AVENTINUS, GATE and Swedish Lingware

Dimitrios Kokkinakis

Språkdata/Dept. of Swedish Language
Göteborg University
SE 405 30 Göteborg, Sweden
svedk@svenska.gu.se

Keywords: AVENTINUS, GATE, Information Extraction, Swedish Lingware

Abstract

This report presents an outline of the Swedish Lingware components integrated in the Language Engineering application development environment of GATE. This is an ongoing work within the framework of the Language Engineering project AVENTINUS.

AVENTINUS, *Advanced Information System for Multilingual Drug Enforcement,* is a multilingual information processing project financed by the European Union, Language Engineering program, with contract reference LE1-2238 10335/0. The AVENTINUS project is a user- and data-oriented project, addressing the needs of European police and law enforcement agencies on prevention and detection of drug-related offenses, such as money laundering, drug trafficking, drug abuse etc. AVENTINUS is also expected to be extended and used in related to drug enforcement fields such as organized crime.

The report will concentrate on a concise description of the ongoing development, integration and evaluation of Swedish lingware components in GATE. These components as well as the textual material used, that is corpora, of the narcotica subcorpora domain will be discussed.

First, a brief overview of the AVENTINUS project, its goals and characteristics are presented. Then a description of GATE and the information extraction, IE, task of the project will be discussed. Finally, the description of Swedish modules, comprising the necessary lingware for the IE task will be given. These modules and the interaction between them will be illustrated with concrete examples. Furthermore the software/lingware specifications, requirements and, when appropriate, some preliminary performance measures will be given.

1. Introduction

This report presents an outline of Swedish Lingware components integrated in the Language Engineering application development environment of GATE. This is ongoing work within the framework of the Language Engineering project AVENTINUS. First, a brief overview of the AVENTINUS project, its goals and characteristics are presented. Then a description of GATE and the information extraction, IE, task of the project will be discussed. Finally, the description of Swedish modules, comprising the necessary lingware for the IE task will be given. These modules and the interaction between them will be illustrated with concrete examples.

Furthermore the software/lingware specifications, requirements and, when appropriate, some preliminary performance measures will be given.

2. The AVENTINUS Project

AVENTINUS[1], *Advanced Information System for Multilingual Drug Enforcement*, is a multilingual information processing project financed by the European Union, Language Engineering program, (LE1-2238 10335/0). The AVENTINUS project is a user- and data-oriented project, addressing the needs of European police and law enforcement agencies on prevention and detection of drug-related offenses, such as money laundering, drug trafficking, drug abuse etc., referred to, in the rest of the report, as *drug enforcement*. AVENTINUS is also expected to be extended and used in related to drug enforcement fields such as organized crime. The economical and social impact of such project is evident, drug trafficking, dealing and consumption poses one of the greatest threats to the European nations. The goal of the project is thus to support drug enforcement with multilingual linguistic expertise. The AVENTINUS users should be able to access information in their own native language and receive the results of search requests in their own native language as well, even if the information is derived from foreign language resources.

AVENTINUS focuses on three main areas:

Translation support and tools
term substitution, translation memory and machine translation;

Information processing
components for IE, named entity recognition; intelligent indexing;
template generation, in standard as well as in Interpol forms (user requirements);

Search support
query expansion, transliteration and name similarity.

In the first phase of AVENTINUS, the languages analyzed and processed are German, English and Spanish, while Swedish is part of the second phase.
It should be emphasized that the main task of AVENTINUS is not concentrating on constructing a new full-fledged information system for the drug domain, as the AVENTINUS users have their operational environment in place already. Instead, AVENTINUS provides modules and components which can be linked to and integrated into these environments. Thus, the two predominant features of AVENTINUS are modularity and integratability.

3. GATE/IE and AVENTINUS

3.1 GATE

GATE, *General Architecture for Text Engineering*, Cunningham *et al.* (1995, 1997), Gaizauskas *et al.* (1996b), is an application development environment which supports natural language engineering tasks. GATE has been chosen by AVENTINUS for the information extraction task. In GATE, heterogeneous natural language components/software may be evaluated and refined

[1] More information about the project can be found:

individually or may be combined into larger applications. GATE support both researchers and developers working on component technologies (e.g. tagging, parsing) and those working on developing end-user applications (e.g. information extraction, text summarization). GATE is an integrated collection of tools built upon a standardized model of storage and retrieval developed by the TIPSTER program. TIPSTER is an ARPA-funded program of research and development in information retrieval and extraction, involving the creation of a standard architecture/interfaces for systems dealing with information retrieval and extraction.

