Towards a semantic extraction of named entities

Diana Maynard, Kalina Bontcheva, Hamish Cunningham
Dept of Computer Science
University of Sheffield
Sheffield, S1 4DP, UK
diana@dcs.shef.ac.uk

Abstract

In this paper, we discuss the new challenges posed by the progression from information extraction to content extraction, as demonstrated by the ACE program. We explore whether traditional IE approaches are sufficient, and describe the adaptation of a generic IE system to this kind of application. Results suggest that a deeper level of processing is necessary to achieve excellent results in all areas, although rule-based systems can still produce results of a reasonable quality with a small amount of adaptation. In particular, the task of entity detection and tracking on texts of varying genre and quality is one of the most challenging.

1 Introduction

US Government initiatives such as MUC (SAIC 98) and TIPSTER (ARPA 96) paved the way for the development of many current Information Extraction (IE) systems. In a short space of time, systems were able to recognise named entities with precision and recall scores in the 90th percentile in narrow domains such as newswires about terrorist attacks. The challenge then became one of adapting systems to new domains, and extended tasks such as co-reference, template filling, question-answering and summarisation. In recent years the demand has been growing for commercial applications to perform such tasks. There are two main consequences of this: first, systems need to be more adaptable and portable than ever before; second, the level of detail of analysis has become more important. It is no longer enough to be able to recognise and classify text at string level – in order to make use of classified entities we need to be able to recognise them at a semantic level. Systems also need to be more robust and able to deal with issues such as degraded texts rather than perfect transcriptions with correct spelling, punctuation, and grammar. The interesting challenge is whether existing methods for information extraction are sufficient to deal with these new demands, or whether deeper processing techniques such as full syntactic and semantic parsing, and deeper forms of knowledge such as dictionaries, ontologies and even pragmatics or real world knowledge will become necessary.

In this paper, we discuss the new challenges posed to IE systems by the ACE (Automatic Content Extraction) program, and describe how a generic system was adapted for this task. We discuss to what extent such a system can be modified in order to perform the deeper level of analysis necessary, and whether the advent of such challenges means that new techniques may be necessary. Our findings reveal that adaptation to the ACE task was relatively straightforward, but in order to gain equivalent scores to those achieved in e.g. MUC, current techniques may not be sufficient.

2 The ACE Program

The ACE program began in September 1999, administered by NSA, NIST, and the CIA. It was designed as "a program to develop technology to extract and characterise meaning from human language". Formal evaluations of ACE algorithm performance are held at approximately 6 month intervals, and are open to all sites who wish to participate, but the results of the evaluations are closed. For this reason we can only publish here details of internal evaluations rather than official scores. ACE includes both Entity Detection and Tracking (EDT) and Relation Detection and Characterisation (RDC). EDT is broadly comparable with the MUC Named Entity (NE) task, while RDC is broadly comparable with the MUC template elements task, although both ACE tasks are more challenging than their MUC forerunners. We shall limit our discussion to the EDT task; however, the inclusion of RDC as a task in ACE supports our claim about the complexity of content extraction.

The main objective of ACE is to produce structured information about entities, events, and rela-

tions among them. The program aims to encourage a powerful new generation of robust and retargetable NLP applications, by promoting faster system development 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. Potential uses of ACE output include more precise forms of information retrieval, data mining, development of large knowledge bases, and automatic large-scale annotation for the Semantic Web.

2.1 Data

The scope of the ACE program is broader from that of MUC in that the texts are of varying source and quality, ranging from standard newswires to degraded texts produced from automatic speech recognition (ASR) and optical character recognition (OCR) output. Although ACE focuses on the core extraction task, rather than on ASR or OCR algorithms, these low quality texts provide a unique challenge for systems.

2.2 Entity Detection and Tracking (EDT)

The EDT set of tasks involves detecting each unique entity (of specified types) mentioned in the source text, and tracking its mentions. 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).

