GATE: A Unicode-based Infrastructure Supporting Multilingual Information Extraction

Kalina Bontcheva and **Diana Maynard** and **Valentin Tablan** and **Hamish Cunningham**
Dept. of Computer Science, University of Sheffield
Regent Court, 211 Portobello St, Sheffield, S1 4DP, UK
[K.Bontcheva, D.Maynard, V.Tablan, H.Cunningham]@dcs.shef.ac.uk

Abstract

NLP infrastructures with comprehensive multilingual support can substantially decrease the overhead of developing Information Extraction (IE) systems in new languages by offering support for different character encodings, language-independent components, and clean separation between linguistic data and the algorithms that use it. This paper will present GATE – a Unicode-aware infrastructure that offers extensive support for multilingual Information Extraction with a special emphasis on low-overhead portability between languages. GATE has been used in many research and commercial projects at Sheffield and elsewhere, including Information Extraction in Bulgarian, Romanian, Russian, and many other languages.

1 Introduction

GATE(Cunningham 02)[1] is an architecture, development environment and framework for building systems that process human language. It has been in development at the University of Sheffield since 1995, and has been used for many R&D projects, including Information Extraction in multiple languages and for multiple tasks and clients.

The GATE architecture defines almost everything in terms of components - reusable units of code that are specialised for a specific task. There are three main types of components:

- *Language Resources (LRs)* store some kind of linguistic data such as documents, corpora, ontologies and provide services for accessing it. At the moment all the predefined LRs are text based but the model doesn't constrict the data format so the framework could be extended to handle multimedia documents as well.
- *Processing Resources (PRs)* are resources whose character is principally programatic or algorithmic such as a POS tagger or a parser. In most cases PRs are used to process the data provided by one or more LRs but that is not a requirement.
- *Visual Resources (VRs)* are graphical components that are displayed by the user interface and allow the visualisation and editing of other types of resources or the control of the execution flow.

The GATE framework defines some basic language resources such as documents and corpora, provides resource discovery and loading facilities and supports various kinds of input output operations such as format decoding, file or database persistence.

GATE uses a single unified model of *annotation* - a modified form of the TIPSTER format (Grishman 97) which has been made largely compatible with the Atlas format (Bird & Liberman 99), and uses the now standard mechanism of 'stand-off markup' (Thompson & McKelvie 97). Annotations are characterised by a *type* and a set of *features* represented as attribute-value pairs. The annotations are stored in structures called *annotation sets* which constitute independent layers of annotation over the text content.

The advantage of converting all formatting information and corpus markup into a unified representation, i.e. the annotations, is that NLP applications do not need to be adapted for the different formats of each of the documents, which are catered for by the GATE format filters (e.g. some corpora such as BNC come as SGML/XML files, while others come as email folders, HTML pages, news wires, or Word documents).

The work for the second version of GATE started in 1999 and led to a complete redesign of the system and a 100% Java implementation. One of the additions brought by version 2 is full support for Unicode data allowing the users to open, visualise and process documents in languages dif-

[1]GATE is implemented in Java and is freely available from http://gate.ac.uk as open-source free software under the GNU library licence.

ferent from the default one for the underlying platform.

2 Information Extraction in GATE

Provided with GATE is a set of reusable processing resources for common NLP tasks. (None of them are definitive, and the user can replace and/or extend them as necessary.) These are packaged together to form ANNIE, A Nearly-New IE system, but can also be used individually or coupled together with new modules in order to create new applications. For example, many other NLP tasks might require a sentence splitter and POS tagger, but would not necessarily require resources more specific to IE tasks such as a named entity transducer. The system is in use for a variety of IE and other tasks, sometimes in combination with other sets of application-specific modules.

ANNIE consists of the following main processing resources: tokeniser, sentence splitter, POS tagger, gazetteer, finite state transducer (based on GATE's built-in regular expressions over annotations language (Cunningham *et al.* 02b)), and orthomatcher. The resources communicate via GATE's annotation API, which is a directed graph of arcs bearing arbitrary feature/value data, and nodes rooting this data into document content (in this case text).

The **tokeniser** splits text into simple tokens, such as numbers, punctuation, symbols, and words of different types (e.g. with an initial capital, all upper case, etc.). The aim is to limit the work of the tokeniser to maximise efficiency, and enable greater flexibility by placing the burden of analysis on the grammars. This means that the tokeniser does not need to be modified for different applications or text types and frequently does not need to be modified for new languages, i.e., tends to be fairly language-independent.

