results of the evaluation are summarized in Fig. 4-10. They show the revision rule hierarchy, with portable classes highlighted in bold. The frequency of occurrence of each rule in the acquisition corpus is given below the leaves of the hierarchy.

Some rules are *same-concept portable*: they are used to attach corresponding concepts in each domain (*e.g.*, **`Adjoin of Frequency PP to Clause`**, as explained in Sect. 4). They could be re-used "as is" in the financial domain. Other rules, however, are only *different-concept portable*: they are used to attach altogether different concepts in each domain. This is the case for example of **`Adjoin Finite Time Clause to Clause`**, as illustrated by the two corpus sentences below, where the added temporal adjunct (in bold) conveys a streak in the sports sentence, but a complementary statistics in the financial one:
T_3^a: "*to lead Utah to a 119-89 trouncing of Denver* **as the Jazz defeated the Nuggets for the 12th straight time at home.**"
T_3^t: "*Volume amounted to a solid 349 million shares* **as advances out-paced declines 299 to 218.**".

For different-concept portable rules, the left-hand side field specifying the concepts incorporable to the draft using this rule will need to be changed when porting the rule to the stock market domain. In Fig. 4-10, the arcs leading same-concept portable classes are full and thick, those leading to different-concept portable classes are dotted, and those leading to a non-portable classes are full but thin.

59% of all revision rule classes turned out to be same-concept portable, with another 7% different-concept portable. Remarkably, *all* eight top-level classes identified in the sports domain had instances *same-concept portable* to the financial domain, even those involving the most complex non-monotonic revisions, or those with only a few instances in the sports corpus. Among the bottom-level classes that distinguish between revision rule applications in very specific semantic and syntactic contexts, 42% are same-concept portable with another 10% different-concept portable. Finally, the correlation between high usage frequency in the acquisition corpus and portability to the test corpus is not statistically significant (*i.e.*, the hypothesis that the more common a rule, the more likely it is to be portable could *not* be confirmed on the analyzed sample). See (Robin, 1994b) for further details on the evaluation results.

There are two main stumbling blocks to portability: thematic role mismatch and side revisions. Thematic role mismatches are cases where the semantic label or syntactic sub-category of a constituent added or displaced by the rule differ in each domain (*e.g.*, **`Adjunctization of Created into Instrument`** *vs.* **`Adjoin of Affected into Instrument`**). They push portability from 92% down to 71%. Their effect could be reduced by allowing STREAK's reviser to manipulate the draft down to the surface syntactic role level (*e.g.*, in both corpora **`Created`** and **`Affected`** surface as object). Currently, the reviser stops at the thematic role level to allow STREAK to take full advantage of the syntactic processing front-end SURGE (Elhadad and Robin, 1996), which accepts such thematic structures as input.

Accompanying side revisions push portability from 71% to 52%. This suggests that the design of STREAK could be improved by keeping side revisions separate from re-structuring revisions and interleaving the applications of the two. Currently, they are integrated together at the bottom of the revision rule hierarchy.

6 Related work

Apart from STREAK, only three generation projects feature an empirical and quantitative evaluation: ANA (Kukich, 1983), KNIGHT (Lester, 1993) and IMAGENE (Van der Linden, 1993).

ANA generates short, newswire style summaries of the daily fluctuations of several stock market indexes from half-hourly updates of their values. For evaluation, Kukich measures both the conceptual and linguistic (lexical and syntactic) coverages of ANA by comparing the number of concepts and realization patterns identified during a corpus analysis with those actually implemented in the system.

KNIGHT generates natural language concept definitions from a large biological knowledge base, relying on SURGE for syntactic realization. For evaluation, Lester performs a Turing test in which a panel of human judges rates 120 sample definitions by assigning grades (from A to F) for:

- Semantic accuracy (defined as "Is the definition adequate, providing correct information and focusing on what's important?" in the instructions provided to the judges).
- Stylistic accuracy (defined as "Does the definition use good prose and is the information it

