Using Corpus-derived Name Lists for Named Entity Recognition

Mark Stevenson and Robert Gaizauskas
Department of Computer Science,
University of Sheffield
Regent Court, 211 Portobello Street,
Sheffield
S1 4DP United Kingdom
{marks, robertg}@dcs.shef.ac.uk

Abstract

This paper describes experiments to establish the performance of a named entity recognition system which builds categorized lists of names from manually annotated training data. Names in text are then identified using only these lists. This approach does not perform as well as state-of-the-art named entity recognition systems. However, we then show that by using simple filtering techniques for improving the automatically acquired lists, substantial performance benefits can be achieved, with resulting F-measure scores of 87% on a standard test set. These results provide a baseline against which the contribution of more sophisticated supervised learning techniques for NE recognition should be measured.

1 Introduction

Named entity (NE) recognition is the process of identifying and categorising names in text. Systems which have attempted the NE task have, in general, made use of lists of common names to provide clues. Name lists provide an extremely efficient way of recognising names, as the only processing required is to match the name pattern in the list against the text and no expensive advanced processing such as full text parsing is required. However, name lists are a naive method for recognising names. McDonald (1996) defines internal and external evidence in the NE task. The first is found within the name string itself, while the second is gathered from its context. For example, in the sentence "President Washington chopped the tree" the word "President" is clear external evidence that "Washington" denotes a person. In this case internal evidence from the name cannot conclusively tell us whether "Washington" is a person or a location ("Washington, DC"). A NE system based solely on lists of names makes use of only internal evidence and examples such as this demonstrate the limitations of this knowledge source.

Despite these limitations, many NE systems use extensive lists of names. Krupke and Hausman (1998) made extensive use of name lists in their system. They found that reducing their size by more than 90% had little effect on performance, conversely adding just 42 entries led to improved results. This implies that the quality of list entries is a more important factor in their effectiveness than the total number of entries. Mikheev et al. (1999) experimented with different types of lists in an NE system entered for MUC7 (MUC, 1998). They concluded that small lists of carefully selected names are as effective as more complete lists, a result consistent with Krupke and Hausman. However, both studies altered name lists within a larger NE system and it is difficult to tell whether the consistency of performance is due to the changes in lists or extra, external, evidence being used to balance against the loss of internal evidence.

In this paper a NE system which uses only the internal evidence contained in lists of names is presented. Section 3 explains how such lists can be automatically generated from annotated text. Sections 4 and 5 describe experiments in which these corpus-generated lists are applied and their performance compared against hand-crafted lists. In the next section the NE task is described in further detail.

2 NE background

2.1 NE Recognition of Broadcast News

The NE task itself was first introduced as part of the MUC6 (MUC, 1995) evaluation exercise and was continued in MUC7 (MUC, 1998). This formulation of the NE task defines seven types of NE: `PERSON`, `ORGANIZATION`, `LOCATION`, `DATE`, `TIME`, `MONEY` and `PERCENT`. Figure 1 shows a short text marked up in SGML with NEs in the MUC style.

The task was duplicated for the DARPA/NIST HUB4 evaluation exercise (Chinchor et al., 1998) but this time the corpus to be processed consisted of single case transcribed speech, rather than mixed case newswire text. Participants were asked to carry out NE recognition on North American broadcast news stories recorded from radio and television and processed by automatic speech recognition (ASR) software. The participants were provided with a training corpus consisting of around 32,000 words of transcribed broadcast news stories from 1997 annotated with NEs. Participants used these text to

```
"It's a chance to think about first-level questions," said Ms. <enamex
type="PERSON">Cohn<enamex>, a partner in the <enamex type="ORGANIZATION">McGlashan
& Sarrail<enamex> firm in <enamex type="LOCATION">San Mateo<enamex>, <enamex
type="LOCATION">Calif.<enamex>
```

Figure 1: Text with MUC-style NE's marked

develop their systems and were then provided with new, unannotated texts, consisting of transcribed broadcast news from 1998 which they were given a short time to annotate using their systems and return. Participants are not given access to the evaluation data while developing their systems.

After the evaluation, BBN, one of the participants, released a corpus of 1 million words which they had manually annotated to provide their system with more training data. Through the remainder of this paper we refer to the HUB4 training data provided by DARPA/NIST as the SHORT_TRAIN corpus and the union of this with the BBN data as the LONG_TRAIN corpus. The data used for the 1998 HUB4 evaluation was kept blind, we did not examine the text themselves, and shall be referred to as the TEST corpus.

