A RULE-BASED NAMED ENTITY RECOGNITION SYSTEM FOR SPEECH INPUT

Ji-Hwan Kim, P. C. Woodland

Cambridge University Engineering Department
Trumpington street, Cambridge, CB2 1PZ, United Kingdom
{jhk23, pcw}@eng.cam.ac.uk

ABSTRACT

In this paper, we propose a rule based (transformation based) named entity recognition system which uses the Brill rule inference approach. To measure its performance, we compare the performance of the rule-based system and IdentiFinder, one of the most successful stochastic systems. In the baseline case (no punctuation and no capitalisation), both systems show almost equal performance. They also have similar performance in the case of additional information such as punctuation, capitalisation and name lists. The performance of both systems degrade linearly with added speech recognition errors, and their rates of degradation are almost equal. These results show that automatic rule inference is a viable alternative to the HMM-based approach to named entity recognition, but it retains the advantages of a rule-based approach.

1. Introduction

Information extraction from speech is a crucial step in making the transition from speech recognition to speech understanding. Part of the information extraction problem is the recognition of named entities in the speech recogniser output. The Named Entity (NE) recognition task was defined in the 6th Message Understanding Conference (MUC) as the recognition of names of locations, persons and organisations, as well as temporal and numeric expressions [1].

NE recognition from speech recogniser output, such as automatic transcriptions of broadcast news, presents significant challenges because of the corruption of input from speech recognition errors, the disfluency in speech, and the difficulties in evaluation. Furthermore, the standard automatic transcription of speech lacks capitalisation and punctuation.

NE recognition systems are generally categorised according to whether they are stochastic (typically HMM-based) or rule-based [2]. In the stochastic method, linguistic information is captured indirectly through large tables of statistics. Because this process is fully automated, the total development time for the stochastic method is greatly reduced. However, in many instances, a stochastic system encounters difficulties in estimating probabilities from sparse training data. Compared to the stochastic method, the rule-based method encodes linguistic information directly in a set of simple rules.

The advantages of the rule-based method over the stochastic method include its smaller storage requirements, no need for less-descriptive models as in back-off, and its ready extension using expert linguistic knowledge due to its conceptually reasonable rules. However, a disadvantage of previous rule-based systems needs is that rules need to be manually constructed [2].

Manually constructed rule-based systems show reasonable performance on normal texts because many NEs have helpful capitalisation information. However, if the input is derived from speech, capitalisation information is no longer available and it is much harder to obtain the necessary linguistic information using manually constructed rules.

In Section 2, we will present a transformation-based rule-based system which generates rules automatically. Then, in Section 3, experiments and their results will be described. Finally, we conclude this paper and discuss future work in Section 4.

2. Transformation-based automatic NE rule generation

Figure 1 illustrates the procedures in our proposed transformation-based rule-based system which generates rules automatically. The procedures are mainly divided into two parts; preprocessing, and automatic rule generation. The preprocessing steps will be explained in Section 2.1. Then the automatic rule generation steps, the general idea of which originated from Brill's part-of-speech tagger [3], will be described in Section 2.2.

2.1. Preprocessing

In this system, an untagged training data file is passed through the initial NE recogniser. The system separates all punctuation marks from their adjacent words, and then treats these punctuation marks as words.

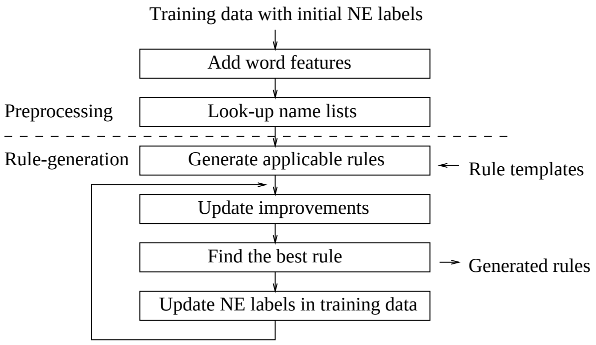

Figure 1: Procedures for preprocessing and rule-generation

Characteristics of a word, called the word features, sometimes give important clues for NE recognition [4]. For example, the capitalisation of the first character of a word, except for the first word of a sentence, gives that word a higher possibility of being a proper noun named entity word. Table 1 lists some possible word features. A deterministic computation must be able to be performed to obtain word features.

