MACHINE TRANSLATION REVIEW

The Periodical
of the
Natural Language Translation Specialist Group
of the
British Computer Society
Issue No. 14
December 2003

The *Machine Translation Review* incorporates the Newsletter of the Natural Language Translation Specialist Group of the British Computer Society and appears annually in December.

The Review welcomes contributions, articles, book reviews, advertisements, and all items of information relating to the processing and translation of natural language. Contributions and correspondence should be addressed to:

Derek Lewis
The Editor
Machine Translation Review
School of Modern Languages
Queen's Building
University of Exeter
Exeter
EX4 4QH
United Kingdom

Tel: +44 (0)1392 264264
E-mail: D.R.Lewis@exeter.ac.uk

ISSN 1358–8346

Please note: From April 2000 this Review will be published electronically and will be available on our web site at the British Computer Society (see page 5). The format will be in HTML, in the same way as some back copies have already been stored electronically, so it will be easy for readers to print copies if they wish. Each section will be separate so readers may print selected parts only.

Some copies will be printed for the Copyright libraries and for purchase at a modest price plus postage and packing for those without electronic access. Members of the Natural Language Translation Specialist Group of the British Computer Society will be advised of each issue.

Contents

Group News and Information 4
The Committee 5
BCS Library 5
Website 5

A Latin Morphological Processor
Paul R. Bowden 6

Blitz Latin - A Machine Translator for Latin to English
William A. Whitaker and John F. White 17

Conferences and Workshops 29

Membership 31

Group News and Information

The British Computer Society
Charity No. 292786

THE NATURAL LANGUAGE TRANSLATION SPECIALIST GROUP

Please reply to:
72 Brattle Wood
Sevenoaks
Kent, TN13 1QU
Tel: 01732 455446
E-mail: wiggjd@bcs.org.uk

The Committee

The telephone numbers and e-mail addresses of the Committee are as follows:

David Wigg (Chair & Treasurer)	Tel.: +44 (0)1732 455446 (H)
	Tel.: +44 (0)207 815 7472 (W)
	E-mail: wiggjd@bcs.org.uk
Monique L'Huillier (Secretary)	Tel.: +44 (0)1276 20488 (H)
	Tel.: +44 (0)1784 443243 (W)
	E-mail: m.l'huillier@rhul.ac.uk
Derek Lewis (Editor)	Tel.: +44 (0)1404 814186 (H)
	Tel.: +44 (0)1392 264325 (W)
	E-mail: d.r.lewis@exeter.ac.uk
Roger Harris (Webmaster)	Tel.: +44 (0)208 800 2903 (H)
	E-mail: rwsh@bcs.org.uk
Douglas Clarke	Tel.: +44 (0)1908 373141
Ian Kelly	Tel.: +44 (0)1276 857599
	E-mail: idkk@idkk.com
Veronica Lawson	Tel.: +44 (0)207 7359060
	E-mail: veronica_1@compuserve.com
Correspondent Members:	
Gareth Evans (Minority Languages)	E-mail: gevans@ubs-wsc.org
Ruslan Mitkov	Tel: +44 (0)1902 322471 (W)
	E-mail: r.mitkov@wlv.ac.uk
Heather Fulford	E-mail: h.fulford@lboro.ac.uk
Mark Stevenson	E-mail: m.stevenson@dcs.shef.ac.uk

BCS Library

Books kindly donated by members are passed to the BCS library at the IEE, Savoy Place, London, WC2R 0BL, UK (tel: +44 (0)207 240 1871; fax: +44 (0)207 497 3557). Members of the BCS may borrow books from this library either in person or by post. All they have to provide is their membership number. The library is open Monday to Friday, 9.00 am - 5.00 pm.

Website

The website address of the BCS-NLTSG is: http://www.bcs.org.uk/siggroup/nalatran

A Latin Morphological Processor

by

Paul R. Bowden

School of Computing and Mathematics
The Nottingham Trent University
Email: paul.bowden@ntu.ac.uk

Abstract

The Latin morphological processor used by the BRUTUS Latin-to-English MT system being developed by the author is described. The processor uses the system's Latin lexicon to find base forms and a database of morphology rules to create a set of possible parts-of-speech (PoS) and other features for each Latin word in the input text. The highly inflected nature of Latin means that most Latin words are labelled with only a single PoS, and so the processor provides the basis for an effective PoS tagger for Latin. Features provided depend on PoS, e.g. gender/case/number for nouns, person/number/tense/voice/mood for verbs. All such possibilities are indicated for each possible PoS. The processor relies upon there being a suitable base form present in the lexicon e.g. infinitive forms for verbs, nominative singular forms for nouns etc, each annotated with standard Latin dictionary category information. The processor is highly effective if the base form is present, finding all inflectional possibilities for each Latin word encountered (together with English meanings).

Introduction

There appears to be a current revival in the fortunes of the Latin language, with many UK schools now teaching it once more, even at the primary level (see e.g. the popular primary level text *Minimus the Mouse*, (Bell, 1999)). The BRUTUS Latin MT system has been initially described by (Bowden, 2001) and is currently being developed, partly with pedagogic needs in mind. BRUTUS is a unidirectional MT system (Latin to English direction only, at present). It is a 'direct' MT system i.e. a multistage system starting with lexical transfer, but involving no parsing (Hutchins and Somers, 1992). The system utilises a morphological processor which aims to present all possible senses of each Latin word in the input text, for later disambiguation down to the gender / case / number level (for nouns and adjectives) and person / number / tense / voice / mood level (for verbs). Figure 1 illustrates the output of the processor for each PoS, and represents the current first-stage output of the MT system:

Secunda legio castra magna in Gallia habet, sed in Britanniam cum imperatore festinabit.

Stage 1 output sentence:

Second[ADJV(us,a,um)-FemNomSing,FemVocSing,FemAblSing,NeuNomPlur,NeuVocPlur,NeuAccPlur]
legions[NOUNFemDec3(?,?is)-NomSing,VocSing]
camps[NOUNNeuPlurDec2(a,orum)-NomPlur,VocPlur,AccPlur]
large[ADJV(us,a,um)-FemNomSing,FemVocSing,FemAblSing,NeuNomPlur,NeuVocPlur,NeuAccPlur]
{**in**[PREP-+Abl]^OR^**on**[PREP-+Abl]^OR^**into**[PREP-+Acc]}
Gaul[NOUNPrpDec1(a,ae)-NomSing,VocSing,AblSing]
he/she/it_have[VERB-3rdSingPresIndAct]
,

but[CONJ]
{**in**[PREP-+Abl]^OR^**on**[PREP-+Abl]^OR^**into**[PREP-+Acc]}
Britain[NOUNPrpDec1(a,ae)-AccSing]
with[PREP-+Abl]
generals[NOUNMscDec3(?,?is)-AblSing]
he/she/it_will_hurry[VERB-3rdSingFutrIndAct]
.

Figure 1. Example BRUTUS stage-1 output.

Most of the input Latin words are looked up directly in the lexicon, which contains both base-form entries and inflected-form entries, all annotated as above with information in the square brackets. However, when a Latin word is encountered which has not been seen by BRUTUS before, i.e. which is not in the lexicon, the morphological processor is automatically applied to it. The processor allows the many tens of inflected forms for each Latin base-word to be generated on-the-fly, i.e. it obviates the need for their manual addition to the lexicon. In order to be able to suggest the square-bracketed parts above for any such new word, as well as the initial English sense, the morphological processor requires access to the system lexicon and the database of lemmatisation rules.

BRUTUS also allows an interactive mode in which the user is asked to view the output of the morphological processor and reject any suggestions which are not correct in Latin. This is necessary because on rare occasions a lemmatisation rule can be applied to the Latin word even though this specific lemmatisation is not grammatically correct in Latin. However, this need be done only once for each Latin lexeme, as the results are then stored in the lexicon for immediate lookup in later runs. An example of this situation will be given later.

The morphological processor was initially intended as a means to aid the building of a very large lexicon containing all inflected forms of all Latin words. The philosophy of BRUTUS is to attempt a very shallow direct-MT approach, the first stage of which is word lookup (with presentation of several senses if these exist – see e.g. the ^OR^ parts in the prepositions in Figure 1.). Later processing stages are intended to refine the first-stage output (Bowden, 2001). However, the processor does identify all the lemmatisations which *could* be applied to any given Latin lexeme, and so is able to suggest a PoS even for unknown base-form words. Thus it may in the future provide the basis for a Latin PoS tagger, even without any Latin lexicon. However, further PoS tag disambiguation stages would be required in such a tagger, since Latin words in different PoS categories may end with the same character(s).