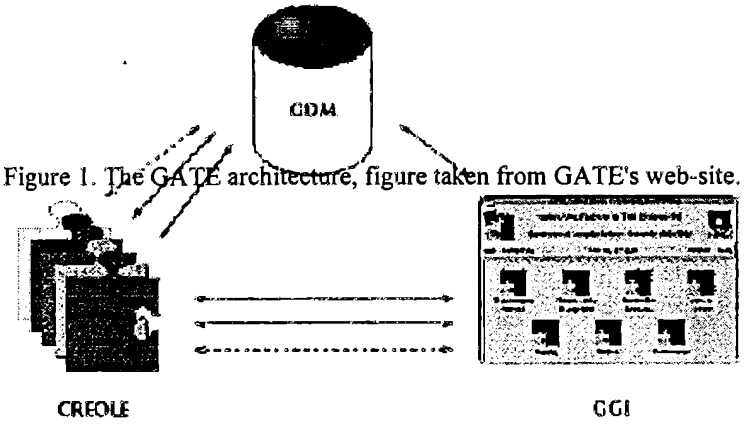

Figure 1. The GATE architecture, figure taken from GATE's web-site. (Site visited 06/11/97)

Figure1. The Gate architecture, figure taken from GATE's web-site. (sote visited 06/11/97

GATE consists of three main parts;

The GATE document manager, **GDM**, based on the TIPSTER document manager, Grishman (1996);

A Collection of REusable Objects for LE, **CREOLE**, which is a set of language engineering modules integrated with the system, such as taggers and parsers;

The GATE Graphical Interface, **GGI**, which provides integrated access to the services of the other components.

3.2 Information Extraction

Information extraction, IE, is the mapping of unseen natural language texts into predefined, fixed-format, structured representations or templates. The templates, when filled, represent an extract of key information from the original text. The resulting data can be further used for intelligent indexing, for storing into a database, for queering or for just displaying it to an end-user. IE systems are often tied to a particular application domain or scenario, and they are computationally as well as knowledge intensive to build.

Four types of IE tasks are identified:

Named entity recognition, identification of all the names of people, places, organizations, dates, amounts of money;

Coreference resolution, identification of identity relations (anaphoric references) between entities in texts;

Template element construction, (TE), association of descriptive information

with the entities, such that "Luleå is_a city in Sweden";

Scenario template construction , linking of TE into events and relation descriptions.

The VIE, *Vanilla IE*, system, following GATE, has been developed at the Univ. of Sheffield. VIE is used for building a model of a text for different purposes, such as MUC-6 tasks, text summarization and for compositionally constructing semantic representations of sentences integrated into a "discourse model". VIE is a modularized version of the lassie system, *Large Scale Information Extraction*, also developed at the Univ. of Sheffield, (Gaizauskas *et al.* 1995). VIE has been the starting point for the development of the Swedish Lingware.

3.3 Multilingual GATE/IE and AVENTINUS

AVENTINUS provides multilingual access to information extracted from texts. Therefore, the IE components have to cope with multiple input languages. This is solved by duplicating parts of the English IE system for each input language. Each language-specific IE system is producing results to each of the AVENTINUS output languages.

