Adapting A Robust Multi-Genre NE System for Automatic Content Extraction

Diana Maynard[1], Hamish Cunningham[1], Kalina Bontcheva[1], and Marin Dimitrov[2]

[1] Dept of Computer Science, University of Sheffield
211 Portobello St, Sheffield, UK S1 4DP
{diana,hamish,kalina}@dcs.shef.ac.uk
http://nlp.shef.ac.uk
[2] Sirma AI Ltd, Ontotext Lab
38A Hristo Botev Blvd, Sofia 1000, Bulgaria
marin@sirma.bg

Abstract. Many current information extraction systems tend to be designed with particular applications and domains in mind. With the increasing need for robust language engineering tools which can handle a variety of language processing demands, we have used the GATE architecture to design MUSE - a system for named entity recognition and related tasks. In this paper, we address the issue of how this general-purpose system can be adapted for particular applications with minimal time and effort, and how the set of resources used can be adapted dynamically and automatically. We focus specifically on the challenges of the ACE (Automatic Content Extraction) entity detection and tracking task, and preliminary results show promising figures.

Keywords: information extraction, named entity recognition, robust NLP

1 Introduction

Most Information Extraction (IE) systems are designed to extract fixed types of information from documents in a specific language and domain [4, 1, 5]. To increase suitability for end-user applications, IE systems need to be easily customisable to new domains [17]. Driven largely by US Government initiatives such as TIPSTER [3] and MUC [18], work on IE, and in particular on named entity recognition (NE), has largely focused on narrow subdomains, such as newswires about terrorist attacks (MUC-3 and MUC-4), and reports on air vehicle launches (MUC-7). In many applications, however, the type of document and domain may be unknown, or a system may be required which will process different types of documents without the need for tuning.

Many existing IE systems have been successfully tuned to new domains and applications - either manually or semi-automatically – but there have been few advances in tackling the problem of making a single system robust enough to

forego this need. It is well-known that the adaptation of existing systems to new domains is hindered by both ontology and rule bottlenecks. A substantial amount of knowledge is needed, and its acquisition and application are non-trivial tasks.

For systems to deal successfully with unknown or multiple types of source material, they must not only be able to cope with changes of domain, but also with changes of *genre*. By this we mean different forms of media (e.g. emails, transcribed spoken text, written text, web pages, output of OCR recognition), text type (e.g. reports, letters, books, lists), and structure (e.g. layout options). The genre of a text may therefore be influenced by a number of factors, such as author, intended audience and degree of formality. For example, less formal texts may not follow standard capitalisation, punctuation or even spelling formats. Most IE systems require manual tuning in order to deal with these different kinds of texts; however, we have developed a system which uses NE technology to detect different text types, and then automatically fires different processing resources depending on the text.

We first describe the default system in Section 2 below. We then describe in Section 3 the background for our work, namely the ACE program. We continue in Section 4 with the challenges posed by ACE, and how the system has been adapted, using the GATE technology, to overcome them. In Section 5 we discuss some aspects of evaluation in IE, and give details of preliminary results with the ACE system. Finally in Section 6 we summarise the approach used and discuss some ongoing improvements to the system.

2 The MUSE System for Named Entity Recognition

The MUSE system (MUlti-Source Entity finder) [14] has been developed within GATE, a General Architecture for Text Engineering [6, 7], which is an architecture, framework and development environment for language processing research and development.

MUSE is based on ANNIE, A Nearly-New IE system, which comes as part of the standard (freely available) GATE package. Figure 1 depicts a full IE pipeline based on a LaSIE[3] backend with ANNIE shallow analysis.

The MUSE system comprises a version of ANNIE's main processing resources: tokeniser, sentence splitter, POS tagger, gazetteer, finite state transduction grammar 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.). It does not need to be modified for different applications or text types.

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.