Lexicon Entries and Morphological Rules

Extracts from the system lexicon are given in Figure 2. Some of the entries were added manually, and some by the morphological processor. The manually added entries include those having -NomSing (for nouns and adjectives) and –PresInfAct (for verbs), which are the primary base forms towards which the morphological processor works. (Positive adverbs and prepositions are also manual additions.) Other base forms are also required (discussed later).

```
ad===at[PREP-+Acc]
ab===from[PREP-+Abl]
per===through[PREP-+Acc]
ad===towards[PREP-+Acc]
agricola===farmers[NOUNMscDec1(a,ae)-NomSing,VocSing,AblSing]
exitus===ends[NOUNMscDec4(us,us)-  NomSing,VocSing,GenSing,NomPlur,VocPlur,AccPlur]
cornu===horns[NOUNNeuDec4(u,us)-NomSing,VocSing,AccSing,DatSing,AblSing]
```

```
domus===houses[NOUNFemDec4Irg(us,us)-NomSing,VocSing,GenSing,NomPlur,VocPlur,AccPlur]
urbs===towns[NOUNFemDec3(?,?is)-NomSing,VocSing]
urbis===towns[NOUNFemDec3(?,?is)-GenSing]
navis===ships[NOUNFemDec3(?,?is)-NomSing,VocSing,GenSing]
classis===fleets[NOUNFemDec3(?,?is)-NomSing,VocSing,GenSing]
canis===dogs[NOUNMscDec3(?,?is)-NomSing,VocSing,GenSing]
canis===dogs[NOUNFemDec3(?,?is)-NomSing,VocSing,GenSing]
bos===oxen[NOUNMscDec3Irg(?,?is)-NomSing]
rex===kings[NOUNMscDec3(?,?is)-NomSing,VocSing]
regis===kings[NOUNMscDec3(?,?is)-GenSing]
dolor===pains[NOUNMscDec3(?,?is)-NomSing,VocSing]
doloris===pains[NOUNMscDec3(?,?is)-GenSing]
amabam===I_was_love[VERB-1stSingImpfIndAct]
amabatis===you_were_love[VERB-2ndPlurImpfIndAct]
amabimus===we_will_love[VERB-1stPlurFutrIndAct]
amabo===I_will_love[VERB-1stSingFutrIndAct]
amabunt===they_will_love[VERB-3rdPlurFutrIndAct]
amamus===we_love[VERB-1stPlurPresIndAct]
amant===they_love[VERB-3rdPlurPresIndAct]
amare===to_love[VERB-PresInfAct]
facere===to_make[VERB-PresInfAct]
facere===to_do[VERB-PresInfAct]
deducere===to_launch[VERB-PresInfAct]
deduxi===I_launch[VERB-1stSingPerfIndAct]
deductum===launch[VERB-SpnAcc/PARTICPerfPas-MscAccSing,NeuNomSing,NeuVocSing,NeuAccSing]
regere===to_rule[VERB-PresInfAct]
rectum===rule[VERB-SpnAcc/PARTICPerfPas-MscAccSing,NeuNomSing,NeuVocSing,NeuAccSing]
communicare===to_share[VERB-PresInfAct]
navigare===to_sail[VERB-PresInfAct]
appellare===to_call[VERB-PresInfAct]
amas===you_love[VERB-2ndSingPresIndAct]
amaveram===I_had_love[VERB-1stSingPlupIndAct]
amaverint===they_will_have_love[VERB-3rdPlurFperIndAct]
amavi===I_love[VERB-1stSingPerfIndAct]
```

Figure 2. Extracts (non-contiguous) from the BRUTUS lexicon.

Note that nouns are classified by PoS (NOUN), gender (Msc,Fem,Neu and Prp for Proper), declension number (e.g. Dec1) and "how they go" (e.g. a,ae). The base form, which includes the string -NomSing, also indicates the other case/number combinations which the base lexeme might also indicate. (Third declension 'increasing' nouns and adjectives also need the GenSing as a base form.) The English sense is always given as a plural, as BRUTUS has available to it a function to create the English singular form as necessary (Bowden et al., 1996). For example, here is the entry for *agricola*:

agricola===farmers[NOUNMscDec1(a,ae)-NomSing,VocSing,AblSing]

Verbs have a simpler base-form entry. They are merely labelled as VERB-PresInfAct. No conjugation number is given (perhaps surprisingly), as this has not yet proved to be vital (but see later comments). (In addition, the 1stSingPerfIndAct is used to cover verbs having a distinct perfect stem, e.g. resurgo, *resurrexi*.) Here is the entry for *numerare*:

numerare===to_count[VERB-PresInfAct]

These lexicon entries, together with the morphological rulebase, allow the morphological processor to explain lexemes such as *agricolae* and *numero*, if they are not already present in the lexicon as separate entries.

Figure 3 gives examples of noun rules from the morphological rulebase, and Figure 4 gives some of the verb rules. In all cases, the notation used is **x**>>>**y**===**text**<<<[…] where **x**

represents the ending characters of the Latin word being investigated, **y** the ending with which **x** is replaced by the processor, and **text** some characters which need to be inserted in the created meaning (used for verbs but not nouns). In the case of nouns, the base form rule always has **x** the same as **y**, and so this rule is not used at present, but has been included for completeness and because of potential future applications (e.g. identifying PoS in a Latin PoS tagger).

```
//-------------------------------------------------------------------
// NOUNS
//-------------------------------------------------------------------
//
// 1st DECLENSION
//
// Note: we morph to the Nom Sing, so we only need this
//       base form in the vocab file.
//
// e.g. mensa===tables[NOUNFemDec1(a,ae)-NomSing,VocSing,AblSing]
//
// mensa
// inaccessible rule: a>>>a===<<<[NOUNFemDec1(a,ae)-NomSing,VocSing,AblSing]
am>>>a===<<<[NOUNFemDec1(a,ae)-AccSing]
ae>>>a===<<<[NOUNFemDec1(a,ae)-GenSing,DatSing,NomPlur,VocPlur]
as>>>a===<<<[NOUNFemDec1(a,ae)-AccPlur]
arum>>>a===<<<[NOUNFemDec1(a,ae)-GenPlur]
is>>>a===<<<[NOUNFemDec1(a,ae)-DatPlur,AblPlur]
abus>>>a===<<<[NOUNFemDec1(a,ae)-DatPlur,AblPlur]
//
// nauta
// inaccessible rule: a>>>a===<<<[NOUNMscDec1(a,ae)-NomSing,VocSing,AblSing]
am>>>a===<<<[NOUNMscDec1(a,ae)-AccSing]
ae>>>a===<<<[NOUNMscDec1(a,ae)-GenSing,DatSing,NomPlur,VocPlur]
as>>>a===<<<[NOUNMscDec1(a,ae)-AccPlur]
arum>>>a===<<<[NOUNMscDec1(a,ae)-GenPlur]
is>>>a===<<<[NOUNMscDec1(a,ae)-DatPlur,AblPlur]
//
//-------------------------------------------------------------------------
// 2nd DECLENSION
//
// Note: we morph to the Nom Sing, so we only need this
//       base form in the vocab file.
//
// e.g. dominus===masters[NOUNMscDec2(us,i)-NomSing]
//
// dominus
// inaccessible rule: us>>>us===<<<[NOUNMscDec2(us,i)-NomSing]
e>>>us===<<<[NOUNMscDec2(us,i)-VocSing]
um>>>us===<<<[NOUNMscDec2(us,i)-AccSing]
i>>>us===<<<[NOUNMscDec2(us,i)-GenSing,NomPlur,VocPlur]
o>>>us===<<<[NOUNMscDec2(us,i)-DatSing,AblSing]
os>>>us===<<<[NOUNMscDec2(us,i)-AccPlur]
orum>>>us===<<<[NOUNMscDec2(us,i)-GenPlur]
is>>>us===<<<[NOUNMscDec2(us,i)-DatPlur,AblPlur]
//
// filius
// inaccessible rule: ius>>>ius===<<<[NOUNMscDec2(ius,i)-NomSing]
ium>>>ius===<<<[NOUNMscDec2(ius,i)-AccSing]
i>>>ius===<<<[NOUNMscDec2(ius,i)-VocSing,GenSing,NomPlur,VocPlur]
ii>>>ius===<<<[NOUNMscDec2(ius,i)-GenSing]
io>>>ius===<<<[NOUNMscDec2(ius,i)-DatSing,AblSing]
ios>>>ius===<<<[NOUNMscDec2(ius,i)-AccPlur]
iorum>>>ius===<<<[NOUNMscDec2(ius,i)-GenPlur]
iis>>>ius===<<<[NOUNMscDec2(ius,i)-DatPlur,AblPlur]
//
// ager
// inaccessible rule: er>>>er===<<<[NOUNMscDec2(er,ri)-NomSing,VocSing]
rum>>>er===<<<[NOUNMscDec2(er,ri)-AccSing]
ri>>>er===<<<[NOUNMscDec2(er,ri)-GenSing,NomPlur,VocPlur]
ro>>>er===<<<[NOUNMscDec2(er,ri)-DatSing,AblSing]
ros>>>er===<<<[NOUNMscDec2(er,ri)-AccPlur]
rorum>>>er===<<<[NOUNMscDec2(er,ri)-GenPlur]
ris>>>er===<<<[NOUNMscDec2(er,ri)-DatPlur,AblPlur]
//
// vir
// inaccessible rule: ir>>>ir===<<<[NOUNMscDec2(ir,iri)-NomSing,VocSing]
irum>>>ir===<<<[NOUNMscDec2(ir,iri)-AccSing]
iri>>>ir===<<<[NOUNMscDec2(ir,iri)-GenSing,NomPlur,VocPlur]
```

```
iro>>>ir===<<<[NOUNMscDec2(ir,iri)-DatSing,AblSing]
iros>>>ir===<<<[NOUNMscDec2(ir,iri)-AccPlur]
irorum>>>ir===<<<[NOUNMscDec2(ir,iri)-GenPlur]
iris>>>ir===<<<[NOUNMscDec2(ir,iri)-DatPlur,AblPlur]
//
// puer
// inaccessible rule: er>>>er===<<<[NOUNMscDec2(er,eri)-NomSing,VocSing]
erum>>>er===<<<[NOUNMscDec2(er,eri)-AccSing]
eri>>>er===<<<[NOUNMscDec2(er,eri)-GenSing,NomPlur,VocPlur]
ero>>>er===<<<[NOUNMscDec2(er,eri)-DatSing,AblSing]
eros>>>er===<<<[NOUNMscDec2(er,eri)-AccPlur]
erorum>>>er===<<<[NOUNMscDec2(er,eri)-GenPlur]
eris>>>er===<<<[NOUNMscDec2(er,eri)-DatPlur,AblPlur]
//
// bellum
// inaccessible rule: um>>>um===<<<[NOUNNeuDec2(um,i)-NomSing,VocSing,AccSing]
i>>>um===<<<[NOUNNeuDec2(um,i)-GenSing]
o>>>um===<<<[NOUNNeuDec2(um,i)-DatSing,AblSing]
a>>>um===<<<[NOUNNeuDec2(um,i)-NomPlur,VocPlur,AccPlur]
orum>>>um===<<<[NOUNNeuDec2(um,i)-GenPlur]
is>>>um===<<<[NOUNNeuDec2(um,i)-DatPlur,AblPlur]
//
//---------------------------------------------------------------------------
// 3rd DECLENSION
//
// Dec3 increasing nouns: we morph to the genitive singular, and the vocab
// file has to contain two entries for these type of nouns: one for Nom/Voc
// (and Acc for Neu nouns), and the Gen Sing. This turns the problem into
// a vocab task (which it really is) rather than a morph task. Non-increasing
// nouns need only the one entry if the Nom is the same as the Gen.
//
// e.g. rex===kings[NOUNMscDec3(?,?is)-NomSing,VocSing]
//    regis===kings[NOUNMscDec3(?,?is)-GenSing]
//
// e.g. civis===citizens[NOUNMscDec3(?,?is)-NomSing,VocSing,GenSing]
//
// e.g. cubile===sofas[NOUNNeuDec3(?,?is)-NomSing,VocSing,AccSing,AblSing]
//     cubilis===sofas[NOUNNeuDec3(?,?is)-GenSing]
// (unusual, because NomSing is same as AblSing, hence latter in first entry)
//
// Masculine
// ?>>>is===<<<[NOUNMscDec3(?,?is)-NomSing,VocSing]
em>>>is===<<<[NOUNMscDec3(?,?is)-AccSing]
// is>>>is===<<<[NOUNMscDec3(?,?is)-GenSing]
i>>>is===<<<[NOUNMscDec3(?,?is)-DatSing]
e>>>is===<<<[NOUNMscDec3(?,?is)-AblSing]
es>>>is===<<<[NOUNMscDec3(?,?is)-NomPlur,VocPlur,AccPlur]
um>>>is===<<<[NOUNMscDec3(?,?is)-GenPlur]
ium>>>is===<<<[NOUNMscDec3(?,?is)-GenPlur]
ibus>>>is===<<<[NOUNMscDec3(?,?is)-DatPlur,AblPlur]
//
// Feminine - as masculine
// ?>>>is===<<<[NOUNFemDec3(?,?is)-NomSing,VocSing]
em>>>is===<<<[NOUNFemDec3(?,?is)-AccSing]
// is>>>is===<<<[NOUNFemDec3(?,?is)-GenSing]
i>>>is===<<<[NOUNFemDec3(?,?is)-DatSing]
e>>>is===<<<[NOUNFemDec3(?,?is)-AblSing]
es>>>is===<<<[NOUNFemDec3(?,?is)-NomPlur,VocPlur,AccPlur]
um>>>is===<<<[NOUNFemDec3(?,?is)-GenPlur]
ium>>>is===<<<[NOUNFemDec3(?,?is)-GenPlur]
ibus>>>is===<<<[NOUNFemDec3(?,?is)-DatPlur,AblPlur]
//
// Neuter
// ?>>>is===<<<[NOUNNeuDec3(?,?is)-NomSing,VocSing,AccSing]
// is>>>is===<<<[NOUNNeuDec3(?,?is)-GenSing]
i>>>is===<<<[NOUNNeuDec3(?,?is)-DatSing]
e>>>is===<<<[NOUNNeuDec3(?,?is)-AblSing]
a>>>is===<<<[NOUNNeuDec3(?,?is)-NomPlur,VocPlur,AccPlur]
ia>>>is===<<<[NOUNNeuDec3(?,?is)-NomPlur,VocPlur,AccPlur]
um>>>is===<<<[NOUNNeuDec3(?,?is)-GenPlur]
ium>>>is===<<<[NOUNNeuDec3(?,?is)-GenPlur]
ibus>>>is===<<<[NOUNNeuDec3(?,?is)-DatPlur,AblPlur]
//
```