There are 5 recognition subtasks of EDT:

- entities (Person, Organization, Location, Facility and GPE[1]);
- entity attributes: type (Name, Nominal or Pronominal);
- entity mentions - entity tracking;
- mention roles - for GPEs, each mention has an optional role associated with it (Person, Organization, Location or GPE);
- mention extents - detection of the whole NP span

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

3 Information vs. content extraction

One reason for the popularity of MUC may be because very high scores were achievable, particularly for the named entity task, in which systems were typically able to achieve scores in the 90th percentile. This meant that there soon became less incentive for sites to push their systems to new developments, particularly as comparing MUC scores soon became one of the gold standards of system evaluation. The ACE program was designed partly to fulfil this need for challenge, such that new advances will be obtained in the development of robust systems capable of fast adaptation to new tasks, and a deeper analysis of language.

3.1 A deeper analysis

One of the main differences between ACE and MUC is that where the MUC NE task dealt with the *linguistic* analysis of text, ACE deals with its *semantic* analysis. The MUC NE task tagged selected segments of text whenever that text represented the name of an entity. In ACE, these names are viewed as *mentions* of the underlying entities. The main task is to detect (or infer) the entities themselves, along with selected attributes (shown in Figure 1).

Entity detection output is in the form of a (unique) ID for the entity, a set of entity attributes, and information about the way in which the entity is mentioned: document ID, mention level (one of Name, Nominal or Pronominal), mention head and mention extent. An example entity output is shown in Figure 2.

3.2 More varied text

Unlike in MUC, where the texts were all related to a specific domain and comprised only texts from one particular source type, the ACE news texts encompass a wide variety of domains, such as sport, politics, business, religion, and popular culture, and cover differing genres and styles such as broadcast news and newspapers. While this may seem trivial, in practice it can have a high impact on the results obtained (Maynard *et al.* 02).

3.3 Human annotation

The difficulty level of the EDT task is confirmed by the experience of human annotators. On the MUC NE task, human annotators could typically

```
ID (a unique ID key)
TYPE (Person, Organization, GPE, Location, Facility)
CLASS (Generic, Specific)
LEVEL (Name, Nominal, Pronoun)
ORIGIN (Database, Corpus)
SUBTYPE (Country, City) - for GPEs
CONTINENT (e.g. Asia, Europe) - for countries
COUNTRY (e.g. Egypt, Australia) - for cities
{names}
{titles}
{mentions}
```

Figure 1: Entity Attributes

```
<entity ID="ft-airlines-27-jul-2001-2"
    GENERIC="FALSE"
    entity_type = "ORGANIZATION">
    <entity_mention ID="M003"
                    TYPE = "NAME"
                    string = "National Air Traffic Services">
    </entity_mention>
    <entity_mention ID="M004"
                    TYPE = "NAME"
                    string = "NATS">
    </entity_mention>
    <entity_mention ID="M005"
                    TYPE = "PRO"
                    string = "its">
    </entity_mention>
    <entity_mention ID="M006"
                    TYPE = "NAME"
                    string = "Nats">
    </entity_mention>
 </entity>
```

Figure 2: Sample entity output

achieve around 97% precision and recall (Marsh & Perzanowski 98). In an experiment conducted by BBN[2], a team of annotators, who were experienced at ACE-style annotation, achieved results on a corpus of 15,000 words worth a value of 82.8%, scoring 4.1% False Alarms, 11.3% misses, and 2% substitutions.

4 Adapting a generic NE system for content extraction

In this section, we describe the development of the MACE system, which was adapted from the ANNIE system available as part of GATE (Cunningham *et al.* 02a). The robust design and flexible architecture of GATE and ANNIE meant that tuning the system to deal with the multiple domains and text types necessary for ACE was very straightforward. In contrast, the adaptation from a linguistic to a semantic analysis of the text involved more substantial effort, not least in terms of understanding the complexities of the task rather than programming time and skill.

