MACHINE TRANSLATION AS A TOOL IN SECOND LANGUAGE LEARNING

Don D. Anderson, Department of Defense

ABSTRACT

The current major Machine Translation (MT) evaluation effort, funded by the Advanced Research Projects Agency (ARPA), shows that when compared to expert human translators, MT systems perform only about 65% as well on the average. In this paper it is argued that despite their overall poor performance, MT software can be used as a powerful focal point to improve second language (L2) skills. The paper describes the evaluation of Computronics Corporation’s Targumatik (Hebrew-->English), a PC-based MT system running under DOS, and shows how each problem and potential obstruction to learning can be overcome by means of discovery procedures using a set of tools and procedures called the 'learning algorithm.'

KEY WORDS

Machine translation, evaluation, Hebrew language, computer assisted language learning, machine translation output problems.

INTRODUCTION

MT viewed as Information Technology

In his keynote address at the CALICO '92 International Symposium, Rüschoff (1993) refers to computational technologies for manipulating information as Information Technology (IT). By extension, the application of these technologies to language learning is termed "IT enhanced language learning." This implies that IT resources should be designed "to facilitate language learning and language acquisition processes" (Rüschoff 1993, 9). Furthermore, he distinguishes between the importance of using language as communication and using it in experimentation and research. This latter use of language by the learner should not be underestimated, he believes.
It seems clear that these views could encompass the use of Machine Translation (MT) in IT enhanced language learning, provided it could be shown that MT can be used to enhance language learning.

**MT state-of-the-art**

MT has existed at least since the 1960s. However, judging from the quality of the output, it might seem to the layman that very little progress has been made since then. In an effort to improve MT core technology, researchers have become increasingly interested in systematically evaluating MT software. Because it funds a number of major MT research efforts, the Advanced Research projects Agency (ARPA) has organized and sponsored a major effort in evaluating the current progress of both research and commercially available MT systems. Results of the evaluation for the third quarter of 1994 (White, et al. 1994) indicate that, on the average, MT systems perform only about 65% as well as expert human translators.

**Overview**

In this paper, it is argued that despite such relatively disappointing results, MT software can be used as a powerful focal point in second language (L2) learning. The paper describes the evaluation of a commercially available Hebrew→English PC-based MT system, Targumatik (Computronics Corporation), for translation output, human-user interface, and software problems, and shows how each problem which has potential for becoming an obstruction to learning can be overcome by means of discovery procedures using a set of tools and procedures called the "learning algorithm." The "Evaluation Methodology" section discusses an evaluation methodology for MT systems. In the section entitled "Targumatik Evaluation and Results," this methodology is used to evaluate Targumatik. Derivation of the learning algorithm is presented in the "Learning Algorithm" section, and in the section entitled "Applying the Learning Algorithm to Targumatik," it is shown how the learning algorithm for Targumatik can be used to address problems uncovered by the Targumatik evaluation.

**Overview of MT main design approaches**

In order to better understand where commercially available MT fits into the MT picture as a whole, it is important to realize that there are several approaches to translation that have been used or are still being used. Mainstream MT researchers generally recognize three general types of system: direct, transfer, and interlingua (cf. Goodman and Nirenburg 1991; Nirenburg, et al. 1992; Hutchins 1989; and Slocum 1988).

A typical direct system is characterized by the processes of replacement and adjustment. There will normally be a bilingual dictionary containing the replacement elements and various rules that govern choosing those elements, putting the words in the right order, and adding or deleting words where necessary.
The transfer approach involves whole sentence analysis, usually syntactic, of the source language during the first step. This analysis is used to replace the source language words by the target language words, but in context. Transfer systems typically involve an intermediate process that maps a source language meaning representation into the equivalent target language meaning representation.

In the interlingua approach, the meaning representation of the source language input is independent of any target language. Thus, once the meaning representation is generated, it should be theoretically possible to map it into any target language. Such systems are generally seen as the most complex of the three approaches because of the very complicated process of generating the meaning representation, or interlingua.

EVALUATION METHODOLOGY

Overview

While MT evaluation has become increasingly important in the past few years, there seems to be a general lack of agreement on evaluation methodology (cf. AMTA 1994; NSF 1992; Balkin, et al. 1991; and Neal, et al. 1992). In general, the majority of MT evaluation methods suggested focused only on the translation output and paid no attention to user interface, software problems, or documentation. Factors such as these, however, may be very important because a poorly designed interface, serious software problems, or poor or nonexistent documentation can possibly obscure whatever useful application a user might find for the system. For this paper, the evaluation and scoring system used is the percentage scale. In each of the three major evaluation areas, the higher the score, the better the translation, human-machine interface, or software is judged to be.

MT output evaluation criteria

Methodology

With respect to MT output, there are three general areas that researchers as a whole seem to agree need to be evaluated: adequacy (also called intelligibility), fluency (sometimes determined using error analysis), and informativeness (also called fidelity). Fluency is the degree to which the MT translation output corresponds to an English speaker’s mental model of fluent English. Adequacy is the degree to which meaning expressed in the expert human translation is present in the MT output, and informativeness is the degree to which the MT output provides needed information. These terms, which are generally accepted throughout the MT research community, are defined and used by ARPA, the major MT evaluation effort (cf. White, et al. 1994).
An ARPA evaluation compares machine translations to expert human translations of general news source texts in terms of the three areas previously defined. However, it does not use error analysis in its evaluations since it was felt that the cost would be prohibitive. The National Academy of Science's Automatic Language Processing Advisory Committee (ALPAC) (ALPAC 1966), the first major MT evaluation effort, was concerned mainly with content and whether the translated passage is understandable. ALPAC did not employ error analysis to arrive at its evaluation results. Other, more recent methodologies (cf. Flanagan 1994; NSF 1992; Isahara, et al. 1994; and Neal, et al. 1992) do not systematically evaluate adequacy or informativeness but do use some form of error analysis.

The ARPA definitions of the evaluation areas appear to overlap to some extent. The main difference between them seems to be in the evaluation method used. Adequacy evaluation compares bracketed fragments from the expert human translation to corresponding sections in the MT output, the evaluator scoring each comparison on a descending 5 to 1 scale: 5 means "all meaning" is present in the translation, while 1 indicates that "almost none or none of the meaning" is present. Fluency evaluation scores sentences, headlines, subheadings, and sentence fragments used by the author for stylistic effect on a 5 to 1 descending scale, where 5 is excellent and 1 is very poor. On the other hand, informativeness uses a set of six multiple choice questions derived from each human expert translation and applies them to the MT output. The evaluator answers the questions and tallies the results.

