Test Suites for Natural Language Processing

Lorna Balkan, Doug Arnold, and Siety Meijer
University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, U.K.

SYNOPSIS

This paper introduces the topic of evaluation of Natural Language Processing systems, and discusses the role of test suites in the linguistic evaluation of a system. The work on test suites that is being carried out within the framework of the TSNLP project is described in detail and the relevance of the project to the evaluation of machine translation systems considered.

INTRODUCTION

Evaluation is a topic that is currently attracting a great deal of interest in the Natural Language Processing community.[1] The science of evaluation is however, relatively speaking, in its infancy. Historically, the United States have been ahead of Europe with their ARPA and DARPA Speech and Natural Language program which started in 1984. Current initiatives in evaluation in Europe include the work of EAGLES (Expert Advisory Group on Language Engineering Standards), which was set up in 1993 and whose primary goal is to improve evaluation methods as a step towards setting up standards for language engineering products.

The Commission of the European Communities is also, within the context of its Linguistic Research and Engineering (LRE) program, currently sponsoring several projects in the field of evaluation, including the project Test Suites for Natural Language Processing (TSNLP) which is the subject of this paper. TSNLP shares with other some other LRE evaluation projects the aim of producing a collection of common test materials. In this case of TSNLP, this constitutes a set of reusable test suites for a range of applications. Further aims of TSNLP are described below.

EVALUATION: SOME TERMINOLOGY

[1] We would like to thank our colleagues in TSNLP for fruitful discussions on this topic: Eva Dauphin, Dominique Estival, Kirsten Falkedal, Sabine Lehmann, Klaus Netter and Sylvie Regnier-Prost.

It is customary to define a number of different evaluation scenarios, depending on the purpose of the evaluation. EAGLES, for example, distinguishes the following three types of evaluation:

- *diagnostic evaluation*, which aims at localizing deficiencies;
- *progress evaluation*, for a comparison between successive stages of development of a system; and
- and *adequacy evaluation*, to determine whether and to what extent a particular system meets some pre-specified requirements.

Developers are chiefly interested in diagnostic and progress evaluation, while users are mainly interested in adequacy evaluation. However, if developers aim eventually to market their products then adequacy evaluation is an issue for them too. Likewise, if a user wants to know not just how a product behaves today, but in its future potential, then they will be interested in performing a diagnostic evaluation. A diagnostic evaluation for the developer will, however, differ from that of the user in that the user is typically in a black box situation with respect to the system (i.e. he does not have access to its internal workings), while a developer will be in a glass box situation, where he will have access to the system rules.

THE ROLE OF TEST SUITES IN EVALUATION

Traditionally there are two main ways of evaluating NLP systems, either by the use of test corpora (i.e. pieces of text) or by test suites (i.e. lists of specially constructed sentences, or sentence sequences or even sentence fragments). Traditionally, test suites are the preferred option of the system developer, since he wants to see how his system will perform on a range of controlled examples. And traditionally the user prefers to test a system against a test corpus that has usually been selected to be representative of the texts he requires his NLP system to process. This is because test suites are useful for diagnostic evaluation, whereas test corpora are a tool traditionally associated with adequacy evaluation. But as the previous paragraph should have made clear, test suites could equally prove useful to the user if he undertakes diagnostic evaluation.

Test suites and test corpora have different roles to play in evaluation and should be seen not as competing tools, but rather as complementary. Test suites are useful for presenting language phenomena in an exhaustive and systematic way. Thus, for example, each different type of noun phrase or adjective phrase can be listed, starting with the simplest and increasing in complexity. Furthermore, combinations of phenomena can be generated in a controlled fashion. For example, coordinated noun phrases can be produced on the basis of simple noun phrases. Negative data, likewise, can be derived systematically from positive data by violating grammatical constraints associated with the positive data item. For example, violation of determiner-noun agreement in English produces ill-formed examples such as **those heavy book* or **that heavy books*. Note that in test suites, the vocabulary, as well as the sort of construction being tested, can be controlled. This allows the evaluator to focus on the way the system deals with the construction without the distraction of problems relating to lexical coverage.

Test corpora, on the other hand, lack the exhaustively and systematicity of test suites. Furthermore, the complexity of many naturally occurring phenomena can make it difficult to isolate the exact phenomenon or phenomena that one is interested in testing. The task is not helped by the fact that most corpora lack any sort of annotation. So, what are the strengths of the test corpora method? Well, firstly, as already mentioned, test corpora represent naturally occurring data, so that one can be sure that the phenomena one is testing for really do occur. A criticism that can be levelled against the test suite technique is that some of the phenomena never ever occur in real life. Note, however, that it is a non-trivial task to ensure that a test corpus is representative of a larger corpus. Text processing tools can give some idea of frequency of phenomena and lexica, sentence length, etc. but the problem is still a hard one.