4.1 System Overview

Like ANNIE, MACE consists of a number of processing resources run sequentially: tokeniser, sentence splitter, part-of-speech tagger, gazetteer lookup, JAPE semantic grammar, and orthographic co-reference. Of these, we modified the gazetteers, semantic grammar and orthographic co-reference modules, and added new modules to perform genre identification, pronominal coreference and nominal coreference. We also added a switching controller mechanism to enable the automatic selection of different processing resources. These modules are all described in the following section.

4.2 Named Entity Detection

Since it is a relatively new program, there is much less training data available for ACE than MUC, and manual annotation is time-consuming. In this respect, traditional rule-based systems have an advantage over machine learning techniques such as BBN's Identifinder (Bikel *et al.* 99), which require very large quantities of annotated data.

On the other hand, rule-based systems suffer from other problems. First, the entity types are different. ANNIE recognises the standard MUC entity types of Person, Location, Organisation, Date, Time, Money and Percent, plus the additional entity types Address and Identifier. ACE has the additional types Facility (which subsumes some entities usually considered to be Organisations and Locations), and GPE (which subsumes some, but not all, entities more usually found as Person, Location and Organisation). This entails both "lumping" and "splitting" of the standard entity types, which means that many rules have to be completely rewritten. For example, an ANNIE rule which recognises Locations which appear in gazetteer lists has to be rewritten for MACE, such that certain types of Locations (rivers, mountains, continents, etc.) remain as Locations, while others such as cities and countries are tagged as GPEs. The GPEs then need to be broken down further into subtypes through role assignment. This means that one ANNIE rule becomes 5 or 6 separate rules in MACE.

However, the modular nature of the GATE architecture makes it relatively straightforward to adapt processing resources such as the grammar and gazetteer lists, firstly because procedural and declarative knowledge in GATE are kept separate, and secondly, because within the processing resources, foreground and background information are largely distinguished. This means 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 it will be unaffected. This is of enormous benefit in reducing the time and effort spent adapting the system to a new application.

The semantic grammars are written in JAPE (Cunningham *et al.* 02b) and consist of phases which are run sequentially and compiled into finite-state transducers. The ANNIE NE recognition grammar contains 21 phases and a total of 187 rules for 9 annotation types – an average of 20.8 rules per entity type. The MACE grammars consist of 15 phases and a total of 180 rules for 5 entity types – an average of 36 rules per entity type. However, an experienced JAPE user may be able to write a dozen new rules in several minutes, so the number of new rules is not itself significant - rather, it is the increased complexity of the new MACE rules that is most important.

[2] personal communication

	PER	ORG	GPE	LOC	FAC
NAME	1.0	0.5	0.25	0.1	0.05
NOM	0.2	0.1	0.05	0.02	0.01
PRO	0.04	0.02	0.01	0.004	0.002

Table 1: ACE Application paramaters

4.3 Entity Tracking

The entity tracking part of the EDT task requires detection and coreference of pronominal and nominal entity mentions and coreference of proper names. In theory, pronominal and nominal entities can also exist in their own right, where there is no named mention of an entity, though in practice, nominal entities are fairly rare, and pronominal entities extremely rare. Tracking of name mentions is handled by the ANNIE orthomatcher module, with a few modifications. Tracking of nominal and pronominal mentions are handled by two new coreference modules. We shall not discuss these resources here due to space restrictions, but more details can be found in (*ref. witheld*).

5 Evaluation

There are several differences in the way evaluation is carried out in MUC and ACE. MUC uses the well known metrics from Information Retrieval of precision, recall and F-measure. The F-measure used (where the balance of precision and recall can be adjusted, though they are usually equally weighted) gives a single percentage which makes it easy to compare the scores of different systems. ACE, however, uses a cost-based scoring metric which favours certain entity types over others, and can potentially be biased towards different types of error. For a more detailed discussion of IE metrics and the cost-based algorithm, see *(ref. withheld)*.