The **sentence splitter** is a cascade of finite-state transducers which segments the text into sentences. This module is required for the tagger. Both the splitter and tagger are domain- and application-independent.

The **POS tagger** is a modified version of the Brill tagger, which produces a part-of-speech (POS) tag as an annotation on each word or symbol. Neither the splitter nor the tagger are a mandatory part of the NE system, but the annotations they produce can be used by the grammar (described below), in order to increase its power and coverage. For languages where no POS tagger is available it can be left out, often without major implications on the system's performance on some IE tasks. Alternatively, the English POS tagger can easily be adapted to a new language using a bi-lingual lexicon (see Section 4.3).

The **gazetteer** consists of lists such as cities, organisations, days of the week, etc. It not only consists of entities, but also of names of useful *indicators*, such as typical company designators (e.g. 'Ltd.'), titles, etc. The gazetteer lists are compiled into finite state machines, which can match text tokens.

The **semantic tagger** consists of hand-crafted rules written in the JAPE (Java Annotations Pattern Engine) language (Cunningham *et al.* 02b), which describe patterns to match and annotations to be created as a result. JAPE is a version of CPSL (Common Pattern Specification Language) (Appelt 96), which provides finite state transduction over annotations based on regular expressions. A JAPE grammar consists of a set of phases, each of which consists of a set of pattern/action rules, and which run sequentially. Patterns can be specified by describing a specific text string, or annotations previously created by modules such as the tokeniser, gazetteer, or document format analysis. Rule prioritisation (if activated) prevents multiple assignment of annotations to the same text string.

The **orthomatcher** is another optional module for the IE system. Its primary objective is to perform co-reference, or entity tracking, by recognising relations between entities. It also has a secondary role in improving named entity recognition by assigning annotations to previously unclassified names, based on relations with existing entities.

3 Support for Multilingual Documents and Corpora in GATE

The most important types of Language Resources that are predefined in GATE are *documents* and *corpora*. A corpus is defined in GATE as a list of documents.

Documents in GATE are typically created starting from an external resource such as a file situated either on a local disk or at an arbitrary location on the Internet. Text needs to be converted to and from binary data, using an *encod-*

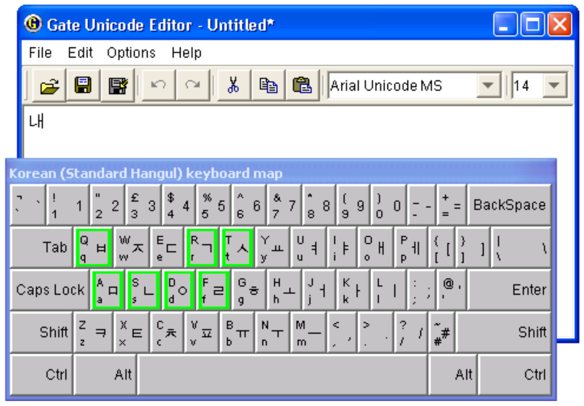

Gate Unicode Editor - Untitled*
File Edit Options Help
Arial Unicode MS
14
Korean (Standard Hangul) keyboard map

Figure 1: The GUK Unicode editor using a Korean virtual keyboard.

ing (or *charset*), in order to be saved into or read from a file. There are many different encodings used worldwide, some of them designed for a particular language, others covering the entire range of characters defined by Unicode. GATE uses the facilities provided by Java and so it has access to over 100 different encodings including the most popular local ones, such as ISO 8859-1 in Western countries or ISO-8859-9 in Eastern Europe, and some Unicode ones e.g. UTF-8 or UTF-16. Once processed, the documents can also be saved back to their original format and encoding.

Apart from being able to read several character encodings, GATE supports a range of popular file formats such as HTML, XML, email, some types of SGML and RTF.

Another important aspect is displaying and editing multilingual documents. GATE uses largely the Java Unicode support for displaying multilingual documents. Editing multilingual documents (and also language-specific grammars, gazetteer lists, etc) is provided by GUK – *GATE's Unicode Kit*. GUK provides input methods for a large number of languages (see Figure 1), allows the definition of new ones, and also provides a Unicode-based text editor.

