A Corpus Study of Negative Imperatives in Natural Language Instructions*

Keith Vander Linden†
Information Technology Research Institute
University of Brighton
Brighton BN2 4AT, UK
knvl@itri.brighton.ac.uk

Barbara Di Eugenio
Computational Linguistics
Carnegie Mellon University
Pittsburgh, PA, 15213 USA
dieugeni@andrew.cmu.edu

Abstract

In this paper, we define the notion of a preventative expression and discuss a corpus study of such expressions in instructional text. We discuss our coding schema, which takes into account both form and function features, and present measures of inter-coder reliability for those features. We then discuss the correlations that exist between the function and the form features.

1 Introduction

While interpreting instructions, an agent is continually faced with a number of possible actions to execute, the majority of which are not appropriate for the situation at hand. An instructor is therefore required not only to prescribe the appropriate actions to the reader, but also to prevent the reader from executing the inappropriate and potentially dangerous alternatives. The first task, which is commonly achieved by giving simple imperative commands and statements of purpose, has received considerable attention in both the interpretation (e.g., (Di Eugenio, 1993)) and the generation communities (e.g., (Vander Linden and Martin, 1995)). The second, achieved through the use of *preventative expressions*, has received considerably less attention. Such expressions can indicate actions that the agent should not perform, or manners of execution that the agent should not adopt. An agent may be told, for example, "Do not enter" or "Take care not to push too hard".

* This work is partially supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant J19221 and by the Commission of the European Union Grant LRE-62009.

† After September 1, Dr. Vander Linden's address will be Dept. of Mathematics and Computer Science, Calvin College, Grand Rapids, MI 49546, USA.

Both of the examples just given involve negation ("do *not*" and "take care *not*"). Although this is not strictly necessary for preventative expressions (e.g., one might say "stay out" rather than "do not enter"), we will focus on the use of negative forms in this paper. We will use the following categorisation of explicit preventative expressions:

- negative imperatives proper (termed *DONT* imperatives). These are characterised by the negative auxiliary *do not* or *don't*.

  (1) *Your sheet vinyl floor may be vinyl asbestos, which is no longer on the market.* **Don't sand it or tear it up** *because this will put dangerous asbestos fibers into the air.*

- other preventative imperatives (termed *negTC* imperatives). These include *take care* and *be careful* followed by a negative infinitival complement, as in the following examples:

  (2) *To book the strip, fold the bottom third or more of the strip over the middle of the panel, pasted sides together,* **taking care not to crease the wallpaper sharply at the fold.**

  (3) *If your plans call for replacing the wood base molding with vinyl cove molding,* **be careful not to damage the walls** *as you remove the wood base.*

The question of interest for us is under which conditions one or the other of the surface forms is chosen. We are currently using this information to drive the generation of warning messages in the DRAFTER system (Vander Linden and Di Eugenio, 1996). We will start by discussing previous work on negative imperatives, and by presenting an hypothesis to be explored. We will then describe the nature of our corpus and our coding schema,

detailing the results of our inter-coder reliability tests. Finally, we will describe the results of our analysis of the correlation between function and form features.

2 Related work on Negative Imperatives

While instructional text has sparked much interest in both the semantics/pragmatics community and the computational linguistics community, little work on preventative expressions, and in particular on negative imperatives, has been done. This lack of interest in the two communities has been in some sense complementary.

In semantics and pragmatics, negation has been extensively studied (cf. Horn (1989)). Imperatives, on the other hand, have not (for a notable exception, see Davies (1986)).

In computational linguistics, on the other hand, positive imperatives have been extensively investigated, both from the point of view of interpretation (Vere and Bickmore, 1990; Alterman et al., 1991; Chapman, 1991; Di Eugenio, 1993) and generation (Mellish and Evans, 1989; McKeown et al., 1990; Paris et al., 1995; Vander Linden and Martin, 1995). Little work, however, has been directed at negative imperatives. (for exceptions see the work of Vere and Bickmore (1990) in interpretation and of Ansari (1995) in generation).