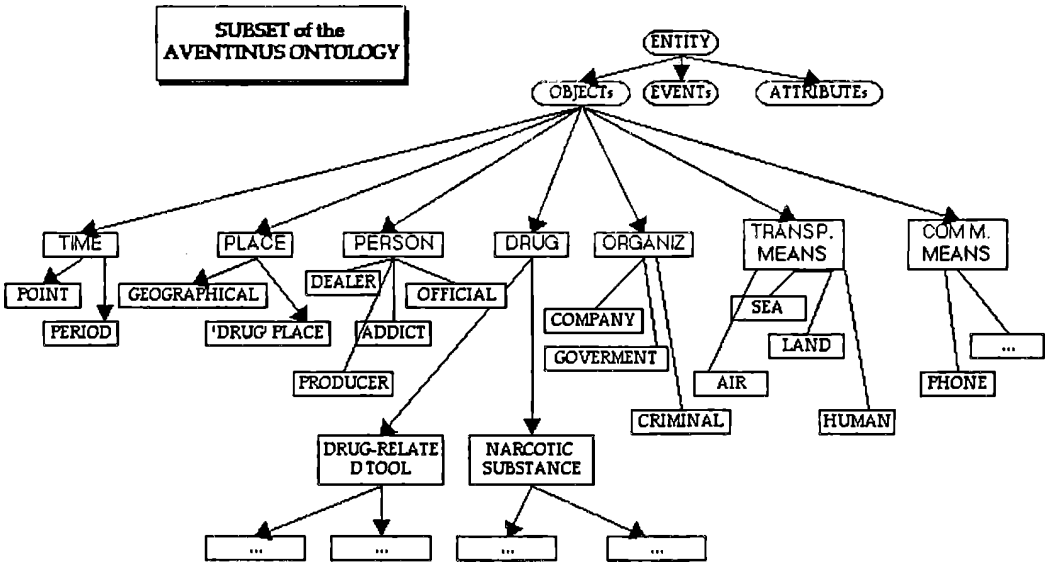

Figure 3. The AVENTINUS ontological objects.

Since AVENTINUS is interested on drug enforcement activities, (cf. Cunningham *et al.* 1996), the entities of interest are persons, communication and transportation means, places, organizations, dates, time sequences and of course drug substances and drug-related tools, (for the latter entities see Lindfors Viklund 1997). A number of relations between these entities, usually in the form of events, is also specified, for instance between people, buyer-seller, between people and companies etc.

4. Swedish Lingware in GATE

4.1 Introduction

The following chapters give a concise description of the ongoing development, integration and evaluation of Swedish lingware components in the language engineering application development environment of GATE. These components as well as the textual material used, that is corpora, of the narcotica subcorpora domain will be discussed. The following stages are carried out by the IE system:

Language Identification	Text type identification
Tokenization	Sentence boundary identification
Part-of-Speech tagging	Morphological analysis
Name entity recognition	Parsing
Coreference resolution	Discourse interpretation

4.2 Corpora, Overview

The text material, corpora, used in the system comprises three different document collections. These are: i)140+ police reports, ii)300+ drug-related newswire-texts downloaded from various Internet web pages and other text collections, and iii)a large number of miscellaneous reports, articles and publications, provided by different sources. The police reports have been provided by the EUROPOL data pool and the Swedish police authorities, Rikspolisstyrelsen. All of them consist of a header identification and a text body. The miscellaneous textual resources consist of machine readable versions of drug related literature, such as the book: "Basfakta om narkotika", provided by SNPF (1996), courtesy of Jonas Hartelius, "Pundartugg", ca 5000 slang words and expressions provided by the author of the book Stefan Holmén (1997), and a large number of on-line publications, reports, wordlists, glossaries and articles about drugs, both encyclopedic and non-encyclopedic.

4.3 Language Identification

4.3.1 Overview

This is a module that uses Mark models for language classification developed by Ted Dunning (1994). No linguistic presuppositions are incorporated for this task. The only assumption used is that a text can be encoded as a set of bytes. The method used is to develop a set of level language models from training data and then to use these language models to estimate the likelihood that a particular test string might have been generated by each of the language models.

4.3.2 Training and Output

For the training of this software, the classifier, a collection of texts for the given language(s) (>5k) is required, after that the creation of a profile file is generated for the given language(s). The output of the classifier is a set of scores plus the profile which had the best match. If a file F is in Swedish, the output will be:

```
"1   swe.3   F  -415.93   -320.63"
```

where 1 is the number of profiles which had best match, swe.3 is the profile with the best match, here the training is based on trigram sequences, F is the file being classified, -415.93 is the score for English and -320.63 is the score for Swedish, assuming that we provide profile files for these two languages.

