TSNLP — Test Suites for Natural Language Processing

Sabine Lehmann♠, Stephan Oepen♡
Sylvie Regnier-Prost♣, Klaus Netter♡, Veronika Lux♣, Judith Klein♡,
Kirsten Falkedal♠, Frederik Fouvry◇, Dominique Estival♠, Eva Dauphin♣,
Hervé Compagnion♠, Judith Baur♡, Lorna Balkan◇, Doug Arnold◇

♠ISSCO	♡DFKI GmbH	◇CL|MT Group	♣Aerospatiale France
Université de Genève	CL Department	University of Essex	Common Research Center
54, route des Acacias	Stuhlsatzenhausweg 3	Wivenhoe Park	12, rue Pasteur BP 76
CH 1227 Genève	D 66123 Saarbrücken	UK Colchester CO4 3SQ	F 92152 Suresnes Cedex
+41 - 22 - 705 79 33	+49 - 681 - 302 52 82	+44 - 1206 - 87 20 86	+33 - 1 - 4697 30 61

Abstract

The growing language technology industry needs measurement tools to allow researchers, engineers, managers, and customers to track development, evaluate and assure quality, and assess suitability for a variety of applications.

The TSNLP (Test Suites for Natural Language Processing) project[1] has investigated various aspects of the construction, maintenance and application of systematic test suites as diagnostic and evaluation tools for NLP applications. The paper summarizes the motivation and main results of TSNLP: besides the solid methodological foundation of the project, TSNLP has produced substantial (i.e. larger than any existing general test suites) multi-purpose and multi-user test suites for three European languages together with a set of specialized tools that facilitate the construction, extension, maintenance, retrieval, and customization of the test data.

The publicly available results of TSNLP represent a valuable linguistic resource that has the potential of providing a wide-spread pre-standard diagnostic and evaluation tool for both developers and users of NLP applications.

1 Background and Motivation

Evaluation of NLP applications plays an increasingly important role in both the academic and industrial NL communities. Two tools traditionally used for evaluating and testing NLP systems are *test suites* and *test corpora*. The two can be seen as serving complementary purposes (see Dauphin et al. (1995a)): in contrast to text corpora, whose main advantage is that they reflect naturally occurring data, the key properties of test suites are (i) *systematicity*, (ii) *control over data*, (iii) *inclusion of negative data*, and (iv) *exhaustivity*.

Among the main motivations for the TSNLP project were the lack of general guidelines for the test suite construction, of adequate and comprehensive test material, and of appropriate tools. The resulting duplication of effort among test suite developers obviously leads to a waste of time and resources. In addition, one of the main conclusions of a study of existing tests suites conducted during the first stage of the project (Estival et al. (1994)) was that the reusability of existing test suites is severely hampered by their lack of structure and annotations. Indeed, despite the pioneering efforts of Flickinger et al. (1987) and Nerbonne et al. (1993), most of the existing test suites were written for some specific system or simply enumerate a number of interesting examples and, thus, do not meet the demand for large, systematic, well-documented, highly-structured and annotated collections of linguistic material, which is now required by a growing number of NLP applications. The TSNLP test suite addresses these demands and provides powerful tools for the construction and manipulation of the test data.

On the one hand, since every NLP system (whether commercial or under development) exhibits specific features which make it unique, and every user (or developer) of an NLP system has specific needs and requirements, the TSNLP approach is based on the assumption that, in order to yield informative and interpretable results, any test suite used for an actual test or evaluation must be *specific* (at least to some degree) to the system and the user. On the other hand, since testing or evaluating NLP systems is performed for a variety of purposes, the TSNLP approach is also guided by the need to provide test material which is easily *reusable*. To achieve these two goals of specificity and reusability, the traditional notion of a test suite as a monolithic set of test items has been abandoned in favour of the notion of a database in which test items are stored together with a rich inventory of associated linguistic and non-linguistic annotations.

Thus, the test suite database serves as a virtual (or meta) test suite that provides the means to extract the relevant subset of the test data suitable for some specific task. Using the explicit struc-

[1] The project was started in December 1993 and completed in March 1996. Most of the project results (documents, bibliography, test data, and software) as well as on-line access to the test suite database and email addresses of the project members can be obtained through the world-wide web from the TSNLP home page '`http://tsnlp.dfki.uni-sb.de/tsnlp/`'.