3 A Priori Hypotheses

Di Eugenio (1993) put forward the following hypothesis concerning the realization of preventative expressions. In this discussion, S refers to the instructor (speaker / writer) who is referred to with feminine pronouns, and H to the agent (hearer / reader), referred to with masculine pronouns:

- **DONT imperatives.** A *DONT* imperative is used when S expects H to be *aware* of a certain choice point, but to be likely to choose the *wrong* alternative among many -- possibly infinite -- ones, as in:

  (4) *Dust-mop or vacuum your parquet floor as you would carpeting.* **Do not scrub or wet-mop the parquet.**

  Here, H is aware of the choice of various cleaning methods, but may choose an inappropriate one (i.e., scrubbing or wet-mopping).

- **Neg-TC imperatives.** In general, *neg-TC* imperatives are used when S expects H to *overlook* a certain choice point; such choice point may be identified through a possible side effect that the wrong choice will cause. It may, for example, be used when H might execute an action in an undesirable way. Consider:

  (5) *To make a piercing cut, first drill a hole in the waste stock on the interior of the pattern. If you want to save the waste stock for later use, drill the hole near a corner in the pattern.* **Be careful not to drill through the pattern line.**

  Here, H has some choices as regards the exact position where to drill, so S constrains him by saying *Be careful not to drill through the pattern line.*

So the hypothesis is that H's *awareness* of the presence of a certain choice point in executing a set of instructions affects the choice of one preventative expression over another. This hypothesis, however, was based on a small corpus and on intuitions. In this paper we present a more systematic analysis.

4 Corpus and coding

Our interest is in finding correlations between features related to the *function* of a preventative expression, and those related to the *form* of that expression. Functional features are the semantic features of the message being expressed and the pragmatic features of the context of communication. The form feature is the grammatical structure of the expression. In this section we will start with a discussion of our corpus, and then detail the function and form features that we have coded. We will conclude with a discussion of the inter-coder reliability of our coding.

4.1 Corpus

The raw instructional corpus from which we take all the examples we have coded has been collected opportunistically off the internet and from other sources. It is approximately 4 MB in size and is made entirely of written English instructional texts. The corpus includes a collection of recipes (1.7 MB), two complete do-it-yourself manuals (RD, 1991; McGowan and R. DuBern, 1991) (1.2 MB)[1], a set of computer games instructions, the Sun Open-windows on-line instructions, and a collection of administrative application forms. As a

[1]These do-it-yourself manuals were scanned by Joseph Rosenzweig.

collection, these texts are the result of a variety of authors working in a variety of instructional contexts.

We broke the corpus texts into expressions using a simple sentence breaking algorithm and then collected the negative imperatives by probing for expressions that contain the grammatical forms we were interested in (e.g., expressions containing phrases such as "don't" and "take care"). The first row in Table 1 shows the frequency of occurrence for each of the grammatical forms we probed for. These grammatical forms, 1175 occurrences in all, constitute 2.5% of the expressions in the full corpus. We then filtered the results of this probe in two ways:

1. When the probe returned more than 100 examples for a grammatical form, we randomly selected around 100 of those returned. We took all the examples for those forms that returned fewer than 100 examples. The number of examples that resulted is shown in row 2 of Table 1 (labelled "raw sample").

2. We removed those examples that, although they contained the desired lexical string, did not constitute negative imperatives. This pruning was done when the example was not an imperative (e.g., "If you **don't** see the Mail Tool window ...") and when the example was not negative (e.g., "Make sure to lock the bit tightly in the collar."). The number of examples which resulted is shown in row 3 of Table 1 (labelled "final coding"). Note that the majority of the "make sure" examples were removed here because they were ensurative.

As shown in Table 1, the final corpus sample is made up of 239 examples, all of which have been coded for the features to be discussed in the next two sections.

4.2 Form

Because of its syntactic nature, the form feature coding was very robust. The possible feature values were: **DONT** — for the *do not* and *don't* forms discussed above; and **neg-TC** — for *take care*, *make sure*, *ensure*, *be careful*, *be sure*, *be certain* expressions with negative arguments.