4.4 Tokenization and Text Type Identification

Tokenization is the process of distinguishing individual tokens and their boundaries in a text-stream and it is of vital interest for lexicon lookup and correct annotation of any kind of labelling, tagging and even for the grammar development. The TOKENIZER module in GATE identifies these boundaries and returns byte offsets to be used as indices in the GDM database. The process is rather advanced and it has the capability to identify over 350 multiword adverbials, prepositions and conjunctions/subjunctions, and a large amount of phrasal verbs. The Text Type Identification (TTI), i.e. if a text is plain, html, email etc., is been carried out at the moment by the tokenizer. This will be a task of an independent module in the future. The TOKENIZER and the TTI are implemented via a set of regular expression patterns which are translated to C++ using flex. The Swedish tokenizer, is following the specifications of the English counterpart, which has been modified in order to account for the Swedish specific set of peculiarities. The TOKENIZER is domain independent. Certain AVENTINUS specific implemented patterns do not influence the analysis of other types of texts. Three major changes have been incorporated into that module.

i) the integration of separate exclusive start states, accounting for the analysis of Swedish newspaper articles and for the analysis of Swedish police reports;
ii) identification of SGML markup;
iii) the integration of the Swedish multiword and phrasal verb recognition.

4.5 Sentence Boundary Identification

4.5.1 Overview

The SENTENCE BOUNDARY IDENTIFICATION (SBI), module identifies sentence start and end byte offsets, making use of SGML sentence markup if present. The module is a variant of the provided English module in GATE. This have been tested and modified accordingly, in order to suit the Swedish needs. The SBI module is implemented as a Perl script.

4.5.2 Processing and Performance

The result of this module is token start and end byte offsets, one pair per line, each followed by the corresponding token string. Newlines from the raw text are also treated as tokens. The byte offset pairs, one per line, represent the sentence start and end positions. The SBI module is reasonably domain independent. Sentence splitting is also affected by the way in which punctuation is handled by the tokenizer, which will vary between different cases. The content of the police reports, for instance, is containing a lot of uppercase tokens, acronyms and abbreviations which require special attention. Some of the stages in SBI are duplicated from the English SBI module, some are modified and some new are added. One of the new stages added is the following:

"A sentence end is assumed if a period is preceded by a sequence of lower case tokens followed by a token starting with a numeral". This is the case of:

"[... bakom smuggling.] [27-åringen häktades ...] "

There is a possibility to analyze texts having SGML markup, in this case, the input is simply searched for "<s>...</s>" pairs, and the corresponding offsets are written out. The performance of the module in the drug enforcement domain is >95% correct identification of sentence boundaries.

4.6 Part-of-Speech Tagging

4.6.1 Overview and Interface

Two taggers for Swedish have been tested in the current system, namely Brill's rule-based part-of-speech tagger, (Brill 1994), and Cooke's semantic tagger, SemanTag, (Cooke 1996). Brill's tagger have been trained in a subset of the SUC corpus, Ejerhed *et al.* (1992), as well as some other newspaper material. For the training in Swedish texts as well as an extensive description of the tagset see Johansson Kokkinakis *et al.* (1996) and Kokkinakis (1997a). The tagset comprises 14 categories, noun, verb etc., as well as SGML markup as a category of its own, 162 tags are used in total. The idea of using SemanTag is mainly threefold. The SemanTag is 9 to 10 times faster than Brill's, there is a possibility of extending the tagger with semantic information and it uses exactly the same resources as the Brill's tagger, so no re-training is necessary.

4.6.2 Processing and Performance

The taggers require four resource files in order to operate: a lexicon, a lexical and a contextual rule list and a list of common bigrams. Various threshold values are decided during training. Tuning flags can be also used. The current lexicon is comprised of 51000 entries, and 750 rules. Multi-word expressions, such as "på_grund_av" are part of the lexicon. The original lexical rule file generated was manually supplemented with rules identifying some common unambiguous endings, not generated during training, for instance "ornas hassuf 5 NCUPGD". The original contextual rule file was also been manually supplemented with rules, for instance with rules distinguishing participles from other verb forms, "VMN0A AP0N(SP)00 PREV1OR2WD blivit". By the addition of these rules the performance has been now improved considerably, after evaluating large Swedish corpora and adding manually rules that resolve that sort of errors. The original performance of 92%-95% correctness is up to 96-99%.
The taggers expect a plain text file as input, formatted with one sentence per line.
Example:

"En 23-årig polsk medborgare häktades i går i Trelleborg misstänkt för grovt narkotikabrott ."