Figure 3. Some NOUN rules from the BRUTUS morphological ruleset.

```
//---------------------------------------------------------------------
// VERBS
//---------------------------------------------------------------------
// Most rules go back to the PresInfAct as the base form, but because
```

```
// of irregular verbs, we go back to the 1stSingPerfIndAct (the 3rd
// principle part of the verb) for perfect and other tenses. A few
// irregulars are built in, though.
//-----------------------------------------------------------------
// PRESENT
// amo
o>>>are===I_<<<[VERB-1stSingPresIndAct]
as>>>are===you_<<<[VERB-2ndSingPresIndAct]
at>>>are===he/she/it_<<<[VERB-3rdSingPresIndAct]
amus>>>are===we_<<<[VERB-1stPlurPresIndAct]
atis>>>are===you_<<<[VERB-2ndPlurPresIndAct]
ant>>>are===they_<<<[VERB-3rdPlurPresIndAct]
//
// moneo
eo>>>ere===I_<<<[VERB-1stSingPresIndAct]
es>>>ere===you_<<<[VERB-2ndSingPresIndAct]
et>>>ere===he/she/it_<<<[VERB-3rdSingPresIndAct]
emus>>>ere===we_<<<[VERB-1stPlurPresIndAct]
etis>>>ere===you_<<<[VERB-2ndPlurPresIndAct]
ent>>>ere===they_<<<[VERB-3rdPlurPresIndAct]
//
// ?eo
eo>>>ire===I_<<<[VERB-1stSingPresIndAct]
es>>>ire===you_<<<[VERB-2ndSingPresIndAct]
et>>>ire===he/she/it_<<<[VERB-3rdSingPresIndAct]
emus>>>ire===we_<<<[VERB-1stPlurPresIndAct]
etis>>>ire===you_<<<[VERB-2ndPlurPresIndAct]
eunt>>>ire===they_<<<[VERB-3rdPlurPresIndAct]
//
//rego, capio
o>>>ere===I_<<<[VERB-1stSingPresIndAct]
io>>>ere===I_<<<[VERB-1stSingPresIndAct]
is>>>ere===you_<<<[VERB-2ndSingPresIndAct]
it>>>ere===he/she/it_<<<[VERB-3rdSingPresIndAct]
imus>>>ere===we_<<<[VERB-1stPlurPresIndAct]
itis>>>ere===you_<<<[VERB-2ndPlurPresIndAct]
unt>>>ere===they_<<<[VERB-3rdPlurPresIndAct]
iunt>>>ere===they_<<<[VERB-3rdPlurPresIndAct]
//
//audio
io>>>ire===I_<<<[VERB-1stSingPresIndAct]
is>>>ire===you_<<<[VERB-2ndSingPresIndAct]
it>>>ire===he/she/it_<<<[VERB-3rdSingPresIndAct]
imus>>>ire===we_<<<[VERB-1stPlurPresIndAct]
itis>>>ire===you_<<<[VERB-2ndPlurPresIndAct]
iunt>>>ire===they_<<<[VERB-3rdPlurPresIndAct]
//
//-----------------------------------------------------------------
// amem
em>>>are===I_<<<[VERB-1stSingPresSubAct]
es>>>are===you_<<<[VERB-2ndSingPresSubAct]
et>>>are===he/she/it_<<<[VERB-3rdSingPresSubAct]
emus>>>are===we_<<<[VERB-1stPlurPresSubAct]
etis>>>are===you_<<<[VERB-2ndPlurPresSubAct]
ent>>>are===they_<<<[VERB-3rdPlurPresSubAct]
//
// moneam
eam>>>ere===I_<<<[VERB-1stSingPresSubAct]
eas>>>ere===you_<<<[VERB-2ndSingPresSubAct]
eat>>>ere===he/she/it_<<<[VERB-3rdSingPresSubAct]
eamus>>>ere===we_<<<[VERB-1stPlurPresSubAct]
eatis>>>ere===you_<<<[VERB-2ndPlurPresSubAct]
eant>>>ere===they_<<<[VERB-3rdPlurPresSubAct]
//
// regam
am>>>ere===I_<<<[VERB-1stSingPresSubAct]
as>>>ere===you_<<<[VERB-2ndSingPresSubAct]
at>>>ere===he/she/it_<<<[VERB-3rdSingPresSubAct]
amus>>>ere===we_<<<[VERB-1stPlurPresSubAct]
atis>>>ere===you_<<<[VERB-2ndPlurPresSubAct]
ant>>>ere===they_<<<[VERB-3rdPlurPresSubAct]
//
// capiam
iam>>>ere===I_<<<[VERB-1stSingPresSubAct]
ias>>>ere===you_<<<[VERB-2ndSingPresSubAct]
iat>>>ere===he/she/it_<<<[VERB-3rd1stSingPresSubAct]
iamus>>>ere===we_<<<[VERB-1stPlurPresSubAct]
iatis>>>ere===you_<<<[VERB-2ndPlurPresSubAct]
iant>>>ere===they_<<<[VERB-3rdPlurPresSubAct]
//
```

Figure 4. Some VERB rules from the BRUTUS morphological ruleset.

Examples of Morphological Processing

I shall now describe the action of the morphological processor. Two examples follow: for a verb, and for a noun.

VERB example:

STEP (1) The word **laudaverant** is encountered in the input, but is not in the lexicon. However, the following entry exists:

laudare===to_praise[VERB-PresInfAct]

STEP (2) The system looks in the morphological rules file to see if it can reduce the encountered verb *laudaverant* to the base form (the infinitive, in the case of verbs) *laudare*, 'to praise'. The following morphological rule is found:

averant>>>are===they_had_<<<[VERB-3rdPlurPlupIndAct]

This says that the ending *–averant* can be reduced to the infinitive stem *–are* where the verb is the 3rd person plural, pluperfect tense, indicative mood, active voice form. In this case the English verb would start "they had". STEP (3) The morphological processor then creates a possible meaning for laudaverant as follows:

laudaverant===they_had_praise[VERB-3rdPlurPlupIndAct]

Note that the word *praise* has been inserted. (A later processing stage changes this to praised .) STEP (4) The user is then asked to inspect the above suggestion. If the user confirms it as being correct, it is added to the lexicon for immediate and future use.