In deciding how to evaluate translation output for this study, a number of points were considered: (1) how the in 'or researchers approached the problem; (2) various evaluation areas; (3) what was feasible, given the available resources; and (4) what it was felt were the most objective methods. As a result of this analysis, an eclectic approach is adopted. This approach turns out to be a modification of the ARPA methodology. The paper assumes the definitions of fluency, adequacy, and informativeness as defined in White, et al. (1994). Error analysis is used in place of the criteria employed by ARPA for fluency. The metric for adequacy is retained. Since single unrelated sentences make up the corpus used with the Targumatiķ learning algorithm, the ARPA methodology for informativeness is clearly inappropriate in that it is not possible most of the time to derive six content questions from one corpus sentence. Hence, a modified ALPAC informativeness metric is used. The paper accepts the ALPAC notion that informativeness refers to how informative the original version is perceived to be after the MT output has been seen. However, in ALPAC, the higher the score, the worse is considered to be the translation. Therefore, to bring it in line with the other evaluation areas, the amount of points assessed is reversed so that the lower the score, the worse the translation is considered. Scoring for these three evaluation areas is discussed in the following section. The three metrics used in this study appear in Appendix A.
Scoring

Adequacy and informativeness

Using the metrics described above, the score for all corpus sentences is totalled and divided by the total number of sentences in the corpus. The result is the average sentence score. This is, in turn, divided by the highest possible score to obtain a percentage in order to facilitate comparison to evaluations in other areas. Thus, if there are 300 sentences in the corpus, for example, and the total score for adequacy was 98C, the average sentence score would be 3.27 (980/300). This result is divided by the high end of the scale, 5, to obtain 65.4% (.654), the final score.

Fluency

Scoring for this metric is different from the others because the sentence score represents the number of error points. Thus, the higher the score, the less fluent the sentence will be. The paper accepts the Quality Panel Evaluation Criteria (NSF 1992) practice of setting the maximum error score for any one sentence at 12. For this reason, the reciprocal of the percent score is the correct version of the score. Thus, if the total fluency score is 1200 and there are 300 sentences in the corpus, then the average sentence score is 4 (1200/300). This result is divided by 12, the total possible error points per sentence, and the answer is subtracted from 1 to obtain the final score of 66.7% (4/12 = .333; 1.0-.333 = .667). In this way, the fluency score represents how well the machine translation was done in this area.

Human-machine interface evaluation criteria

Methodology

Many researchers in the field list principles of sound interface design. Most researchers tacitly agree that the user-interface includes the documentation. For the Targumatik evaluation, a checklist developed by Ravden and Johnson (1989) was used. The checklist is divided into ten functional areas: visual clarity, consistency, compatibility, informative feedback, explicitness, appropriate functionality, flexibility and control, error prevention and correction, user guidance and support, and system usability problems. Appendix D lists these functional areas with their definitions. Each functional area has a list of checklist questions which are answered by assigning weights of 3 to 0 on a descending scale, where a weight of 3 indicates "always," a weight of 2 means "most of the time," etc. Checklist questions are fairly subjective as they deal with the user's reaction to the interface in actual use. In view of the subjective nature of the interface, however, a subjective evaluation technique is probably unavoidable and may even be desirable.
Questions which do not apply are not answered and not counted in the result.

**Scoring**

The weighted scores are totaled, the total is divided by the total possible score for each section and the result is a percentage which can be used in comparison to other evaluation areas. For example, there are 18 questions under Informative Feedback. If the total weighted score is 38, the final percentage score of 70.4% would be obtained by dividing 38 by the total possible score (18'3 = 54).

**Software problem evaluation criteria**

**Methodology**

For this study, a type of operational testing is used in which it is attempted to test the limits of the system in an operational environment and document the results. Typical questions that can be posed are: What are the range of values that the program will accept as input? What if the user inputs something 'de this range? Will it cause the system to crash? Software problems encountered are judged in terms of the impact they might have on the user and are categorized as to minor impact (score = 1), medium impact (score = 2), or major impact (score = 3). Two types of software problems can be distinguished: those resulting from a design flaw, and those resulting from software errors in the program code.

**Scoring**

Scoring for this section is similar to scoring for fluency in the MT output section: since 100% indicates that no problems were encountered and 0% indicates that the total number of problems encountered had a total weighted score of 100 or more, the reciprocal of the initial result expresses the error-free percentage score. Scoring calculation is as follows: the number of software problems in each weight category is multiplied by the weight for that category to obtain the total category weighted score. The category weighted scores are totaled to obtain the overall weighted error score and this result is subtracted from 100 to obtain the final percentage problem-free score. Thus, if there are 4 major impact problems, 5 medium impact problems, and 8 minor impact problems, the total weighted error score will be (4'3 = 12) + (5'2= 10) + (8'1 = 8) = 30. The final percentage problem-free score is 100-30=70%.

**TARGUMATIK EVALUATION AND RESULTS**

**General description of Targumatik and overview**
Targumatik is a PC-based Hebrew→English direct MT system that runs under DOS 3.0 or above. It requires an IBM or compatible 286 processor or better, a VGA screen, 640K RAM, and a hard disk with at least 3MB or more of available space. In order to print the Hebrew characters, a dot matrix printer (Epson or compatible with graphics capability), a laser printer (HP Laserjet or compatible), or a Deskjet printer is required. The software may be run under Windows as a non-Windows application. Notable features include the capability of updating the dictionary with additional words or expressions and translating either single sentences or entire text files. Moreover, the system comes with a set of keyboard overlays for Hebrew letters, and in Hebrew mode the keyboard is set up to match a standard Israeli Hebrew keyboard.

Targumatik evaluation and scoring followed the principles and techniques outlined in the Evaluation Methodology section, above. This section summarizes each of the major evaluation areas as applied to Targumatik and notes problem areas which can potentially adversely affect learning. Appendix B contains a more detailed account of the Targumatik MT output evaluation with illustrative examples.

MT output

The overall average MT output score for Targumatik was 59.4%. This figure was obtained by averaging the scores for each of the evaluation areas. As indicated in Figure 1, Targumatik scored 64.2% for fluency, 68.2% for adequacy, and 45.6% for informativeness. The most common error was the inability of the system to deal effectively with homographs. The second most common error was the mistranslation of prepositions and other words. Other errors include untranslated words, lack of agreement between subject and verb, incorrect placement of a lexical modifier, leaving out the correct form of "be" in interrogatives and predicate adjectives, incorrect translation of verb tense, leaving out the correct form of "do" in yes-no questions and negative sentences, inability to recognize proper nouns, and incorrect negation. The following sample sentence illustrates a number of these errors.