The TSNLP project was funded within the Linguistic Research Engineering (LRE) programme of the European Commission (DG XIII) under research grant LRE-62-089 and by the Swiss Federal Government.

ture of the data and the TSNLP annotations, the database engine allows searching and retrieving data from the virtual test suite, thereby creating a concrete test suite instance according to arbitrary linguistic and extra-linguistic constraints. Since, additionally, there are tools provided for the maintenance and extension of the test suite database, the TSNLP virtual test suite approach is an essential innovation leading the way to a new generation of highly-structured reusable test suites.

2 Test Suite Design and Methodology

Based on a survey of existing test suites and an analysis of the diagnostic and evaluation requirements of both NL technology developers and users, TSNLP has developed the methodology for the construction of *core test data*, that is, test items reflecting central language phenomena and that are applicable to a wide range of applications, including parsers, grammar checkers, and controlled language checkers (Balkan et al. (1996)).

The TSNLP methodology is designed to optimize (i) *control over test data*, (ii) *progressivity*, and (iii) *systematicity*. These are necessary qualities for an adequate, reusable test suite, which are difficult to find in test corpora. The methodology also addresses the specific goals of TSNLP to produce multi-purpose, multi-user, and multilingual test suites.

Control over test data What makes test suites valuable in comparison to corpora is that they can focus on specific linguistic phenomena and that each phenomenon can be presented both in isolation and controlled combinations in which as many linguistic parameters as possible are being kept under control. This is particularly the case when a phenomenon is illustrated by systematic variation over the parameters used to describe this phenomenon, while all other parts of the test items remain constant.

Vocabulary is an aspect of the test data that needs to be controlled. TSNLP achieves this by restricting the vocabulary in size as well as in domain. Categorially and semantically ambiguous words are avoided where possible and only included when ambiguity is explicitly tested for.

Additionally, TSNLP attempts to control the interaction of phenomena by keeping the test items as small as possible. Therefore, a number of guidelines for this purpose (such as *use declarative sentences* and *avoid modifiers and adjuncts*) is provided.

Progressivity Progressivity is the principle of starting from simple test items and increasing their complexity. In TSNLP, this aspect is addressed by requiring that each test item focuses only on a single phenomenon (or rather subphenomenon or even feature) which distinguishes it from all other test items. This principle not only ensures systematicity during the test data construction but also allows test data users to apply the test data in a progressive order obtained from the special attribute *presupposition* in the phenomena classification. Thus, the precise identification of the coverage of a system and of its deficiencies is rendered easier.

Systematicity Systematicity refers to the depth of coverage of a test suite, with respect to both well-formed and ill-formed items. Systematicity in TSNLP is achieved for well-formed items by the explicit classification of test items according to phenomena and sub-phenomena. Negative test data permits testing for overgeneration as well as for coverage. Ill-formed items are derived from well-formed ones by systematic variation of the parameters through the application of one (or more) of four operations, namely:

- REPLACEMENT (e.g. change of person)
  (French) *L' ingénieur vient.*
  (French) **L' ingénieur viens.*
- ADDITION (e.g. of an object NP)
  (German) *Der Manager arbeitet.*
  (German) **Der Manager arbeitet den Vortrag.*
- DELETION (e.g. of an obligatory complement)
  (German) *Der Manager hält den Vortrag.*
  (German) **Der Manager hält.*
- PERMUTATION (e.g. inverting word order)
  (English) *He saw the boy.*
  (English) **He the boy saw.*

In general, the systematicity of test data was greatly enhanced through the use of special-purpose tools in the data construction and validation process (see section 5 below).

Multilinguality Multilinguality is achieved in the TSNLP test suites by covering the same range of phenomena in English, French and German, and adopting the same classification for these phenomena in the three languages. Furthermore, the choice of related terminology for the categorial and structural description contributes to the comparability and consistency of the test items (see section 4 for details).

Documentation To enhance the usability and extensibility of TSNLP results, a three-volume user guide is under preparation providing clear instructions for the assessment of the methodology, test data, and tools developed.

3 TSNLP Annotation Schema

A detailed annotation schema was designed for the test data which does not presuppose a specific linguistic theory, a particular evaluation situation or application type.