The default output is a version of the input file with a part-of-speech tag appended to each token.
Example:

"En/DIUS0 23-årig/A0PUSNI polsk/A0PUSNI medborgare/NCUPNI häktades/VMISP i_går/R i/S Trelleborg/NP misstänkt/AP0USNI för/S grovt/A0PNSNI narkotikabrott/NCNSNI ./F"

The lexicon used by the taggers is slightly biased towards the drug enforcement domain, because their resource files have been extensively tested on drug related texts. The tagger can be used in other domains with the same lexicon and rule sets, without serious performance drawbacks.

4.7 Annotated and Unannotated Morphological Analysis

4.7.1 Overview

The morphological analysis is producing the stem or root form for each token it is given, plus an affix. This module can be used either with part-of-speech annotated texts or just plain texts and only processing of noun and verb tokens is considered. The morphological analysis is implemented via a set of regular expression patterns which are translated to C++ using flex. These patterns cover over 90% of all the possible patterns for Swedish nouns and verbs, and originate from the analysis and the morphological grouping of the 11th edition of the Swedish Academy word list (SAOL), see Kokkinakis *et al.* (1997). In this analysis 208 morphological classes of verbs and 244 classes of nouns have been distinguished, very unusual or elderly patterns have been omitted for this task, though.

4.7.2 Performance

The input of this module is plain text on standard input with each token optionally annotated by VERB or NOUN to restrict the search time. The performance of the module in the annotated material is over 98% accuracy, while for unannotated texts is considerably lower and it ranges from 70-95%. This depends to a great extend on the ambiguity between identical verbal and noun suffixes, "`ar/er`", as well as homography, "`hamnar/händer`".
Example: "`...föraren har gripits...` => `...förare+n ha+r gripa+its...`"

4.8 Name Entity Recognition, Gazetteer Lookup

The Gazetteer lookup module is trying to identify keywords and phrases related to named entities, as defined for AVENTINUS. This is accomplished by searching a series of pre-stored lists of organizations, locations, person names, date forms, currency names, etc. Most of the lists have been derived from various Internet resources, such as the *Genet Names Server* for a taxonomic list of several thousand of Swedish cities. These lists are translated into a series of finite-state recognisers using flex. The output of this process is a tag and a type attribute for every recognized named entity pattern. Most of the gazetteers are domain independent. Consider for instance a small subset of the content of the title gazetteer for the Swedish police enforcement agencies:

```
Rikspolischef                                   ställföreträdande Rikspolischef
Biträdande (Rikspolischef| Länspolismästare)    Länspolismästare
(byråchef|Rektor) på PHS                        kriminal(kommissarie|inspektör)
polis(kommissarie| inspektör| assistent| aspirant|chefsaspirant|\
            sekreterare| mästare|överintendent|intendent)
```

The performance of this module is over 97% correct identification, while AVENTINUS requires that the quality recognition is >80% in recall and precision.

4.9 Parsing

4.9.1 Overview

Two separate grammars are used for the parsing task. First a named entity grammar is applied to construct proper noun phrases, then a full sentence grammar is applied. The parser is a bottom-up chart parser which builds a semantic representation compositionally, and a best parse algorithm is applied to each final chart, providing a partial parse if no complete sentence span can be constructed. The parser is written in Prolog. There are about 50 rules for the named entities with terminals like `title_NP`, `person_NP`, and 110 for the sentence grammar. The sentence level grammar has been derived by first acquiring a surface level partial grammar which has been manually completed by inspecting the results of applying it on the drug enforcement corpus, Kokkinakis (1997b).

The formal chart/5 term in the grammar is defined as follows:

```
chart(sentence:N1,
      first_token:T1,
      last_token:Tn,
      edges: [edge(Start_token,End_token,
                   Category(f1:v1,f2:v2,...),
            Level,Start_offset,End_offset,Category,Parent,Child,ID)]
      next_edge_number:Ne).
```