4.3 Function Features

The design of semantic/pragmatic features usually requires a series of iterations and modifications. We will discuss our schema, explaining the reasons behind our choices when necessary. We coded for two function features: INTENTIONALITY and AWARENESS, which we will illustrate in turn using α to refer to the negated action. The conception of these features was inspired by the hypothesis put forward in Section 3, as we will briefly discuss below.

4.3.1 Intentionality

This feature encodes whether the agent consciously adopts the intention of performing α. We settled on two values, CON(scious) and UNC(onscious). As the names of these values may be slightly misleading, we discuss them in detail here:

CON is used to code situations where S expects H to intend to perform α. This often happens when S expects H to be aware that α is an alternative to the β H should perform, and to consider them equivalent, while S knows that this is not the case. Consider Ex. (4) above. If the negative imperative *Do not scrub or wet-mop the parquet* were not included, the agent might have chosen to *scrub* or *wet-mop* because these actions may result in deeper cleaning, and because he was unaware of the bad consequences.

UNC is perhaps a less felicitous name because we certainly don't mean that the agent may perform actions while being unconscious! Rather, we mean that the agent doesn't realise that there is a choice point It is used in two situations: when α is totally accidental, as in:

(6) *Be careful not to burn the garlic.*

In the domain of cooking, no agent would consciously burn the garlic. Alternatively, an example is coded as UNC when α has to be intentionally planned for, but the agent may not take into account a crucial feature of α, as in:

(7) *Don't charge – or store – a tool where the temperature is below 40 degrees F or above 105 degrees.*

While clearly the agent will have to intend to perform *charging* or *storing a tool*, he is likely to overlook, at least in S's conception, that temperature could have a negative impact on the results of such actions.

4.3.2 Awareness

This binary feature captures whether the agent is AWare or UNAWare that the consequences of α are bad. These features are detailed now:

	DONT		Neg-TC			
	don't	do not	take care	make sure	be careful	be sure
Raw Grep	417	385	21	229	52	71
Raw Sample	100	99	21	104	52	71
Final Coding	78	89	17	3	46	6
	167		72			

Table 1: Distribution of negative imperatives

UNAW is used when H is perceived to be unaware that α is bad. For example, Example (7) ("Don't charge – or store – a tool where the temperature is below 40 degrees F or above 105 degrees") is coded as UNAW because it is unlikely that the reader will know about this restriction;

AW is used when H is aware that α is bad. Example (6) ("Be careful not to burn the garlic") is coded as AW because the reader is well aware that burning things when cooking them is bad.

4.4 Inter-coder reliability

Each author independently coded each of the features for all the examples in the sample. The percentage agreement is 76.1% for intentionality and 92.5% for awareness. Until very recently, these values would most likely have been accepted as a basis for further analysis. To support a more rigorous analysis, however, we have followed Carletta's suggestion (1996) of using the K coefficient (Siegel and Castellan, 1988) as a measure of coder agreement. This statistic not only measures agreement, but also factors out chance agreement, and is used for nominal (or categorical) scales. In nominal scales, there is no relation between the different categories, and classification induces equivalence classes on the set of classified objects. In our coding schema, each feature determines a nominal scale on its own. Thus, we report the values of the K statistics for each feature we coded for.

If $P(A)$ is the proportion of times the coders agree, and $P(E)$ is the proportion of times that coders are expected to agree by chance, K is computed as follows:

$$K = \frac{P(A) - P(E)}{1 - P(E)}$$

Thus, if there is total agreement among the coders, K will be 1; if there is no agreement other than chance agreement, K will be 0. There are various ways of computing $P(E)$; according to Siegel and Castellan (1988), most researchers agree on the following formula, which we also adopted:

$$P(E) = \sum_{j=1}^{m} p_j^2$$

where m is the number of categories, and p_j is the proportion of objects assigned to category j.

The mere fact that K may have a value k greater than zero is not sufficient to draw any conclusion, though, as it must be established whether k is significantly different from zero. While Siegel and Castellan (1988, p.289) point out that it is possible to check the significance of K when the number of objects is large, Rietveld and van Hout (1993) suggest a much simpler correlation between K values and inter-coder reliability, shown in Figure 2.