Test data and annotations in TSNLP test suites are organized at four distinct representational levels:

- **Core Data** The core of the test data consists of the individual *test items* together with all general, categorial and structural information that is independent of a token phenomenon or application. Besides the actual input string, annotations at this level include (i) bookkeeping and documentation information (author, date, id number), (ii) the item format, its length, category and well-formedness code, (iii) the (morpho-)syntactic categories and string positions of the lexical and phrasal elements constituting the test item, and (iv) an (underspecified) representation of its functional structure. Encoding a dependency or functor-argument graph rather than a phrase structure tree allows generalizations over potentially controversial phrase structure configurations and, thus, avoids imposing a specific constituent structure but still can be mapped onto one.
- **Phenomenon-Related Data** Based on a hierarchical classification of linguistic (and extra-linguistic) *phenomena* (e.g. verb valency as a subtype of general complementation), each phenomenon is identified by a phenomenon id and by its supertype(s). Interaction with other phenomena as well as the phenomena which must be presupposed are also given. In addition, the (syntactic) parameters which are relevant for the phenomenon (e.g. the number and type of complements in the case of verb valency) are described. Individual test items can be assigned to one or several phenomena and annotated according to the corresponding parameters.
- **Test Sets** Test items can optionally be grouped into *test sets*. A test set is a group of test items containing typically one positive example and one or more negative examples. The relation between positive and negative test items has been one of the most challenging questions in designing test data and, as has been mentioned, is based on the systematic variation of phenomenon-specific parameters.
- **User and Application Parameters** Information that typically correlates with the use of a test suite for different types of evaluation and for different applications (e.g. ratings of frequency or relevance for a particular domain) is factored from the remainder of the data into *user & application profiles*. As part of the customization process users of the TSNLP test suites are encouraged to extend this part of the test suite database and add whatever (formal or informal) information is necessary for their specific requirements.

In addition to the parts of the annotation schema that follow a formal specification, there is room for textual comments at the various levels to accommodate information that cannot or need not be formalized.

Test Item		
item id: *24020101*	author: *issco*	date: *jan-95*
register: *formal*	format: *none*	origin: *invented*
difficulty: *1*	wellformedness: *1*	category: *S*
input: *L' ingénieur vient .*		length: *3*
comment:		

position	instance	category	function	domain
0:2	*L' ingénieur*	*NP_sg*	*subj*	*2:3*
2:3	*vient*	*V_3-sg*	*func*	*0:3*

Phenomenon		
phenomenon id: *2402*	author: *issco*	date: *jan-95*
name: *C_Complementation_subj(NP)_V*		
supertypes: *C_Complementation*		
presupposition: *C_Agreement, NP_Agreement*		
restrictions: *neutral*	interaction: *none*	purpose: *test*
comment: *Intransitive verb (valency:1)*		

Figure 1: Sample instance of the TSNLP annotation schema for one test item: the annotations are given in tabular form for the *test item*, *analysis*, and *phenomenon* levels.

4 Test Data Construction

Following the TSNLP test suite guidelines (Estival et al. (1994)) and using the annotation schema sketched above, the construction of test data was based on a classification of the (syntactic) phenomena to be covered. From judgements on the linguistic relevance and frequency for the individual languages, the following list of *core phenomena* for TSNLP was compiled:

- complementation;
- agreement;
- modification;
- diathesis;
- modality, tense, and aspect;
- sentence and clause types;
- word order;
- coordination;
- negation; and
- extragrammatical (e.g. parentheticals and temporal expressions).

A further sub-classification of phenomena is made according to the relevant *syntactic domains* in which a phenomenon occurs (e.g. sentences (S), clauses (C), noun phrases (NP) et al.). Figure 2 gives an overview of the test material available. For each of the three languages some 5000 test items are provided. Therefore, TSNLP has already achieved a substantially broader and deeper coverage than previous general-purpose test suites (the still very popular Hewlett-Packard test suite, for instance, has a coverage of 3000 test items for English only).

In order to enforce consistency of annotations across the three languages, canonical lists of the categories and functions used in the description of categorial and dependency structure were established (see Lehmann et al. (1996)). The dimensions chosen in the classification attempt to avoid

Phenomenon	English		French		German	
C_Complementation	148	863	188	567	218	246
C_Agreement	68	55	104	183	224	175
C_Modification			329	63		
NP_Complementation	10	27	12	28		
NP_Agreement	201	995	272	1082	299	1732
NP_Modification	301	484			53	60
Diathesis	157	124	176	119	147	148
Tense Aspect Modality	157	39	77	275	186	134
Sentence Types	80	100	389	387	105	14
Coordination	147	186	379	319	105	429
Negation	289	129	68	100	82	210
Word Order			7	7	60	160
Extragrammatical	24	34			253	0
Total	1582	3036	2001	3130	1732	3308

Figure 2: Status of the TSNLP data (December 1995): relevance and breadth of individual phenomena present language-specific variation (the numbers given are for grammatical vs. ungrammatical items). Individual phenomena are often further sub-classified according to phenomenon-internal dimensions.

the presupposition of very specific assumptions of a particular theory of grammar (or of a language), and rather try to capture those distinctions that seem to be relevant across the set of TSNLP core phenomena.