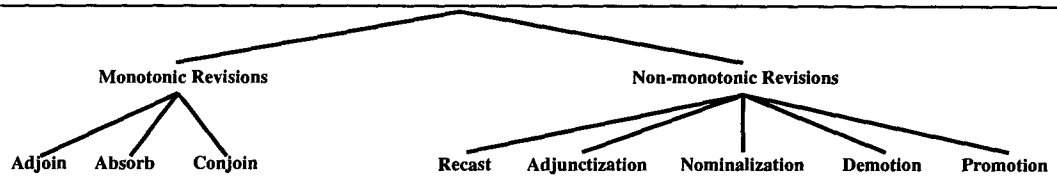

Figure 4: Revision rule hierarchy: top-levels

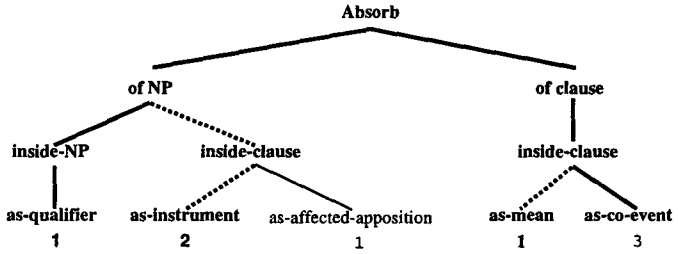

Figure 5: Absorb revision rule sub-hierarchy

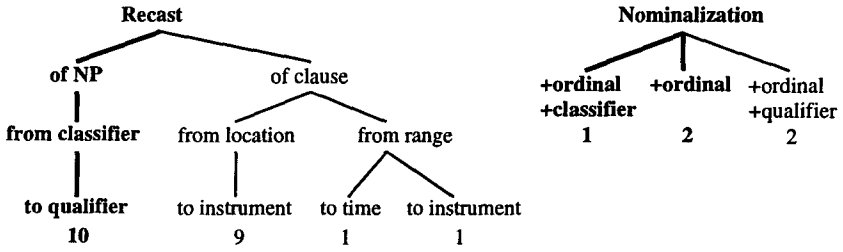

Figure 6: Recast and Nominalize revision rule sub-hierarchy

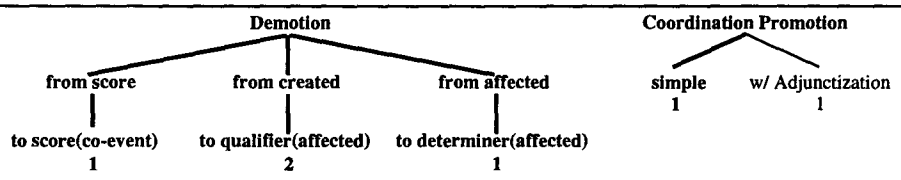

Figure 7: Demotion and Promotion revision rule sub-hierarchy

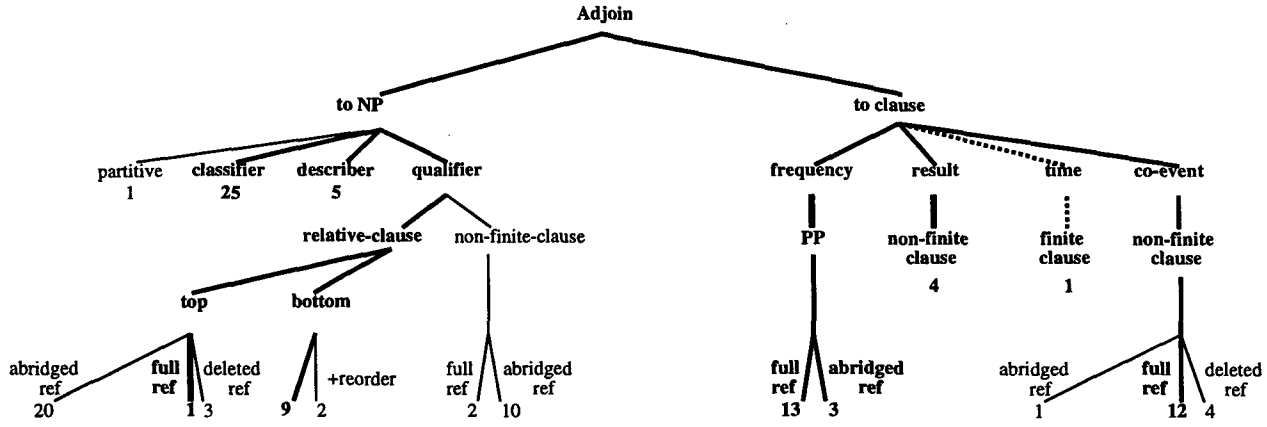

Figure 8: Adjoin revision rule sub-hierarchy

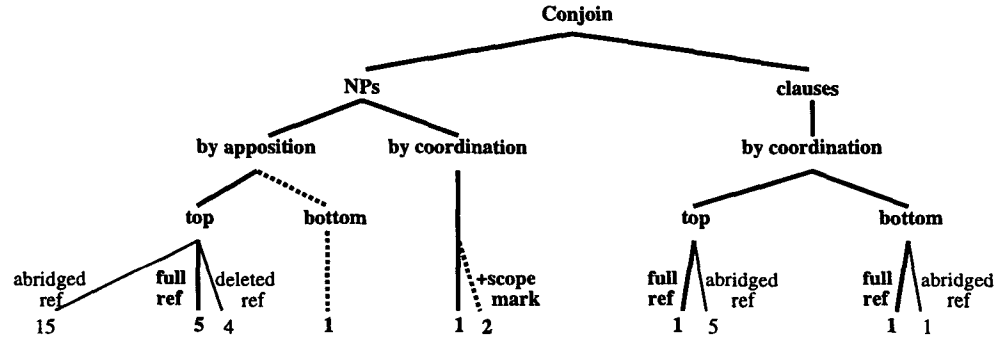

Figure 9: Conjoin revision rule sub-hierarchy

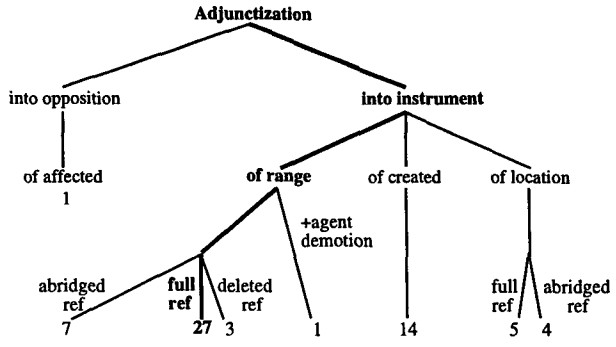

Figure 10: Adjunctization revision rule sub-hierarchy

	Object of Evaluation	Evaluated Properties	Empirical Basis	Evaluation Procedure
ANA	knowledge structures	conceptual coverage	textual corpus	manual
	"	linguistic coverage	"	"
KNIGHT	output text	semantic accuracy	human judges	manual
	"	stylistic accuracy	"	"
IMAGENE	output text	stylistic accuracy	textual corpus	manual
	"	stylistic robustness	"	"
STREAK	knowledge structures	**cross-domain portability**	textual corpus	semi-automatic
	"	same-domain robustness	"	"
	"	same-domain scalability	"	"

Figure 11: Empirical evaluations in language generation

conveys well organized" in the instructions provided to the judges).

The judges did not know that half the definitions were computer-generated while the other half were written by four human domain experts. Impressively, the results show that:

- With respect to semantic accuracy, the human judges could *not* tell KNIGHT apart from the human writers.
- While as a group, humans got statistically significantly better grades for stylistic accuracy than KNIGHT, the best human writer was singlehandly responsible for this difference.

