Learning Decision Trees for Named-Entity Recognition and Classification

Georgios Paliouras[1], **Vangelis Karkaletsis**[1], **Georgios Petasis**[1] and **Constantine D. Spyropoulos**[1]

Abstract. We propose the use of decision tree induction as a solution to the problem of customising a named-entity recognition and classification (NERC) system to a specific domain. A NERC system assigns semantic tags to phrases that correspond to named entities, e.g. persons, locations and organisations. Typically, such a system makes use of two language resources: a recognition grammar and a lexicon of known names, classified by the corresponding named-entity types. NERC systems have been shown to achieve good results when the domain of application is very specific. However, the construction of the grammar and the lexicon for a new domain is a hard and time-consuming process. We propose the use of decision trees as NERC "grammars" and the construction of these trees using machine learning. In order to validate our approach, we tested C4.5 on the identification of person and organisation names involved in management succession events, using data from the sixth Message Understanding Conference. The results of the evaluation are very encouraging showing that the induced tree can outperform a grammar that was constructed manually.

1 INTRODUCTION

Machine learning techniques have recently been proposed as a promising solution to a major problem in language engineering: the construction of lexical resources. Most of the real-world language engineering systems make use of a variety of lexical resources, in particular grammars and lexicons. The use of general-purpose resources is ineffective, since in most applications a specialised vocabulary is used, which is not supported by general-purpose lexicons and grammars. For this reason, significant effort is currently put into the construction of generic tools that can quickly adapt to a particular thematic subdomain. The adaptation of these tools mainly involves the acquisition of domain-specific semantic lexical resources.

Named-entity recognition and classification (NERC) is the identification of proper names in text and their classification as different types of named entity, e.g. persons, organisations, locations, etc. The NERC system tags phrases in text with their corresponding named-entity type. This is an important subtask in most language engineering applications, in particular information retrieval and extraction. The categories into which the named entities are classified constitute semantic information that varies significantly in different thematic domains. For instance the identification of organisation names makes sense in financial news, but not in the scientific literature. The lexical resources that are typically included in a NERC system are a lexicon, in the form of gazetteer lists, and a grammar, responsible for recognising the entities that are either not in the lexicon or appear in more than one gazetteer lists. The manual adaptation of those two resources to a particular domain is very time-consuming and in some cases impossible, due to the lack of experts. Thus, the automatic acquisition of the resources from a training corpus, i.e., text data, is highly desirable. This article deals with one half of this problem, namely the acquisition of the NERC grammar.

Research on the problem of automatically acquiring a NERC grammar from text data is still at the initial exploratory stage, with a small number of solutions having been proposed so far. On the other hand, the problem of acquiring automatically recognition and classification models has been studied extensively in machine learning, giving rise to a variety of successful methods. Among these, the decision-tree induction algorithm C4.5 [1] is admittedly the most widely used. Among the merits of the algorithm are: its applicability in a variety of learning problems, its computational efficiency and the human-readable format of the induced models, i.e., the decision trees. These properties of C4.5 have led us to select it for our experiments in learning domain-specific NERC "grammars". In our experiments, we used data from the sixth Message Understanding Conference (MUC-6) [2]. The Message Understanding Conferences have been established as the main event for evaluating new information extraction systems on a common task. The work presented in this article has been performed in the context of the research project ECRAN,[2] which focused on the adaptation of an information extraction system to new thematic domains and languages.

Related work on the adaptation of NERC systems is presented in section 2. The NERC task, as realised in our approach, is presented in Section 3. Section 4 presents the experimental results and section 5 uses the conclusions of this work to start drawing the picture for the future.

[1] Institute of Informatics and Telecommunications, NCSR "Demokritos", 15310, Aghia Paraskevi, Attikis, GREECE, email: {paliourg, vangelis, costass}@iit.demokritos.gr

[2] ECRAN (Extraction of Content: Research at Near-market) was a Language Engineering project (LE-2110, Telematics Applications Programme) funded partially by the European Commission and involving Thomson (FR), SIS (GE), Univ. of Sheffield (UK), NCSR "Demokritos" (GR), Univ. of Ancona (IT), Univ. of Tor Vergata (IT) and Univ. of Fribourg (SU).