NOUN example:

STEP (1) The word **tabernam** is encountered in the input, but is not in the lexicon. However, the following entry exists:

taberna===inns[NOUNFemDec1(a,ae)-NomSing,VocSing,AblSing]

This states that *taberna* is a first declension feminine noun like those having Nominative singular ending *–a* and Genitive singular ending *–ae*, and that the word *taberna* might be Nominative, Vocative or Ablative singular. Note that only the plural version of the English noun is stored (inns), since BRUTUS contains a reliable function to create the singular form if this is later found to be needed. STEP (2) The system looks in the morphological rules file to see if *tabernam* might be reduced to *taberna*. The rules file is found to contain an entry for the same type of noun:

am>>>a===<<<[NOUNFemDec1(a,ae)-AccSing]

STEP (3) Applying this rule does indeed reduce *tabernam* to *taberna*, so the morphological processor suggests the meaning for *tabernam* is as follows:

tabernam===inns[NOUNFemDec1(a,ae)-AccSing]

STEP (4) When confirmed by the user, this is then added to the vocab file and can be used immediately. When used, since it can only be Accusative Singular (according to the square-bracketed part), *inns* is changed to *inn*.

Discussion

The morphological ruleset is largely complete. One of the problems addressed in the early stages of the design of the morphological processor concerned the handling of 'increasing' 3rd declension nouns, such as gladiator, gladiatoris and lex, legis. The problem here is that the nominative singular has a shorter form than the genitive singular and that this shorter form is not easily predictable and hence is not easily generated by a few morphological rules. Although there are groups of nouns having similar patterns (e.g. rex*,* regis is like lex*,* legis*)* the problem is that there are many tens of such patterns. Each pattern (-o,-onis; -as,-atis; -as,-adis; -ens,-entis; -ex,-egis; -ex,-icis; -s,-ris; -s,-ssis; -ix,-icis; -or,-oris; -en,-inis etc) would require its own ruleset (for each gender), and every time a new pattern was encountered, not only would the lexicon entry be required but also a new morphological ruleset to go with it. On the other hand, most of the 3rd declension endings themselves are quite regular (those from the accusative singular onwards), and could be described by only three rulesets, one for each gender. It was realised that this problem is not really a morphological problem – it is actually a vocabulary problem (i.e. knowing what the NomSing looks like). Therefore it was decided to morph back to the genitive singular form for the 'increasing' 3rd declension nouns, and have two entries provided in the lexicon, e.g. one for lex and one for legis. Happily, this approach also covers the non-increasing 3rd declension nouns, such as clades, cladis*,* and in some cases only requires one entry in the lexicon e.g. for *civis, civis.*

The ruleset also contains all the "standard" rules for the conjugation of all the main categories of verb (in all their tenses, voices and moods, as well as participles, supines, gerunds etc - there are about 1,500 individual rules for verbs) and for adjective declensions. Adverbs are also handled (comparative and superlative forms morphing back to positives). As such the ruleset encapsulates almost all of the inflectional grammar of Latin. This represents a useful teaching resource in its own right, and in fact the potential pedagogical aspects of BRUTUS were a motivating factor for the research right from the start. The Stage-1 output (see Figure 1) is intended to be helpful to learners of Latin, in that it gives *all* the possible meanings for each Latin word. Particularly useful for a learner is that this includes all the possible number/case possibilities for the encountered Latin noun, for example.

The morphological processor does sometimes suggest bad lemmatisations. The system may suggest more than one meaning for any unknown Latin word, because there may be multiple morphological rules that can be applied, *or* multiple vocab entries e.g. where one stem has different meanings, *or both* of these may apply. It is possible for one of the suggestions to be wrong. For example, here are two suggestions made for the same Latin word:

reges===you_will_rule[VERB-2ndSingFutrIndAct]
reges===kings[NOUNMscDec3(?,?is) –NomPlur,VocPlur,AccPlur]

Both of these suggestions are correct. However, there was in fact a third suggestion for *reges* along with the above two:

reges===you_rule[VERB-2ndSingPresIndAct] *WRONG!*

This bad suggestion was made because currently the system does not store information about which *conjugation* a verb is in. It is interesting to note that the lexical information given in

both traditional and more modern Latin dictionaries and textbooks is usually good enough to prevent errors such as that above (e.g. by indicating first person singular present indicative active form of the verb (or the PresInfAct), plus the conjugation number, perfect form and supine.) See e.g. (Balme and Morwood, 1996), (Bell, 1999), (Jones, 1997), (Jones and Sidwell, 1986), (Paterson and Macnaughton, 1968), (Morwood, 2001). However, the error situation illustrated above has been rare and so unless its frequency of occurrence increases as the lexicon grows, the current arrangement with verbs (using the infinitive) will remain. (The advantage of the current system is that it is not necessary to divide the VERB category into five separate categories VERBCnj1, VERBCnj2, VERBCnj3, VERBCnj4, VERBMixd and so the number of verb morphological rules can be kept lower than that required for the latter situation, and the morphological processor is kept as simple as possible.) Furthermore, since this situation always arises with two conflicting VERB meanings[1] being suggested, the morphological processor itself can detect its occurrence. Therefore the possibility of human error can be signalled to the user in the form of an extra 'caution' message.

The description given above concerning the working of the morphological processor with respect to what needs to be in the lexicon (as a base form) has omitted some detail e.g. for some 3rd declension nouns and for irregular verbs. More detail is given in the Appendix, which contains the header of the lexicon file explaining what is required for every part of speech.

Conclusion

The morphological processor built into the BRUTUS system has the potential to aid in the construction of a very large look-up lexical transfer stage for a Latin MT system. In addition, it is capable of forming the basis for a Latin part-of-speech tagger, even in the absence of any lexicon. However, its successful use depends upon (a) the existence of base-form lexemes in the lexicon, (b) existence of the correct lemmatisation rule in the rulebase, and (c) an expert (Latin-fluent) human user to confirm suggested lemmatisations, as these are incorrect in a very few cases.

Future experiments will examine the morphological processor's ability to PoS tag Latin texts in the absence of a lexicon. In addition, missing rules will be added as necessary (which will mostly be at the same time as the addition of new vocabulary). Vocabulary building will also continue as a separate activity alongside the addition of new rules. It is hoped also that BRUTUS will eventually become a useful teaching aid, particularly for school pupils. To this end, it is planned to provide a website so that single Latin sentences may be submitted, and the useful Stage 1 output returned to the student. The morphological processor is central to these aims.

References

Balme, M. and Morwood, J. (1996) *Oxford Latin Course Vol. I.* Oxford University Press

Bell, B. (1999) *Minimus the Mouse*. Cambridge University Press

[1] The problem in fact only occurs for the 2nd/3rd persons singular and 1st/2nd/3rd persons plural for 2nd and 3rd conjugation verbs, where the endings for 2nd conjugation present tense and 3rd conjugation future tense are the same, i.e. –es, -et, -emus, -etis, -ent. The situation arises because both 2nd and 3rd conjugation verbs have present infinitives ending *–ere*.

Bowden, P. R. (2001) 'Latin to English Machine Translation – A Direct Approach'. In *Machine Translation Review*, December 2001 (online via BCS Natural Language Translation Specialist Group at http://www.bcs.org.uk/siggroup/nalatran/mtreview/mtr-12/5.htm)

Bowden, P. R., Halstead, P. and Rose, T. G. (1996) 'Dictionaryless English Plural Noun Singularisation Using A Corpus-Based List of Irregular Forms'. In *Corpus-based Studies in English - Papers from the Seventeenth International Conference on English Language Research on Computerized Corpora (ICAME 17)* Stockholm, May 15 - 19 1996 (Rodopi)

Hutchins, W. J. and Somers, H. L. (1992) *An Introduction to Machine Translation.* Academic Press

Jones, P. (1997) *Learn Latin – The Book of The Daily Telegraph QED Series.* Duckworth

Jones, P. V. and Sidwell, K. C. (1986) *Reading Latin – Grammar, Vocabulary and Exercises.* Cambridge University Press

Morwood, J. (Ed.) (2001) *Pocket Oxford Latin Dictionary (2nd edition).* Oxford University Press

Paterson, J. and Macnaughton, E. G. (1968) *The Approach to Latin (First Part)* (revised 1968). Oliver and Boyd

Appendix

This is the explanatory header for the BRUTUS lexicon:

```
//-----------------------------------------------------------------
// Brutus' VOCAB FILE
//
// Paul R. Bowden
//
// A.D. MMII
//
//-----------------------------------------------------------------
// WARNING! DO NOT SORT THIS FILE!
//-----------------------------------------------------------------
//
// ENGLISH NOUNS MUST ALWAYS BE PLURAL IN HERE!
// (EXCEPT PROPER NOUNS)
// ENGLISH VERBS MUST ALWAYS HAVE AUX's BUT NO TENSE CHANGES
//
// Example entries:
// When adding new vocab manually, there is a minimum amount
// of information you must add in order for the morphological
// processor to be able to do the rest. The minimum information
// required for each part of speech is given below.
//
// Note: make multiple entries for different senses (applies to all
//       parts of speech)
//
//
// VERB
// Regular: e.g. amare===to_love[VERB-PresInfAct]
//          e.g. minari===to_threaten[VERB-PresInfDep]
// Irregular: dare===to_give[VERB-PresInfAct]
//            dedi===I_give[VERB-1stSingPerfIndAct]
//            datum===give[VERB-SpnAcc/PARTICPerfPas-MscAccSing,
//                                    NeuNomSing,NeuVocSing,NeuAccSing]
// i.e. give any principle part that is not what you'd expect
// Note: some irregular/defective verbs are here in their entirety.
//
// NOUN
// 1st, 2nd, 4th, 5th declensions: Nom Sing needed:
//        mensa===tables[NOUNFemDec1(a,ae)-NomSing,VocSing,AblSing]
```

```
// 3rd declension: Must have Nom and Gen Sing forms:
//       One-liner:
//        civis===citizens[NOUNMscDec3(?,?is)-NomSing,VocSing,GenSing]
//       Two-liner:
//        rex===kings[NOUNMscDec3(?,?is)-NomSing,VocSing]
//        regis===kings[NOUNMscDec3(?,?is)-GenSing]
//
// ADJV
// 1st/2nd declension: Need MscNomSing form:
//      bonus===good[ADJV(us,a,um)-MscNomSing]
// 3rd dec. need 2-line entries:
//      audax===bold[ADJVDec3-MscNomSing,MscVocSing,FemNomSing,FemVocSing,
//                              NeuNomSing,NeuVocSing,NeuAccSing]
//      audacis===bold[ADJVDec3-MscGenSing,FemGenSing,NeuGenSing]
// IRREGULAR comparative and superlatives are also needed:
//     melior===more_good[ADJV(us,a,um)-COMPARA-MscNomSing,MscVocSing,
//                           FemNomSing,FemVocSing]
//     optimus===most_good[ADJV(us,a,um)-SUPERL-MscNomSing]
// Note: (us,a,um) comes from base form bonus.
// Later processing will change "more-","most-" into required word.
//
// ADVB
// Only positive forms required, except for irregular comparatives
// and superlatives:
//   Regular:
//    vere===correctly[ADVB]
//   Irregular:
//    bene===well[ADVB]
//    melius===more_well[ADVB-COMPARA]
//    optime===most_well[ADVB-SUPERL]
// Later processing takes care of more/most, as for ADJVs.
//
// INTJ
//     ecce===behold[INTJ]
//
// PREP
// Need multiple entries if more than one sense:
//      in===in[PREP-+Abl]
//      in===on[PREP-+Abl]
//      in===into[PREP-+Acc]
//
// PRON
// All pronouns are in this vocab list;
// none are generated by the morphological processor.
// Need case/number info.
//     ego===I[PRON-NomSing]
// Note: there are some ADJV-PRON entries - see below.
//
// In what follows, where you see more than the minimum information
// present, the morphological processor or a person has added the
// other lines over the course of time. The morphological processor
// always adds to the end of the file. The order of the entries is
// not important.
```