The systems were evaluated in terms of the complementary precision (P) and recall (R) metrics. Briefly, precision is the proportion of names proposed by a system which are true names while recall is the proportion of the true names which are actually identified. These metrics are often combined using a weighted harmonic called the F-measure (F) calculated according to formula 1 where β is a weighting constant often set to 1. A full explanation of these metrics is provided by van Rijsbergen (1979).

$$F = \frac{(\beta + 1) \times P \times R}{(\beta \times P) + R} \quad (1)$$

The best performing system in the MUC7 exercise was produced by the Language Technology Group of Edinburgh University (Mikheev et al., 1999). This achieved an F-measure of 93.39% (broken down as a precision of 95% and 92% recall). In HUB4 BBN (Miller et al., 1999) produced the best scoring system which achieved an F-measure of 90.56% (precision 91%, recall 90%) on the manually transcribed test data.

2.2 A Full NE system

The NE system used in this paper is based on Sheffield's LaSIE system (Wakao et al., 1996), versions of which have participated in MUC and HUB4 evaluation exercises (Renals et al., 1999). The system identifies names using a process consisting of four main modules:

List Lookup This module consults several lists of likely names and name cues, marking each occurrence in the input text. The name lists include lists of organisations, locations and person first names and the name cue lists of titles (eg. "Mister", "Lord"), which are likely to precede person names, and company designators (eg. "Limited" or "Incorporated"), which are likely to follow company names.

Part of speech tagger The text is the part of speech tagged using the Brill tagger (Brill, 1992). This tags some tokens as "proper name" but does not attempt to assign them to a NE class (eg. PERSON, LOCATION).

Name parsing Next the text is parsed using a collection of specialised NE grammars. The grammar rules identify sequences of part of speech tags as added by the **List Lookup** and **Part of speech tagger** modules. For example, there is a rule which says that a phrase consisting of a person first name followed by a word part of speech tagged as a proper noun is a person name.

Namematching The names identified so far in the text are compared against all unidentified sequences of proper nouns produced by the part of speech tagger. Such sequences form candidate NEs and a set of heuristics is used to determine whether any such candidate names match any of those already identified. For example one such heuristics says that if a person is identified with a title (eg. "President Clinton") then any occurrences without the title are also likely to be person names (so "Clinton" on it own would also be tagged as a person name).

For the experiments described in this paper a restricted version of the system which used only the **List Lookup** module was constructed. The list lookup mechanism marks all words contained in any of the name lists and each is proposed as a NE. Any string occurring in more than one list is assigned the category form the first list in which it was found, although this did not occur in any of the sets of lists used in the experiments described here.

3 List Generation

The **List Lookup** module uses a set of handcrafted lists originally created for the MUC6 evaluation. They consisted of lists of names from the gazetteers provided for that competition, supplemented by manually added entries. These lists

evolved for the MUC7 competition with new entries and lists being added. For HUB4 we used a selection of these lists, again manually supplementing them where necessary. These lists included lists of companies, organisations (such as government departments), countries and continents, cities, regions (such as US states) and person first names as well as company designators and person titles. We speculate that this ad hoc, evolutionary, approach to creating name lists is quite common amongst systems which perform the NE task.

In order to compare this approach against a simple system which gathers together all the names occurring in NE annotated training text, a program was implemented to analyse text annotated in the MUC SGML style (see Figure 1) and create lists for each NE type found. For example, given the NE `<enamex type="LOCATION">SAN MATEO<enamex>` an entry SAN MATEO would be added a list of locations.

This simple approach is certainly acceptable for the LOCATION, ORGANIZATION and, to a more limited extent, PERSON classes. It is less applicable to the remaining classes of names (DATE, TIME, MONEY and PERCENT) because these are most easily recognised by their grammatical structure. For example, there is a rule in the NE grammar which says a number followed by a currency unit is as instance of the MONEY name class – eg. FIFTY THREE DOLLARS, FIVE MILLION ECU. According to Przbocki et al. (1999) 88% of names occurring in broadcast news text fall into one of the LOCATION, ORGANIZATION and PERSON categories.

Two sets of lists were derived, one from the SHORT_TRAIN corpus and a second from the LONG_TRAIN texts. The lengths of the lists produced are shown in Table 1.

