Multi-Source and Multi-Lingual Information Extraction

Diana Maynard

3rd September 2003

Abstract

This paper describes a robust and easily adaptable system for named entity recognition from a variety of different text types. Most information extraction systems need to be customised according to the domain, either by collecting a large set of training data or by rewriting grammar rules, gazetteer lists etc., both of which methods can be costly and time-consuming. The MUSE system incorporates a modular set of resources from which different subsets can be mixed and matched as required. The process of selecting the correct resources depending on the text type is fully automatic. This method could be easily extended to deal with different languages in the same way. Results show figures in the 90th percentile for news texts, and slightly lower for other text types.

1 Introduction

MUSE is an Information Extraction (IE) system developed within the GATE architecture [3], designed to perform named entity recognition (NE) and coreference on a variety of different types of text, such as news reports, emails, spoken transcriptions, etc. Named entity recognition is an important pre-requisite for many other tasks related to Language Engineering, such as Question Answering, Summarisation, Cross-Language Information Retrieval, Machine Translation and the development of the Semantic Web. In this paper we detail the basic components of the MUSE system, and discuss its adaptation to different text types and languages.

2 Processing Resources

In this section we describe the components (processing resources) which constitute the MUSE system. The application consists of a conditional controller operating over a pipeline of processing resources (PRs), shown in Figure 1. Most of these resources rely on finite-state algorithms and the JAPE (Java Annotations Pattern Engine) language [4]. More detailed descriptions of all these resources can be found in [8]. Each PR produces annotations which serve to label the text in some way, and may be used by following PRs in the pipeline.

2.1 Tokeniser

The tokeniser splits the text into simple tokens such as numbers, punctuation, words of different types (e.g. capitalised, mixed-case, etc), and white space. This resource is almost totally language- and domain-independent.

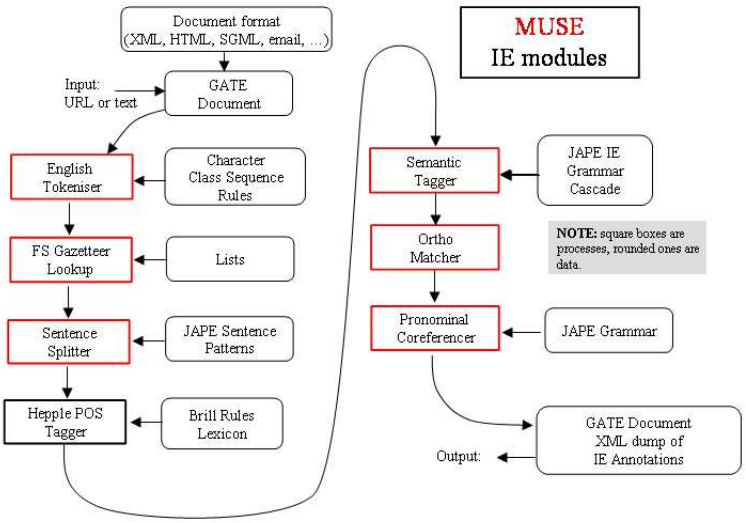

Figure 1: Basic MUSE architecture

2.2 Sentence Splitter

The sentence splitter is a cascade of Finite State Transducers (FSTs) which segments running text into sentences. This is required for other PRs such as the tagger, and is implemented in JAPE.

2.3 Part-of-Speech (POS) Tagger

The POS tagger [6] is a modified version of the Brill tagger [2], which produces a POS tag (e.g. Verb, Noun etc.) as an annotation on each token. The POS tagger uses a default lexicon and ruleset, created by training on an annotated corpus (the Wall Street Journal), though it can be retrained on any appropriate corpus as necessary, or modified manually.