IMAGENE generates instructions on how to operate household devices relying on NIGEL (Mann and Matthiessen, 1983) for syntactic realization. The implementation focuses on a very limited aspect of text generation: the realization of purpose relations. Taking as input the description of a pair <operation, purpose of the operation>, augmented by a set of features simulating the communicative context of generation, IMAGENE selects, among the many realizations of purpose generable by NIGEL (*e.g.,* fronted to-infinitive clause *vs.* trailing for-gerund clauses), the one that is most appropriate for the simulated context (*e.g.,* in the context of several operations sharing the same purpose, the latter is preferentially expressed *before* those actions than after them). IMAGENE's contextual preference rules were abstracted by analyzing an acquisition corpus of about 300 purpose clauses from cordless telephone manuals. For evaluation, Van der Linden compares the purpose realizations picked by IMAGENE to the one in the corresponding corpus text, first on the acquisition corpus and then on a test corpus of about 300 other purpose clauses from manuals for other devices than cordless telephones (ranging from clock radio to automobile). The results show a 71% match on the acquisition corpus[16] and a 52% match on the test corpus.

The table of Fig. 11 summarizes the difference on both goal and methodology between the evaluations carried out in the projects ANA, KNIGHT, IMAGENE and STREAK. In terms of goals, while Kukich and Lester evaluate the coverage or accuracy of a particular *implementation*, I instead focus on three properties inherent to the use of the revision-based generation *model* underlying STREAK: robustness (how much of other text samples from the same domain can be generated without acquiring new knowledge?) and scalability (how much more new knowledge is needed to fully cover these other samples?) discussed in (Robin and McKeown, 1995), and portability to another domain in the present paper. Van der Linden does a little bit of both by first measuring the stylistic accuracy of his system for a very restricted sub-domain, and then measuring how it degrades for a more general domain.