A fundamental restriction of the corpus-based approach to name finding is the relatively small number of names (of people, places, organisations etc.) observed in even a large training corpus [2]. Even with the use of an unknown word model, identification of these entities depends largely upon the presence of signalling words. An extension to this approach in our system is the use of lists of location names, first names, well-known surnames, organisations etc. The advantage of this approach is that many names can be included very quickly: an enormous corpus would be necessary in order to include the same number of names from normal text.

In our system, word features from name lists can be added as word features at the preprocessing stage. Table 2 shows word features from name lists.

Type	Descriptions
Init_Cap	Words with capitalised first character except first words of sentences
All_Cap	Words with all capitalised characters (such as BBC) and having a word length greater than 2 letters
Not_In_Ent	Words which are never used inside NEs
Ent_In_L	Words which are Not_In_Ent and which have the possibility of having an entity word on their left side
Ent_In_R	Words which are Not_In_Ent and which have the possibility of having an entity word on their right side
Numeric	Numeric words in the numeric dictionary

Table 1: Word features

Type	Description
In_P_List	Words in the persons' name list
In_L_List	Words in the locations' name list
In_O_List	Words in the organisations' name list

Table 2: Word features from name lists

2.2. Rule-generation and testing

After these preprocessing steps are completed, automatic rule-generation starts with the assignment of the named entity class of every word with a non-named entity tag. Once the training data file has been passed through the initial NE recogniser, its assigned named entity classes are compared to the true named entity classes and errors are then counted. For all words whose named entity types are different from those which they should be, applicable rules to recognise these words correctly are generated and stored, and then applied, and the resulting number of improvements on the whole training data calculated. Applicable rules are generated according to their appropriate rule templates.

Table 3 shows examples of the rule templates. Rule templates consist of pairs of characters and a subscript. w, f, t denotes that templates are related to words, word features and word classes respectively. b indicates whether the word is combined with the previous word into a single NE word (if combined, b=1 and if not, b=0). Subscripts show the relative distance from the current word; that is 0 means the current word, -1 means the previous word and 1 means the next word. Each rule template has its own applicable range where the conditions of the rule are met. Our system uses 53 rule templates. The details of these 53 rules are described in [2].

Among all possible rules at each stage, the rule which causes the greatest improvement is applied to the current training data and the training data file is updated. If there is any change in word classes which affects any of the other rules, then the improvement of that other rule is also updated. In our system, the improvement is defined as the number of words which can obtain their correct word class after the rule is applied. These steps are repeated until no changes can be made to the rules so as to reduce the number of errors between the current named entity labels for the training data and the true named entity labels.

Rule+Range		
w_0 f_0 [0 0],	w_0 f_{-1} [-1 0],	w_0 f_1 [0 1]
w_0 w_1 [0 1],	w_0 w_{-1} [-1 0],	w_0 t_1 [0 1]
w_0 t_{-1} [-1 0],	w_1 t_0 [0 1],	w_{-1} t_0 [-1 0]
t_0 t_1 [0 1],	t_0 t_{-1} [-1 0],	w_0 f_{-1} [-1 0]

Table 3: Examples of developed rule templates (w:words; f:word features; t:word classes). Subscripts define the distance from the current word and bracketed numbers indicate the range of rule application [start-offset from current word, end-offset from current word].

In testing, the rules are applied to the input text one-by-one according to a given order. If the conditions for a rule are met, then the rule is triggered and the word classes of the words are changed if necessary.

Particular importance must be given to the effect of words encountered in the test data which have not been seen in the training data. One way of improving the situation is to build separate rules for unknown words. In our system, unknown words are defined as those words which appear only once in the whole training data. If a sentence in the training data has more than one unknown word, then this sentence is duplicated and the unknown words are changed into the index for unknown words and the same rule generation procedures are then applied.