2 RELATED WORK

As mentioned above, the NERC task involves the exploitation of gazetteers and named-entity grammars, which need to be updated when the NERC system is adapted to a new domain. The exploitation of learning techniques to support the adaptation task has recently attracted the attention of researchers in language engineering. Nymble [3], Alembic [4,5], AutoLearn [6], RoboTag [7] and the NYU system for MUC-7 [8] are examples of NERC systems exploiting supervised learning techniques (either statistical or symbolic). The approach presented here belongs also in this category. On the other hand, the NERC system developed for Italian in the project ECRAN [9] and the approach of multi-level bootstrapping in [10] are examples of systems exploiting unsupervised learning.

Nymble [3] uses statistical learning to acquire a Hidden Markov Model (HMM) that recognises named entities in text. The HMM labels each word either with one of the desired named-entity types, e.g., person, organisation, etc., or with the label 'NOT-A-NAME'. The HMM states are grouped into regions, one region for each desired type plus one for the other text. Within each region, a statistical bigram language model is used, emitting exactly one word upon entering each state. In addition to the generation of the word, states may also generate features for that word pertaining to numeric expressions, capitalisation and membership in lists of important words, e.g., company designators, person titles, etc. Evaluation results reported for Nymble in [3] are of the order of 90%. In the MUC-7 [11] competition the system achieved 89% recall and 92% precision. The success of the system is attributed to the use of the correct features in the encoding of words, e.g. capitalisation, and the probabilistic modelling of the recognition system.

Named-entity recognition in Alembic [4] uses the transformation-based rule learning approach introduced in Brill's work on part-of-speech tagging [12]. The approach aims at discovering automatically phrase rule sets in a maximum error-reduction scheme for selecting the next rule in a set. The search for a rule set in a training corpus starts by applying an initial labelling function. The learning procedure needs then to consider every possible rule *r*, computing the improvement in phrase labelling caused by applying *r* to the current state. The rule that causes the larger reduction of the residual error in the training data is selected as the next rule in the set. Learning continues until a criterion is fulfilled which is usually taken as the point where performance improvement falls below a threshold. According to the article [4], an important aspect of this approach is the fact that the system learns rules that can be freely intermixed with hand-engineered ones. This system has also been quite successful in evaluation tests. The results presented in [4], are 88% recall and 83% precision, while a manually constructed system on the same data achieved 91% recall and 92% precision.

The AutoLearn system [6] uses the ID3 algorithm [13]. The learning algorithm uses the hand-tagged training data to construct decision trees that detect the start and end points of specific types of named entity. For the training, the data are converted into tuples of five words. Each tuple is marked as having the start (or end) of a specific named-entity type at the middle word, i.e., the third word of the tuple. Following a path from the root to the leaves of the tree, a sequence of tests is performed resulting in a decision about whether a word is the start (or end) of a specific named-entity type. The Autolearn system did not perform well in the MUC-6 evaluation. It achieved only 47% recall and 81% precision. This was mainly due to the limited use of lexical resources, such as the gazetteer lists. Improved methods, based on the approach of decision tree induction, are presented in [14] and [7]. These methods use an improved version of the ID3 algorithm, known as C4.5 [1]. The method presented in [14] is of limited interest, due to the fact that it only deals with half of the NERC task, namely the classification of NEs to the correct semantic class. The identification of NEs is done using manually created patterns. The RoboTag system presented in [7], formulates the learning task in manner similar to Autolearn, i.e., it classifies words as being potentially the start/end of a particular NE type. RoboTag's performance on the MUC-6 data is better than that of Autolearn, due to the use of gazetteer lists and other lexical resources. A variant of the same approach was used in the system presented by the New York University (NYU) in the Multilingual Entity Task (MET-2) of MUC-7 [8,15]. The results for this system are only for Japanese texts and therefore not comparable to most of the other systems. RoboTag, which was also evaluated on Japanese texts for MET-1, achieves slightly worse results than those presented in [15]. RoboTag's overall F-measure was 83.6%, while the NYU system reaches 85%.