Consider for instance the syntactic encoding and representation of the sentence:

```
"SILK skulle vilja använda låttare droger."
chart(sentence_n:1, 1,7,edges:[

       edge(1, 2, pn(s_form:'SILK', m_root:'silk', number:sing),1,...),
       edge(1, 2, list_np(s_form:'SILK', m_root:'silk', ne_tag:person,
                   ne_type:person_last),2,...),
       edge(2, 3, md(s_form:'skulle', m_root:'ska'),1,...),
       edge(3, 4, md(s_form:'vilja', m_root:'vilja'),1,...),
       edge(4, 5, v(s_form:'använda', m_root:'använda', tense:none,
                   voice:active,vform:base),1,...),
       edge(5, 6, adj(s_form:'låttare',m_root:'lätt',degree:_),1,...),
       edge(6, 7, n(s_form:'droger',m_root:'drog',number:plural),1,...),
       edge(7, 8, period(s_form:'.', m_root:'.'),1,...)]
       next_edge_number:8).

syntax 0 37 s
       syntax 0 4 np
              syntax 0 4 basic_np
       syntax 5 37 vp
              syntax 5 24 vpcore
                     syntax 5 24 modal_vpcore
                            syntax 5 11 md
                                                  syntax 12 17 md
                                                  syntax 18 24 nonmodal_vpcore1
                                   syntax 18 24 vpcore1
                                   syntax 18 24 v
              syntax 29 37 np
                     syntax 29 31 adj
                     syntax 32 37 n
```

4.9.2 QLF

The best parse selected during parsing is returned in a predicate-argument representation or quasi-logical form, QLF, for each sentence processed. The QLF produced marks the point where a common representation begins to emerge between the different AVENTINUS languages, Gaizauskas *et al.* (1997). The QLF is just conjunctions of first order logical terms. Some of the

QLF predicates are the unary `city` and `organization`, the binary and language independent `time` and `name`, and the relational binary `logical subject` and `logical object`, Azzam *et al.* (1997). All NP's and VP's lead to the introduction of a unique instance constant in the semantics which serves as an identifier for the object or event referred to in the text.
Consider for instance the semantic/QLF representation of the sentence:

```
"SILK skulle vilja använda lättare droger."
name 0 4 person e5
semantics 0 37
        [name(e5,'SILK'), ne_tag(e5,offsets(0,4)), person(e5),
                realisation(e5,offsets(0,4)), använda(e4), modal(e4,skulle,vilja),
                time(e4,present), aspect(e4,simple), voice(e4,active), lobj(e4,e6),
                drog(e6), number(e6,plural), head(e6,droger), adj(e6,lätt),
                realisation(e6,offsets(29,37)), realisation(e4,offsets(5,37)), lsubj(e4,e5)]
```

4.10 Coreference Resolution

The coreference resolution module is not trying to recognize new proper names but adds identity relations between those found by the parser. Some of the rules are: i) if one of the tokens of one name match a name in a text, then they are identical: "`Barclays Bank/banken, Nattklubben 'Sports'/klubb/klubben`"; ii) Aliases are considered equal: "`Ove Lennart Andersson alias 'Armar och Ben'`".

4.11 Discourse Interpretation

The DISCOURSE INTERPRETER, (DI), module translates the QLF representation produced by the parser into a semantic net representation of instances, their AVENTINUS specific ontological classes and their attributes. The AVENTINUS domain model is using, at the moment, the XI knowledge representation language, Gaizauskas *et al.* 1996a. The XI is written in Prolog. The principal task for the discourse processing module is to integrate the semantic representations of multiple sentences into a single model of the text from which the information required for filling a template may be derived.
This is the only module not been implemented yet for Swedish in any form, i.e. prototypical or fully operational. This depends on the fact that the AVENTINUS domain model or ontology is, at this moment (November 1997), on the last phases of its integration into GATE.

5. Template Generation

The next step of the monolingual IE task is to search the discourse model for all the instances of objects and their (inherited) attributes, which are formatted according the AVENTINUS user specifications, e.g. Interpol forms. This phase is not implemented yet, since it is dependent on the DI. The accuracy required for the template element production in f-measure, i.e. minimum precision and recall is 65%.

6. Conclusion and Further Development

This paper has described the ongoing work for building an information extraction system for Swedish, within the framework of the LE project AVENTINUS. Swedish lingware components have been presented and discussed. The deadline for accomplishing the task is the summer '98, and there is still a lot of work needed in order to meet the AVENTINUS goals stated in the specifications. Nevertheless, there is a strong belief that the time framework allows us for a

successful accomplishment of the task. By mid-summer '98 the system will be operational into unix and NT environments.

Acknowledgment

Many thanks to the Sheffield IE team for all the support regarding the installation of GATE.