5 Test Suite Technology

Because the test data construction proper as well as the customization and application of a general-purpose test suite to a specific NLP system or domain are laborious, cost-intensive and error-prone tasks, TSNLP put strong emphasis on supplying suitable special-purpose tools to facilitate both the development as well as usage of the TSNLP test data (Oepen et al. (1996a) give an overview).

5.1 Test Data Construction

To ease the time-consuming test data construction and to reduce erratic variations in filling in the TSNLP annotation schema, a graphical test suite construction tool (tsct) was implemented. The tool instantiates the annotation schema (see section 3) as a form-based input mask and provides for (limited) consistency checking of the field values. Additionally, tsct allows reusing previously constructed and annotated data, as quite often — when constructing a series of test items — it can be easier to duplicate and adapt a similar item rather than produce annotations from scratch. For some of the test data a DCG-based test suite generation tool (Arnold et al. (1994)) was deployed to automatically produce systematically varied (i.e. both grammatical and ungrammatical) test items together with some part of the annotations.

5.2 Test Data Maintenance and Retrieval

To implement the TSNLP virtual test suite approach (see section 1), the test data is mounted on a relational database to satisfy the following key desiderata:

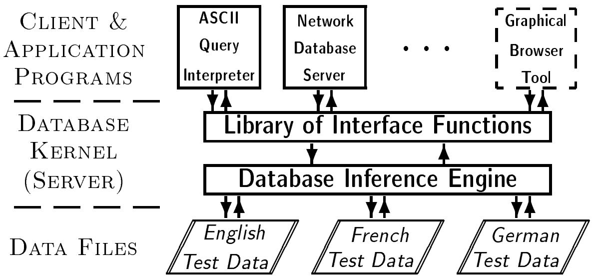

Figure 3: Sketch of the modular tsdb_1 design: the database kernel is separated from client programs through a layer of interface functions.

- **usability**: to facilitate the application of the methodology, technology, and test data developed in TSNLP to a wide variety of diagnosis and evaluation purposes for different applications by developers or users with varied backgrounds;
- **suitability**: to meet the specific necessities of storing and maintaining natural language test data (e.g. in string processing) and to provide maximally flexible interfaces;
- **adaptability and extensibility**: to enable and encourage users of the database to add test data and annotations according to their needs without changes to the underlying data model; and
- **portability and simplicity**: to make the results of TSNLP available on several different hard- and software platforms and easy to use.

To account for the potentially different requirements of NLP developers and users and in order to provide suitable interfaces to human test suite users as well as to external application programs, a dual database implementation was carried out: (i) while a proprietary implementation (called tsdb_1) allowed the fine-tuning of both the query language and interfaces, (ii) a second version (tsdb_2) builds on a commercial database product and, thus, is compliant to common industry standards allowing (industrial) users of the TSNLP test suite to acquire on-site technical support where necessary.

The tsdb_1 implementation is a small and efficient relational database engine in ANSI C. It was designed with an open and documented interface layer (see figure 3) that enables test suite users to bidirectionally link an application being tested to the database and run automated retrieve, process, and compare cycles. Diagnostic results obtained can be stored in the database as part of the *user & application profile* for use in continuous progress evaluation (section 6 gives an example).

An ASCII-based command shell interprets a simplified SQL-style query language and provides editing, completion, and command and query result history. A network database server gives remote (though read-only) access to the test data.

For the alternative implementation tsdb_2 the competitively priced database package Microsoft

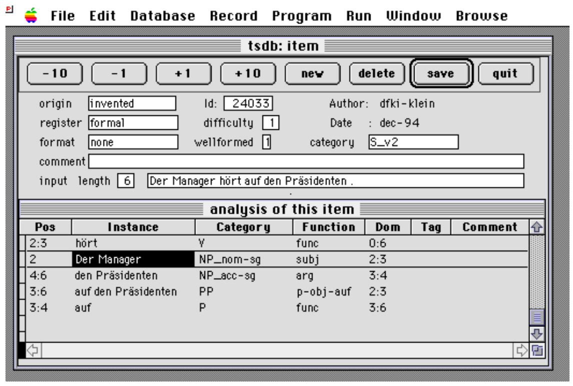

Figure 4: Screen dump of the tsdb_1 *test item* window; the underlying relational database allows parallel browsing and editing of multiple relations.