We said above that test suites and test corpora are complementary techniques. The test suite method is particularly useful for testing syntactic phenomena (see for example the Hewlett Packard test suite (see Flickinger et al. (3), perhaps the best known test suite to date), where the range of phenomena is relatively well-understood and well-documented. Semantic and pragmatic phenomena are less accessible to the test suite method, since the phenomena are less easy to characterise, and are frequently context-dependent. This means that many phenomena, such as anaphora resolution need to be tested within a sequence of sentences, rather than in isolated sentences. This is where test corpora are useful, because they just are a sequence of sentences. Some suggestions for what should go into a semantic test suite are discussed in Hoard (6).

It is also the case that some applications are less well suited to being tested by the test suite method than by test corpora. Message understanding systems, for example, need whole sequences of sentences as input, so are better suited to the test corpora method. Test suites are useful for any system which has a large syntactic analysis component. Furthermore, they are best suited to applications where it is possible to specify not just the nature of the input, but also the nature of the output. A good example is a grammar checker. Generation systems, on the other hand, are less well-suited to this method, since it is difficult to specify not only what the input to a generation system should be, but also what constitutes an appropriate output.

There is a long tradition of using test suites to evaluate machine translation systems. Recent examples include Gambäck (4), Heid and Hildenbrand (5), and Way (10). Arnold et al. (2), Chapter 9 provides a general introduction, and useful discussions on the role of test suites in the evaluation of machine translation systems can also be found in King and Falkedal (7). Of course, test suites can be used straightforwardly in the evaluation of many machine translation system components (e.g. syntactic parsers). However, their use in relation to machine translation systems raises a number of interesting issues.

First, as King and Falkedal (*ibid*) point out, most existing test suites are designed for monolingual applications. However, in the case of machine translation systems, "bilingual" test suites are required that probe the capacity of systems to deal with particular translation problems (such as the problem of lexical and structural mismatch, e.g. the classical *like– plaire* case, where the arguments of the verb are reversed in translation: *John likes Mary* translates as *Mary plaît à John*). Such "bilingual" test suites will have to be specially constructed, and in general their construction requires some rather detailed insight into the nature of translation problems. Of course, "bilingual" test suites must be distinguished from any test suites that are to be used to test purely monolingual components, where the test items should be translationally unproblematic, so that they do not introduce irrelevant difficulties.

Second, as with generation systems, there is what one might call the "output" problem. For some applications, one is only interested in whether a system accepts or rejects a test item. For such applications, the evaluation process can be automated and a high degree of objectivity (relative to the particular test suite) is possible. With a machine translation system this is not the case: one is typically interested not just in whether a system accepts an input, but also in the correctness of the output it produces. Of course, one cannot simply specify what the "correct" translation of any particular test item is – there is in general no single "correct" translation of any expression. This makes the evaluation process rather subjective, and difficult to automate. One interesting suggestion here (due to Henry Thompson (9), and currently being investigated as part of a research project in Edinburgh) is to assume that, though a wide range of translations may be possible, "good" translation will tend to be more similar to each other than bad ones – "good" translation will tend to cluster together. As regards test suites, a possible application of this idea would be to associate a "central" member of this cluster with each test item, and compare this to what the system under test actually produces. If the degree of difference is within the range that one finds among the cluster of "good" translations, one may assume that the system has performed satisfactorily on this item.

Finally, test suites need to be supplemented by corpus methods to test semantic and pragmatic phenomena. Despite these limitations, test suite based evaluation is unquestionably a useful component in the evaluation of machine translation systems, both for developers and end users. It is to be expected that the development of multilingual test suites, as in the present project, will be a useful step towards overcoming these limitations, and making them more useful still.

THE TSNLP PROJECT

The aim of TSNLP is to develop a methodology for the design and development of test suites, and to actually produce test suites for a range of NLP applications. These test suites will be of medium size (several hundred items) for English, French and German. The applications are, specifically, parsers, grammar checkers and controlled language checkers, all of which contain large syntactic components, and are thus, as we have seen, particularly suited to the test suite method of evaluation. However, it is expected that the results will be usable for other application types. The fact that the data is being constructed in three languages (English, French and German) means that it should be of particular relevance to multilingual applications, including machine translation. The results of the project, both scientific reports and actual test suites, will be in the public domain.

The project started in December 1993 and has a duration of 20 months. The partners involved are The University of Essex, UK who are the coordinators, plus Aerospatiale, France, Deutsches Forschungszentrum fuer Kuenstliche Intelligenz GmbH. (DFKI), Saarbruecken, Germany, and Istituto per gli Studii Semantici e Cognitivi (ISSCO), Geneva, Switzerland.

This project has the following aims:

- To define a set of guidelines for the construction of test suites for a range of NL products, including machine translation systems, concentrating on grammar checkers, parsers and controlled language checkers.