So far GATE has been used successfully to create corpora and process documents in a wide range of languages – Slavic (e.g., Bulgarian, Russian), Germanic (Maynard *et al.* 01; Gambäck & Olsson 00), Romance (e.g., Romanian) (Pastra *et al.* 02), Asian (e.g., Hindi, Bengali) (McEnery *et al.* 00), Chinese, Arabic, and Cebuano (Maynard *et al.* 03).

4 Adapting the GATE Components to Multiple Languages

The use of the Java platform implies that all processing resources that access textual data will internally use Unicode to represent data, which means that all PRs can virtually be used for text in any Unicode supported language. Most PRs, however, need some kind of linguistic data in order to perform their tasks (e.g. a parser will need a grammar) which in most cases is language specific. In order to make the algorithms provided with GATE (in the form of PRs) as language-independent as possible, and as a good design principle, there is always a clear distinction between the algorithms - presented in the form of machine executable code - and their linguistic resources which are typically external files. All PRs use the textual data decoding mechanisms when loading the external resources so these resources can be represented in any supported encoding which allows for instance a gazetteer list to contain localised names. This design made it possible to port our information extraction system ANNIE from English to other languages by simply creating the required linguistic resources.

4.1 The Unicode Tokeniser

One of the PRs provided with GATE is the tokeniser which not only handles Unicode data but is actually built around the Unicode standard, hence its name of "*GATE Unicode Tokeniser*".

Like many other GATE PRs, the tokeniser is based on a finite state machine (*FSM*) which is an efficient way of processing text. In order to provide a language independent solution, the tokeniser doesn't use the actual text characters as input symbols, but rather their categories as defined by the Unicode standard.

As part of our work on multilingual IE, the ruleset of the tokeniser was improved, because originally it was intended for Indo-European languages and therefore only handled a restricted range of characters (essentially, the first 256 codes, which follow the arrangement of ISO-8859-1 (Latin1). We created a modified version of this which would deal with a wider range of Unicode characters, such as those used in Chinese, Arabic, Indic languages etc. There is some overlap between the Unicode characters used for different languages. Codes which represent letters, punctuation, symbols, and diacritics that are generally shared by multiple languages or scripts are grouped together in several locations, and characters with common characteristics are grouped together contiguously (for example, right-to-left scripts are grouped together). Character code allocation is therefore not correlated with language-dependent collation.

In order to enable the tokeniser to handle other Unicode characters, we had to find the relevant character types and their symbolic equivalents (e.g. type 5 has the symbolic equivalent "OTHER_LETTER"; type 8 characters are usually at the beginning or end of a word and have the symbolic equivalent "COMBINING_SPACING_MARK" or "NON-SPACING_MARK"). Rules covering these types were added to the tokeniser in order to discover the tokens correctly in a variety of other languages. Even more importantly, having discovered the technique for extending the tokeniser in this way, it will be easy to add any further new types as necessary, depending on the language (since we have not covered all possibilities).

4.2 Localising the Gazetteer Lists

The GATE gazetteer processing resource enables gazetteer lists to be described in 3 ways: majorType, minorType and language. The major and minor types enable entries to be classified according to two dimensions or at 2 levels of granularity – for example a list of cities might have a majorType "location" and minorType "city". The language classification enables the creation of parallel lists, one for each language.

For example, for our Cebuano[2] IE experiment (see Section 5.4) we had the same structure for the Cebuano lists as for their English counterparts, and simply altered the language label, to differentiate between the two. This is useful for languages where names of English entities can be found in the texts in the other language (e.g. for Cebuano – "Cebu City Police Office"). To recognise these successfully we required both the English gazetteer (to recognise "Office") and the Cebuano gazetteer (to recognise "Cebu City", which is not in the English gazetteer). Using both gazetteers improved recall and did not appear to affect precision, since English entities did not seem to be ambiguous with Cebuano entities or proper nouns. However, this might not be the case for other, closer languages.

[2]A language spoken in the South Philippines.

4.3 Multilingual Adaptation of the POS Tagger

The Hepple POS tagger, which is freely available in GATE as part of ANNIE, is similar to the Brill's transformation-based tagger (Brill 92), but differs mainly in that it uses a decision list variant of Brill's algorithm. This means that in classifying any instance, only one transformation can apply. It is also written in Java.