FoxPro was deployed because it is available for both Apple Macintosh and personal computers running MS Windows[2] and has a very wide distribution. The database provides convenient graphical browsing and editing of the data (using pull-down menus for finite domain fields; see figure 4) as well as standard import and export facilities to exchange data with external applications.

5.3 Query and Retrieval: An Example

To illustrate the capacity and flexibility of the TSNLP annotation schema in conjunction with a relational database retrieval engine, a query example in the simplified SQL-like query language interpreted by tsdb_1 together with an informal English paraphrase is given:[3]

- find all grammatical test items that are associated with the phenomenon of clausal (i.e. subject verb) agreement and have pronominal subjects:

```
select i-id i-input
  where i-wf = 1 &
        p-name = "C_Agreement" &
        a-function = "subj" &
        a-category ~ "^PRON"
```

6 Customization and Testing

To validate the TSNLP annotation methodology, test data, and tools, the project results have been tested against three different application types, viz. a commercial grammar checker for French, a controlled language checker (SECC) for English and a parser (the PAGE system developed at DFKI) for German. As in this setup the evaluation situations ranged from user-level black box evaluation of a commercial product to glass box diagnosis of a research prototype under development (the DFKI system), a number of interesting results were obtained on both the adequacy of the TSNLP approach as well as the quality of the systems being tested.

French Grammar Checker The real life evaluation scenario (i.e. the diagnostic evaluation of a commercial NLP product) enabled Aerospatiale to give a precise account of the type of information obtainable from the use of TSNLP.

The following major performance characteristics were revealed:

- TSNLP ill-formed test items are frequently not detected as such.
- The system performs well on (both well-formed and ill-formed) test items illustrating the phenomenon of agreement, in clauses as well as in noun phrases.
- The system does not master the phenomenon of complementation, especially not in adjectival phrases.
- Sentential test items produce better results than subsentential ones.
- The analysis capabilities of the system are limited (19% of the TSNLP test items were not fully analysed).

The interpretation of the results produced by the system and the comparison with the linguistic information provided in the TSNLP annotations led to an identification of the major shortcomings of the system in terms of systematicity, lexical and morpho-syntactic deficiencies, and interference with other system components.

English Controlled Language Checker Essex tested the controlled language checker SECC (Adriaens (1994)). Like Aerospatiale, Essex was mostly in a black box situation with respect to the system, except that they had access to the controlled grammar language descriptions (but not to the system rules). The testing involved the writing of a large number of customised test items, due to the fact that many CL rules are lexically based, whereas the core test suite concentrates on syntactic phenomena. The testing proved very valuable in highlighting deficiencies in the system performance, as well as in the rule descriptions and gave pointers to the possible source of those errors.

German Parser In connecting the German TSNLP test suite to the DFKI PAGE parser[4] both

[2] Building on the popular database package MS Access, another implementation of the test suite database is currently being developed. This version will provide a similar functionality to tsdb_2 and be available for the MS Windows world.

[3] Additional sample queries and more details on the database schema (including relation and attribute names) can be found in Oepen et al. (1996b) and on the TSNLP World-Wide Web home page `http://tsnlp.dfki.uni-sb.de/tsnlp/`.

[4] The DFKI PAGE (Platform for Advanced Grammar Engineering) system is a state-of-the-art NL core engine and grammar engineering platform; it is in active use at several international research institutions, primarily for HPSG-style grammar development for German, English, Japanese, and Italian.

the test data as well as the TSNLP technology were validated. Building on the C version of the TSNLP database (tsdb_1), a bidirectional interface to the application was established allowing the instantiation of a DFKI user & application profile for the storage of application-specific data (including performance measures and a semantic specification of the expected output).

The seamless coupling between the test suite and the NL system allows running fully automated *retrieve, process, and compare* cycles in the continuous progress evaluation of the grammar and software such that — after making changes to the system — the impact on coverage and performance can be determined in an overnight batch job. The TSNLP test data and database technology proved to be a highly adequate tool for glass-box diagnostic evaluation; besides, the testing experience provided valuable feedback for both the test suite and the application tested (Dauphin et al. (1995b)).