- To produce substantial test suite fragments covering core syntactic phenomena in three languages (English, French and German). The project includes a testing phase for each of the three applications and revisions to the guidelines are foreseen in the light of test results.
- To identify and develop a number of tools which will facilitate the construction and use of test suites, namely:
  - A database in which the test suite will be stored which will allow easy access and manipulation of the data.
    TSNLP is inspired by the DITO test collection (see Nerbonne et al. (8)) in its use of a database on which to mount and manipulate the data. The aim is to make the test data easy to access and flexible in the type of configuration that can be retrieved.
  - An automatic test suite generation tool.
    Little previous work has been done on the automatic generation of test suites, but the endeavour seems worthwhile, given the labour-intensive and error-prone business of constructing test suites by hand. The project will take as a starting point work by Arnold et al. (1) on test suite generation.
  - A lexical replacement tool.
    This will be helpful in the customisation that will be necessary to test system performance against a user's own corpora.

TSNLP began by reviewing publicly available test suites, to see in what ways test suite design could be improved.

Despite the frequent reference to test suites in the NLP literature, surprisingly few test suites are publicly available. The test suites investigated differed greatly with respect to:

- Purpose (diagnostic/adequacy/progress evaluation):
- Intended application (parsers, MT systems, etc)
- Depth and Breadth of coverage
- Presentation of data

TSNLP is above all interested in producing a test suite that is flexible and reusable. The review of existing publicly available test suites revealed the following characteristics that are important for flexibility and reusability:

- Systematic annotation scheme:
  An explicit characterisation of the test data, not merely section headings.
- Support tools:
  Software tools to assist in the creation or use of test suites
- Documentation:
  Documentation is useful on both the design and content of the test suite.

Few of the test suites we examined or which are reported on in the literature contain any or all of these characteristics. They are however, a key focus of TSNLP. Systematicity, as we have seen, is important for negative as well as positive data, and for combinations of phenomena. However, in the case of negative data and combinations of phenomena, the possibilities are numerous and some method is needed for their selection. One selection criterion might be, for example, frequency of occurrence. A proper annotation scheme is required, not just in view of the database, but to make the data maximally explicit and therefore reusable in general.

The availability of validated test data that is fully annotated and accessible, by means of the database, is expected to be of benefit to developers and users of NLP products, even outside the applications for which the data is principally designed (i.e. grammar checkers, controlled language checkers and parsers). Test suites as a tool are, as we have discussed, of interest to anyone, developer or user, that is interested in diagnostic evaluation. Test suites, as we have seen, are most useful for systems that contain a large syntactic component, and this includes many MT systems. The multilingual nature of the project means that it should be possible to extract parallel data across different languages, and, potentially locate where there is non-parallelism in structure. Other phenomenon of important to MT, such as lexical mismatches, however, remain outside the scope of the present project.

REFERENCES

1. Arnold, D., Moffat, D., Sadler, L., Way, A., 1993, “Automatic Test Suite Generation”, in *Machine Translation*, Volume 8, nos 1-2, pp29-38.

2. Arnold, D., Balkan, L., Humphreys, R. L., Meijer, S., Sadler, L., 1993, *Machine Translation – An Introductory Guide*, NCC Blackwell, Oxford.

3. Flickinger, D., Nerbonne, J., Sag, I.A. and Wasow, T., 1987, "Toward Evaluation of NLP Systems", Hewlett-Packard Laboratories, distr. at the 24th Annual Meeting of the Association for Computational Linguistics (ACL), Stanford.

4. Gambäck, B., 1992, "Developer Oriented Evaluation: Two Experiments", talk presented at the *Workshop on the Strategic Role of Evaluation in Natural Language Processing and Speech Technology*, Record of the ESPRIT DANDI-ELSNET-HCRC Workshop, Edinburgh, 1992.

5. Heid, U., and Hildenbrand, E., 1991, "Some practical experience with the use of test suites for the evaluation of SYSTRAN" in the *Proceedings of the Evaluators' Forum*, Les Rasses, April 21-24 1991, available from ISSCO, University of Geneva, Switzerland.

6. Hoard, J., 1919, "Preliminaries to the Development of Evaluation Metrics for Natural Language Semantic and Pragmatic Analysis Systems", in Neal, J.G. and Walter, S.M. (eds), 1991, *Natural Language Processing Systems Evaluation Workshop*, Report RL-TR-91-362, Rome Laboratory.

7. King, M. and Falkedal, K., 1990, "Using Test Suites in the Evaluation of Machine Translation Systems", in *Proceedings of the 13th International Conference on Computational Linguistics (COLING)*, Helsinki.

8. Nerbonne, J., Netter, K., Kader Diagne, A., Klein, J., and Dickman, L., 1992, "A Diagnostic Tool for German Syntax" Report DFKI D-92-03, Saarbrücken. Also in: Neal, J. and Walter, S. eds., 1991, *Natural Language Processing Systems Evaluation Workshop*, Berkeley, Rome Laboratory, Report RL-TR-91-362, New York.

9. Thompson, Henry, S. 1991, "Automatic Evaluation of Translation Quality: Outline of Methodology and Report on Pilot Experiment", *Proceedings of the Evaluators' Forum*, April 21-4, Less Rasses, Vaud, Switzerland, 215-224.

10. Way, A., 1991, "A Practical Developer-Oriented Evaluation of Two MT Systems", *Department of Language and Linguistics Working Papers in Language Processing*, 26, Department of Language and Linguistics, University of Essex, Colchester, UK.