The NERC system developed for Italian in ECRAN [9], uses unsupervised learning to expand a manually constructed system and improve its performance. The manually constructed system is used to identify the named entities and classify as many of them as possible. About 20% remain unclassified and are fed to the unsupervised learning algorithm. The algorithm uses untagged (raw text) learning corpus, a shallow syntactic parser, a "seed" gazetteer and a dictionary of synonyms. The parser extracts elementary syntactic relations (ESL) from the corpus, e.g. subject-object, noun-preposition-noun, etc. The ESLs are used to characterise the named entities that have been classified by the manually constructed system. This characterisation is then used to classify the remaining named entities.

The multi-level bootstrapping approach presented in [10] can be characterised as partially supervised, because it learns from a small number of tagged examples and a larger volume of untagged data. The aim is to induce a set of information extraction patterns, which can be used to identify and classify named entities in text. The system starts off by generating exhaustively all candidate extraction patterns, using an earlier system called AutoSlog [16]. Additionally, a small number of seed examples of named entities are provided. The most useful pattern for recognising the seed examples is selected and used to expand the set of classified named entities. This process is repeated for a pre-specified number of iterations. The end result is a dictionary of named entities and the extraction patterns that correspond to them.

Our approach resembles that used in AutoLearn, RoboTag and the NYU system for Japanese, since we are using the C4.5 algorithm. Contrary to the rule-learning method used in Alembic, C4.5 is a general-purpose and ready-to-use system. At the same time it provides comprehensible rules, like the ones used in Alembic and unlike the statistical HMMs, which are harder to interpret. Our main difference with previously published work using C4.5 for NERC is in the representation of the problem. All of the existing approaches that were mentioned above aim to identify the components of a phrase belonging to a particular NE type, especially its start and end points. This piece-wise approach requires a further post-processing stage, which constructs a phrase based on its individual components. This task is non-trivial, due to the many peculiarities of NEs. For instance, phrases that correspond

to organisation names very often include person names, which should not be recognised as such. The best solution presented so far to that problem [15] is to search for the most probable sequence of word-specific tags that provide a valid combined solution. Our main objection to this approach is that it introduces knowledge that is external to the induced decision tree. As a result, the decision tree - and the associated rule set to which it can be translated – ceases to be of direct use to the human expert, as it cannot be used on its own to identify NEs. In this manner, the decision-tree approach loses its advantage of direct interpretability by humans. In contrast to this post-processing approach, we propose a further pre-processing step, in which noun phrases are identified by a separate parser. Under the weak assumption that NEs are noun phrases, the decision tree can then focus on these phrases and classify them into the required NE types. A special class *non-NE* is used for phrases that are not NEs.

3 THE NERC TASK IN MUC-6

For the evaluation of C4.5 we used part of the corpus that was used for the evaluation of the systems in the MUC-6 conference [2]. The thematic domain in MUC-6 was *management succession events*, involving several types of named entity, such as person, organisation, location, date, time, money, etc. The general consensus (e.g. [17,7]) is that person and organisation types are more difficult to identify and classify. For this reason, our study focuses on these two types of entity. Our data contain 461 organisation and 373 person instances.

The objective of our work is to minimise human effort in the adaptation of the NERC system to a given domain, in this case management succession events. The NERC architecture that we used is the VIE NERC system developed at Sheffield University [18]. This system makes use of a set of gazetteer lists, consisting of person names, organisation names, company designators (such as Ltd. and Co.), person titles (such as Mr. and MD), etc., and a grammar. The information taken into account by the grammar consists of tags assigned by looking up the gazetteer lists, part-of-speech and syntactic properties of the words in a phrase. A simple bottom-up chart parser uses this grammar to identify phrases of interest in the text. Adaptation of such a system to a particular domain involves the update of the gazetteer lists and the NERC grammar. In this work, we simplify the adaptation problem, by considering only the NERC grammar. The gazetteer lists need to have been constructed beforehand. In our study we used the following lists: organisation (2,559), org_base (55), org_key (80), cdg (94), person (476), title (163), location (2,114), money (101), time (360), where the numbers in the brackets correspond to the number of entries in each list.