In order to adapt the POS tagger to a new language, one would typically need to re-train it on a big part-of-speech annotated corpus. However, there are no such corpora for many languages, so we experimented with adapting the POS tagger to a new language without such training data, only using a bilingual lexicon.

As part of the ANNIE adaptation to Cebuano (see Section 5.4), we tested whether we could adapt the Hepple tagger to Cebuano using a bilingual Cebuano-English lexicon with POS information. On first appearances it seemed that Cebuano word order and morphology is similar to English, and it also has similar orthography. The rules for English (derived from training on the Wall Street Journal) would clearly not be applicable for Cebuano, so we used an empty ruleset, but we decided that many of the default heuristics might still be appropriate. The heuristics are essentially as follows:

1. look up the word in the lexicon
2. if no lexicon entry found:
   - if capitalised return NNP
   - if word containes "-" return JJ
   - if word contains a digit return CD
   - if word ends in "ed", "us", "ic", "ble", "ive", "ish", "ary", "ful", "ical", "less" return JJ
   - if word ends in "s" return NNS
   - if word ends in "ly" return RB
   - if word ends in "ing" return VBG
   - if none of the above matched return NN
3. apply the trained rules to make changes to the assigned categories based on the context

Some of these heuristics make little sense for Cebuano because it is unusual for Cebuano words to have endings such as "ic","ly", "ing" etc. This means that in most cases, when a word is not in the lexicon, the tag returned will be NNP (proper noun) if capitalised, or NN (common noun) if not, which is appropriate. However, these heuristics cannot be changed without modifying the code of the POS tagger itself, therefore we left them unchanged even though most of them did not apply.

Adapting the tagger did have a number of problems, mostly associated with the fact that while the English lexicon (used for the tagger) consists only of single-word entries, the Cebuano lexicon contained many multi-word entries (such as *mahitungod sa honi* (musical)). The tagger expects lexicon entries to have a single word entry per line, followed by one or more POS tags, each separated by a single space.

We therefore modified the lexicon so that the delimiter between the lexical entry and the POS tag(s) was a "#" rather than a space, and adapted the tagging mechanism to recognise this. As a result, the ANNIE POS tagger now has the option of processing multi-word entries, which are very important in a number of languages, e.g., the ANNIE adaptation to Hindi also required this.

In order to evaluate the portability and usefulness of such low-overhead adaptation of the POS tagger, we repeated the same experiment for Hindi, using a relatively small English-Hindi bilingual lexicon. The results were 67% correctness as evaluated by a native Hindi speaker. Whereas such correctness may not be sufficient for deeper linguistic processing (e.g., parsing), it is sufficient for named entity recognition.

Next we discuss how these multilingual processing resources were used to perform information extraction in a variety of languages.

5 Information Extraction in Multiple Languages

Robust tools for multilingual information extraction are becoming increasingly sought after now that we have capabilities for processing texts in different languages and scripts. While the ANNIE IE system in GATE is English-specific, some of the modules can be reused directly (e.g. the Unicode-based tokeniser can handle Indo-European languages), and/or easily customised for new languages (Pastra *et al.* 02). So far, ANNIE has been adapted to do IE in Bulgarian, Romanian, Bengali, Greek, Spanish, Swedish, German, Italian, French, Hindi, and Cebuano, and we are currently porting it to Arabic, Chinese and Russian, as part of the MUSE project[3].

5.1 NE in Slavonic languages

The Bulgarian NE recogniser (Paskaleva *et al.* 02) was built using three main processing resources: a tokeniser, a gazetteer and a semantic grammar built using JAPE. There was no POS tagger available in Bulgarian, and consequently we had no need of a sentence splitter either. The main changes to the system were in terms of the gazetteer lists (e.g. lists of first names, days of the week, locations etc. were tailored for Bulgarian), and in terms of some of the pattern matching rules in the grammar. For example, Bulgarian makes far more use of morphology than English does, e.g. 91% of Bulgarian surnames could be directly recognised using morphological information. The lack of a POS tagger meant that many rules had to be specified in terms of orthographic features rather than parts of speech. An example Bulgarian text with highlighted named entities is shown in Figure 2.