Blitz Latin - A Machine Translator for Latin to English

by

William A. Whitaker (McLean, VA, USA) and John F. White (Wokingham, UK)

Abstract

We have created a machine translator "*Blitz Latin*" for the direct translation of Latin texts to English, combining a 32,000+ word electronic dictionary and a novel programming approach derived from computer chess algorithms. Translation is by stages on a sentence-by-sentence basis. The result is an extremely fast translator that can consequently bear an huge number of translation heuristics. The product has been available commercially since July 2001. Programming principles and retail experience are outlined. In particular, we describe the difficulties of translation caused by the highly ambiguous Latin language, relative to its modern derived languages.

Introduction

The difficulties of machine translation of natural languages from one to another have been very well documented, not least within the pages and articles of the BCS Natural Languages web-site (*www.bcs.org.uk/siggroup/nalatran/*). A principal, common, problem is that caused by the translating program's inability to comprehend the widespread general knowledge employed by the speakers of a language in order to separate alternative meanings.

The question is also asked: who will employ such machine translators, when their standard of translation is inadequate for a professional translator of the language, and often too garbled for a casual reader? However, such translators have in practice been found to be of value to the former as an aid to translation, and to the latter as a means to grasp the gist of an otherwise unreadable text. One of the problems which plague those who translate Latin is the number of non-Latin speakers who request a 'quick translation' of a short motto or inscription - but never for payment. Cumulatively, these people become a serious burden for professional translators, whose web-sites generally try to discourage such approaches.

There is also a third group of potential beneficiaries. During years 2000-2001 one of us (JFW) was engaged in reading background Latin material for a book about a little-known Roman emperor. While English translations were available for many of the original sources, for some only the Latin could be obtained. JFW had learned (and received a qualification in) Latin about 32 years prior to this research, but through lack of use had forgotten most of the Latin vocabulary. Unexpectedly, though, he could remember the grammar. Thus translation of the Latin texts required the continual looking-up of words in a paper Latin dictionary - always tiresome and almost impossible if the stem happened to be irregular.

It appeared that machine translation would provide a better option, but it proved to be difficult to find a satisfactory commercial product. One amateur offering was no longer supported; another was far too slow (20 minutes at 166 MHz to translate one long paragraph), and the result was both bug-ridden and inadequate. The only professional offering was as part of a "Universal Translator", where the size of the Latin dictionary in Kbytes was about 5% of the size of the French and German dictionaries, and the product could not even find all the words for '*rex amat reginam*' (the king loves the queen). As such it was useless, and the

manufacturer withdrew the claim to translate Latin after the inevitable complaint. We may mention in passing the academic research tool 'Brutus' program developed by Nottingham University, UK (1), which apparently also has a deficient vocabulary.

However, an American ex-military scientist/ computer programmer (WAW) had produced, as a hobby over seven years, a very well-organised Latin to English electronic dictionary with separate tables of stems and inflections that would enable rapid construction of Latin words. The dictionary at that time amounted to 28,000 unique Latin words, and therefore already exceeded all but the most monumental of paper Latin dictionaries for size. The electronic dictionary additionally provided extra information about each Latin stem: in which Area it is used (general, ecclesiastical, legal, military, biological, agricultural, dramatic or poetical, scientific or technological); in which Age it was predominantly used (general, antiquity, classical era, post-classical, medieval, post-medieval, modern); and with what Frequency the stem is cited in conventional dictionaries (from very common to extremely rare).

Example of *ars, artis*, fem, from WAW's electronic dictionary:

```
ars  art   N 3 3 F T    X X X A O skill/craft/art; trick, wile; science,
knowledge; method, way; character;
```

We joined forces in early 2001 with a view to creating a superior Latin translator. JFW previously had acquired skills in programming Artificial Intelligence (AI), honed largely on computer chess and neural networks. A characteristic of computer chess is that speed of attaining the correct move from a position dominates virtually all other considerations, so that the programmer acquires the mental set of writing programs that execute extremely fast. There would be no waiting 20 minutes to translate single paragraphs with even the slowest of computers.

The Latin Language

The ancient Latin language has two principal features that distinguish it from most modern European languages:

1. It is an inflected language, so that the ending of each word provides a crucial part of the meaning. Thus the word '*dominus*' means 'the lord' (singular, nominative case), whereas '*domini*' can mean 'of the lord' or 'the lords' (plural, nominative or vocative).

2. It is a massively over-loaded language, where simple words can have several unrelated meanings. A good example is '*plaga*', which can mean 'snare', 'blow' or 'tract of land' (partly separated by pronunciation, which cannot be distinguished when the word is written). A serious later complication was the medieval word '*plagis*', meaning a certain type of musical chord. All these variants of *plaga* and *plagis* share some common inflections. '*Plagis*' might mean 'to the snares/blows/tracts', 'with the snares/blows/tracts', 'O chord', 'the chord' or 'of the chord'. How is a translator to pick between the nine alternatives?

All languages have ambiguity of meaning for some words (eg English has 'right', which can mean 'not left' or 'not wrong', both ultimately derived from Roman superstition about the left way being the unlucky way), but Latin is particularly well endowed with this translator's problem. Latin is, indeed, a much harder language to translate into English than the modern west-European languages that are derived from it, such as French, Spanish and Italian. In fact, inflected Latin is so ambiguous that the spoken form ('Vulgar Latin') is known to have added many prepositions to clarify the structure and meaning (2), while its derivative languages abandoned inflections altogether. Prepositions were also added to the modern German language, which is lightly inflected. Latin was actually a retrograde step from the older Greek

language (from which Latin derived very many words), which, although also inflected, at least had a definite article ('the'). Latin cannot easily distinguish between 'the king loves the queen' and 'a king loves a queen'.

Any book about Latin translation tells the reader to identify the nominative (subject) noun in the sentence, then pick out the verb which matches the noun. You do not find this instruction in books about the translation of modern European languages! In fact, modern European languages may generally be translated word-for-word into English, while retaining their original sense. This is certainly not the case with Latin, where the word order is used for emphasis. Examples:

English: 'against the followers of Christ' (a common theme in the early Christian Latin literature).

Italian: 'contro i discepoli di Christo.'

Latin: '*contra Christi cultores.*'

Note that the Latin sentence has fewer words and their order is different from the Italian/English. Further, '*cultores*' can mean 'inhabitants, cultivators, supporters'.

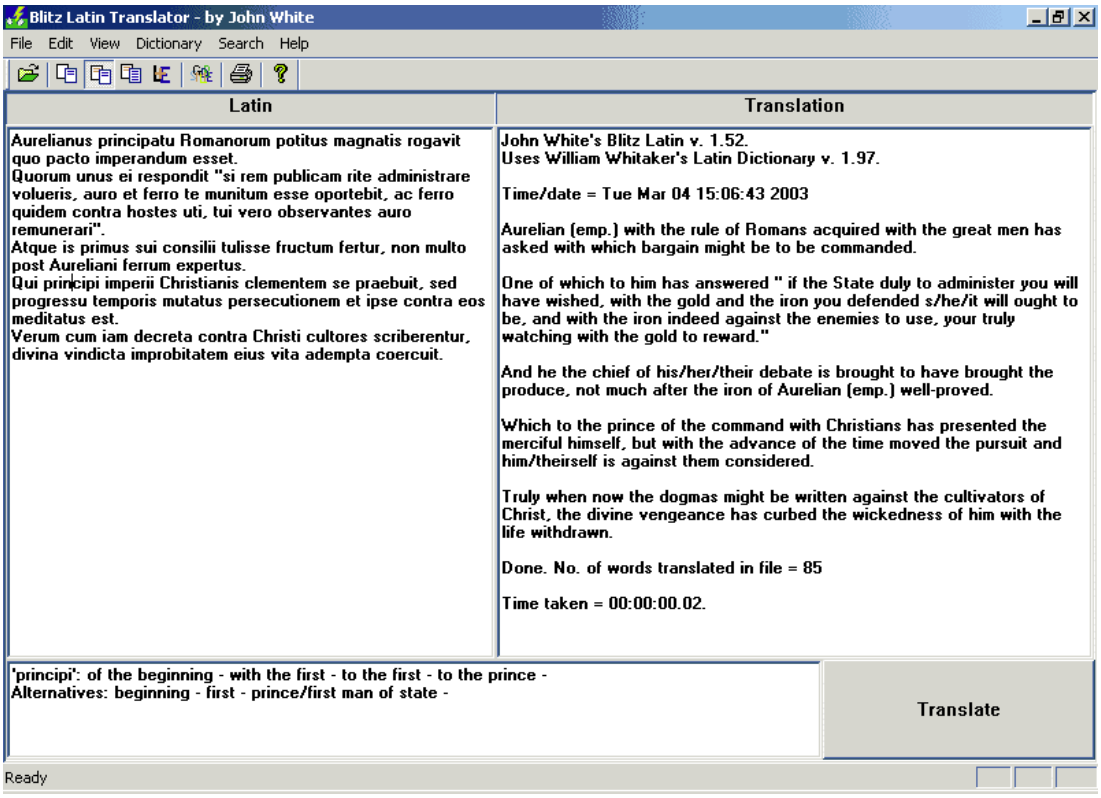
Blitz Latin

We first created our machine translator "*Blitz Latin*" over several months in 2001. It was written with Microsoft's Visual C++, and is a standard application with a GUI interface designed for use with Microsoft's Windows 95™ or later variant running on an IBM™-compatible PC. There is no Unix or Apple Macintosh™ version. The current release of the program requires about 4.5 Mbytes of RAM and some 12 Mbytes of hard-disk space. The user can type in Latin text for translation, or can load it as a pre-typed or computer-scanned text file. Note that *Blitz Latin* cannot translate from English to Latin, for which there appears to be very little demand.