References

Azzam S., Cunningham H, Wilks Y., Humphreys K. and Gaizauskas R., (1997), *The AVENTINUS Multilingual QLF Interface*, AVENTINUS internal technical report, Univ. of Sheffield.

Brill E., (1994), *Some Advances In Rule-Based Part of Speech Tagging*, In Proc. of the 12th AAAI '94, Seattle Wa.

Cooke G., (1996), *The SemanTag Project*, http://www.rt66.com/gcooke/SemanTag/. Site visited 07/11/97.

Cunningham H., Gaizauskas R., Wilks Y. (1995) *A General Architecture for Text Engineering (GATE) - A New Approach to Language Engineering R&D*, Techn. rep. CS - 95 - 21, Univ. of Sheffield, Dept of Computer Science, http://www.dcs.shef.ac.uk/research/groups/nlp/gate/. Site visited 06/11/97.

Cunningham H., Azzam S. and Wilks Y., (1996), *Domain Modelling for AVENTINUS, (WP4.2)*, AVENTINUS internal technical report, Univ. of Sheffield.

Cunningham H., Freeman M., Black W.J. (1997) *Software reuse, object-oriented frameworks and NLP*, In New Methods in Language Processing, Jones D. and Somers H. (eds), Studies in Computational Linguistics, UCL Press, pp. 357-366.

Dunning T., (1994), *Statistical Identification of Language*, CRL, New Mexico State University.

Ejerhed E., Källgren G., Wennstedt G. and Åström M., (1992), *The Linguistic Annotation of the Stockholm-Umeå Corpus project*, Technical Report No. 33, Univ. of Umeå.

Gaizauskas R., Wakao T., Humphreys K., Cunningham H., and Wilks Y., (1995) *Description of the LaSIE System as used for MUC-6*, In Proc. of the Sixth Message Understanding Conference (MUC-6), Morgan Kaufmann, 1995, pp. 207-220, http://www.dcs.shef.ac.uk/research/groups/nlp/funded/lasie.html. Site visited 11/11/97.

Gaizauskas R. and Humphreys K., (1996a), *XI: A Simple Prolog-based Language for Cross-Classification and Inheritance*, In Proceedings of the 7th International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA96), Sozopol, Bulgaria, pp. 86-95.

Gaizauskas R., Cunningham H., Wilks W., Rodgers P., and Humphreys K., (1996b) *GATE: An Environment to Support Research and Development in Natural Language Engineering*, In

Proceedings of the 8th IEEE International Conference on Tools with Artificial Intelligence, Toulouse, France.

Gaizauskas R., Humphreys K., Azzam S., Wilks Y., (1997), *Concepticons vs Lexicons: An Architecture for Multilingual Information Extraction*, AVENTINUS internal technical report, Univ. of Sheffield *GEOnet Names Server*, http://www.nima.html/gns/html. Site visited 26/10/97.

Grishman R., (1996), *TIPSTER Text Phase II Architecture Design*, Version 2.3, New York University.

Holmén S., (1997), *Pundartugg, Narkotikarelaterade slanguttryck*, Polishögskolan, Solna.

Johansson-Kokkinakis S., Kokkinakis D., (1996), *Rule-Based Tagging in Språkbanken*, Research Reports from the Department of Swedish, Göteborg University, GU-ISS-96-5.

Kokkinakis D., (1997a), *Linguistic Toolset for Swedish: Tokenisation, Tagging and Lemmatization, (WP4.3)*, AVENTINUS System Specifications, Vol. 2.

Kokkinakis D., (1997b), *Partial Parsing using a mini-lexicalized CFG, as seed to a bottom-up chart parser*, unpublished manuscript.

Kokkinakis D., Johansson-Kokkinakis S., (1997), *A Robust, Modularized Lemmatizer/Tagger for Swedish Based on Large Lexical Resources,* Research Reports from the Department of Swedish, Göteborg University, GU-ISS-97-1.

Lindfors Viklund M., (1997), *Drug Terms*, Dep. of Swedish, Göteborg Univeristy.

SNPF (Svenska Narkotikapolisföreningen), (1996), *Basfakta om narkotika,* Göteborg, Sweden.

TIPSTER, http://cs.nyu.edu/cs/faculty/grishman/tipster.html, and http://www.tipster.org, Sites visited 19/10/97.