5.2 NE in Romanian

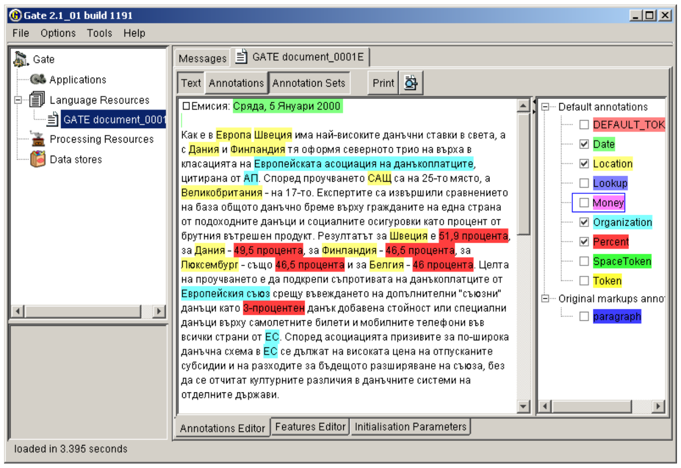
The Romanian NE recogniser (Hamza *et al.* 03) was developed from ANNIE in a similar way to the Bulgarian one, using tokeniser, gazetteer and a JAPE semantic grammar (see Figure 3).

Romanian is more flexible language than English in terms of word order; also agglutinative e.g. definite articles attach to nouns, making a definite and indefinite form of both common and proper nouns.

As with Bulgarian, the tokeniser did not need to be modified, while the gazetteer lists and grammar rules needed some changes, most of which were fairly minor. For both Bulgarian and Romanian, the modifications necessary were easily implemented by a native speaker who did not require any other specialist skills beyond a basic grasp of the JAPE language and the GATE architecture. No Java skills or other programming knowledge was necessary. The Gate Unicode kit was invaluable in enabling the preservation of the diacritics in Romanian, by saving them with UTF-8 encoding.

In order to evaluate the language-independence of ANNIE's named entity recognition rules we ran an experiment comparing the performance of the English grammars with the Romanian gazetteer

[3]http://www.dcs.shef.ac.uk/nlp/muse/



Емисия: Сряда, 5 Януари 2000

Как е в Европа Швеция има най-високите данъчни ставки в света, а с Дания и Финландия тя оформя северното трио на върха в класацията на Европейската асоциация на данъкоплатците, цитирана от АП. Според проучването САЩ са на 25-то място, а Великобритания - на 17-то. Експертите са извършили сравнението на база общото данъчно бреме върху гражданите на една страна от подоходните данъци и социалните осигуровки като процент от брутния вътрешен продукт. Резултатът за Швеция е 51,9 процента, за Дания - 49,5 процента, за Финландия - 46,5 процента, за Люксембург - също 46,5 процента и за Белгия - 46 процента. Целта на проучването е да подкрепи съпротивата на данъкоплатците от Европейския съюз срещу въвеждането на допълнителни "съюзни" данъци като 3-процентен данък добавена стойност или специални данъци върху самолетните билети и мобилните телефони във всички страни от ЕС. Според асоциацията призивите за по-широка данъчна схема в ЕС се дължат на високата цена на отпусканите субсидии и на разходите за бъдещото разширяване на съюза, без да се отчитат културните различия в данъчните системи на отделните държави.

Figure 2: Bulgarian named entities in GATE

Entity Type	Precision	Recall
Address	0.81	0.81
Date	0.67	0.77
Location	0.88	0.96
Money	0.82	0.47
Organisation	0.75	0.39
Percent	1	0.82
Person	0.68	0.78
Overall	0.82	0.67

Table 1: Average P + R per entity type, obtained with English NER grammar set

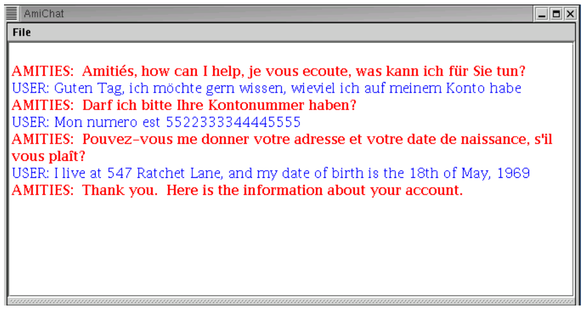
Entity Type	Precision	Recall
Address	0.96	0.93
Date	0.95	0.94
Location	0.92	0.97
Money	0.98	0.92
Organisation	0.95	0.89
Percent	1	0.99
Person	0.88	0.92
Overall	0.95	0.94

Table 2: Average P + R per entity type, obtained with Romanian NER grammar set

lists to the performance of the Romanian grammars, which were an extended set containing some rules specific to the language. The results are shown in Tables 1 and 2 respectively and show that without any adaptation of the grammars, only by collecting gazetteer lists for Romanian, ANNIE was able to achieve 82% precision and 67% recall. Once the system was customised to Romanian the performance was in line with that for English, i.e., 95% precision and 94% recall.