![Figure 1. Targumatik Output Evaluation](image-url)
1) /ló natáta lí hizdamnút ledaβer/
   [not yo-gave to-me a-chance to-speak]
   “You didn’t give me a chance to speak”
T: “Not (natáta?) me opportunity to talk”

The fluency evaluation sentence score is 10 (final percentage score: 16.7%) due to the following errors:

   missing subject pronoun (2)
   untranslated word (2)
   incorrect placement for negation (2)
   missing auxiliary, did (2)
   missing determiner, the (2)

The adequacy evaluation score is 3 (final percentage score: 60%). Four words are missing from a nine word sentence. However, a moderate amount of the meaning expressed in the human translation appears to be present in the MT output for this sentence.

The score for the informativeness evaluation 3, which translates to 33.3% for the final percentage score. It appears that by correcting sentence structure, words, and phrases, the program makes a moderate change in the reader’s impression of the meaning intended, although not so much as to change or reverse the meaning completely.

**Human-machine interface**

The overall average human-machine interface score for Targumatik of 50.4% was obtained by computing the average of the nine major evaluation areas.5 Figure 2 shows the comparative scores for these areas in a bar chart format. All of these evaluation areas are concerned primarily with how the user perceives the system rather than how the software translates an input sentence. Only one evaluation area deals with input errors, that of error prevention and correction, and would thus be of interest because of the potential to adversely affect learning. Coincidentally, this was the area in which Targumatik scored the lowest: 23%. The software contains virtually no edit checks or validation routines that can be run on the input to reduce the likelihood of errors in the output.

**Software problems**

Targumatik obtained an overall average score of 68% in the area of software problems. This figure resulted from computing the average of the scores in the areas of major impact (71.9), medium impact (62-5) and minor impact (65.6). A total of 20 software problems were encountered — three in the major impact area, 6 in the medium impact
area, and 11 in the minor impact area. Figure 3 presents a summary chart showing how Targumatik scored in the major evaluation areas. Four software problems - one in the major impact area and three in the medium impact area — were judged to have the potential to adversely affect learning. All of these problems are related to the Dictionary Add feature of the software.
When adding multi-word expressions into the dictionary, version 2.2 of the software appeared to accept them, but when it was later attempted to use the new expressions, the system was unable to find them. This was classified as major impact because there is only one way to add multi-word expressions and there are many expressions that Targumatik cannot translate.

In the Dictionary Add feature, data fields are restricted to a single screen. The nif'al form of the Hebrew verb (corresponding roughly to the English passive) causes a problem in that there is only one field for the English past tense. Thus, if the user inputs "was" + past participle, "were" is automatically excluded. This may result in a mistranslation in which subject and verb do not agree.

The other two problems also have to do with the Dictionary Add feature where important screen messages are displayed in Hebrew. Because they should be in English, they may be easily overlooked by a user with only Level 1 Hebrew language skills.

THE LEARNING ALGORITHM

Overview

At this point, the question is "How can MT be used to enhance language learning?" The learning algorithm addresses this question by defining a set of requirements, which include ancillary reference tools, and by establishing a set of iterative, step-by-step procedures which the student can follow. These requirements, ancillary tools, and procedures all work together to optimize the learning process through the use of discovery procedures. This section begins with a discussion of a learning algorithm development strategy, which includes defining the requirements and developing the set of step-by-step procedures. The Targumatik learning algorithm is outlined to conclude the section.

Defining the requirements

Defining the requirements for the learning algorithm includes defining selection criteria for the MT software, the set of ancillary tools, the text corpus, the language pair, the minimum student skill levels, and the establishment of target goals. Some of these tasks should be done in order. For example, defining the selection criteria for the MT software should be done before defining the selection criteria for the ancillary tools since it is probably easier to find ancillary tools that cover material in the MT dictionary and expressions than the reverse.

MT software selection
Selection criteria for the MT software should include, at least, the following:

- The bilingual dictionary of words and expressions should be in a general subject domain. It should be expandable and/or modifiable by the user. The software should, optionally, allow addition of specific domain vocabularies when and if desired.

- The character set for the L2 should be the L2 country standard character set, if there is one, or, at least, the one most commonly used.

- The keyboard layout for L2 should be the L2 country standard, if there is one, or, at least, the one most commonly used.

- The software should be evaluated to ensure that any major design or operational problems that could inhibit learning are noted so that they can be incorporated into the learning algorithm. The reviewer should also be aware of any major software problems that prevent normal operation of the software.

Selection of ancillary tools

The set of ancillary tools used in a learning algorithm are the L2 learner's resources for determining and correcting MT output errors. As such, they must include accurate information on the words and expressions in the MT bilingual dictionary. The kinds of information the learner will need in order to correctly use the step-by-step procedures includes information on verb forms, word and expression meanings, verb, adjective and noun inflection paradigms, and L2 sentence structure and syntax. This type of information can normally be found in verb declension books, reference grammars, bilingual dictionaries of words and expressions, and the like.

Text corpus selection

The text corpus is the set of L2 sentences that will be used in the learning algorithm. Some quality control should be exercised in their selection in order for the learner to derive the maximum benefit from the system.

- Text that includes a parallel professional English translation is necessary. If the text does not consist of individual sentences but of some type of connected narrative, the parallel translation should be aligned so that the learner can line up the L2 phrases with their L1 equivalents. This procedure will be necessary so that the learner can judge whether the L1 output from the MT system contains any errors.
• The subject domain, if not general, should, at least, match the subject domain of the MT bilingual dictionary. This is the least time-consuming approach. If a domain match cannot be located, much time will be needed to update the MT bilingual dictionary.

• The learning algorithm requires the learner to speak the L2 sentences aloud. For this reason, it is important to select a text corpus that represents spoken, conversational style.

• The text corpus should contain graded linguistic constructs, progressing from simple sentence structures to the more complex.

Language pair selection

It is important in the early stages of L2 acquisition using MT and the learning algorithm that the L2 → L1 language pair be used. This enables the learner to benefit from verification of the MT output based on his or her native speaker intuition. The specific problems presented by the L1 → L2 language pair for learners with lower skill levels should be addressed separately and are beyond the scope of this paper.

Criteria for student skill levels requirement

In general, the student who uses MT and a learning algorithm should have a minimum ILR skill level of S1-L0-R1-W1 or higher to derive maximum benefit from the system. The learning algorithm requires the student to read aloud, type, write cursively, and look up linguistic information in the L2. The closer the L2 is to English, the more this requirement can be relaxed. If the L2 uses the Roman alphabet, it will generally be easier for the student who is a native speaker of English to acquire rudimentary reading skills than if the L2 uses a non-Roman alphabet, such as Cyrillic or Hebrew, or if it uses ideograms, such as Chinese and Japanese do.