2.4 Gazetteer Lists

A set of gazetteer lists is used to aid NE recognition. Each list represents a set of names, such as cities, organisations, days of the week, first names, etc. An index file is used to access these lists, which are compiled into Finite State Machines. It is important to note that such lists are not sufficient in themselves to perform NE recognition, for two main reasons. First, it is impossible to create exhaustive lists, especially for open-ended sets such as names of organisations, surnames (for English) etc. Second, such names can be highly ambiguous. For example, "India" could be the name of a country or a person; "May" could be the month or a person, or not a named entity at all but a modal verb; "David Lloyd" could be the name of a company or a person. The lists are therefore just used as a starting point from which NEs can be recognised. We also use lists of context words, such as company endings (Ltd, plc, etc) to help recognise boundaries of NEs.

2.5 Semantic Tagger

The Semantic tagger consists of a set of JAPE grammars which use the annotations previously generated to recognise NEs. The grammars contain hand-written pattern-action rules which recognise e.g. annotations from the POS tagger and gazetteer, and combine them to produce new NE annotations over patterns. For example, a rule might recognise a first name (from the gazetteer module) followed by a proper noun (from the POS tagger), and annotate this pattern as a Person. This rule could be written in JAPE as follows:

```
Rule: Person1
(
 {Lookup.majorType == firstname}
 {Token.category == NNP}
 ):label
 -->
 :label.Person = {rule = "Person1"}
```

2.6 Orthomatcher

The orthomatcher performs orthographic coreference between NEs in the text, i.e. it might recognise that "James Smith" and "Mr Smith" are the same person. It uses a set of hand-crafted rules, some of which apply for all entities, others applying for only specified types of entities.

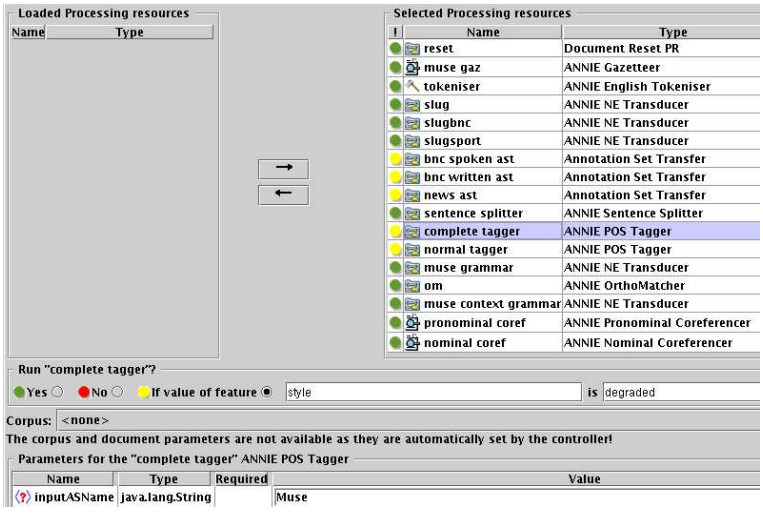

Figure 2: Switching Controller Mechanism

3 Switching Controller

The switching controller is a mechanism which operates over the pipeline of processing resources, and enables different PRs to be substituted in the application depending on a set of criteria.

Type	P	R	F
Person	81.7	93.7	87.3
Org	92.7	84.2	88.2
Location	96.2	93.5	95.0
Date	89.9	85.5	87.7
Money	97.8	98.2	98.0
Percent	99.4	98.4	98.9
Total	**93.5**	**92.3**	**92.9**

Table 1: Results for News Corpus

Instead of having a fixed chain of resources, each resource can be made to fire only if a certain feature is present in the document. For example, a different POS tagger might be used for text in all lowercase as opposed to text in mixed case. The controller enables the correct tagger to be used on a certain text, depending on its features. A special categorisation module is used to assign features to texts based on factors such as capitalisation, text type, text source, language, domain etc. Figure 2 shows a screenshot of the switching controller, where the "complete tagger" is only used if the document has a feature "style" with value "degraded".