5.3 NE in other languages

ANNIE has also been adapted to perform NE recognition on English, French and German dialogues in the AMITIES project[4], a screenshot of which is shown in Figure 4. Since French and German are more similar to English in many ways than e.g. Slavonic languages, it was very easy to adapt the gazetteers and grammars accordingly.



AMITIES: Amitiés, how can I help, je vous ecoute, was kann ich für Sie tun?
USER: Guten Tag, ich möchte gern wissen, wieviel ich auf meinem Konto habe
AMITIES: Darf ich bitte Ihre Kontonummer haben?
USER: Mon numero est 5522333344445555
AMITIES: Pouvez-vous me donner votre adresse et votre date de naissance, s'il vous plaît?
USER: I live at 547 Ratchet Lane, and my date of birth is the 18th of May, 1969
AMITIES: Thank you. Here is the information about your account.

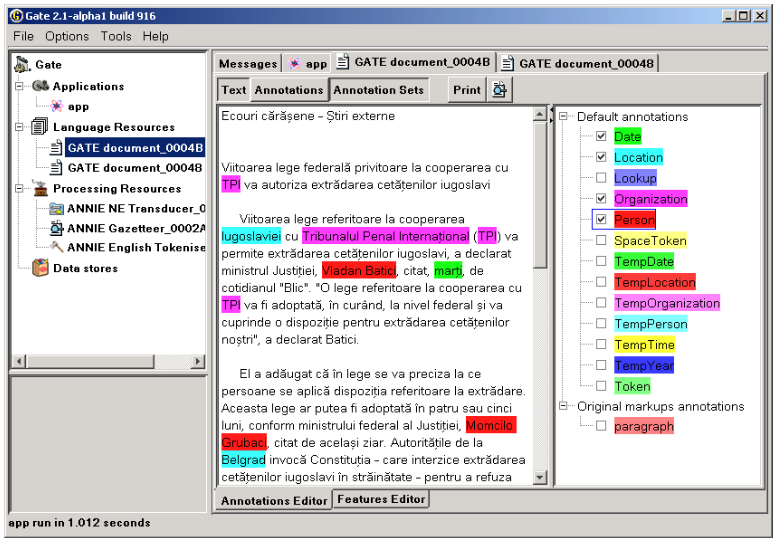
Figure 4: AMITIES multilingual dialogue

5.4 Surprise languages

We carried out further experiments as part of the TIDES-based "surprise language program", which requires various NLP tasks such as IE, IR, summarisation and MT to be carried out in a month on a surprise language, the nature of which is not known in advance.

Here we will concentrate on the dry-run experiment which ran for 10 days on the Cebuano language, which is spoken by 24% of the population in the Philippines, and is the lingua franca of the South Philippines. As part of that effort, we adapted ANNIE to the Cebuano language.

[4]http://www.dcs.shef.ac.uk/nlp/amities



Ecouri cărăşene - Ştiri externe

Viitoarea lege federală privitoare la cooperarea cu TPI va autoriza extrădarea cetăţenilor iugoslavi

Viitoarea lege referitoare la cooperarea Iugoslaviei cu Tribunalul Penal Internaţional (TPI) va permite extrădarea cetăţenilor iugoslavi, a declarat ministrul Justiţiei, Vladan Batici, citat, marţi, de cotidianul "Blic". "O lege referitoare la cooperarea cu TPI va fi adoptată, în curând, la nivel federal şi va cuprinde o dispoziţie pentru extrădarea cetăţenilor noştri", a declarat Batici.