[3] the original IE system developed within the first version of GATE

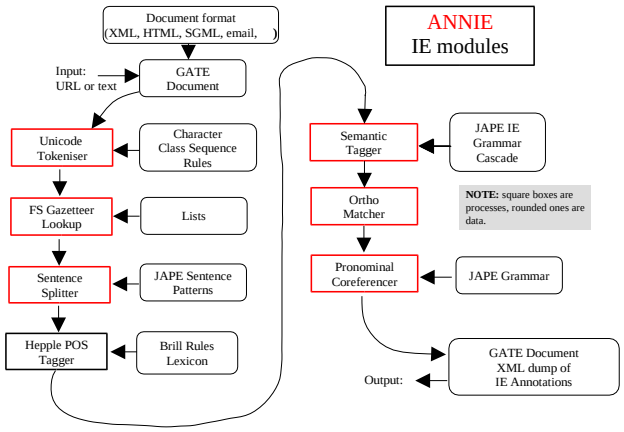

Fig. 1. ANNIE, A Nearly-New IE system

The **tagger** is a modified version of the Brill tagger, which adds a part-of-speech tag as a feature to each Token annotation. Neither the splitter nor the tagger are a mandatory part of the NE system, but the annotations they produce can be used by the semantic tagger (described below), in order to increase its power and coverage.

The **gazetteer** consists of lists such as cities, organisations, days of the week, etc. It contains some entities, but also names of useful key words, such as company designators (e.g. 'Ltd.'), titles, etc. The lists are compiled into finite state machines, which can match text tokens.

The **semantic tagger** (or JAPE transducer) consists of hand-crafted rules written in the JAPE pattern language [8], which describe patterns to match and annotations to be created. Patterns can be specified by describing a specific text string or annotation (e.g. those created by the tokeniser, gazetteer, document format analysis, etc.).

The **orthomatcher** performs co-reference, or entity tracking, by recognising relations between entities. It also has a secondary role in improving NE recognition by assigning annotations to previously unclassified names, based on relations with existing entities.

3 The ACE Program

The ACE program aims to encourage the development of robust NLP applications, by promoting faster system development from given linguistic resources, which encourages the development of general purpose retargetable systems, using a variety of methods from richly annotated corpora. It also aims to promote

the design of more general purpose linguistic resources, and the development of general purpose standalone systems.

The ACE entity detection and tracking (EDT) task goes beyond existing NE tasks, in that all mentions of an entity (in the form of a name, description or pronoun) must be recognised and classified (based on reference to the same entity). The aim is to produce structured information about entities, events, and relations among them. Although, as with MUC, the texts to be used for the tasks are newswires, the scope of the task is widened by measuring results not only on standard written texts, but also on texts produced from automatic speech recognition (ASR) and optical character recognition (OCR) output. ACE focuses on the core extraction challenge however, rather than on ASR or OCR algorithms.

The ACE Program aims to create a powerful new generation of algorithms capable of extracting information accurately and robustly from human language data, and to represent that information in a form suitable for subsequent automatic analysis. Potential uses of ACE output include more precise forms of information retrieval, data mining, and the development of large knowledge bases.

3.1 The Entity Detection and Tracking (EDT) Task

The EDT task is divided into the following 5 recognition subtasks:

- entities (Person, Organization, Location, Facility and GPE[4]);
- entity attributes: type (Name, Nominal or Pronominal);
- entity mentions [optional] - entity tracking (similar to co-reference);
- mention roles [optional] - for GPEs, each mention has an optional role associated with it (Person, Organization, Location or GPE);
- mention extents [optional] - detection of the whole NP span, rather than just the head.

One of the main differences between ACE and MUC is that where MUC dealt with the *linguistic* analysis of text, ACE deals with the *semantic* analysis of text. Discussion of how the system was adapted to perform this deeper level of analysis can be found in Section 4.

3.2 Data

The ACE tasks are carried out on the following types of input data:

- Text from newswire
- (Degraded) text produced from broadcast news by ASR
- (Degraded) text produced from newspapers by OCR
- Clean versions of text produced from broadcast news
- Clean versions of text produced from newspapers

[4] Geo-Political Entity (essentially, any kind of location which has a government, such as a city or country)

Unlike in MUC, where the texts were all related to a specific domain, the ACE news texts encompass a wide variety of domains, such as sport, politics, religion, popular culture, etc.

3.3 Participation

The University of Sheffield team has been participating in the ACE program, and has therefore faced the challenge of adapting the core MUSE system to deal specifically with the ACE tasks. We focus here on the EDT task, since we are not currently participating in the RDC (relation detection and characterisation) task.

4 Adapting MUSE to perform EDT

The MUSE system is designed to process multiple types of text in a robust fashion, with minimal adaptation. However, it does require some tuning in order to deal with new applications where either the guidelines for entity recognition are different, or where new tasks are involved. In this section, we describe the adaptation of the MUSE system to build the ACE system, used to perform the EDT task. We describe the two parts of the task (named entity detection and tracking) separately.