3. Experiments

Broadcast news provides a good test-bed for speech recognition and information extraction systems, because such systems are intended to handle unanticipated speakers, a large vocabulary and various domains. A set of annotated broadcast news training data is available from BBN. It consists of the entire second 100 hours of the LDC-distributed Hub-4 broadcast news training data with NE tags, and was used in these experiments as training data. For testing, we used 3 hours of data from the NIST 1998 Hub-4 broadcast news benchmark tests. Further details about the databases are given in [2].

The systems are evaluated based on how well the entities in the hypothesis file agree with those in the reference file. Two important metrics for assessing the performance of an information extraction system are recall and precision. Recall (R) refers to how much of the information that should have been extracted is correctly extracted. Precision (P) refers to the reliability of the information extracted. These are defined as:

$$P = \frac{\text{number of correct responses}}{\text{number of hypothesised responses}} \qquad (1)$$

and

$$R = \frac{\text{number of correct responses}}{\text{number of tags in reference}} \qquad (2)$$

The F-measure is the uniformly weighted harmonic mean of precision and recall:

$$F = \frac{RP}{(R+P)/2} \qquad (3)$$

The F-measure was used as the metric for assessing the performance. As a scorer, version 0.7 of the NIST HUB-4 IE scoring pipeline package was used.

3.1. Experimental results

In order to measure the performance of our system, we compared its performance against IdentiFinder, BBN's HMM-based system which gave the best performance among the five systems that participated in the 1998 Hub-4 broadcast news benchmark tests [5, 6]. Compared to the results in the benchmark tests, the results of IdentiFinder shown in this paper differs slightly, because of differences in the amount of the training data [7] and preprocessing steps for the texts. Also, there may be a difference in the version of IdentiFinder used.

The last row in Table 4 shows the performance of each system for the baseline case (no punctuation, no capitalisation and no name lists). Compared to IdentiFinder, our rule-based system showed a small improvement of 0.12 in F-measure.

Next, the effect of punctuation was measured. Punctuation has a positive effect in NE recognition and increased the performance in F-measure for the rule-based system by 0.43 and for the IdentiFinder system by 0.74. The effect of capitalisation was also measured, and was shown to be helpful for NE recognition. It contributed 1.46 in F-measure for the rule-based system and 1.54 for the IdentiFinder system. Rows 7 and 4 of Table 4 show these results. The conditions of 'Baseline+Punc' are punctuation, no capitalisation and no name lists. The conditions of 'Baseline+Cap' are capitalisation, no name lists and no punctuation.

The effect of name lists was then tested. If more than one name list's elements are overlapped, then the system prefers the longer element. If the same word appears on more than one name list, then a precedence rule is applied. The location name list has the highest priority, the person name list has the next, and the organisation name list has the lowest. The effects of name lists are shown in row 6 of Table 4. They improve the performance of the rule-based system by 1.04 in terms of F-measure and that of IdentiFinder by 1.08.

Table 4 summarises the effects of capitalisation, punctuation and name lists on performance. From this summary,

Conditions	F-measure	
	RBS	IDF
Baseline+Cap+NL+Punc	91.34	91.45
Baseline+Cap+NL	91.05	91.21
Baseline+Cap+Punc	90.86	90.87
Baseline+Cap	90.04	90.00
Baseline+NL+Punc	90.07	90.10
Baseline+NL	89.62	89.52
Baseline+Punc	89.01	89.20
Baseline	88.58	88.46

Table 4: Comparison of results (Cap: Capitalisation; NL: Name Lists; Punc: Punctuation; RBS: Rule-based system; IDF: IdentiFinder)

it is observed that the performances of both systems are very similar.

3.2. Effects of speech recognition errors

Trained patterns for NE recognition are designed to account for the variety of syntactic and semantic structures. So, patterns with several required elements are quite sensitive to errors in the input text. If any of the required elements are missing, or if an extra token intervenes between the elements, then the pattern will not match the input.