Setting target goals

Setting target goals involves identifying the subset of language skills which can be improved with this method and defining the learning goals. These processes will depend on whether the learning algorithm is used in a classroom situation or as self study, and whether or not the text corpus has an accompanying set of audio tapes. They will also depend, in part, on developing drills and carrying out periodic testing and evaluation of the student’s skill levels.
Developing the procedures

Development of the learning algorithm step-by-step procedures should involve identifying the functional steps and determining the order in which they should be applied, testing the procedures and modifying, if necessary, and evaluating to ensure that the description of each step can be clearly understood by the student.

The functional steps may be seen as an enhanced version of the process of evaluating the MT output where the goal of the enhancement is strictly pedagogical. The evaluation process could involve the following steps, for example:

- Selecting a text corpus suitable for testing the range of capabilities of the MT system
- Developing an evaluation plan which will indicate what is to be tested, in which order, and how it is to be evaluated
- Manually entering sentences or phrases one by one into the MT system, initiating the translation process, and noting the results
- Using native speaker intuition and L2 reference works to identify and correct the MT output errors
- Updating the MT bilingual dictionary of words and expressions, if necessary, retranslating the sentence or phrase, and observing the MT output for differences

To enhance these steps pedagogically, one might, for instance, add audio to the text corpus, speak the L2 sentence or phrase aloud several times while typing it into the system, or cursively write the sentence or phrase after analyzing the output for errors and determining why the errors were made. There are a number of other possibilities which should include developing a testing and evaluation scheme for the L2 structures, syntax, and pronunciation.

Deciding on the order in which the set of functional steps are performed is the next stage and will, most likely, parallel the natural order in which the MT evaluation steps are taken. For example, the L2 sentence or phrase must, obviously, be entered into the system before one can initiate the translation process. Similarly, initiation of the translation process must be done before one can identify the MT output errors and correct them.

Finally, while testing and evaluating the step-by-step procedures may be of little or no importance to an MT evaluator, it is important pedagogically.
The Targumatik learning algorithm

Requirements

• Language pair Hebrew->English should be used since the learner can benefit from verification of the translation output using native speaker intuition. English→Hebrew presents specific problems for learners with lower skill levels and should be addressed much later.

• Learners should have minimum ILR skill level of S1-L0-R1-W1 or higher to derive maximum benefit from this system. Targumatik expects input in Israeli Hebrew — i.e., without vowel diacritics (nikud) — and Hebrew words will have to be looked up in reference works. The learning algorithm also requires that the learner read sentences aloud as they are being typed into the system and write them cursively later. This clearly implies that a prior beginning skill level will be needed to work through the learning algorithm.

• The Targumatik learning algorithm requires that the learner use reference works to learn why the system mistranslated or failed to translate words or expressions. This study used two verb declension references (Halkin 1970, Tarmon and Uval 1991); three reference Hebrew grammars (Blumberg and Lewittes 1982, Yetiv 1973, and Birnbaum 1966); and two bilingual Hebrew-English dictionaries (Inbal 1992, and Lauden and Weinbach 1993).

• The Targumatik learning algorithm requires that the text used be a parallel corpus - i.e., each Hebrew sentence should have a corresponding English translation. Moreover, the text must contain some type of standard phonemic or phonetic transcription for each sentence, and the sentences in the corpus should be standard conversational Israeli Hebrew. Furthermore, the ensemble of the sentences should present graded linguistic structures. Finally, it is important that the domain of the text corpus match that of the systems bilingual dictionary. The text chosen for this study that meets the majority of these requirements is from Reif and Levinson (1965).

Step-by-step procedures

These procedures are patterned after a computer algorithm; that is, they are iterative and describe the steps necessary to solve a given problem. To work the algorithm, each step is followed in succession according to the instructions.

To begin, go to Targumatik's Files menu and select the Translations option. When the appropriate window comes up with further options, select the Translate Sentences by Hand option.
1. If you want to quit, or there are no more sentence pairs from the parallel corpus, go to step 17.

2. Type in the Hebrew sentence from a sentence pair, reading aloud several times; repeat the human English translation aloud.

3. Check the Hebrew sentence just typed for spelling errors (before hitting the Enter key!). If there are no errors, go to step 5.

4. Correct spelling errors by comparing your input with the original, spelling aloud the correctly spelled word in Hebrew. Repeat steps 3 and 4 until all words in the sentence are correctly spelled.

5. Run the Targumatik translation routine on the sentence you just typed and checked for spelling errors.

6. Check Targumatik’s translation for untranslated words. If there are none, or all untranslated words in Targumatik’s dictionary have been accounted for, go to step 12.

7. Look up untranslated words in printed reference works, noting information about structure, form, and part of speech.

8. Look up an untranslated word in Targumatik’s dictionary. If the word is found, go to step 11.

9. Add the word to Targumatik’s dictionary. Repeat steps 8 and 9 until all untranslated words not in Targumatik’s dictionary have been added.


11. Check in the printed reference material for homographs and note grammatical information, structure, and part of speech as appropriate, making notes to try to explain the problem. Repeat this step until all untranslated words in Targumatik’s dictionary and all mistranslated words from steps 12 and 13 have been accounted for.

12. Compare Targumatik’s translation of the Hebrew sentence with the human English translation in the parallel corpus. If there are no mistranslated or missing words or expressions, go to step 15.

13. Locate a Hebrew word or expression that was mistranslated.

14. Go to step 11.
15. Write cursive the Hebrew sentence you typed, checked for spelling errors, translated, and for which you corrected the translation; then copy the human translation in the parallel corpus.


17. Optional: write up a report with comments based on your notes.

18. End.

APPLYING THE LEARNING ALGORITHM TO TARGUMATIK

Overview

Application of the Targumatik learning algorithm to problems discovered as a result of the evaluation involves decisions regarding the potential capability of a problem to obstruct L2 learning. Each problem was analyzed to determine whether or not it was capable of obstructing learning. This section describes how the potential adverse effect of these problems can be neutralized or reversed through use of the learning algorithm and its inherent discovery procedures.