Since *Blitz Latin's* translations often are ambiguous, or key Latin words such as *plaga* may have context-sensitive meanings, *Blitz Latin* has an interactive editing mode. The user can find suggested alternatives just by clicking on any Latin word, over-type corrections and save the result. The Latin dictionary may also be consulted from the on-screen menu. Words missing altogether from the dictionary may be added by a knowledgeable Latin user to a User File, which is a simple text file with pre-worked examples.

The huge dictionary and (selectable) grammatical detail provided by *Blitz Latin* for single Latin words make the program potentially of value to professional translators. We added a unique Latin spell-checker to the program, after finally becoming exasperated with the number of mis-spellings or scan-errors in published Latin texts. Previously there was no basic Latin spell-checker to match those commonly found in conventional word processors. Professional translators or proof readers can use a different searching arrangement to isolate multiple errors from many Latin files into a single output file for later examination. The program has also the ability to search many texts for Latin words as text, or even as identified stems and inflections.

Figure 1 - Screen View of *Blitz Latin*



Computer chess has been described as the 'drosophila fly' of Artificial Intelligence. One of the great discoveries of the 1970s to 1980s was that attention to acceleration of the move generator and the deep search routines produced far better programs than those where attention was paid to improving chess knowledge. Whisper it softly, but the best chess programs use the least input from chess grandmasters.

Machine translators traditionally have brought together a skilled linguist, who knows no computing, and a general programmer who knows little of natural languages. We felt that a novel approach based on the above discovery from computer chess might fare better: we used AI techniques from computer chess to make the fastest possible substitutions of English words for Latin words. This exploratory program, created in Microsoft's MS-DOS, was so fast that measurement of its speed became worthless. The time of word substitution depended heavily on the program's rate of access to the hard-drive, the screen and other system resources. The program compiled tables of alternative meanings (from combinations of stems and inflections) for each individual Latin word in a sentence.

Subsequent application of simple grammatical rules from Latin now eliminated completely many alternative meanings from the table for each ambiguous candidate Latin word. Then we added a few basic rules for which points were scored, and the best scoring alternative for each original Latin word was taken as the correct choice. Now we sought to broaden the program's comprehension with 'Reviews'. The best choice for each original Latin word was reviewed to see if it made a rule-based sense. For example, if the sentence contained a nominative noun, an accusative noun, but no verb, the score weightings would be altered to favour a verb and the scoring evaluation repeated. In theory this re-assessment might go on indefinitely,

following some kind of converging algorithm, but in practice little extra benefit is seen after just three reviews.

And now we come to the deepest irony: Because *Blitz Latin* was still executing its translations at a speed far beyond what was required, we felt able to add more and more translation heuristics to the scoring algorithms. This improved the quality of the selection of the best candidate meaning for each original Latin word to the extent that the number of necessary reviews fell. The average number of reviews over tens of thousands of clauses is now very close to 1.1, reducing the time spent on multiple searching of the scoring algorithms. Thus the net effect of *Blitz Latin's* speed is that it now has *more* translation heuristics than slower programs, with an associated improved quality of translation! And thus, too, we have discovered that speed alone does not paper over all of a machine translator's defects - knowledge is required also.

Like many machine translators, Blitz Latin carries out translations on single sentences. We spent some effort trying to 'remember' information from one sentence to another, so that a 3rd person verb could be tied to its subject matter, and elementary books of Latin Grammar even state that there is a rule: 'in Latin, the subject is continued from the previous sentence unless there is clear indication otherwise' (3). Unfortunately, the 'subject' is not necessarily the Latin word with the nominative case in the preceding sentence, and this rule is heuristically worthless for real Latin sentences, as we soon discovered.

Blitz Latin now carries out automatic translation in several stages:
1. 'Load-Text'. Delineation of sentences and sub-clauses. Intelligent tidying-up is done at this stage, checking for punctuation. An intermediary text file is created, which replaces the user's input file in all the following stages.
2. 'Parser'. Construction of tables for each word in a sentence. One original word may have several Latin stems (e.g. nouns, pronouns, verbs, adjectives) each with several inflections modifying the meanings. All the possible combinations are stored for each original word provided by the user. If the full translation option has not been selected, this is the output that will be displayed.
3. 'Clear-out'. Removal of improbable words or meanings using grammatical principles.
4. 'AI-Select'. Use of AI heuristics to determine which surviving meanings for a word are the most probable. The most probable combination of stem and inflection is selected. Reviews are carried out to determine whether the results are satisfactory. If not, the weightings are adjusted intelligently and the clause re-examined.
5. 'Elaborate'. Use of the best meaning from the best word's translation and its inflection. For example, a 2nd person future of the verb '*amare*' will be constructed as 'you will love'.
6. 'Polish'. Use of look-up tables to polish dis-jointed meanings as far as possible.
7. 'Best Order'. Analysis of the best polished meanings to improve the word order from Latin sequence to English sequence. This can make a big difference to comprehension of the final output.
8. 'Output'. Output of the improved translation to a text file which is subsequently displayed on screen.

While the speed of automatic translation naturally depends on the CPU speed, translation of complete text files (such as book chapters) is completed in at most seconds. This speed is achieved in part by a very efficient search algorithm for Latin stems and inflections within the electronic dictionary and, to a lesser extent, by use of high speed hash techniques.

Tables are constructed for each possible combination of stem and inflection for every individual Latin word in a sentence. Frequently, the table for a single word is a long one. For

example, the very common word *multum* has 10 different entries. *Decreta* has 15 entries! These are not artificial examples: both words have been taken from *Blitz Latin's* standard test, a short authentic historical Latin text extract provided with the translator, that illustrates many points for the novice user (see Figure 1). While we have no direct knowledge of the average number of entries required to store alternative English meanings for a single word of Italian, German, French or Spanish, we would be surprised indeed if more than three entries are often encountered. These are 'rich' languages, where single words commonly have single meanings and there are no complicating inflections.

We found that storage of complete tables for individual Latin words conferred some speed advantage (up to 10%) over direct indexing into the electronic dictionary, but required an enormous outlay of RAM (or, worse, of slow hard-drive space) to accommodate the tables. This would not be feasible for many users with older computers. However, small hash tables in RAM are particularly effective at handling the rarer occurrences of proper names and other words not previously found in the dictionary. These words would otherwise have been fruitlessly sought and re-sought in the main dictionary, and then examined by slow use of 'tricks' (see below), whenever encountered. It was particularly interesting to discover that, when proper names are first encountered, they tend to be re-encountered very quickly in succeeding sentences and paragraphs. By contrast, words completely unknown to the dictionary - even when not typing or scanning errors - tended not to recur.

In order to test the performance of *Blitz Latin*, we accumulated as many test files as we could. There are several excellent web-sites from which Latin HTML or TXT files may be freely downloaded (eg *www.theLatinLibrary.com* and *www.fh_augsburg.de/~harsch/a_chron.html*). At present our files are loosely divided as follows:

Test Files	Million words of Latin
Classical Latin	5.7 (includes Vulgate Latin Bible)
Justinian Legal Latin	1.7 (6th Century)
Medieval Latin	3.6 (includes some modern Latin)
Bracton Medieval Law	0.6 (13th Century)
Medieval Music theory	1.0 (3rd-11th Centuries)
PHI CD-ROM #5.3	7.5 (all Latin texts to 200 AD, plus some later)
Total:	20+ million words

Note: there is a large overlap between the PHI texts (tested courtesy of Packard Humanities Institute, USA) and the 'Classical Latin' test files.

Blitz Latin translates ALL the above test files within 25 minutes (1.4 GHz CPU PC), or within 4½ hours on a six-year-old 166 MHz PC. The program is exceptionally robust, handling not only true Latin text, but also a mish-mash of scanner errors, typing mistakes, Greek words, English or other foreign part-translations, as well as test rubbish like code files and even binary files disguised as text files. Naturally, it is only the Latin that gets translated.

Blitz Latin's Electronic Dictionary

For a non-inflected language such as English, it does not matter too much if a single word is missing from the translator's vocabulary. If we write 'a long black car was driven down the road', then striking out as unknown any single word in the above sentence does not affect the meaning of the remaining words.

For an inflected language, the consequences of not knowing a single word can be far more serious, and may cause mis-interpretation of the inflections on other words. Thus *Blitz Latin* makes every effort to find words in its dictionary, and maintenance of the dictionary is arguably the single most important part of machine translation of the Latin language.