In this section, in order to examine the effects of speech recognition errors, experiments are conducted using the outputs from 11 speech recognition systems for the 1998 Hub-4 evaluation. These outputs are available from [5]. Experiments were performed with no punctuation, no capitalisation, but still using name lists. The rule-based system and IdentiFinder are trained using the human transcribed training data.

Using our rule based system and IdentiFinder, the performance have been checked on the output of the speech recognition systems for the 1998 Hub-4 evaluation. The results are plotted in Figure 2.

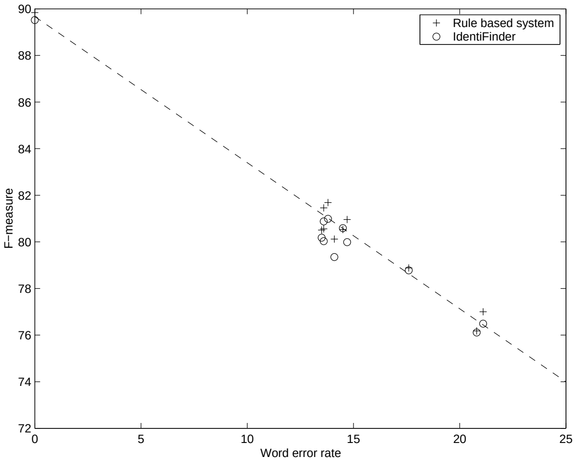

Figure 2: Effects of speech recogniser output errors. The line indicates the line-of-best-fit for the rule-based system's results

Although the points in Figure 2 are sparse, it appears that the performance degrades linearly. The line in Figure 2 is the line-of-best-fit for the results of the rule-based system, estimated by the least squares method. It appears that both systems lose about 0.62 points in F-measure per 1% of additional errors. Table 5 shows the decrease in F-measure for each percentage increase in Word Error Rate (WER). These experiments show the ability to label NE words correctly despite the fact that some words are mistranscribed. The effects of word recognition errors appear to be almost the same for both systems.

System	F-measure loss
RBS	0.627
IDF	0.622

Table 5: Decrease in F-measure for each percentage increase in WER, estimated by the least squares method (RBS: Rule-based system; IDF: IdentiFinder)

4. Conclusions and future work

In this paper, we devised a rule-based system which generates rules automatically. Then we compared its performance with BBN's commercial implementation called IdentiFinder. For the baseline case, both systems showed almost equal performance, and are also similar in the case of additional information such as punctuation, capitalisation and name lists. When input texts were corrupted by speech recognition errors, the performances of both systems were degraded by almost the same level. Although our approach is different from the stochastic method, which is recognised as one of the most successful methods, our rule-based system shows the same level of performance.

5. Acknowledgements

Ji-Hwan Kim is supported by the British Council, LG company and GCHQ. The authors wish to express their thanks to BBN for the use of the IdentiFinder software.

6. REFERENCES

1. Named Entity Task Definition. In *Proc. the Sixth Message Understanding Conference*, 1995.
2. Ji-Hwan Kim and P. C. Woodland. Rule Based Named Entity Recognition. Technical Report CUED/F-INFENG/TR.385, Cambridge University Engineering Department, 2000.
3. E. Brill. *A Corpus-Based Approach to Language Learning.* PhD thesis, University of Pennsylvania, 1993.
4. D. Bikel, S. Miller, and R. Schwartz. Nymble: a High-Performance Learning Name-finder. In *Proc. Applied Natural Language Processing*, pages 194–201, 1997.
5. 1999 NIST Hub-4 Information Extraction (Named Entity) Broadcast News Benchmark Test Evaluation. ftp://jaguar.ncsl.nist.gov/csr98/h4iene_98_official_scores_990107/index.htm.
6. M. Przybocki, J. Fiscus, J. Garofolo, and D. Pallett. 1998 Hub-4 Information Extraction Evaluation. In *Proc. DARPA Broadcast News Workshop*, pages 13–18, 1999.
7. D. Miller, R. Schwartz, R. Weischedel, and R. Stone. Named Entity Extraction from Broadcast News. In *Proc. DARPA Broadcast News Workshop*, pages 37–40, 1999.