C4.5 requires text data to be represented in a feature-vector format. In our case, an example is a named-entity (NE) phrase, consisting of one or more words, plus some external information, i.e., words in the close neighbourhood of the NE phrase. Thus, each organisation and person instance in the MUC-6 data is represented by a feature vector. Two features are used for each word: its gazetteer tag, if it has one, and its part of speech. The feature vector consists of 14 words: 10 words for the NE phrase plus the two adjacent words on each side of the phrase. Therefore, each vector consists of 28 features, 14 part-of-speech and 14 gazetteer tags. When the NE phrase is shorter than 10 words, the remaining features are assigned a special value (a question mark), treated as a label for missing information by the algorithm. Words that are not assigned a gazetteer tag are not treated in the same manner. They are given a special tag called `NOTAG` instead. As an example of the way in which NE phrases are coded into feature vectors consider the following phrase:

… of the *Securities and Exchange Commission* in the …

where the organisation phrase is shown in italics. The vector corresponding to this phrase is the following:

```
[IN, NOTAG, DT, NOTAG, NNP,
org_key+organisation, CC, organisation, NNP,
organisation, NNP, org_base+organisation, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, IN, NOTAG, DT, NOTAG]
```

where the part-of-speech tags are to be interpreted as follows:
`IN`: preposition, `DT`: determiner, `NNP`: noun phrase, `CC`: conjunct.
The gazetteer tags appearing in the above example are: `organisation`, `org_key`, `org_base` and `NOTAG`. The phrase *Securities and Exchange Commission* appears in the list of organisations and as a result all of its component words are assigned the tag `organisation`. Note that more than one gazetteer tag may be given to a word, meaning that the word exists in more than one gazetteer list, as in the case of the word *Securities*, which is both an `org_key` and part of an `organisation` (*Securities and Exchange Commission*). Multiple tags are joined by the plus sign in the symbolism that we use.

In addition to the training examples corresponding to person and organisation NE phrases, a number of negative, i.e., non-NE, example phrases are constructed from the data. This is needed, in order to capture the dual nature of the NERC task, namely the identification *and* classification of NE phrases. By providing the learning algorithm simply with person and organisation phrases, a decision tree will be constructed that distinguishes between person and organisation names. In other words, all phrases, up to 10 words would be either an organisation or a person name for the constructed tree. By adding the negative examples, the NERC system will capture patterns that distinguish between NE and non-NE phrases. The negative examples in our study are constructed using all *noun phrases* that are not NE phrases. The number of these examples in our data is 4,333. It should be noted here that some of the non-NE noun phrases might overlap or even subsume each other. More importantly they may subsume or be subsumed by NE phrases! For instance, the noun phrase *George Black's garden* is not a named entity, but subsumes the person name *George Black.* Similarly the phrase *Greek Society for the Protection* is not a named entity, but part of the organisation name *Greek Society for the Protection of Forests.* The latter case poses an important problem for the learning algorithm, which needs to identify what is missing from the phrase, i.e., the words *of Forests*, rather than what should be in it, in order for it to be a named entity.

4 RESULTS

The aim of the experiments presented here is to evaluate the performance of the decision trees generated by C4.5 on the NERC task and gain an insight on the NERC "grammar" generated by C4.5, i.e., what information is included in the constructed decision trees.

For this purpose, the NERC task is represented as a three-class problem. The three classes are: *person*, *organisation* and *non-NE*. Thus, the decision tree generated by C4.5 performs both parts of the NERC task, i.e., identification and classification of NE phrases. In

the experiments we utilise a special facility of C4.5, which is particularly suitable for processing many-valued features. This facility allows subsets of feature values to be constructed automatically, rather than examining each feature value individually whenever the feature is used in the decision tree. The separation of feature values into subsets is based on the mutual information heuristic that C4.5 uses in the construction of decision trees. The effect of this facility on the tree constructed by C4.5 is a significant reduction of the branching factor. In this manner, the induced tree becomes more comprehensible to humans.