For the form feature, the Kappa value is 1.0, which is not surprising given its syntactic nature. The function features, which are more subjective in nature, engender more disagreement among coders, as shown by the K values in Table 3. According to Rietveld and van Hout, the awareness feature shows "substantial" agreement and the intentionality feature shows "moderate" agreement.

Kappa Value	Reliability Level
.00 – .20	slight
.21 – .40	fair
.41 – .60	moderate
.61 – .80	substantial
.81 – 1.00	almost perfect

Table 2: The Kappa Statistic and Inter-coder Reliability

feature	K
INTENTIONALITY	0.51
AWARENESS	0.75

Table 3: Kappa values for function features

5 Analysis

In our analysis, we have attempted to discover and to empirically verify correlations between the

feature	χ^2	significance level
intentionality	51.4	0.001
awareness	56.9	0.001

Table 4: χ^2 statistic and significance levels

function features and the form feature. We did this by computing χ^2 statistics for the various functional features as they compared with form distinction between DONT and neg-TC imperatives. Given that the features were all two-valued we were able to use the following definition of the statistic, taken from (Siegel and Castellan, 1988):

$$\chi^2 = \frac{N(|AD - BC| - \frac{N}{2})^2}{(A+B)(C+D)(A+C)(B+D)}$$

Here N is the total number of examples and A-D are the values of the elements of the 2×2 contingency table (see Figure 5). The χ^2 statistic is appropriate for the correlation of two independent samples of nominally coded data, and this particular definition of it is in line with Siegel's recommendations for 2×2 contingency tables in which $N > 40$ (Siegel and Castellan, 1988, page 123). Concerning the assumption of independence, while it is, in fact, possible that some of the examples may have been written by a single author, the corpus was written by a considerable number of authors. Even the larger works (e.g., the cookbooks and the do-it-yourself manuals) are collections of the work of multiple authors. We felt it acceptable, therefore, to view the examples as independent and use the χ^2 statistic.

To compute χ^2 for the coded examples in our corpus, we collected all the examples for which we agreed on both of the functional features (i.e., intentionality and awareness). Of the 239 total examples, 165 met this criteria. Table 4 lists the χ^2 statistic and its related level of significance for each of the features. The significance levels for intentionality and awareness indicate that the features do correlate with the forms. We will focus on these features in the remainder of this section.

The 2×2 contingency table from which the intentionality value was derived is shown in Table 5. This table shows the frequencies of examples marked as conscious or unconscious in relation to those marked as DONT and neg-TC. A strong tendency is indicated to prevent actions the reader is likely to consciously execute using the DONT form. Note that the table entry for conscious/neg-TC is 0, indicating that there were no examples marked as both CON and neg-TC. Similarly, the neg-TC form is more likely to be used to prevent actions the reader is likely to execute unconsciously.

	Conscious	Unconscious	*Total*
DONT	61 (A)	45 (B)	106
neg-TC	0 (C)	59 (D)	59
Total	61	104	165 (N)

Table 5: Contingency Table for Intentionality

	Aware	Unaware	*Total*
DONT	3	103	106
neg-TC	32	27	59
Total	35	130	165

Table 6: Contingency Table for Awareness

In Section 3 we speculated that the hearer's awareness of the choice point, or more accurately, the writer's view of the hearer's awareness, would affect the appropriate form of expression of the preventative expression. In our coding, awareness was then shifted to awareness of bad consequences rather than of choices per se. However, the basic intuition that awareness plays a role in the choice of surface form is supported, as the contingency table for this feature in Table 6 shows. It indicates a strong preference for the use of the DONT form when the reader is presumed to be unaware of the negative consequences of the action to be prevented, the reverse being true for the use of the neg-TC form.

The results of this analysis, therefore, demonstrate that the intentionality and awareness features do co-vary with grammatical form, and in particular, support a form of the hypothesis put forward in Section 3.

6 Application

We have successfully used the correlations discussed here to support the generation of warning messages in the DRAFTER project (Paris and Vander Linden, 1996). DRAFTER is a technical authoring support tool which generates instructions for graphical interfaces. It allows its users to specify a procedure to be expressed in instructional form, and in particular, allows them to specify actions which must be prevented at the appropriate points in the procedure. At generation time, then, DRAFTER must be able to select the appropriate grammatical form for the preventative expression.