7 Conclusion and Future Work

The TSNLP project has laid the foundations for building large scale reference data for diagnostic and evaluation purposes. The project has produced a substantial set of test items for three different languages, which are based on a systematic and controlled methodology, comprehensively annotated, and embedded in an environment allowing for easy access and maintenance of the data. The approach has been successfully tested against commercial and research NLP applications and components.

However, while this work can be seen as an important step in the right direction, we are very well aware of future developments which will be essential for a widespread acceptance of the system in a broad user community. These developments comprise amongst others further extensions of the test data (possibly taking into account aspects of morphology and discourse), customization tools, which support the adaptation of the test data to specific domains and applications, as well as tools and methods which relate the isolated test items to corpora in order to determine their frequency and relevance. While the members of the project will continue this work, outside developers and users of NLP applications are invited to contribute to these resources which can become a reference standard only if they are truly public domain.

Acknowledgements

In its initial specification and in the early phase of the project, TSNLP was greatly inspired by the conceptional and administrative contributions of Siety Meijer of University of Essex. Additionally, substantial parts of the implementation work at DFKI and the University of Essex have been carried out by Tom Fettig, Fred Oberhauser, and Martin Rondell. We especially want to thank Roger Havenith, the TSNLP project officer at DG XIII, for his help throughout the project and the two external reviewers, Dan Flickinger and John Nerbonne, for their constructive comments and suggestions.

References

Adriaens, Geerd. 1994. SECC: Simplified English Checker and Style Correction in an MT Framework. In *Proceedings of the Language Engineering Convention.* Paris.

Arnold, Doug, Martin Rondell, and Frederik Fouvry. 1994. Design and Implementation of Test Suite Tools. Report to LRE 62-089 D-WP5.1. University of Essex, UK.

Balkan, Lorna, Frederik Fouvry, and Sylvie Regnier-Prost (editors). 1996. TSNLP User Manual. Volume 1: Background, Methodology, Customization, and Testing. Technical report. University of Essex, UK.

Dauphin, Eva, Veronika Lux, Sylvie Regnier-Prost (principal authors), Doug Arnold, Lorna Balkan, Frederik Fouvry, Judith Klein, Klaus Netter, Stephan Oepen, Dominique Estival, Kirsten Falkedal, and Sabine Lehmann. 1995a. Checking Coverage against Corpora. Report to LRE 62-089 D-WP3.2. University of Essex, UK.

Dauphin, Eva, Veronika Lux, Sylvie Regnier-Prost, Lorna Balkan, Frederik Fouvry, Kirsten Falkedal, Stephan Oepen (principal authors), Doug Arnold, Judith Klein, Klaus Netter, Dominique Estival, Kirsten Falkedal, and Sabine Lehmann. 1995b. Testing and Customisation of Test Items. Report to LRE 62-089 D-WP4. University of Essex, UK.

Estival, Dominique, Kirsten Falkedal, Lorna Balkan, Siety Meijer, Sylvie Regnier-Prost, Klaus Netter, and Stephan Oepen. 1994. Survey of Existing Test Suites. Report to LRE 62-089 D-WP1. University of Essex, UK.

Flickinger, Daniel, John Nerbonne, Ivan A. Sag, and Thomas Wassow. 1987. Toward Evaluation of NLP Systems. Technical report. Hewlett-Packard Laboratories. Distributed at the 24^{th} Annual Meeting of the Association for Computational Linguistics (ACL).

Lehmann, Sabine, Dominique Estival, Kirsten Falkedal, Hervé Compagnion, Lorna Balkan, Frederik Fouvry, Judith Baur, and Judith Klein. 1996. TSNLP User Manual. Volume 3: Test Data Documentation. Technical report. Istituto Dalle Molle per gli Studii Semantici e Cognitivi (ISSCO) Geneva, Switzerland.

Nerbonne, John, Klaus Netter, Kader Diagne, Ludwig Dickmann, and Judith Klein. 1993. A Diagnostic Tool for German Syntax. *Machine Translation* 8:85–107.

Oepen, Stephan, Frederik Fouvry, Klaus Netter, Tom Fettig, and Fred Oberhauser. 1996a. TSNLP User Manual. Volume 2: Core Test Suite Technology. Technical report. Deutsches Forschugnszentrum für Künstliche Intelligenz (DFKI) Saarbrücken, Germany.

Oepen, Stephan, Klaus Netter, and Judith Klein. 1996b. TSNLP — Test Suites for Natural Language Processing. In *Linguistic Databases*, ed. John Nerbonne. CSLI Lecture Notes. Center for the Study of Language and Information. forthcoming.