For a few, very rare, Latin words, we know all the grammatical structure but cannot find the meaning in a conventional Latin dictionary. These words are nevertheless added to the electronic dictionary, so that the translator can at least get the context of use of the word in order to enable correct translation of the surrounding words. The meaning of the missing word is set equal to its stem in upper case, for example '*trimatus, trimatus*, masc' (TRIMAT).

When *Blitz Latin* cannot find a word in its vocabulary, it then tries to create 'slurred' words, eg '*inmitis*' will be slurred to give '*immitis*', in the dictionary. Failing here, the translator will attempt to create synthetic words from a similar dictionary stem and a prefix or a suffix. For example, '*superstruo'* will be created from 'super-' + *struo* (in the dictionary), or '*tralationem*' from *tralat* (in the dictionary) and '-ionem'. These 'Tricks' (as we call them; others use the word 'morphing') mop up many of the Latin words not found in the electronic dictionary. Since tricks are slow to implement, we try to identify Latin words commonly translated as a result of their use, and add them to the dictionary.

We have also the option to enable *Blitz Latin* to spew out into a special file all words that it does not recognise. These failed words are then automatically processed to collate them and sort them into descending numerical order. Then we identify the most common 'unknown' words and add them to the electronic dictionary. We may therefore claim that no word, in all of our 20 million test words, will be described as 'unknown' by the electronic translator if it occurs more than 15 times altogether (most words occurring fewer than 15 times will also be translated). Proper names, abbreviations and words of foreign origin are excepted.

Thus the dictionary continues to grow, and our program continues to become faster.

The Super-Adjective

Generally speaking, *Blitz Latin* uses the stem-types (noun, pronoun, adjective, verb and so on) as presented to it by WAW's original electronic dictionary. The dictionary distinguishes - whenever known from actual example - the substantive use of an adjective to give a noun; thus *bonus* is entered in the dictionary as an adjective ('good') and as a noun ('good man'). However, it became necessary to introduce an artificial stem-type, 'Super-Adjective', into the dictionary to describe certain adjectives that are declined as adjectives but used as pronouns. Examples include *qualis*, *talis*, *plerus*, *multus*, *malus* and *alius*. This differentiation is, we believe, novel, and was not made lightly. The change was found to be necessary to render more accurate *Blitz Latin's* grammatical eliminations of improbable word candidates, and is unlikely to be applicable outside the Latin language.

This is an example of how the lightning speed of *Blitz Latin*, which enables sweeps of thousands of Latin files to be made for items of interest within minutes, has altered our understanding of how the Latin language should be translated, as well as causing changes to

be made in the electronic dictionary. A more common occurrence is that inflections are found for a Latin stem that are inconsistent with those previously reported. This also has to be corrected in the dictionary.

Medieval Latin

As the Roman world fragmented under the onslaught of barbarians in the 5th Century AD, the dialects of the inhabitants around the Roman Empire diverged, ultimately leading to the modern west-European languages. However, during the 'Middle Ages' (defined by Gibbon as the period from the Fall of Rome in 476 AD to the Fall of Constantinople in 1453 AD) monks and scribes wrote extensively their own dialects in Latin-as-it-sounds. To take just one example, the word listed in Lewis and Short's Latin Dictionary as *synemmenon* is variously spelled as *synnemenon*, *synemenon*, *sinemenon* and *sinemmenon*. Worse, in Britain and perhaps elsewhere, Norman writers turned their everyday French-Saxon speech into Latin by the dubious expedient of placing Latin inflections onto their native words. The results must often have been incomprehensible even to readers of Latin in the next town, let alone to readers in Italy.

Medieval Latin thus introduced its own ambiguities. The most senior Norman feudal overlords were known as 'barons'. Norman scribes invented a new Latin word '*baro, baronis*, masc', meaning 'baron', apparently oblivious to the fact that there was already a perfectly good, identical, Latin word that meant 'blockhead'. Thus important late medieval Latin legal documents, such as the 13th Century '*Magna Carta*' (the famous pact between an English king and his barons), are filled with references to the king's loyal blockheads. It was not until the Enlightenment (around the 15th-16th Centuries AD) that the spelling and use of Latin was again standardised.

How should *Blitz Latin* respond to this challenge, bearing in mind that far more Latin survives from the medieval period than from the Classical Age? We have made pragmatic changes. A large number of common medieval Latin words have gone into the electronic dictionary. We have created medieval 'tricks' to compensate for the most common medieval mis-spellings. And, most ambitiously, we have created phonetics lists, which try to match a phonetically mis-spelled medieval Latin stem (ie, excluding the inflection) with its classical spelling. The last procedure is quite effective, but rather slow. It has three weaknesses:

1. Sometimes there are several matches with a phonetically mis-spelled medieval Latin word, and it is not clear which match should be used. Thus *Blitz Latin* now keeps a short-list of the most promising (defined heuristically) match options and tries them in turn until both candidate stem and its inflection (computed by subtraction of the candidate stem from the original Latin word) are accepted by the dictionary as a legitimate Latin word.

2. There is no systematic way to search for phonetic Latin mis-spellings in any dictionary. In order to save an excruciating search of the entire electronic dictionary, *Blitz Latin* selects (currently) up to four areas of the dictionary for systematic search. This is fast, but sometimes a phonetic match elsewhere in the dictionary will be missed.

3. There is the serious problem that the spell-it-as-I-speak-it writer might have a pronounced local accent. We have recently added the 'Historia Francorum' of St. Gregory of Tours (6th Century) to our test files. This writer, descended of a Romano-Gallo family but living in a Frank-controlled area surrounded by civil strife, has persistently, but not consistently, mis-spelled many Latin words. For example:

victoria, cognosc-, territorium, oratorius ⇒ victuria, cognusc-, territurium, oraturius.

monasterium, religiosus, itinere, obsidere ⇒ monastirium, relegiosus, itenere, obsedere.

And those are just counting the most common mis-spellings. It would appear that the author has altered those vowels on which a stress is placed in speech. *Blitz Latin* now contains an adaption that will test changes to stressed vowels as part of its phonetics search. This is again rather slow, so that the medieval phonetics search has to be actively engaged by the user.

Modern Latin

More Latin has been written in the past 50 years than in all previous history. Two main reasons may be put forward: the (Latin-speaking) bureaucracy known as 'The Vatican' and Internet chat rooms. This created a demand for new words for such inventions as the motor car and the aeroplane. Dictionaries of such modern Latin words have been compiled, and one such dictionary is the 'Calepinus Novus', compiled by the Belgian 'Melissa Foundation'. An electronic version of the Calepinus Novus dictionary, adapted by the authors, is now incorporated into *Blitz Latin*, by generous permission of Guy Licoppe of Melissa (*guy.licoppe@pophost.eunet.be*).

Blitz Latin thus covers all bases with its electronic dictionary. It is arguably the fullest (ie, considering number of words and time spread) Latin dictionary in existence, although the Oxford Latin Dictionary (£225) and Lewis and Short's Latin Dictionary (£100) retain their superiority for classical Latin (for now…). An unexpected by-product of our trials with Blitz Latin with so many Latin test files has been the ability to increase accurately and aggressively the number of words now inserted into the dictionary, after they had previously been rejected too frequently as 'unknown'.

Standard of Latin Translation and Resolution of Ambiguities

We should like to be able to say that the standard of translation by *Blitz Latin* is perfect, but clearly that is not possible for such an ambiguous language as Latin. Nevertheless, the quality of Latin translation is easily good enough to follow the gist of Latin texts. JFW has been able to translate his difficult historical and legal Latin texts referred to at the beginning of this article.

Often use of an ambiguous Latin stem can be resolved on grammatical principles (whether the word should be a noun, adjective, verb or other) but, if this is not possible, the translator will use the most-cited word (see 'Frequency', previously). We have found it sometimes necessary to change the frequency values originally allocated, on the basis of our experiences in translating Latin texts. For example, the stem 'mult-' is cited in standard dictionaries as very frequent whether encountered as *multus* (adjective, 'much') or as *multa* (noun, 'penalty'). In practice, the stem 'mult-' occurs far more often as part of *multus*.

As a consequence of the ambiguity of Latin, *Blitz Latin* frequently provides sentences that have been translated with grammatical accuracy, but with the wrong meanings. An extreme example of this can be seen with the very common, two-word, clause: '*liber primus*'. This phrase is placed at the head of a multi-volume Latin text, and means 'The First Book'. However, it can be translated with equal accuracy as 'The Free Leader' or 'The Free First-Man'. And why should it not be so translated? The only reason we prefer the first translation 'The First Book' is because we happen to know that the next book begins '*liber secundus*' -

'The Second Book'. But the first book could have been a biography of the freed South African president, Nelson Mandela.

A related difficulty can be seen in a short extract from the 19th Century Latin translation of the Greek text of the Byzantine historian Zonaras: '*ad fossam quamdam'*. The intended translation is 'at a certain ditch'. *Blitz Latin*'s translation is 'to the dug certain'. This is, in fact, a legal translation, despite the confusion as to whether '*fossam*' should be translated as a noun qualified by an adjective, or as a verb participle qualifying a pronoun (note that '*ad*' is yet another overloaded Latin word).

A further difficult example of ambiguity is found with *Blitz Latin's* test file:

'*Qui principi imperii Christianis clementem se praebuit, ...*'. *Blitz Latin* translates this sentence as 'Which to the prince of the command with Christians has presented the merciful himself, …' This translation seems to be perfectly good, since we know from other Latin sources that the emperor ('which') initially enjoyed good relations with the Pope, who might well be described as the 'princeps' of the Christian Command by the original pagan writer. Indeed, we believed for some time that this was the correct translation. But '*principi*' ought to be translated as 'of the beginning'. This alters the whole meaning of the sentence: 'Which of the beginning of the (his) command to Christians has presented the merciful himself'. Since this ambiguity fooled even us, it is clear that we shall have difficulty getting the translator to make the correct translation.