Summary of Targumatik problems

MT Output

The following MT output problems were judged to have the potential to adversely affect L2 acquisition:

- Inability to distinguish homographs
- Mistranslation of prepositions and other words
- Untranslated words or expressions
- Lack of agreement between subject and verb
- Incorrect placement of intensifiers
- Leaving out the correct form of "be" in interrogatives and predicate adjectives
- Leaving out the correct form of "do" in interrogatives and predicate adjectives
- Inability to recognize proper nouns
- Incorrect negation
Human-machine interface

Only one of the functional areas in the Targumnik human-machine interface evaluation seemed to have the potential to obstruct L2 learning:

• Error prevention and correction

Software

These Targumnik software problems were judged to have the potential to obstruct L2 learning:

• Inability to add expressions to dictionary

• Dictionary Add treatment of nif’al form of Hebrew verbs which allows only one form for translation into English passive

• Hebrew screen labels and messages in the Dictionary Add feature which are important but can be overlooked by the student with rudimentary L2 language skills

Applying the learning algorithm

This section describes how the learning algorithm satisfactorily answers each of the problems summarized in the section entitled "Summary of Targumnik problems." The learning algorithm anticipates that there will be problems with the typical MT package and, consequently, solving such problems is implicit in its structure and operation.

MT output problems

To illustrate how the learning algorithm handles the MT output problems outlined in the "Summary" section above, the Targumnik learning algorithm is applied to two typical text corpus sentences. The first is relatively simple and contains only one error. The second sentence is complex and illustrates translations containing several errors.

In the first example, the student will learn — in addition to spelling, writing, pronunciation, and reading aloud — about Hebrew homographs — words that are spelled identically but which have different functions and pronunciations:

1. /yéÓlánu Ḥnéy sfarím/
   [there-is to-us two books]
"We have two books"

T: "We have second books"

Note first that there is no gloss in the MT output, of course. This has been added for clarification purposes only. Steps 1 through 11 of the Targumatik learning algorithm will have already been performed at this point. During these steps the student will have learned Hebrew typing, spelling for the words in the sentence, pronunciation and reading aloud. Step 12 says to compare the translation with the human English translation from the parallel corpus and look for mistranslated or missing words or expressions. If there are none, skip to step 15. Otherwise, step 13, which is next, says to locate a Hebrew word or expression that was mistranslated.

The first thing the student should notice is that the MT output looks correct except for the use of "second" in place of "two." A logical place to start would be to look up "second" in one of the bilingual Hebrew-English dictionaries. The first entry for "second" in the Inbal (1992) User-Friendly English-Hebrew Hebrew English Dictionary — henceforth called Inbal — is /Óéni/. This word is spelled identically to /Ónéy/ — shin-nun-yod — when one takes into consideration that Israeli Hebrew does not use vowel diacritics. For this reason, the two words are homographs. Thus, the student understands why Targumatik made the error. Since there are no more errors in the MT output, the student is instructed in step 12 to go to step 15. This step instructs the student to write the Hebrew sentence cursively. Step 16 tells the student to go to step 1 where the option is to go to step 17 and write up a short report prior to ending in step 18.

The second example is considerably more complex. In this sentence, the student will learn about the following points of Hebrew grammar:

- possessive pronouns
- the construct state
- the English indefinite article in Hebrew
- BE in the present tense in Hebrew
- verb/noun homographs

2. /hú ló yadá Óemiryám betelavív/
   [he no knew that-Miriam [is]-in-Tel-Aviv]
   “He did not know that Miriam is in Tel Aviv”

T: “He not a knowledge of yeast in Tel Aviv”
Again, steps 1 through 11 of the algorithm will have already been performed. The student is now at step 12. Since there are mistranslated words, step 13 is next performed. Probably the most noticeable problem is that the MT output has a noun in place of the verb. The next step, step 14, instructs the student to go back to step 11. Here, the student is instructed to check in the printed reference material for homographs, etc. A good starting place might be to look up "knowledge" in one of the bilingual Hebrew-English dictionaries. Checking this entry in Inbal, the student will note that one of the definitions is /yedá/. Again, since Israeli Hebrew does not use vowel diacritics, these two words would be spelled identically — yod-dalet-ayin — and are, thus, considered homographs.

At this point, the verb "know" should be looked up in a verb declension book. The declension is found under the root form spelling, yod-dalet-ayin, in 201 Hebrew verbs by Halkin (1970). The student with rudimentary knowledge of Hebrew has only to find a match between the verb form in the text corpus Hebrew sentence and a verb form under this root. This is found under the kal form of the verb, past tense masculine singular. This confirms that /hú lá yadá/ means "he did not know." This takes care of the problem of DO deletion in the MT output. At this point, the student has learned that there is a nominal form of the verb, yod-dalet-ayin, that is spelled identically to the past masculine singular of the kal form of the verb, but pronounced differently: /yedá/. Since the Inbal entry for the Hebrew translation of "knowledge" includes the vowel diacritic marks, he or she will also know how the word is pronounced.

Following the instructions in steps 13 and 14, it appears that /Óemiryám/ has been mistranslated "of yeast" so the student returns to step 11. A starting point might be to look up "yeast" and try to find a match of the Hebrew words. Indeed, this is the case. The student will note that "yeast" is /Ómarím/, which is spelled identically to /Óemiryám/ — shin-mem-resh-yod-mem — but pronounced differently.

At this point, the student may not understand the reason Targumatik used "of" in translating /yadá Óemiryám/ as "a knowledge of yeast." There is also the unexplained presence of the indefinite article, "a." This is where the Hebrew grammar references are needed. Looking up "of" in Inbal, the student will note that the first meaning is /Óél/. Here a student might here consult a Hebrew grammar reference on /Óél/ to see how "a knowledge of yeast" might be translated. Birnbaum (1966) provides a clue. Constructions of the form noun1 + /Óél + noun2 translate as noun1 "of" noun2. It should also be noted that the juxtaposition of two nouns — known as the "construct state" — can occur with the same meaning as the /Óél/ construction. Thus, "a knowledge of yeast" could be translated into Hebrew either as /yedá Óél Ómarím/ or /yedá Ómarím/. As far as Targumatik is concerned, the latter is spelled identically to
the input phrase — יוד-דלת-אين shin-mem-resh-yod-mem — and, therefore, matches the input. As for the indefinite article, "a," a search in Blumberg and Lewittes (1982) indicates that there is no indefinite article in Hebrew. A noun such as /סֵפֶר/ , for example, may be translated either as "book" or "a book" — depending on the context. So at this point, the student has added some knowledge of the construct state in Hebrew and the way in which Hebrew handles the English indefinite article.