	Corpus	
Category	SHORT_TRAIN	LONG_TRAIN
ORGANIZATION	245	2,157
PERSON	252	3,947
LOCATION	230	1,489

Table 1: Lengths of lists derived from SHORT_TRAIN and LONG_TRAIN corpora

4 List Application

The SHORT_TRAIN and LONG_TRAIN lists were each applied in two ways, alone and appended to the original, manually-created, lists. In addition, we computed the performance obtained using only the original lists for comparison. Although both sets of lists were derived using the SHORT_TRAIN data (since the LONG_TRAIN corpus includes SHORT_TRAIN), we still compute the performance of the SHORT_TRAIN lists on that corpus since this provides some insight into the best possible performance which can be expected from NE recognition using a simple list lookup mechanism. No scores were computed for the LONG_TRAIN lists against the SHORT_TRAIN corpus since this is unlikely to provide more information.

Table 2 shows the results obtained when the SHORT_TRAIN lists were applied to that corpus. This first experiment was designed to determine how well the list lookup approach would perform given lists compiled directly from the corpus to which they are being applied. Only PERSON, LOCATION and ORGANIZATION name classes are considered since they form the majority of names occurring in the HUB4 text. As was mentioned previously, the remaining categories of name are more easily recognised using the NE parser. For each configuration of lists the precision, recall and F-measure are calculated for the each name class both individually and together.

We can see that the original lists performed reasonably well, scoring an F-measure of 79% overall. However, the corpus-based lists performed far better achieving high precision and perfect recall. We would expect the system to recognise every name in the text, since they are all in the lists, but perfect precision is unlikely as this would require that no word appeared as both a name and non-name or in more than one name class. Even bearing this in mind the calculated precision for the ORGANIZATION class of names is quite low. Analysis of the output showed that several words occurred as names a few times in the text but also as non-names more frequently. For example, "police" appeared 35 times but only once as an organisation; similarly "finance" and "republican" occur frequently but only as a name a few times. In fact, these three list entries account for 61 spuriously generated names, from a total of 86 for the ORGANIZATION class. The original lists do not include words which are likely to generate spurious entries and names like "police" would only be recognised when there was further evidence.

The SHORT_TRAIN lists contain all the names occurring in that text. When these lists are combined with the original system lists the observed recall remains 100% while the precision drops. The original system lists introduce more spurious entries, leading to a drop of 3% F-measure.

The results of applying the corpus-derived lists to the texts from which they were obtained show that, even under these circumstances, perfect results cannot be obtained. Table 3 shows a more meaningful evaluation; the SHORT_TRAIN lists are applied to the TEST corpus, an unseen text. The original system lists achieve an F-measure of 83% on this text and the corpus-derived lists perform 8% worse. However, the configuration of lists which performs best is the union of the original lists with those derived from the

Lists	Original			SHORT_TRAIN			Combination		
Name Type	P	R	F	P	R	F	P	R	F
ALL	86	73	79	94	100	97	88	100	94
ORGANIZATION	84	49	62	83	100	90	79	100	88
PERSON	78	71	74	99	100	99	88	100	94
LOCATION	92	88	90	98	100	99	95	100	97

Table 2: SHORT_TRAIN lists applied to SHORT_TRAIN corpus

corpus. This out-performs each set of lists taken in isolation both overall and for each name category individually. This is clear evidence that the lists used by the system described could be improved with the addition of lists derived from annotated text.

It is worth commenting on some of the results for individual classes of names in this experiment. We can see that the performance for the ORGANIZATION class actually increases when the corpus-based lists are used. This is partially because names which are made up from initials (eg. "C.N.N." and "B.B.C.") are not generally recognised by the list lookup mechanism in our system, but are captured by the parser and so were not included in the original lists. However, it is also likely that the organisation list is lacking, at least to some level. More interestingly, there is a very noticeable drop in the performance for the PERSON class. The SHORT_TRAIN lists achieved an F-measure of 99% on that text but only 48% on the TEST text. In Section 2.1 we mentioned that the HUB4 training data consists of news stories from 1997, while the test data contains stories from 1998. We therefore suggest that the decrease in performance for the PERSON category demonstrates a general property of broadcast news: many person names mentioned are specific to a particular time period (eg. "Monica Lewinksi" and "Rodney King"). In contrast, the locations and organisations mentioned are more stable over time.

Table 4 shows the performance obtained when the lists derived from LONG_TRAIN were applied to the TEST corpus. The corpus-derived lists perform significantly worse than the original system lists, showing a large drop in precision. This is to be expected since the lists derived from LONG_TRAIN contain all the names occurring in a large body of text and therefore contain many words and phrases which are not names in this text, but spuriously match non-names. Although the F-measure result is worse than when the SHORT_TRAIN lists were used, the recall is higher showing that a higher proportion of the true names can be found by analysing a larger body of text. Combining the original and corpus-derived lists leads to a 1% improvement. Recall is noticeably improved compared with the original lists, however precision is lowered and this shows that the corpus-derived lists introduce a large number of spurious names.