4 Evaluation

The MUSE system has been evaluated in terms of Precision, Recall and Fmeasure on the following kinds of NEs: Person, Organisation, Location, Date, Time, Money, Percent, Address. On news texts it achieves 92.9% Fmeasure across all entity types. The breakdown for different entity types is shown in Table 1. Results differ according to the text type, performing best on news texts and least well on spoken transcriptions (where incorrect punctuation, spelling and capitalisation cause problems for the system).

5 Other Approaches to IE

The MUSE system is entirely rule-based; however, many systems have turned to machine learning (ML) techniques either as an alternative or in addition to using hand-coded rules. The most common method is to use Hidden Markov Models, e.g. Identifinder [1] and Phrag [10]. These methods work by training the system on an annotated training corpus, and have the advantage that they can be easily adapted to a new domain or language, without a linguistic expert. The main disadvantages, however, are that they require large amounts of training data, which is not always readily available, and that they can be difficult to finetune without an intricate knowledge of how the algorithms work.

6 Adaptation to Other Languages

Recently we have focused on adapting MUSE to other languages. The advantage of the modular architecture is that only those components which are not language-independent need to be replaced. Usually this would entail the POS tagger, gazetteer lists, and parts of the semantic

Figure 3: Screenshot of Chinese System

tagger, while the other components can be left untouched. We have developed systems for Romanian [5], Bulgarian [11], Cebuano [9], Hindi [7], French and German[1],Chinese, Arabic and Russian. Figure 3 shows a screenshot of the Chinese system.

Speed and ease of adaptation are becoming increasingly important factors in the choice of system. We recently took part in a surprise language exercise where we had 10 days to develop a system for Cebuano (spoken in the S. Philippines) and a month for Hindi (though dealing with font and encoding issues for the latter meant that practically, we only had about 10 days to develop the Hindi system).These two languages are quite different in terms of both their linguistic properties and in the amount of data available. Other participating systems (who all used ML techniques) found the latter to be a more important factor in terms of system adaptation, for such techniques are highly dependent on at least some, and preferably a large amount, of training data. Rule-based techniques have an advantage in this respect, as was the case for Cebuano, but on the other hand the linguistic properties of the language may be more problematic to deal with for such techniques than for machine learning ones. For Hindi we scored an Fmeasure of 62%, while for Cebuano we achieved 78%.

7 Conclusions

In this paper we have described a rule-based system for Named Entity Recognition from different text types and languages. It is the first system we are aware of that is able to deal with multiple genres inside a single application with no manual intervention. Results are on a level with machine learning approaches, but MUSE has the advantage of requiriring very small amounts of training data. The system is fast, flexible and robust, and easy to adapt to new applications as required.

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

References

[1] 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.

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

[3] 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.

[4] H. Cunningham, D. Maynard, and V. Tablan. JAPE: a Java Annotation Patterns Engine (Second Edition). Research Memorandum CS–00–10, Department of Computer Science, University of Sheffield, November 2000.

[5] O. Hamza, D. Maynard V.Tablan, C. Ursu, H. Cunningham, and Y. Wilks. Named Entity Recognition in Romanian. Technical report, Department of Computer Science, University of Sheffield, 2002.

[6] Mark Hepple. Independence and commitment: Assumptions for rapid training and execution of rule-based POS taggers. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000)*, Hong Kong, October 2000.

[7] D. Maynard, V. Tablan, K. Bontcheva, and H. Cunningham. Rapid customisation of an information extraction system for surprise languages. *Submitted to Special issue of ACM Transactions on Asian Language Information Processing: Rapid Development of Language Capabilities: The Surprise Languages*, 2003.

[8] D. Maynard, V. Tablan, K. Bontcheva, H. Cunningham, and Y.Wilks. Multi-source entity recognition – an information extraction system for diverse text types. Research Memorandum CS–03–02, Department of Computer Science, University of Sheffield, April 2003.

[9] 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.

[10] D. Palmer, J. Burger, and M. Ostendorf. Information extraction from broadcast news speech data. In *Proceedings of the DARPA Broadcast News and Understanding Workshop*, 1999.

[11] 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.