One more problem remains: Targumatik did not use a form of BE where it should have in translating the last part of the input sentence / יָדָא עֶמְרִיָּם בֶּתֶל אֲבִיב/ as “a knowledge of yeast in Tel Aviv” instead of "a knowledge of yeast is in Tel Aviv." Yetiv (1973) provides an answer: in the present tense, the equivalent of BE in Hebrew is not used. Thus, a sentence such as "this is a book" would be translated /זֶה סֵפֶר/ in Hebrew — with no equivalent for either "is" or "a." Now the student understands why the MT output did not match the human parallel corpus translation.

As /בֶּתֶל אֲבִיב/, a prepositional phrase, was translated correctly as "in Tel Aviv," the iteration of steps 11 through 14 is finished and the next step is step 15. The student is instructed here to write the Hebrew input sentence cursively, followed by copying the expert human translation of the sentence. If this is the end of the session, the student goes to step 17 and may write up a short report on what was learned with comments. Otherwise, the next Hebrew sentence from the text corpus is input into Targumatik and the process begins again at step 1.

During the entire session, there has been only positive learning and reinforcement since the input sentence — and model used in pronunciation, reading, typing, and writing — was in grammatically correct Hebrew. All the errors were in English and were used to learn new points of Hebrew grammar and vocabulary or reinforce ones already known.

**Human-machine interface problems**

Only one area of the human-machine interface evaluation — error prevention and correction — was judged to be a potential obstruction to learning since it deals with user input errors and because Targumatik scored the lowest in that area. Since virtually no edit checks are performed by the software at the time of user input, this appears as a difficult problem. The learning algorithm, however, takes possible user input errors into account by "training" the student to check for errors in the input before initiating the translation process (see step 3 in the step-by-step procedures for the Targumatik learning algorithm listed in Selection of ancillary tools). Thus, the necessity for edit checking and validation routines is significantly reduced and the potential obstruction to learning is virtually eliminated because the student controls the input process. By
checking the input for errors before initiating the translation process, the student continually reinforces his or her knowledge of Hebrew orthography.

**Software problems**

The software problems summarized above do not adversely affect L2 learning because the learning algorithm accounts for them. The first two problems that appear to have the potential to obstruct learning have to do with updating the dictionary. By the time the student is ready for this step, he or she is already at step 9 of the step-by-step procedures. All mistranslated or untranslated words have already been looked up in the ancillary tools and learning has taken place. Adding the new words or expressions and retranslating, only to find that the system has not accepted the expressions, should have little or no effect on the student's ability to learn the new material. In fact, repeating the procedure will likely reinforce the newly learned material.

The final two problems were noted because messages that should have been displayed in English appeared, instead, in Hebrew. Note first that once these messages have been deciphered, they no longer pose any problem to the student. To ensure that these messages are understood, the student has only to use the Targumatik resources and the ancillary tools to determine the meaning. This is in keeping with the spirit of the learning algorithm. For example, at the top of the Dictionary Add screen is a Hebrew message (reading right to left): he-kof-shin Fl lamed-ayin-zayin-resh-he. A strong clue to the meaning of the phrase is the English "F 1" which the student will note refers to the Fl function key on the computer keyboard. The phrase probably is an instruction to press the Fl key to initiate some action. Using the Targumatik on-line Hebrew dictionary for the first word yields the English "HIT [PRESS]." This confirms the guess with respect to "Fl." Since there is no English instruction for bringing up a "Help" screen, one might assume that lamed-ayin-zayin-resh-he means something like "help." Indeed, a check in the dictionary shows this to be the case. So the screen message means "Press Fl for Help." One can conclude that using the methodology of the learning algorithm on this potential problem turns it into a learning experience.

**SUMMARY AND CONCLUSIONS**

Rüschoff (1993) refers to the application of computational technologies to language learning as 'IT enhanced language learning" and stresses the importance of using language in experimentation and research. MT would likely come under this classification if it can be shown that it can be used to enhance language learning.
But the current major MT evaluation effort (ARPA), shows that MT performs only about 65% as well as a human expert translator. Despite such a poor showing, however, it is argued that MT can be used as a powerful focal point in L2 learning. This can be accomplished by developing a set of requirements and procedures called a learning algorithm.

MT evaluation is important in understanding how the learning algorithm works. MT output is evaluated by mainstream researchers in terms of fluency (by means of error analysis), adequacy, and informativeness. These three metrics measure how close it comes to an English speaker's mental model of fluent English, the degree to which meaning expressed in the expert human translation is present in the MT output, and the degree to which the output provides the meaning expressed in the original. Two additional areas are found to be useful in evaluating the MT software: the human-machine interface (using a checklist of weighted questions in nine functional areas) and an analysis of software problems in terms of whether they are of major, medium or minor impact on the user.

With these tools, Targumatik, a Hebrew-English DOS-based direct design MT system for the PC (286 and above) is evaluated. The results of the evaluation indicate that Targumatik scored 59.4% for overall MT output - 64.2% for fluency, 68.2% for adequacy, and 45.6% for informativeness. The human-machine interface score was 50.4% and the software problems error-free score was 68%. From the evaluation, a list of problems with the potential to obstruct learning was selected.

The learning algorithm is a set of requirements, a set of ancillary tools, and a set of step-by-step procedures that work together. Defining the requirements involves defining selection criteria for MT software, the ancillary tools, the text corpus, the language pair, the minimum student skill levels, and the establishment of target goals. Development of the procedures consists of identifying the functional steps, determining their order of application, testing, modifying, and evaluating them.

The Targumatik learning algorithm is applied to the set of problems identified in the Targumatik evaluation. The results show that since the text corpus input sentences are all grammatically correct Hebrew sentences and serve as the model used in pronunciation, reading, typing, and writing, then there is only positive learning and reinforcement. MT output errors are only in English, and there is an expert human translation to which the student can compare them to determine why they occurred. This, too, is a positive learning experience since it allows the student to learn new points of Hebrew grammar and vocabulary or reinforce points already discovered.
Finally, applying the Targumatik learning algorithm to Targumatik evaluation problems demonstrates that MT with a properly constructed and applied learning algorithm can definitely be used to enhance language learning and, hence, must qualify as a candidate for Rüschoff’s “IT enhanced language learning.”

NOTES

1 See the Quality Panel Evaluation Criteria for the 1992 MT workshop.
2 The ARPA term for this is “degrading.”
3 An expanded version of the error analysis metric used in the Quality Panel Evaluation Criteria (NSF 1992) was used in which some notions from Flanigan (1994) were added.
4 ALPAC adds a 0 level which is defined as “The original contains, if anything, less information than the translation. The translator has added certain meanings, apparently to make the passage more understandable.” This level is not used as the assumption (justified or not) is made that the human English translation member of the corpus pair is a perfect translation. This is done for the purpose of simplifying the comparison.
5 The tenth functional area mentioned under “Methodology” in the “Human-machine interface evaluation criteria” section is not actually part of the checklist. See Appendix D for further details.
6 Interagency Language Roundtable language skill level scale.