For ACE, the application parameters applied to each entity type and level were chosen according to their perceived importance, and are shown in Table 1. The cost parameters for Miss, False Alarm and Error are all equal.

Aside from the metric, there are two other major factors – related to the task definitions themselves – which impact the scores of MUC and ACE and mean that the two cannot be directly compared. First, the way coreference is handled directly affects the score. In MUC, coreference is treated as a separate task from NE recognition, and is scored separately. This means that if two entities are correctly identified and classified, but the co-reference between them is missed, they still generate a perfect score for the NE task. In ACE, however, entity tracking (which is essentially coreference) is an integral part of the entity recognition task. Although a separate score is given for entity mention recognition, the two subtasks are related because if two entity mentions are recognised, but not related to the same entity, a spurious entity will be generated, which will negatively affect the entity score. Let us say that we have two strings in the text: "Bush" and "George Bush", and that our system correctly identifies both as a Person entity, but fails to make any coreference between them. In MUC, we get a perfect score for NE recognition, but get no score for coreference. In ACE, instead of generating one entity with two mentions, we generate two entities, each with one mention. We therefore do not get a perfect score for entity recognition, because we have a spurious entity generated which counts against us. So in ACE, poor coreference resolution will lower our entity score. In MUC, scores for the coreference task were quite low, which explains why coreference has such an impact on the results in ACE.

The second factor is the impact of metonymy and problematic entities. In MUC, these issues were skirted over by allowing multiple possible entity types, or optional annotations, where the case was unclear. Getting either one of the possibilities correct was sufficient. In ACE, however, there is only ever one correct answer, and in cases where it is unclear, there is no leeway for a different interpretation from that of the official annotators.

5.1 Results for the MACE system

Since the ACE evaluation is closed, we cannot release official figures on our ACE score. However, internal evaluations on ACE data, using Precision and Recall, gave figures of 82.4% Precision and 82.0% Recall on newswire texts. We then performed some experiments with the MUC data, in order to get some reference for how well our system performs in comparison to others. We made some minor modifications to the system in order to recognise the different entity types used in MUC, and some of the different guidelines (for example, MUC does not deal with metonymy as ACE does). We did not score the system on Dates and Numbers, since previous experiments with our system (*ref. witheld*) have shown that we can

Text	P	R	F
ACE	82.4	82	82.2
MUC	89	90	89.5

Table 2: Comparison of results on ACE and MUC texts

	ANNIE system			MACE system		
Text	P	R	F	P	R	F
ACE	55.8	59.7	57.8	82.4	82	82.2
News	89.0	91.8	90.4	61.9	60.0	61.0

Table 3: Comparison of results on ACE and NEWS texts

achieve near perfect precision and recall for these types, and because there are no corresponding entities in ACE. The results for each evaluation are detailed in Table 2.

Note that the scores on the MUC texts are for ENAMEX only. Had we scored TIMEX and NUMEX as well, our overall results would have improved. The best systems in MUC-7 achieved F-measures in the mid-90s, and without even any tuning to the domain, we have achieved a higher score than our original system entered in MUC-7 (*ref witheld*). If MACE had been tuned to the task and domain, it is likely we could have achieved an even higher score.

We also compared our default ANNIE system with the MACE system on a set of news texts (NEWS) and on the official ACE texts (ACE). The news texts were a blind test set of 92 articles about business news taken from the Web, similar to those on which the ANNIE system was trained. The ACE texts were the 86 broadcast news texts used for the September 2002 ACE evaluation (another blind set). Table 3 shows the results for Precision, Recall and F-measure. The aim was to see how much improvement the MACE system produced over ANNIE on [ACE], and how well the MACE system worked on MUC-style annotations in [NEWS]. We ran the systems with no modifications at all, so for example the MACE systems recognised most of the [NEWS] Locations as GPEs. Responses were only measured where there was both a Key and Response for that annotation type, i.e. we did not include cases where the system produced GPE annotations but there were no Key GPE annotations (as when running the ACE results on the [NEWS] texts).