We have used the correlations discussed in this paper to build the text planning rules required to generate negative imperatives. This is discussed in more detail elsewhere (Vander Linden and Di Eugenio, 1996), but in short, we input our

coded examples to Quinlan's C4.5 learning algorithm (Quinlan, 1993), which induces a decision tree mapping from the functional features to the appropriate form. Currently, these features are set manually by the user as they are too difficult to derive automatically.

7 Conclusions

This paper has detailed a corpus study of preventative expressions in instructional text. The study highlighted correlations between functional features and grammatical form, the sort of correlations useful in both interpretation and generation. Studies such as this have been done before in Computational Linguistics, although not, to our knowledge, on preventative expressions. The point we want to emphasise here is a methodological one. Only recently have studies been making use of more rigorous statistical measures of accuracy and reproducibility used here. We have found the Kappa statistic critical in the definition of the features we coded (see Section 4.4).

We intend to augment and refine the list of features discussed here and hope to use them in understanding applications as well as generation applications. We also intend to extend the analysis to ensurative expressions.

References

Richard Alterman, Roland Zito-Wolf, and Tamitha Carpenter. 1991. Interaction, Comprehension, and Instruction Usage. Technical Report CS-91-161, Dept. of Computer Science, Center for Complex Systems, Brandeis University.

Daniel Ansari. 1995. Deriving Procedural and Warning Instructions from Device and Environment Models. Master's thesis, University of Toronto.

Jean Carletta. 1996. Assessing agreement on classification tasks: the kappa statistic. *Computational Lingustics*, 22(2).

David Chapman. 1991. *Vision, Instruction and Action.* Cambridge: MIT Press.

Eirlys Davies. 1986. *The English Imperative.* Croom Helm.

Barbara Di Eugenio. 1993. *Understanding Natural Language Instructions: a Computational Approach to Purpose Clauses.* Ph.D. thesis, University of Pennsylvania, December. Technical Report MS-CIS-93-91 (Also Institute for Research in Cognitive Science report IRCS-93-52).

Laurence Horn. 1989. *A Natural History of Negation.* The University of Chicago Press.

J. McGowan and editors R. DuBern. 1991. *Home Repair.* London: Dorlin Kingersley Ltd.

Kathleen R. McKeown, Michael Elhadad, Yumiko Fukumoto, Jong Lim, Christine Lombardi, Jacques Robin, and Frank Smadja. 1990. Natural language generation in COMET. In Robert Dale, Chris Mellish, and Michael Zock, editors, *Current Research in Natural Language Generation,* chapter 5. Academic Press.

Chris Mellish and Roger Evans. 1989. Natural language generation from plans. *Computational Linguistics,* 15(4):233–249, December.

Cécile Paris and Keith Vander Linden. 1996. Drafter: An interactive support tool for writing multilingual instructions. *IEEE Computer.* to appear.

Cécile 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,* August 20 25, Montréal, Canada, pages 1398–1404. Also available as ITRI report ITRI-95-11.

J. Ross Quinlan. 1993. *C4.5: Programs for Machine Learning.* Morgan Kaufmann.

1991. Reader's Digest New Complete Do-It-Yourself Manual.

T. Rietveld and R. van Hout. 1993. *Statistical Techniques for the Study of Language and Language Behaviour.* Mouton de Gruyter.

Sidney Siegel and N. John Castellan, Jr. 1988. *Nonparametric statistics for the behavioral sciences.* McGraw Hill.

Keith Vander Linden and Barbara Di Eugenio. 1996. Learning micro-planning rules for preventative expressions. In *Proceedings of the Eighth International Workshop on Natural Language Generation,* Herstmonceux, England, 13–15 June 1996, June. To appear.

Keith Vander Linden and James Martin. 1995. Expressing Local Rhetorical Relations in Instructional Text. *Computational Linguistics,* 21(1):29–57.

Steven Vere and Timothy Bickmore. 1990. A Basic Agent. *Computational Intelligence,* 6:41–60.