REFERENCES


APPENDIX A: EVALUATION METRICS FOR MT OUTPUT

Fluency

Syntactic errors (incorrect subject/object/indirect object, incorrect prepositional phrase, interclausal error, incorrect negation) are assessed a weight of 4.

Lexical errors were assessed a weighted score of 2. The following are considered lexical errors: mistranslated words— including proper nouns; incorrect noun inflection; misplaced lexical modifiers; deletion or insertion of a word; incorrect tense or incorrectly formed verb; lack of subject-verb agreement; absent or unneeded article; untranslated word; and incorrect, absent or unneeded pronoun. Also included in this category would be incorrect translation of a multi-word expression (if not covered by any of the preceding subcategories).

Stylistic errors receive 1 point each and included errors of style, usage, or thematization/focus.

Punctuation and orthography errors are worth 1/2 point each.

Adequacy

The expert human English translation member of the parallel corpus sentence pair is compared to the machine translation member and points are assessed as follows:

5 = All meaning expressed in the expert human translation member of the corpus pair is present in the machine translation version.
4 = Almost all meaning expressed in the expert human translation member of the corpus pair is present in the machine translation version.
3 = A moderate amount of the meaning expressed in the expert human translation member of the corpus pair is present in the machine translation version.
2 = Not much of the meaning expressed in the expert human translation member of the corpus pair is present in the machine translation version.
1 = Almost none or none of the meaning expressed in the expert human translation member of the corpus pair is present in the machine translation version.

Informativeness

The expert human translation member of the parallel corpus sentence pair is compared to the machine translation member and points are assessed as follows:

9 = Not informative at all; no new meaning is added, nor is the reader’s confidence in his understanding increased or enhanced.
8 = No really new meaning is added by the original, either at the word level or the grammatical level, but the reader is somewhat more confident that he apprehends the meaning intended.
7 = By correcting one or two possible critical meanings, chiefly on the word level, it gives a slightly different “twist” to the meaning conveyed by the translation. It adds no new information about sentence structure.
6 = In contrast to 7, adds a certain amount of information about the sentence structure and syntactical relationships; it may also correct minor misapprehension about the general meaning of the sentence or the meaning of individual words.
5 = (Between 4 and 6)
4 = Clearly informative. Adds considerable information about the sentence structure and individual words, putting the reader “on the right track” as to the meaning intended.
3 = (Between 2 and 4.)
2 = Very informative. Contributes a great deal to the clarification of the meaning intended. By correcting sentence structure, words, and phrases, it makes a great change in the reader’s impression of the meaning intended, although not so much as to change or reverse the meaning completely.
1 = Extremely informative. Makes “all the difference in the world” in comprehending the meaning intended. (A rating of 1 should always be assigned when the original completely changes or reverses the meaning conveyed by the translation.)

APPENDIX B: TARGUMATIK MT OUTPUT EVALUATION

- Total sentences in text corpus: 284.
- Total text corpus points for fluency using error analysis: 1,229.5
- Average error points per sentence: 4.3 on a scale of 1-12
- Total error percentage score: 35.8
- Overall error-free percentage score: 64.2

Sample sentences:

Homograph confusion:
/\aní efné bapíná hazót/ /efné/ is spelled alef-fe-nun-hey
[I will-turn at-corner this]
T: “I’ll turn at this corner”
Mistranslation of prepositions and other words:
/hitrasámnu mehabulím haxadasím/
[we-were-impressed from-the-stamps new]
“We were impressed by the new stamps”
T: “We was impressed from the new stamps:
Untranslated words: 93
/emórna lí bevakasa eyfó hadóar/
[tell to-me please where-[is] the-post]
“tell me please, where is the postoffice?”
T: “emórna me, please, where the mail?”

Lack of agreement between subject and verb:
/hitrasámnu meód mibét haséfer/
[we-were-impressed very from-house-of the-book]
“we were very impressed by the school”
T: “we was impressed very from the school”

Sample sentences:

Homograph confusion:
/\aní efné bapíná hazót/ /efné/ is spelled alef-fe-nun-hey
[I will-turn at-corner this]
T: “I am a fashion in this corner” /ofná/ “fashion,” can be spelled alef-fe-nun-hey

Mistranslation of prepositions and other words:
/hitrasámnu mehabulím haxadasím/
[we-were-impressed from-the-stamps new]
“We were impressed by the new stamps”
T: “We was impressed from the new stamps:

Untranslated words: 93
/emórna lí bevakasa eyfó hadóar/
[tell to-me please where-[is] the-post]
“tell me please, where is the postoffice?”
T: “emórna me, please, where the mail?”

Lack of agreement between subject and verb:
/hitrasámnu meód mibét haséfer/
[we-were-impressed very from-house-of the-book]
“we were very impressed by the school”
T: “we was impressed very from the school”
Incorrect placement of lexical modifier:
/hú mocé sehabáit yafé meód/
[he finds that-the-house [is]-pretty very]
“he finds that the house is very nice”
T: “he finds that the house is pretty very”

Deletion of BE:
/haím zé raxók mikán/
[(question marker) it far from-here]
“Is it far from here?”
T: “It far from here?”

Deletion of DO:
/beéyze bet séfer lamádet/
[in-which school [you (fem. Sing.)]-studied]
“In which school did you study?”
T: “In which school studied?”

Incorrect verb tense:
/hú yesév bamisrád ád xamés/
[he will-sit in-office until 5:00]
“he will sit in the office until 5:00”
T: “he sat in the office until five”

Inability to recognize proper nouns:
/baalí gár bemalón dán/
[husband-to-me lives in-hotel Dan]
“my husband lives in Hotel Dan”
T: “in on me lives in hotel discusses”

• Total text corpus points for adequacy using ARPA methodology: 969
• Average points per sentence: 3.41 on a scale of 1-5
• Overall percentage score: 68.2

Sample sentence:
/át medabéret ivrít mamás kmó israelít/
[you (fem.) speak Hebrew really like [an]-Israeli (fem.)]
“You speak Hebrew just like an Israeli.”
T: “You talk Hebrew quite a such as israeli.”

• Analysis: Because Targumatik missed the expression “just like,” the meaning is fairly garbled. The sense of “talk” in English is not exactly the same as “speak” — as “to speak a language.”
• Score: 3 (1-5 scale).
• Total text corpus points for informativeness using the modified ALPAC metric: 1171
• Average points per sentence: 4.1 on a scale of 1-9
• Overall percentage score: 45.6
Sample sentence:
/raínu et yisraél haxadisá/
[we-saw [dir.obj.marker] Israel the-modern]
“We saw the modern Israel”

T: “Interview Israel (haxadisá/?)”