We have suggested elsewhere (see reference 4) that some of the ambiguities for individual Latin words might be cleared up by training of a neural network by human Latin experts, employing many example Latin sentences containing the ambiguous Latin word. The correct meaning could be assigned for each ambiguous Latin word by the experts, thus providing the training. This process would be time-consuming for each word trained, and would then require the addition of a great deal of complex code to *Blitz Latin*, if the latter were to make use of the results. There is also a faint hope that perhaps rules might become apparent for the deployment of each ambiguous Latin word by this process, so that *Blitz Latin* could use simple rules instead of a neural network whose parameters had to be changed for each trained Latin word.

The present authors, who have other responsibilities, lack the requisite time to investigate this further. But certainly we would be interested to know whether the method will be successful in principle, if tested with perhaps 10-20 very ambiguous Latin words (such as *plaga*). Such an investigation - requiring training of a neural network for each separate ambiguous Latin word - is a job for a PhD student.

Another potential means to resolve the ambiguities of Latin stems is to consider the area (or field) in which the Latin word is used. It may be recalled that the electronic dictionary stores such areas as part of its structure. Frequently only the general meaning of a word is required, but specialist meanings will apply in some sentences. Examples include the Latin words '*plaga, plagae*, fem' (snare/blow/tract of land) and '*plagis, plagis*, fem' (music chord), mentioned previously, and '*ars, artis*, fem' ('skill' or 'wile', but with scientific meaning 'science' or 'knowledge').

Blitz Latin now allows the user to select the area or field (general, ecclesiastical, legal etc, as listed earlier) before attempting the translation. Even more ambitious are the program's attempts to auto-change the area according to its perception of the Latin content. If *Blitz Latin* discovers that most of its translation was (say) legal Latin, the user is advised and recommended to re-translate with the manual setting for 'legal area'.

Both of these techniques for selecting the translation area work quite well (surprisingly so, in the case of the auto-changing facility), but there remain serious defects.

1. The original writer may have intended that a Latin word retain its general meaning even when used in a specialist text. This is by far the most serious problem.

2. The auto-changing translation lags a change in Latin content. If a text changes from a discussion of astronomy (scientific area) to its implication for theology (ecclesiastical area), some 30-50 sentences may pass before *Blitz Latin* auto-changes its translation area from scientific to ecclesiastical.

Finally, we point up that a very ambiguous language is likely to be formula-driven; that is, the native speakers would use standard formulae to express certain phrases. Therefore a very small part (currently) of *Blitz Latin* is devoted to making standard phrase substitutions, when encountered. Probably this section should be expanded, but it requires identification of suitable, common, target phrases, and requires quite a time overhead to implement. A trivial example is the recognition and translation of the various derivative forms of '*res publica*' as 'State'.

Blitz Latin and 'Latin Grammar'

This article has addressed the programming issues of *Blitz Latin*. For a discussion of the grammatical issues, we refer the reader to our article in the JACT Review (4).

After the latter article appeared in print, a reader made a thought-provoking point: we had attended too much to unravelling the intricacies of classical Latin texts, which had been preserved as much for their eloquence and grammatical complexity as for their content. The reader suggested that we should have much less difficulty with simpler Latin texts, such as modern or even medieval texts (after compensating for spelling problems). This is a valid point that other would-be programmers of Latin translators should also bear in mind.

'Blitz Latin and Latin Inscriptions'

Large numbers of Latin inscriptions remain on Roman monuments that have survived through the millennia. These were initially collated by the German, Mommsen, in the 19th Century, but are now available electronically. A good source is Frankfurt University (Germany) at *www.rz.uni-frankfurt.de/~clauss/index-e.html*.

Blitz Latin can translate expanded inscriptions (such as those supplied by Frankfurt; the originals are often abbreviations only), after a special toggle is set. The big difficulty is that the translation routines rely rather heavily on finding a verb present. Verbs tend to be absent in most inscriptions, making the problem of ambiguity of translation even worse.

Experience with 'Blitz Latin' in Service

Blitz Latin has been commercially available since July 2001 (version 1.35) via the web-site of the independent retail distributors Software Partners (*www.software-partners.co.uk*). The down-loaded version comes with an auto-install (and auto-uninstall!) facility and is free for 10 uplifts from the hard-drive into RAM. Thereafter a licence has to be purchased (at present £29 or foreign equivalent). The current version is 1.52, released in May 2003. Existing

licence holders are entitled to free upgrades with each new release. To date, there have been five such upgrades.

The problem of legal, but inaccurate, translation of ambiguous Latin text has proved to provide by far the most common source of complaint about *Blitz Latin*. Experienced Latin translators do not seem to understand that their knowledge of Latin depends at least as much on their general knowledge as on their Latin skills. They complain that *Blitz Latin's* translation sometimes makes no 'sense'. What is 'sense', and how is a computer program to acquire it?

There has not been a single word of complaint about the stability of *Blitz Latin*. It simply does not crash.

Summary

We have created a lightning-fast machine translator of Latin which we call "*Blitz Latin*". Its high speed of operation has enabled the introduction of a very high number of translation heuristics, so that it is the most accurate commercially-available Latin translator, as well as the fastest.

The translator provides a quality of translation sufficient for the gist of the Latin text to be grasped, and to provide a fast aid to professional (human) translators. The ambiguities of the Latin language, far greater than those experienced with modern west-European languages, are such that we believe that it will not be possible to improve significantly on *Blitz Latin's* translations without a means of conferring 'context' on each word as it is translated. Regrettably, 'context' usually means 'wide general knowledge', which is not at present accessible to computer programs.

References

1. Bowden, P.R. 'Latin to English Translation - A Direct Approach'. In *Machine Translation Review,* No. 12, December 2001
2. *Columbia Electronic Encyclopaedia. 6th Edition, 2000. The Latin Language*. Columbia University Press. [www.encyclopedia.com/articles/07250.html.]
3. *Collins Latin Dictionary Plus Grammar.* Harper-Collins, 1997:134
4. Whitaker, W.A. and White, J.F. 'Blitz Latin - Experiments with automatic translation of Latin'. In *JACT Review*, No. 32, Autumn 2002: 2-8.

Conferences and Workshops

The following is a list of forthcoming conferences and workshops. E-mail addresses and websites are given where known.

The online Language and Speech Calendar maintained by ELSNET (European Network in Human Language Technologies) contains announcements of events in the area of Natural Language and Speech, and related areas. This is an extensive website which provides links to a wide range of conferences, meetings and workshops. The site may be accessed at: http://www.elsnet.org.

A list of 2004 Computational Linguistics Conferences is also accessible at: http://www.cs.rochester.edu/u/tetreaul/conferences.html

26-27 April 2004
Ninth Workshop of the European Association for Machine Translation (EAMT)
The themes are machine-translation-related issues concerning Semitic languages, and the languages of the newly accessioned states of the European Union
Foundation for International Studies in Valletta, Spain
E-mail: mike.rösner@um.edu.mt
http://www.eamt.org

21-23 May 2004
AAALC 2004
The American Association for Applied Corpus Linguistics
The Fifth North American Symposium on Corpus Linguistics
Montclair State University, Upper Montclair, New Jersey, USA
E-mail: aaacl@mail.montclair.edu
http://www.chss.montclair.edu/linguistics/aaacl/index.html

24-30 May 2004
LREC 2004
Centro Cultural de Belem, Lisbon, Portugal
http://www.lrec-conf.org

14-16 July 2004
INLG04
Third International Conference on Natural Language Generation
Careys Manor, Brockenhurts, New Forest, UK
E-mail: inlg04@itri.brighton.ac.u
http://www.itri.brighton.ac.uk/inlg04

21-26 July 2004
ACL-2004
42nd Annual Meeting of the Association for Computational Linguistics
Forum Convention Centre, Barcelona, Spain
http://www.acl2004.org

23-27 August 2004 (main conference)
28-29 August 2004 (workshops)

COLING 2004
Conference on Computational Linguistics
University of Geneva, Switzerland
http://www.issco.unige.ch/coling2004/

5-7 September 2004
11th International CALL Conference (Computer Assisted Language Learning)
Call and Research Methodologies
Conference Centre Elzenveld, Antwerp, Belgium
E-mail: mathea.simons@ua.ac.be

28 September - 2 October 2004
AMTA 2004
6th Biennial Conference of the Association for Machine Translation in the Americas
Georgetown University, Levy Conference Center, Washington DC, USA
http://www.amtaweb.org/AMTA2004

18-19 November 2004
26th Aslib Conference: Translating and the Computer
http://www.aslib.com/conference

MEMBERSHIP: CHANGE OF ADDRESS

If you change your address, please advise us on this form, or a copy, and send it to the following (this form can also be used to join the Group):

Mr. J.D.Wigg
BCS-NLTSG
72 Brattle Wood
Sevenoaks, Kent TN13 1QU
U.K. Date: ...../...../.....

Name: .....
Address: .....
.....
Postal Code: ..... Country: .....
E-mail: ..... Tel.No: .....
Fax.No: .....

Note for non-members of the BCS: your name and address will be recorded on the central computer records of the British Computer Society.

Questionnaire

We would like to know more about you and your interests and would be pleased if you would complete as much of the following questionnaire as you wish (please delete any unwanted words).

1.
   a. I am mainly interested in the computing/linguistic/user/all aspects of MT.
   b. What is/was your professional subject? .....
   c. What is your native language? .....
   d. What other languages are you interested in? .....
   e. Which computer languages (if any) have you used? .....

2. What information in this Review or any previous Review have you found:
   a. interesting? Date .....
      .....
      .....
   b. useful (i.e. some action was taken on it)? Date .....
      .....
      .....

3. Is there anything else you would like to hear about or think we should publish in the *MT Review*?
      .....
      .....
      .....
      .....

4. Would you be interested in contributing to the Group by,

   a. Reviewing MT books and/or MT/multilingual software
   b. Researching/listing/reviewing public domain MT and MNLP software .....
   c. Designing/writing/reviewing MT/MNLP application software .....
   d. Designing/writing/reviewing general purpose (non-application specific) MNLP ..... procedures/functions for use in MT and MNLP programming .....
   e. Any other suggestions? .....
      .....
      .....
      .....

Thank you for your time and assistance.