From this first set of experiments it can be seen that perfect results will not be obtained even using lists contain all and only the names in a particular text, thus demonstrating the limitations of this naive approach to named entity recognition. We have also demonstrated that it is possible for the addition of corpus-derived lists to improve the performance of a NE recognition system based on gazetteers. However, this is not guaranteed and it appears that adding too many names without any restriction may actually lead to poorer results, as happened when the LONG_TRAIN lists were applied.

5 Filtering Lists

The results from our first set of experiments led us to question whether it is possible to restrict the entries being added to the lists in order to avoid those likely to generate spurious names. We now go on to describe some methods which can be used to identify and remove list entries which may generate spurious names.

Method 1: Dictionary Filtering The derived lists can be improved by removing items in the list which also occur as entries in a dictionary.

We began by taking the *Longman Dictionary of Contemporary English* (LDOCE)(Procter, 1978) and extracting a list of words it contained including all derived forms, for example pluralisation of nouns and different verb forms. This produced a list of 52,576 tokens which could be used to filter name lists.

Method 2: Probability Filtering The lists can be improved by removing names which occur more frequently in the corpus as non-names than names.

Another method for filtering lists was implemented, this time using the relative frequencies of phrases occurring as names and non-names. We can extract the probability that a phrase occurs as a name in the training corpus by dividing the number of times it occurs as a name by the total number of corpus occurrences. If this probability estimate is an accurate reflection of the name's behaviour in a

Lists	Original			SHORT_TRAIN			Combination		
Name Type	P	R	F	P	R	F	P	R	F
ALL	86	79	83	90	65	75	83	86	84
ORGANIZATION	82	57	67	76	66	71	79	81	80
PERSON	77	80	78	93	32	48	79	83	81
LOCATION	93	89	91	97	81	88	92	94	93

Table 3: SHORT_TRAIN lists applied to TEST corpus

Lists	Original			LONG_TRAIN			Combination		
Name Type	P	R	F	P	R	F	P	R	F
ALL	86	79	83	64	86	73	62	91	74
ORGANIZATION	82	57	67	44	85	58	43	88	58
PERSON	77	80	78	55	75	63	53	86	66
LOCATION	93	89	91	87	92	89	84	94	89

Table 4: LONG_TRAIN lists applied to TEST corpus

new text we can use it to estimate the accuracy of adding that name to the list. Adding a name to a list will lead to a recall score of 1 for that name and a precision of Pr (where Pr is the probability value estimated from the training corpus) which implies an F-measure of $\frac{2Pr}{1+Pr}$.[1] Therefore the probabilities can be used to filter out candidate list items which imply low F-measure scores. We chose names whose corpus probabilities produced an F-measure lower than the overall score for the list. The LONG_TRAIN lists scored an F-measure of 73% on the unseen, TEST, data (see Table 4). Hence a filtering probability of 73% was used for these lists, with the corpus statistics gathered from LONG_TRAIN.

Method 3: Combining Filters These filtering strategies can be improved by combining them.

We also combined these two filtering strategies in two ways. Firstly, all names which appeared in the lexicon *or* whose corpus probability is below the filtering probability are removed from the lists. This is dubbed the "or combination". The second combination strategy removes any names which appear in the lexicon *and* occur with a corpus frequency below the filtering probability are removed. This second strategy is called the "and combination".

These filtering strategies were applied to the LONG_TRAIN lists. The lengths of the lists produced are shown in Table 5.

The strategies were evaluated by applying the filtered LONG_TRAIN lists to the TEST corpus, the results of which are shown in Table 6. There is an improvement in performance of 4% F-measure when lists filtered using the "and" combination are used compared to the original, hand-crafted, lists. Although this approach removes only 108 items from all the lists there is a 14% F-measure improvement over the un-filtered lists. Each filtering strategy used individually demonstrates a lower level of improvement: the dictionary filtered lists 12% and the probability filtered 10%.

The "and" combination is more successful because filtering lists using the dictionary alone removes many names we would like to keep (eg. country names are listed in LDOCE) but many of these are retained since both filters must agree. These experiments demonstrate that appropriately filtered corpus-derived lists can be more effective for NE recognition than hand-crafted lists. The difference between the observed performance of our simple method and those reported for the best-performing HUB4 system is perhaps lower that one may expect. The BBN system achieved 90.56% overall, and about 92% when only the PERSON, LOCATION and ORGANIZATION name classes are considered, 5% more than the method reported here. This difference is perhaps lower than we might expect given that name lists use only internal evidence (in the sense of Section 1). This indicates that simple application of the information contained in manually annotated NE training data can contribute massively to the overall performance of a system. They also provide a baseline against which the contribution of more sophisticated supervised learning techniques for NE recognition should be measured.