In the experiment different levels of tree pruning are examined, leading to decision trees of various sizes. At each size, 10-fold cross-validation is performed to gain an unbiased estimate of the performance of the system on unseen data. According to this evaluation method, the dataset is split into ten, equally-sized subsets and the final result is the average over ten runs. In each run nine of the ten subsets of the data are used to construct the decision tree and the tenth is held out for the evaluation.

The measures that were chosen for the evaluation are those typically used in the language engineering literature: *recall* and *precision*. Recall measures the number of items of a certain type (e.g. organisation) correctly identified, divided by the total number of items of this type in the training data. Precision is the ratio of the number of items of a certain type correctly identified to all items that were assigned that particular type (e.g. organisation) by the system. In total four measures are used in the experiment: recall of organisations, recall of persons, precision of organisations and precision of persons.

Finally, as a basis for comparing the results in the experiments we can use the performance of the manually constructed set of rules in the VIE NERC system [18]. The results of this system on our data are shown in Table 1.

Table 1. Performance of the manually constructed set of rules on the whole dataset.

Recall (o)	*Precision (o)*	*Recall (p)*	*Precision (p)*
69.25%	83.42%	84.97%	92.5%

Note that these results are significantly lower than the aggregate results presented for VIE in MUC-6. This is due to the difficulty in identifying person and organisation names. VIE's performance was also considerably lower than the best-performing system in MUC-6, which achieved recall and precision results above 90%, even for persons and organisations. The results in Table 1 are better for persons than for organisations. This is a more general observation in the MUC-6 results and it is due to the fact that person names are shorter and are usually either included in the gazetteers, or preceded by a person title. As a result, their identification is easier than for organisation phrases, which can be lengthy and may consist of words of various parts of speech and gazetteer types.

The performance of the decision trees in the experiment is shown in Figures 1 and 2, which display average recall and precision results from the cross-validation runs for different sizes of the decision tree, i.e., different levels of pruning. Figure 1 displays the results for the organisation phrases and Figure 2 those for the person cases. Each point in the graph is the average of the 10 values acquired in the corresponding 10-fold cross-validation experiment. Similar to the manually constructed VIE NERC system, the performance for organisations is substantially lower than that for persons. For organisations, both recall and precision fluctuate around 80% for different sizes of the tree. There does not seem to be significant fluctuation in the curves, as the size of the decision tree increases. This means, that small and simple trees perform equally well as larger trees in this task. A similar phenomenon is observed for the person entities, where the curves for recall and precision almost overlap each other at 90%, for sizes of the decision tree larger than 50 nodes. Smaller trees seem to be missing some person entities, leading to smaller recall values. In summary, the performance that is observed is around 80% recall and precision for organisations and around 90% for persons. In comparison to the manually constructed VIE system, precision is lower by about 3.5 percentage points for organisations and 2.5 for persons. However, recall is higher by about 10 percentage points for organisations and 5 points for persons. Overall, the automatically induced trees compare favourably to the manually constructed grammar.

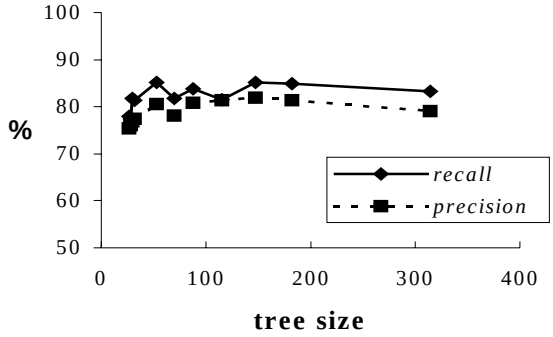

Figure 1. Recall and precision results for organisations.