On [ACE], the MACE system achieved an F-measure of 82.2% compared with the ANNIE system's 57.8%. The initial adaptation took approximately 6 person weeks (excluding work on the new modules for coreference etc), although the system was further improved in the 6 months between the first and second ACE evaluations (by adding a nominal coreference module and fine-tuning some of the rules and lists). On [NEWS], the ANNIE system achieved an F-measure of 90.4%, while MACE achieved 61%.

The main reason why there is a greater difference in score between the systems on [NEWS] than on [ACE] is because MACE is in many ways a much more specific system, designed specifically for ACE rather than a more flexible multi-purpose system. Also, as discussed earlier, the difference in semantics of the Location entities in the two systems caused a great deal of error when running systems on their non-respective texts. We therefore also ran an experiment where we modified the MACE grammars to transform all GPE entities into Locations, and reran the experiment on [NEWS]. The scores increased to 72% Precision and 84% Recall, giving an F-measure of 78% (a more realistic measure of the differences between the systems).

These results all show that substantial improvements can be made on a baseline system in a relatively short space of time, taking into account the quite substantial differences in requirements for the two systems.

6 Conclusions

The MACE system we have described here is entirely rule-based, in contrast with the majority of current IE systems which combine rule-based with learning mechanisms, e.g. (Bikel *et al.* 99; Borthwick *et al.* 98; Day *et al.* 98). While rule-based systems can perform at equivalent levels on mixed-case text (Bikel *et al.* 99), rule construction is generally time-consuming. On the other hand, learning mechanisms such as Hidden Markov Models (HMMs) generally require large volumes of training data.

MACE, demonstrates that if a system and its underlying architecture are well-designed, the effort needed for adaptation from a generic system is not necessarily expensive. The debate remains open as to the intrinsic superiority of any particular method for more intricate language pro-

cessing tasks such as the ACE program. We are currently experimenting with adding a learning-based approach using HMMs on top of the existing rule-based system, and preliminary results suggest that this will bring some improvements to the score.

For NE tasks such as that of MUC, there is no clear winner any more, but the more semantically-based tasks that are now emerging, and which are difficult even for humans, will bring new surprises, and certainly new challenges. Clearly, however, a single score in a single evaluation does not prove a system's overall superiority. In the real world, there are many other factors to be considered, such as speed of processing, overall ease of use, adaptability and portability to new (and even unknown) languages, domains and text types all play an important role. For the end user, one of the most important factors may indeed be not the system's overall performance, but the ease with which minimally trained users can adapt the system to their own needs without assistance from the developers.

References

(ARPA 96) Advanced Research Projects Agency. *Proceedings of the TIPSTER Text Program (Phase II)*. Morgan Kaufmann, California, 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.

(Borthwick *et al.* 98) A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. Description of the Mene named entity system as used in MUC-7. In *Proceedings of the MUC-7 Conference*, NYU, 1998.

(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.

(Day *et al.* 98) D. Day, P. Robinson, M. Vilain, and A. Yeh. MITRE: Description of the *Alembic* System Used for MUC-7. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, 1998.

(Marsh & Perzanowski 98) E. Marsh and D. Perzanowski. Muc-7 evaluation of ie technology: Overview of results. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*. `http://www.itl.nist.gov/iaui/894.02/-related_projects/muc/index.html`, 1998.

(Maynard *et al.* 02) D. Maynard, V. Tablan, H. Cunningham, C. Ursu, H. Saggion, K. Bontcheva, and Y. Wilks. Architectural elements of language engineering robustness. *Journal of Natural Language Engineering – Special Issue on Robust Methods in Analysis of Natural Language Data*, 8(2/3):257–274, 2002.

(SAIC 98) SAIC. Proceedings of the Seventh Message Understanding Conference (MUC-7). `http://www.itl.nist.gov/iaui/894.02/-related_projects/muc/index.html`, 1998.