Although the two tasks are considerably different from the MUSE basic NE task, the time and effort spent tuning the system was remarkably small, because of the robust design and flexible architecture of GATE. The ACE system is just one example of this: we have also implemented similar adaptations to build the HaSIE system for IE and summarisation from company reports [15], the OldBaileyIE system for information extraction from old English court reports [2], and the Romanian NE system [12], among others.

4.1 Named Entity Detection

There are a number of features of the EDT task that have required adaptation to the original MUSE system.

1. The entity types are different. MUSE recognises the standard MUC entity types of Person, Location, Organisation, Date, Time, Money and Percent, plus the additional entity types Address (including email, phone numbers, urls, etc.) and Identifier. ACE has the first three, plus the additional types Facility (which subsumes some entities previously belonging to the MUSE types Organisation and Location), and GPE, which subsumes some, but not all, entities from the MUSE types Person, Location and Organisation). This means that on the one hand, some entities are grouped together, and on the other hand, that finer distinctions are made (for example, the division into Location and GPE).

2. A word or string does not consistently belong to an entity type in the same way that it (usually) does in MUSE; for example, in ACE "English" could be annotated as a Person or Organization, depending on the situation. Contextual information and intended meaning are very important, and world knowledge, intuition or pragmatic information may be necessary to categorise a particular occurrence of an entity correctly.
3. Entities may be used metonymously. This means that they must also be classified as such, by means of the use of roles. A metonymous mention of an entity is given a literal and an intended role. For example, in *the museum announced its new exhibit*, the entity *museum* is a facility that houses art, but in this context it is being used to describe the organisation behind the museum, and the mention should therefore be annotated as having the literal role Facility and the Intended role Organisation.
4. For some domains, such as Sport, a string may have a different entity type. For example, names of cities and countries are often used to represent team names, and should therefore be annotated as Organisations and not GPEs.

Due to the modular nature of the GATE architecture, it is relatively straightforward to adapt processing resources such as the grammar and gazetteer lists of MUSE in order to deal with the first problem. Firstly, procedural and declarative knowledge in GATE are separate, which minimises the adaptation necessary. Secondly, within the processing resources, foreground and background information are largely distinguished, so that background knowledge (such as that required for the tokenisation, name matching etc.) can remain untouched and only foreground information (that which is very specific to the domain or application) needs to be modified. For example, changes can be made to specific parts of the grammar while ensuring that the remaining parts of the grammar will be unaffected. This is of enormous benefit in reducing the time and effort spent adapting the system to a new application.

To deal with different text types, we introduced a conditional controller mechanism, which enables the user to set up the chain of processing resources according to features found in the text. A JAPE transducer is first run over the text to determine its type, by identifying salient features of the document.

The transducer adds a feature to the document indicating its domain. The conditional controller is set up so that depending on the presence or absence of certain features (e.g. a "sport" feature), a processing resource can be fired or not. In this way, we can set the controller to fire, for example, a particular sports grammar if the sports feature is present, and a regular grammar if it is not. The sports grammar annotates certain locations as Organisations, whereas the regular grammar annotates them as GPEs. The same mechanism can also be used for dealing with any kind of metonymy: if certain features are detected in the text, we can set the controller to run certain extra grammars, or to omit existing ones. The firing of other resources such as gazetteer lists or POS taggers can also be handled in the same way.

4.2 Entity Tracking

The entity tracking part of the EDT task has required the construction of some entirely new components to the MUSE system, and some further adjustments to existing parts. The main problems for the system were detection of pronominal entity mentions, coreference of proper names, and anaphora resolution.

The detection of pronouns is quite straightforward, since MUSE already contains a POS tagger which enabled us to recognise them by simply making some minor additions to the JAPE grammars.

Recognising the name mentions (i.e. finding coreference chains between proper nouns) required use of the MUSE orthomatcher, with some minor modifications to ensure that the correct entity types were considered. For example, we added new rules to match locations with their respective adjectives (e.g. France and French), and we extended the rules for Organisations so that they also took care of matches between Facilities (since Facility was not an entity type used in MUSE).