- Analysis: Because Targumatik mistranslated /raínu/ and was unable to decipher /haxadisá/, the meaning is changed completely from what was originally meant. The only thing known for certain is that the sentence has something to with Israel.
- Score: 1 (scale of 1-9).

APPENDIX C: HUMAN-MACHINE INTERFACE EVALUATION CHECKLIST EXCERPT

<table>
<thead>
<tr>
<th>Checklist question</th>
<th>Always</th>
<th>Most of the time</th>
<th>Some of the time</th>
<th>Never</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Is each screen clearly highlighted with an informative title or description?</td>
<td>3</td>
<td>X</td>
<td></td>
<td></td>
<td>Special not well designed; HELP screens inconvenient</td>
</tr>
<tr>
<td>2. Is important information highlighted on the screen? (e.g. cursor position, instructions, errors)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. When the user enters information on the screen is it clear where the information should be entered?</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. When the user enters information on the screen is it clear in what format it should be entered?</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>Mixed language problem</td>
</tr>
<tr>
<td>5. Where the user overtypes information on the screen, does the system clear the previous information, so that it does not get confused with the updated input?</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>no automatic feature to clear information</td>
</tr>
<tr>
<td>6. Does information appear to be organized logically on the screen? (e.g. menus organized by probable sequence of selection, or alphabetically)</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>See notes on Special and Files options</td>
</tr>
<tr>
<td>7. Are different types of information clearly separated from each other on the screen? (e.g. instructions, control options, data displays)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. Where a large amount of information is displayed on one screen, is it clearly separated into sections on the screen?</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>General Help screen and Dictionary entry screen are crowded and confusing</td>
</tr>
<tr>
<td>9. Are columns of information aligned on the screen? (e.g. columns of alphanumerics left-justified, columns of integers right-justified)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10. Are bright or light colors displayed on a dark background, and vice versa?</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11. Does the use of color help to make the displays clear?</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>can be changed to black and white</td>
</tr>
<tr>
<td>12. Where color is used, will all aspects of the display be easy to see if used on a monochrome or low</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13. Is the information on the screen easy to see and read?</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14. Do screens appear uncluttered?</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>Final word list screen is cluttered</td>
</tr>
<tr>
<td>15. Is it easy to find the required information on a screen?</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>Except for the few cluttered screens</td>
</tr>
<tr>
<td>Subtotals:</td>
<td>8</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>Total: 15</td>
</tr>
<tr>
<td>Weighted subtotals (total * weight)</td>
<td>24</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>Total: 35</td>
</tr>
</tbody>
</table>

Visual Clarity Total Score

<table>
<thead>
<tr>
<th>Very satisfactory</th>
<th>Moderately satisfactory</th>
<th>Neutral</th>
<th>Moderately unsatisfactory</th>
<th>Very unsatisfactory</th>
</tr>
</thead>
<tbody>
<tr>
<td>81-100</td>
<td>61-80</td>
<td>41-60</td>
<td>21-40</td>
<td>0-20</td>
</tr>
</tbody>
</table>

77.7*  

*The value of 77.7 was derived by dividing the total possible weighted score (15*3 = 45) by the total evaluated score of 35.;;
APPENDIX D: CHECKLIST FUNCTIONAL AREAS

**Visual clarity** concerns the way in which information is displayed on screen. Good visual clarity should make a screen appear uncluttered and enable the user to find the required information as well as draw the user’s attention to any important information. (14 applicable questions)

**Consistency** is concerned with creating/reinforcing the user’s expectations by maintaining predictability across the interface. A consistent interface enables the user to learn more quickly/efficiently, reduces confusion as well as search/response times, reduces the user working memory load, and reduces the likelihood of user errors. (12 applicable questions).

**Compatibility** is defined as being concerned with ensuring that the interface conforms with existing user expectations where they are appropriate. Likely sources for this are conventions such as date format and the use of certain colors such as red for danger/stop. (15 applicable questions.)

**Informative feedback** is concerned with providing users with appropriate, clear, meaningful, and timely information about the system in the context of the task being carried out. The interface should provide status information to the user regarding the actions they have taken and whether or not these actions have been successful. (15 applicable questions.)

The checklist authors define **explicitness** as being concerned with helping the user develop a clear/accurate understanding of the interface/structure/function. **Explicitness** helps to make the interface “transparent” to the user. If the interface is **explicit**, the user should be able to complete tasks more easily because he or she has a clearer understanding of how the system is structured and how it functions. (13 applicable questions.)

The **appropriate functionality** metric concerns the user’s perceived requirements of the interface when carrying out tasks. Poor design in this area can lead to the user’s experiencing difficulty and frustration, and it can reduce user efficiency. (11 applicable questions.)

**Flexibility and control** refer to the ability of the system to accommodate the needs and requirements of different users in different situations. An experienced user, for example, should have short cuts made available to him or her to accomplish a task via the interface. This might involve eliminating unnecessary steps when appropriate, allowing customization of the system to fit user needs, and reducing any time-consuming activities such as typing repeated information. (14 applicable questions.)

The **error prevention and correction** metric concerns the ability of the system to prevent, detect and handle input errors. A good design in this area should reduce the likelihood of errors and ensure that those that occur are corrected before they cause problems. (13 applicable questions.)
**User guidance and support** concerns the presence, usability, content, organization and guidance in the use of on-line help and hard-copy documentation. A well-designed system interface in this area will provide informative, easy-to-use and relevant guidance and support—both on the computer and in hard-copy format—to help the user understand and use the system effectively. (11 applicable questions.)

Problems with **system usability** represents the need for a follow-up to the interface evaluation checklists to determine level of user problems encountered. This questionnaire is not actually part of the checklist, and the method used differs from the checklist metrics already discussed. For this reason, the results are not included in the overall average human-machine interface score. The 25 items were rated as “No problems,” “Minor problems” or “Major problems.”

**AUTHOR’S BIO DATA**

Don D. Anderson received his doctorate in Languages and Linguistics from Georgetown University in 1974. He taught for a number of years prior to joining the U.S. Department of Defense, where he works as a computational linguist. His primary research interests include natural language processing and computer assisted language learning.

**AUTHOR’S ADDRESS**

12902 Old Chapel Road
Bowie, MD 20720

Home phone: 301-464-9246
E-mail: donande@romulus.ncsc.mil