El a adăugat că în lege se va preciza la ce persoane se aplică dispoziţia referitoare la extrădare. Aceasta lege ar putea fi adoptată în patru sau cinci luni, conform ministrului federal al Justiţiei, Momcilo Grubaci, citat de acelaşi ziar. Autorităţile de la Belgrad invocă Constituţia - care interzice extrădarea cetăţenilor iugoslavi în străinătate - pentru a refuza

Figure 3: Romanian news text annotated in GATE

Cebuano system	**P**	**R**	**F**	**Baseline system**	**P**	**R**	**F**
Person	71	65	68	Person	36	36	36
Organization	75	71	73	Organization	31	47	38
Location	73	78	76	Location	65	7	12
Date	83	100	92	Date	42	58	49
Total	**76**	**79**	**77.5**	**Total**	**45**	**41.7**	**43**

Table 3: NE results on the news texts

We succeeded in our adaptation task, without the help of a native speaker, because most of the rules for NE recognition in English are based on POS tags and gazetteer lookup of candidate and context words (more detail is given in e.g. (Cunningham *et al.* 02b)). Assuming similar morphological structure and word order, the default grammars are therefore not highly language-specific (as discussed in Section 5.2). We did not have time to make a detailed linguistic study of Cebuano, however it turned out that after we adapted the ANNIE part-of-speech (POS) tagger and gazetteer lists to Cebuano, the rules were performing successfully without any adaptation.

The performance was boosted further by using ANNIE's orthographic coreference module (orthomatcher) to boost recognition of unknown words. This works by matching entities tagged as Unknown (i.e., unclassified entity) with other types of entities (Person, Location etc.) if they match according to the coreference rules. For example, "Smith" on its own might be tagged as Unknown, but if "John Smith" is tagged as a Person, the orthomatcher will match the two entities and retag "Smith" as a Person. Our experiment showed that the rules were not particularly language-specific, given a language with similar morphology and word order, so the orthomatcher can be used directly, without modification in such languages (we had a similar experience with Bulgarian, Romanian and Chinese). Manual inspection of texts showed that the orthomatcher was helpful in improving recall. For example, it recognised "Pairat" as a Person due to coreference with "Leo Pairat" which was correctly recognised as a Person by the named entity grammars. Although we were not focusing on coreference per se, we noticed that many coreferences were correctly identified, which proves indeed that the rules used are not particularly language-specific.

We evaluated the performance of the adapted system on 21 news documents from two different Cebuano web sites. These texts were annotated by a local Cebuano speaker prior to our experiment, and the automated scoring tools in GATE (Cunningham *et al.* 02a) were used to evaluate the results of the system. The results (in terms of Precision, Recall and F-measure) are shown in

Table 3, together with with the results from our baseline system, the default ANNIE system for English, which we ran on the same test set. ANNIE typically scores for Precision and Recall in the 90th percentile for English news texts.

6 Conclusion

In this paper we presented GATE – a Unicode-based NLP infrastructure particularly suitable for the creation and multilingual adaptation of Information Extraction systems. The different pre-processing components, i.e., tokeniser, gazetteer, and POS tagger are designed to be easily adaptable to new languages. As demonstrated by our experience with adapting ANNIE to a variety of languages, the named entity recognition and coreference algorithms are relatively language independent and also easy to adapt or extend to new languages.

The advantages of our approach is that it requires little involvement of native speakers (mainly for evaluation purposes and possibly for gazetteer creation) and a small amount of annotated data. Therefore, fast adaptations from one language to another are possible with relatively low overhead, unlike many machine learning-based IE systems (e.g., (Bikel *et al.* 99)) which require big amounts of annotated data. However, for languages where such big amounts of annotated data does exist, we have now created an automatic gazetteer acquisition method that can be used to reduce further the overhead of porting ANNIE to new languages.

Future work will continue in the direction of improving multilingual support, among other things. The most important issues to be addressed are integration of morphological tools for improved support for inflected languages (e.g., (Declerck & Crispi 03)), automatic language and encoding identification (e.g., (Ignat *et al.* 03)), and further work on automatic acquisition of gazetteer lists from annotated corpora.

Acknowledgements

Work on GATE has been funded by EPSRC grants GR/K25267 (GATE), GR/M31699 (GATE2), GR/N15764/01 (IRC AKT), and GR/N19106 (EMILLE) and several smaller grants.

We would like to thank Markus Kramer from Max Planck Institute, Nijmegen, for providing us with the IPA and Chinese input methods.