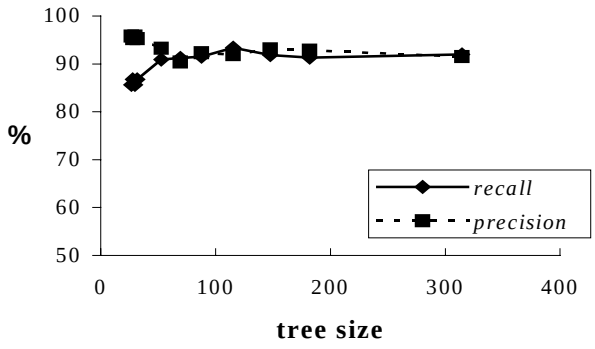

Figure 2. Recall and precision results for persons.

Furthermore, our results are comparable to the results presented for RoboTag in [7]. Unfortunately, a similar comparison with the NYU system is not possible, as we have not evaluated our system on the Japanese texts.

The good performance of our method is interesting, considering the fact that it does not require a post-processing stage, i.e., the decision trees provide directly the classification of NE phrases. Due to this fact, the decision-tree classifiers are directly usable by humans. In fact, they can easily be translated to IF – THEN – ELSE rules.[3] Figure 3 presents a set of rules, which correspond to a

[3] C4.5 incorporates a facility (C4.5rules) to automate this translation.

decision tree of 58 nodes that was generated in our experiments. At this size the performance of the trees has already reached the levels mentioned above. In particular, the performance of the tree presented in Fig. 3 on the whole data set (both training and test) is shown in Table 2.

Table 2. Performance of the representative classifier on the whole dataset.

Recall (o)	*Precision (o)*	*Recall (p)*	*Precision (p)*	*Tree Size*
89.6%	86.6%	93.0%	95.6%	58 nodes

The representative classifier in Fig. 3 captures some very interesting rules, such as the fact that a person name is usually preceded by a title (Gtag(-1) **IN** {`title, org_key+title`}). Particularly interesting are the rules that reject subphrases of organisation and person names, i.e., they classify them as *non-NE* phrases. These rules include special cases for incomplete named entities. An example of such a rule is that which says that if the gazetteer tag of the first word in the phrase is potentially an `organisation` (Gtag(1) **IN** {`location+organisation, org_key+organisation, organisation+person`}), but the gazetteer tag of the word succeeding the phrase is a company designator (`cdg`), an `organisation,` or a `person` then the phrase is a *non-NE*, because it is only part of the organisation phrase.

5 CONCLUDING REMARKS

In this article we evaluated the behaviour of C4.5 on the task of learning decision trees to recognise and classify named entities in text. This approach reduces significantly the effort needed for customising a NERC system to a particular domain. The experiments that were performed led to a variety of useful conclusions about the usability of C4.5 in this task:

- The first important result is that the performance of named-entity recognisers generated by C4.5 compares favourably to that of a manually constructed using the same lexicon. Thus, the

Representative classifier: (abbreviations: POS=part-of-speech tag, Gtag=gazetteer tag)