Finding the pronominal mentions (anaphora resolution) required more extensive work in the form of an entirely new module, the pronominal coreference module, which was built using the JAPE formalism. The GATE framework provided the basis for this to be designed, developed and slotted into the architecture with minimum effort. Detailed analysis of the data revealed that a few simple rules could account for the vast majority of pronominal cases. For example, 80-85% of the occurrences of [he,his,she,her] referred to the closest person of the same gender in the same sentence, or, if unavailable, the closest preceding one. In most cases, they referred back to named entities rather than nominal references. Likewise, [it,its] are handled in the same way, but with scope restriction (because there are many nominals). Currently the rules do not allow for cataphora, but occurrences of these were rare. Pronouns occurring in quoted speech are handled by a separate grammar, and require slightly more complex rules. More details of the pronominal coreference module can be found in [9].

5 Evaluation

We have evaluated the ACE system using Precision and Recall, which we calculated using the evaluation facilities developed within GATE: the AnnotationDiff tool and the Benchmarking Tool. These are particularly useful not just as a final measure of performance, but as a tool to aid system development by tracking progress and evaluating the impact of changes as they are made. The evaluation tool (AnnotationDiff) enables automated performance measurement and visualisation of the results, while the benchmarking tool enables the tracking of a system's progress and regression testing.

5.1 The AnnotationDiff Tool

Gate's AnnotationDiff tool enables two sets of annotations on a document to be compared, in order to either compare a system-annotated text with a reference

(hand-annotated) text, or to compare the output of two different versions of the system (or two different systems). For each annotation type, figures are generated for precision, recall, F-measure and false positives.

5.2 Benchmarking tool

GATE's benchmarking tool differs from the AnnotationDiff in that it enables evaluation to be carried out over a whole corpus rather than a single document. It also enables tracking of the system's performance over time. Performance statistics are output for each text in the corpus, and overall statistics for the entire corpus, in comparison with a reference corpus.

5.3 Results

The latest evaluations for the ACE system scored 82-86% precision and recall, depending on the text type (newswire scored the highest, though it also had the most substitution errors[5] (7%). For detection of pronominal mentions (anaphora resolution) the recall was low (around 40%) but precision was high (83% for broadcast news, and slightly less for the other text types). The main reason for recall being low was that we currently do not attempt to identify all types of pronouns. Detection of name mentions was also high, with the precision for newswires at 93%, and a slightly lower score for the other text types.

5.4 Other Evaluation Metrics

The most commonly used evaluation metrics in IE - precision, recall, error rate and F-measure - all stem from the IR field, and consequently so does much of the literature on this topic, e.g. [19, 13, 11]. Typically in IR, people want to know how many relevant documents are to be found in the top N percent of the ranking. This is reflected well by the precision metric. In IE, however, people typically want to know for each entity type how many entities have been correctly recognised and classified. In IE therefore, the proportion of entities belonging to each type has an impact on the outcome of the evaluation, in a way that the proportion of relevant documents in the collection does not in IR. Evaluation mechanisms in IE can also be affected by the notion of *relative document richness*, i.e. the relative number of entities of each type to be found in a set of documents. For this reason, error rate is sometimes preferred in the IE field, because, unlike precision, it is not dependent on relative document richness.

5.5 Cost-based Evaluation

Using error rate instead of precision and recall means, however, that the F-measure can no longer be used. An alternative method of getting a single bottom-line number to measure performance is the cost-based metric. This appears to

[5] where an entity was correctly detected but allocated the wrong entity type

be becoming a favourite with the DARPA competitions, such as TDT2 [10], and is the method used in ACE. The model stems from the field of economics, where the standard model "Time Saved Times Salary" measures the use of the direct salary cost to an organisation as a measure of the value [16].

Another advantage of this type of evaluation is that it enables the evaluation to be adapted depending on the user's requirements. A cost-based model characterises the performance in terms of the cost of the errors (or the value of the correct things, depending on whether you see the glass as half-empty or half-full). For any application, the relevant cost model is applied, and expected prior target statistics are defined.

For a cost-based error model, a cost would typically be associated with a miss and a false alarm, and with each category of result (e.g. recognising Person might be more important then recognising Date correctly). Expected costs of error would typically be based on probability (using a test corpus). This makes the assumption that a suitable test corpus is available, which has the same rate of entity occurrence (or is similar in content) to the evaluation corpus. If necessary, the final score can be normalised to produce a figure between 0 and 1, where 1 is a perfect score.