In itself, measuring the accuracy and coverage of a particular implementation in the sub-domain for which it was designed brings little insights about what generation approach should be adopted in future work. Indeed, even a system using mere canned text can be very accurate and attain substantial coverage if enough hand-coding effort is put into it. However, all this effort will have to be entirely duplicated each time the system is scaled up or ported to a new domain. Measuring how much of this effort duplication can be avoided when relying on revision-based generation was the very object of the three evaluations carried in the STREAK project.

[16] This imperfect match on the acquisition corpus seems to result from the *heuristic* nature of IMAGENE's stylistic preferences: individually, none of them needs to apply to the whole corpus.

In terms of methodology, the main originality of these three evaluations is the use of CREP to partially automate reverse engineering of corpus sentences. Beyond evaluation, CREP is a simple, but general and very handy tool that should prove useful to speed-up a wide range of corpora analyses.

7 Conclusion

In this paper, I presented a quantitative evaluation of the portability to the stock market domain of the revision rule hierarchy used by the system STREAK to incrementally generate newswire sports summaries. The evaluation procedure consists of searching a test corpus of stock market reports for sentence pairs whose (semantic and syntactic) structures respectively match the triggering condition and application result of each revision rule. The results show that at least 59% of all rule classes are fully portable, with at least another 7% partially portable.

Since the sports domain is not closer to the financial domain than to other quantitative domains such as meteorology, demography, business auditing or computer surveillance, these results are very encouraging with respect to the general cross-domain reusability potential of the knowledge structures used in revision-based generation. However, the present evaluation concerned only one type of such knowledge structures: revision rules. In future work, similar evaluations will be needed for the other types of knowledge structures: content selection rules, phrase planning rules and lexicalization rules.

Acknowledgements

Many thanks to Kathy McKeown for stressing the importance of empirically evaluating STREAK. The research presented in this paper is currently supported by CNPq (Brazilian Research Council) under post-doctoral research grant 150130-95.3. It started out while I was at Columbia University supported by of a joint grant from the Office of Naval Research, by the Advanced Research Projects Agency under contract N00014-89-J-1782, by National Science Foundation Grants IRT-84-51438 and GER-90-2406, and by the New York State Science and Technology Foundation under this auspices of the Columbia University CAT in High Performance Computing and Communications in Health Care, a New York State Center for Advanced Technology.

References

Duford, D. 1993. CREP: a regular expression-matching textual corpus tool. Technical Report CU-CS-005-93. Computer Science Department, Columbia University, New York.

Elhadad, M. and Robin, J. 1996. An overview of SURGE: a re-usable comprehensive syntactic realization component. Technical Report 96-03. Computer Science and Mathematics Department, Ben Gurion University, Beer Sheva, Israël.

Kukich, K. 1983. Knowledge-based report generation: a knowledge engineering approach to natural language report generation. PhD. Thesis. Department of Information Science. University of Pittsburgh.

Lester, J.C. 1993. Generating natural language explanations from large-scale knowledge bases. PhD. Thesis. Computer Science Department, University of Texas at Austin.

Mann, W.C. and Matthiessen, C. M. 1983. NIGEL: a systemic grammar for text generation. Research Report RR-83-105. ISI. Marina Del Rey, CA.

Robin, J. and McKeown, K.R. 1993. Corpus analysis for revision-based generation of complex sentences. In *Proceedings of the 11th National Conference on Artificial Intelligence*, Washington DC. (AAAI'93).

Robin, J. and McKeown, K.R. 1995. Empirically designing and evaluating a new revision-based model for summary generation. *Artificial Intelligence*. Vol 85. *Special Issue on Empirical Methods*. North-Holland.

Robin, J. 1993. A revision-based generation architecture for reporting facts in their historical context. In *New Concepts in Natural Language Generation: Planning, Realization and System*. Horacek, H. and Zock, M., Eds. Frances Pinter.

Robin, J. 1994a. Automatic generation and revision of natural language summaries providing historical background In *Proceedings of the 11th Brazilian Symposium on Artificial Intelligence*, Fortaleza, Brazil. (SBIA'94).

Robin, J. 1994b. Revision-based generation of natural language summaries providing historical background: corpus-based analysis, design, implementation and evaluation. PhD. Thesis. Available as Technical Report CU-CS-034-94. Computer Science Department, Columbia University, New York.

Van der Linden, K. and Martin, J.H. 1995. Expressing rhetorical relations in instructional texts: a case study of the purpose relation. *Computational Linguistics*, 21(1). MIT Press.