IF Gtag(-1) **IN** {`title, org_key+title`} **THEN** *person*
ELSEIF Gtag(-1) **IN** {`NOTAG, currency_unit, date, location, org_key+organisation, organisation, person`} **THEN:**
  IF Gtag(1) = `NOTAG` **THEN:**
    IF POS(1) **IN** {`NNP, VBG`} **THEN**:
      IF POS(+2) **IN** {`RP, VB, WP`} **THEN** *person*
      ELSEIF POS(+2) **IN** {`CD, MD, NN, NNS, RB, TO`}
        AND POS(-1) **IN** {`CC, NN, PERIOD, SYM, VB, VBD, VBZ`}
        THEN *person*
      ELSE *organisation*
    ELSE *non-NE*
  ELSEIF Gtag(1) **IN** {`cdg, govern_key, location, location+title, org_base, org_key, title`} **THEN:**
    IF POS(1) **IN** {`NNP, VBG`} **AND** POS(-1) **IN** {`DT, JJ`} **THEN** *organisation*
    ELSE *non-NE*
  ELSEIF Gtag(1) **IN** {`location+organisation, org_key+organisation, organisation+person`} **THEN:**
    IF Gtag(+1) **IN** {`cdg, organisation, person`} **THEN** *non-NE*
    ELSEIF Gtag(+1) = `NOTAG` **THEN:**
      IF POS(+1) **IN** {`CC, COMMA, DT, IN, JJ, JJR, NN, NNS, POS, SYM, TO, VB, VBD, VBZ, WP`} **THEN:**
        IF POS(4) **IN** {`CC, IN, JJ, NN, NNS, VBD, VBZ`}
          AND POS(2) **IN** {`COMMA, IN, NN, NNS, POS`}
          AND POS(-2) **IN** {`CC, CD, NN, NNS, PERIOD, PRP, VB, VBZ, WP, JJ, NNP, SYM, TO, VBD, VBN`}
          THEN *non-NE*
        ELSE *organisation*
      ELSEIF POS(+1) **IN** {`CD, MD, NNP, NNPS, PERIOD, RB, VBG, VBN, VBP`} **THEN:**
        IF POS(-1) **IN** {`DT, JJ`} **THEN** *organisation*
        ELSEIF POS(-1) **IN** {`CC, COMMA, NNP, PERIOD, PPS, SYM, TO, VBN`} **AND** POS(+2) **IN** {`NN, VBZ`} **THEN** *organisation*
        ELSE *non-NE*
      ELSE *organisation*
    ELSE *organisation*
  ELSEIF Gtag(1) = `person` **THEN:**
    IF POS(+1) **IN** {`CD, NNP, VBP`} **THEN** *non-NE*
    ELSE *person*
  ELSE *non-NE*
ELSE *non-NE*

Figure 3. A representative classifier, corresponding to a decision tree containing 58 nodes.

use of C4.5 for the adaptation of a NERC system is certainly recommended.

- Furthermore, the recognisers that C4.5 builds are rather simple and can easily be translated into a small number of comprehensible rules. Rule simplicity is enhanced with the use of a special facility provided by C4.5, which allows the automatic subsetting of feature values for many-valued features. The classification rules generated by C4.5 can be examined and refined further by human experts.

The results that we obtained were comparable to these of similar systems that use the C4.5 algorithm for NERC. At the same time our representation of the NERC problem removes the need for a post-processing stage, which is common in all systems that use decision trees. As a result, the decision tree provides directly the NERC rules.

An interesting issue for further investigation is the comparative evaluation of alternative learning methods. The first set of candidates could be learning methods that use the same feature-vector representation as C4.5, e.g. AQ15 [19] and CN2 [20]. Alternative methods could be those performing grammar induction explicitly [21,22,23]. These methods are able to construct grammars from data. This is particularly interesting for NERC, which has traditionally been performed by parsers, using grammars. Furthermore unsupervised learning methods [24,25] are worth further study as they could reduce even further the involvement of humans in the learning process. Removing the need for manual tagging of the training data, without significant loss in recognition performance would be of great use to the designer of the NERC system. A different way to reduce human effort is to automate the construction of the lexicon used in NERC. We are also pursuing work in this direction [26]. Finally, our goal is to devise a method for learning NERC systems that perform as well as the best manually constructed systems. Towards this goal we are trying to enrich our representation for the training data, using more informative features about the words that make up a named entity.

In conclusion, the results presented in this paper show that the customisation of NERC systems to specific domains can be achieved efficiently with the use of learning methods. For this reason, we consider the work presented here one step in the direction of providing NERC systems that can easily be adapted to a variety of real-world applications.

6 REFERENCES

[1] Quinlan, J. R., *C4.5: Programs for machine learning*, Morgan-Kaufmann, San Mateo, CA, 1993.

[2] Defense Advanced Research Projects Agency. Proceedings of the Sixth Message Understanding Conference (MUC-6), Morgan Kaufmann, 1995.

[3] Bikel, D.M., Miller, S., Schwartz, R., and Weischedel, R. "Nymble: a High-Performance Learning Name-finder." In *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP-97)*, Washington, D.C., pp. 194 – 201, 1997.

[4] Vilain, M., and Day, D. "Finite-state phrase parsing by rule sequences". In *Proceedings of the 16th International Conference on Computational Lingustics (COLING)*, vol. 1, pp. 274-279, 1996.