The work on adapting ANNIE to Bulgarian was carried out in collaboration with Elena Paskaleva, Milena Yankova, and Galia Angelova, as part of the EC-funded project BIS-21 BULGARIAN INFORMATION SOCIETY, CENTER OF EXCELLENCE FOR EDUCATION, SCIENCE AND TECHNOLOGY IN 21 CENTURY (ICA1-2000-70016).

References

(Appelt 96) D.E. Appelt. The Common Pattern Specification Language. Technical report, SRI International, Artificial Intelligence Center, 1996.

(Bikel *et al.* 99) D. Bikel, R. Schwartz, and R.M. Weischedel. An Algorithm that Learns What's in a Name. *Machine Learning, Special Issue on Natural Language Learning*, 34(1-3), Feb. 1999.

(Bird & Liberman 99) S. Bird and M. Liberman. A Formal Framework for Linguistic Annotation. Technical Report MS-CIS-99-01, Department of Computer and Information Science, University of Pennsylvania, 1999. http://xxx.lanl.gov/-abs/cs.CL/9903003.

(Brill 92) E. Brill. A simple rule-based part-of-speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, Trento, Italy, 1992.

(Cunningham 02) H. Cunningham. GATE, a General Architecture for Text Engineering. *Computers and the Humanities*, 36:223–254, 2002.

(Cunningham *et al.* 02a) H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.

(Cunningham *et al.* 02b) H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, and C. Ursu. *The GATE User Guide.* http://gate.ac.uk/, 2002.

(Declerck & Crispi 03) T. Declerck and C. Crispi. Multilingual Linguistic Modules for IE Systems. In *Proceedings of Workshop on Information Extraction for Slavonic and other Central and Eastern European Languages (IESL'03)*, Borovets, Bulgaria, 2003.

(Gambäck & Olsson 00) B. Gambäck and F. Olsson. Experiences of Language Engineering Algorithm Reuse. In *Second International Conference on Language Resources and Evaluation (LREC)*, pages 155–160, Athens, Greece, 2000.

(Grishman 97) R. Grishman. TIPSTER Architecture Design Document Version 2.3. Technical report, DARPA, 1997. http://www.itl.nist.gov/div894/894.02/-related_projects/tipster/.

(Hamza *et al.* 03) O. Hamza, K. Bontcheva, D. Maynard, V. Tablan, and H. Cunningham. Named Entity Recognition in Romanian. In *Proceedings of Workshop on Information Extraction for Slavonic and other Central and Eastern European Languages (IESL'03)*, Borovets, Bulgaria, 2003.

(Ignat *et al.* 03) C. Ignat, B. Pouliquen, A. Ribeiro, and R. Steinberger. Extending and Information Extraction Tool Set to Eastern-European Languages. In *Proceedings of Workshop on Information Extraction for Slavonic and other Central and Eastern European Languages (IESL'03)*, Borovets, Bulgaria, 2003.

(Maynard *et al.* 01) D. Maynard, V. Tablan, C. Ursu, H. Cunningham, and Y. Wilks. Named Entity Recognition from Diverse Text Types. In *Recent Advances in Natural Language Processing 2001 Conference*, pages 257–274, Tzigov Chark, Bulgaria, 2001.

(Maynard *et al.* 03) D. Maynard, V. Tablan, and H. Cunningham. Ne recognition without training data on a language you don't speak. In *ACL Workshop on Multilingual and Mixed-language Named Entity Recognition: Combining Statistical and Symbolic Models*, Sapporo, Japan, 2003.

(McEnery *et al.* 00) A.M. McEnery, P. Baker, R. Gaizauskas, and H. Cunningham. EMILLE: Building a Corpus of South Asian Languages. *Vivek, A Quarterly in Artificial Intelligence*, 13(3):23–32, 2000.

(Paskaleva *et al.* 02) E. Paskaleva, G. Angelova, M.Yankova, K. Bontcheva, H. Cunningham, and Y. Wilks. Slavonic named entities in gate. Technical Report CS-02-01, University of Sheffield, 2002.

(Pastra *et al.* 02) K. Pastra, D. Maynard, H. Cunningham, O. Hamza, and Y. Wilks. How feasible is the reuse of grammars for Named Entity Recognition? In *Proceedings of 3rd Language Resources and Evaluation Conference*, 2002.

(Thompson & McKelvie 97) H. Thompson and D. McKelvie. Hyperlink semantics for standoff markup of read-only documents. In *Proceedings of SGML Europe'97*, Barcelona, 1997.