The official ACE evaluations are carried out using a cost-based function based on error rate, for the reasons described above. However, since these evaluations are closed (i.e. we are not able to divulge any results other than those of our system), it is not very informative to discuss these cost-based results in isolation, since a single value means little without direct comparison, and therefore we have given our results in terms of the more widely recognised Precision and Recall.

6 Conclusions

In this paper, we have described a robust general-purpose system for NE across different kinds of text, and its adaptation for use in a specific application, the ACE EDT task. We have shown that the flexibility and open design of the GATE architecture and MUSE system enables this kind of adaptation to be carried out with minimal time and effort. Approximately 8 person weeks were spent on the development of the coreference module (which was not ACE-specific, but intended for general use within GATE); 6 person weeks were spent on the adaptation, (including those modules developed specifically for ACE), and a further 2 person weeks on familiarisation with the task and guidelines.

The conditional controller mechanism enables the system to be adapted automatically and dynamically according to the characteristics of the text being processed. Current results are promising and we aim to improve on them in the near future, with modifications to coreference and metonymy, and the use of learning mechanisms for error corrections and ambiguity resolution.

References

1. D. Appelt. An Introduction to Information Extraction. *Artificial Intelligence Communications*, 12(3):161–172, 1999.

2. K. Bontcheva, D. Maynard, H.Saggion, and H. Cunningham. Using human language technology for automatic annotation and indexing of digital library content. In *submitted to European Conference on Digital Libraries*, 2002.
3. J. Cowie, L. Guthrie, W. Jin, W. Odgen, J. Pustejowsky, R. Wanf, T. Wakao, S. Waterman, and Y. Wilks. CRL/Brandeis: The Diderot System. In *Proceedings of Tipster Text Program (Phase I)*. Morgan Kaufmann, California, 1993.
4. J. Cowie and W. Lehnert. Information Extraction. *Communications of the ACM*, 39(1):80–91, 1996.
5. H. Cunningham. Information Extraction: a User Guide (revised version). Research Memorandum CS–99–07, Department of Computer Science, University of Sheffield, May 1999.
6. H. Cunningham. GATE, a General Architecture for Text Engineering. *Computers and the Humanities*, 36:223–254, 2002.
7. 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.
8. H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, and C. Ursu. *The GATE User Guide*. `http://gate.ac.uk/`, 2002.
9. M. Dimitrov. *A Light-weight Approach to Coreference Resolution for Named Entities in Text.* MSc Thesis, University of Sofia, Bulgaria, 2002. `http://www.ontotext.com/ie/thesis-m.pdf`.
10. Jonathan G. Fiscus, George Doddington, John S. Garofolo, and Alvin Martin. Nist's 1998 topic detection and tracking evaluation (tdt2). In *Proc. of the DARPA Broadcast News Workshop*, Virginia, US, 1998.
11. W.B. Frakes and R. Baeza-Yates, editors. *Information retrieval, data structures and algorithms*. Prentice Hall, New York, Englewood Cliffs, N.J., 1992.
12. O. Hamza, V. Tablan, D. Maynard, C. Ursu, H. Cunningham, and Y. Wilks. Named Entity Recognition in Romanian. Technical report, Department of Computer Science, University of Sheffield, 2002. Forthcoming.
13. C.D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT press, Cambridge, MA, 1999. Supporting materials available at `http://www.sultry.arts.usyd.edu.au/fsnlp/` .
14. 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*, Tzigov Chark, Bulgaria, 2001.
15. Diana Maynard, Kalina Bontcheva, Horacio Saggion, Hamish Cunningham, and Oana Hamza. Using a text engineering framework to build an extendable and portable IE-based summarisation system. In *Proceedings of the ACL Workshop on Text Summarisation*, 2002.
16. Peter Sassone. Cost-benefit methodology for office systems. *ACM Transactions on Office Information Systems*, 5(3):273–289, 1987.
17. S. Soderland. Learning to extract text-based information from the world wide web. *Proceedings of Third International Conference on Knowledge Discovery and Data Mining (KDD-97)*, 1997.
18. Beth Sundheim, editor. *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, Columbia, MD, 1995. ARPA, Morgan Kaufmann.
19. Yiming Yang. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1:67–88, 1998.