[5] Day, D., Robinson, P., Vilain, M., and Yeh, "A. Description of the ALEMBIC system as used for MUC-7." In [11], 1998.

[6] Cowie, J. "Description of the CRL/NMSU System Used for MUC-6." In [2], 1995.

[7] Bennett, S.W., Aone, C. and Lovell, C. "Learning to Tag Multilingual Texts Through Observation." In *Proceedings of the Second Conference on Empirical Methods in NLP*, pp. 109-116, 1997.

[8] Sekine, S., "NYU: Description of the Japanese NE System used for MET-2." In [11], 1998.

[9] Cuchiarelli, A., Luzi, D., and Velardi, P. "Automatic Semantic Tagging of Unknown Proper Names." In *Proceedings of COLING-98*, Montreal, 1998.

[10] Riloff, E. and Jones, R., "Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping." In *Proceedings of the National Conference on Artificial Intelligence*, pp. 474-479, 1999.

[11] Defense Advanced Research Projects Agency. Proceedings of the Seventh Message Understanding Conference (MUC-7), Morgan Kaufmann, 1998.

[12] Brill, E. "A corpus-based approach to language learning." *PhD Dissertation*, Univ. of Pennsylvania, 1993.

[13] Quinlan, J.R. "Machine Learning: Easily Understood Decision Rules." In *Computer Systems that Learn*, eds. Weiss, S.M. and Kulikowski, C.A., Morgan Kaufmann, 1991.

[14] Gallippi, A., "Recognizing Names Across Languages." In *Proceedings of the 16th International Conference on Computational Lingustics (COLING)*, 1996.

[15] Sekine, S., Grishman, R. and Shinnou, H., "A Decision Tree Method for Finding and Classifying Names in Japanese Texts." In *Proceedings of the Sixth Workshop on Very Large Corpora*, 1998.

[16] Riloff, E., "Automatically Constructing a Dictionary for Information Extraction Tasks." In *Proceedings of the National Conference on Artificial Intelligence*, pp. 811-816, 1993.

[17] Appelt, D.E., Hobbs, J.R., Bear, J., Israel, D., Kameyama, M., Kehler, A., Martin, D., Myers, K., Tyson, M. "SRI International FASTUS System MUC-6 Test Results and Analysis." In [2], 1995.

[18] Humphreys, K., Gaizauskas, R., Cunningham, H., and Azzam, S. VIE Technical Specifications. Department of Computer Science, University of Sheffield, 1997.

[19] Michalski, R. S., Mozetic, I., Hong, J. and Lavrac, N., "The multi-purpose incremental learning system AQ15 and its testing application to three medical domains." In *Proceedings of the National Conference on Artificial Intelligence*, pp. 1041-1045, 1986.

[20] Clark, P. and Niblett, T., " The CN2 algorithm." *Machine Learning*, 3(4), pp. 261-283, 1989.

[21] Langley, P. "Machine learning and Grammar induction." *Machine Learning*, v. **2**, pp. 5-8, 1987.

[22] Langley, P. and Stromsten, S. "Learning Context-Free Grammars with a Simplicity Bias." In *Proceedings of the Eleventh European Conference on Machine Learning*, Lecture Notes in Artificial Intelligence, **1810**, eds. R. L. de Mántaras and E. Plaza, pp. 220-228, 2000.

[23] Lari, K. and Young S. J. "The estimation of stochastic context-free grammars using the Inside-Outside algorithm." *Computer Speech and Language*, **4,** 1990.

[24] Mannes, C. "Self-organising grammar induction using a neural network model." In *New trends in neural computation*, Lecture Notes in Computer Science, **686**, eds. J. Mira *et al*, pp. 198-203, 1993.

[25] Kohonen, T. *Self-organisation and associative memory*. 3rd edition, Springer-Verlag, Berlin, 1989.

[26] Petasis, G., Cuchiarelli, A., Velardi, P., Paliouras, G., Karkaletsis, V. and Spyropoulos C.D. "Automatic adaptation of Proper Noun Dictionaries through cooperation of machine learning and probabilistic methods." In *Proceedings of the 23rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2000.