[1] Analysis of the behaviour of the function $f(Pr) = \frac{2Pr}{1+Pr}$ shows that it does not deviate too far from the value of Pr (ie. $f(Pr) \approx Pr$) and so there is an argument for simply filtering the lists using the raw probabilities.

NE Category	Un-Filtered List	Dictionary Filtered	Probability Filtered	Or Combined	And Combined
ORGANIZATION	2,157	1,978	2,000	1,964	2,049
PERSON	3,947	3,769	3,235	3,522	3,809
LOCATION	1,489	1,412	1,364	1,382	1,449

Table 5: Lengths of corpus-derived lists

	Original Lists			Un-Filtered Lists			Dictionary Filtered			Probability Filtered			Or Combination			And Combination		
Name Type	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
ALL	86	79	83	64	86	73	95	79	85	96	73	83	95	73	83	93	81	87
ORGANIZATION	82	57	67	44	85	58	86	72	78	85	74	79	84	60	70	84	76	80
PERSON	77	80	78	55	75	63	96	66	78	96	40	56	100	49	66	94	66	78
LOCATION	93	89	91	87	92	89	98	89	93	97	90	93	98	90	94	97	92	94

Table 6: Filtered and un-filtered LONG_TRAIN lists applied to TEST corpus

6 Conclusion

This paper explored the role of lists of names in NE recognition, comparing hand-crafted and corpus-derived lists. It was shown that, under certain conditions, corpus-derived lists outperform hand-crafted ones. Also, supplementing hand-crafted lists with corpus-based ones often improves their performance. The reported method was more effective for the ORGANIZATION and LOCATION classes of names than for PERSON, which was attributed to the fact that reportage of these names does not change as much over time in broadcast news.

The method reported here achieves 87% F-measure, 5% less than the best performing system in the HUB4 evaluation. However, it should be remembered that this technique uses only a simple application of internal evidence.

References

E. Brill. 1992. A simple rule-based part of speech tagger. In *Proceeding of the Third Conference on Applied Natural Language Processing (ANLP-92)*, pages 152–155, Trento, Italy.

N. Chinchor, P. Robinson, and E. Brown. 1998. Hub-4 named entity task definition (version 4.8). Technical report, SAIC. http://www.nist.gov/speech/hub4_98.

G. Krupke and K. Hausman. 1998. Isoquest Inc: description of the NetOwl(TM) extractor system as used for MUC-7. In *Message Understanding Conference Proceedings: MUC 7.* Available from http://www.muc.saic.com.

D. McDonald. 1996. Internal and external evidence in the identification and semantic categorization of proper names. In B. Boguraev and J. Pustejovsky, editors, *Corpus Processing for Lexical Aquisition*, chapter 2, pages 21–39. MIT Press, Cambridge, MA.

A. Mikheev, M. Moens, and C. Grover. 1999. Named entity recognition without gazeteers. In *Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–8, Bergen, Norway.

D. Miller, R. Schwartz, R. Weischedel, and R. Stone. 1999. Named entity extraction from broadcast news. In *Proceedings of the DARPA Broadcast News Workshop*, pages 37–40, Herndon, Virginia.

MUC. 1995. *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, San Mateo, CA. Morgan Kaufmann.

1998. Message Understanding Conference Proceedings: MUC7. http://www.muc.saic.com.

P. Procter, editor. 1978. *Longman Dictionary of Contemporary English.* Longman Group, Essex, UK.

M. Przbocki, J. Fiscus, J. Garofolo, and D. Pallett. 1999. 1998 HUB4 Information Extraction Evaluation. In *Proceedings of the DARPA Broadcast News Workshop*, pages 13–18, Herndon, Virginia.

S. Renals, Y. Gotoh, R. Gaizausaks, and M. Stevenson. 1999. Baseline IE-NE Experimants Using the SPRACH/LASIE System. In *Proceedings of the DAPRA Broadcast News Workshop*, pages 47–50, Herndon, Virginia.

C. van Rijsbergen. 1979. *Information Retrieval.* Butterworths, London.

T. Wakao, R. Gaizauskas, and K. Humphreys. 1996. Evaluation of an algorithm for the recognition and classification of proper names. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*, pages 418–423, Copenhagen, Denmark.