Statistical Named Entity Recognizer Adaptation

John D. Burger and **John C. Henderson** and **William T. Morgan**
The MITRE Corporation
MS K309
202 Burlington Road
Bedford, MA 01730-1420
{john,jhndrsn,wmorgan}@mitre.org

1 Introduction

Named entity recognition (NER) is a subtask of widely-recognized utility of information extraction (IE). NER has been explored in depth to provide rapid characterization of newswire data (Sundheim, 1995; Palmer and Day, 1997). The NER task involves both identification of spans of text referring to named entities, and categorization of these entities into classes based on the role they fill *in context*. The sentence "Washington announced that Washington ate seven hotdogs in Washington" provides an example in which a single name can arguably refer to three different entities: an organization, a person, and a location.

Following the paradigm introduced by Ramshaw and Marcus (1999), many researchers reduce the NER problem to a word-tagging problem, and address it with techniques similar to those used for part of speech tagging (Meteer et al., 1991; Brill, 1995). Borthwick explores the maximum entropy approach in his dissertation (1999). Collins and Singer (1999) investigate semi-unsupervised methods for named entity categorization. Cucerzan and Yarowsky (1999) produce a unified technique for producing NER systems for several languages, utilizing extensive bootstrapping from small amounts of supervised data with an EM-style algorithm. Miller et al. (2000) produce a statistical Hidden Markov Model (HMM) for NER which is similar to the one used by Palmer et al. (1999); the latter system, named `phrag`, is the NER engine utilized in the work described in this paper.

The experiments described herein explore unsupervised approaches to NER, with an eye toward using unannotated corpora consisting of a few hundred million words. Recent word sense disambiguation results suggest that some simple techniques can scale well with increased data sizes (Banko and Brill, 2001). This paper presents several experiments in adapting a HMM-based named entity recognizer to a target data set. Our core learning engine is a word-based HMM, and we show two techniques, informed smoothing and iterative adaptation, for incorporating unsupervised data into the model, which provide overall gains in performance.

1.1 `phrag`

`phrag`[1] is a trainable phrase tagger based on HMMs. `phrag` uses bigram language models, i.e., state emissions are conditioned on the previous word only. State transitions are similarly conditioned, which allows the model to capture context words, such as "Mr". All models are smoothed with type c Witten-Bell discounting (Bell et al., 1990). For named entity recognition, the typical HMM topology has two states per phrase type. The first word of each phrase is generated by the first state, and any subsequent words are generated by the second state. This is essentially the *BII* scheme employed by many chunkers, e.g. (Tjong Kim Sang, 2002). Figure 1 presents a sketch of this topology.

The lexicon is constructed from the training corpus, excluding the least frequent words. These words are pooled to form unknown word models, e.g., `unknown-number`, `unknown-punctuation`, `unknown-alphabetic`. `phrag` can also use auxiliary resources such as word lists to form additional equivalence classes. These are consulted, in order of preference, when a word is not in the main lexicon. If the word is not in any word list, it is relegated to the appropriate "truly unknown" class described above. Language model statistics for

[1] `http://www.openchannelsoftware.org/projects/Qanda/`.

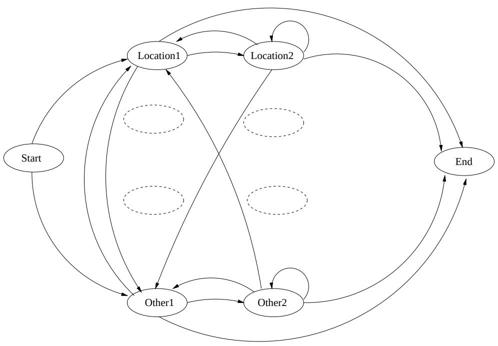

Figure 1: HMM topology in `phrag`

these equivalence classes are informed by the pooled words from the training data, then consulted at tagging time.

`phrag` has been used as a named entity finder, a POS tagger, and a syntactic chunker. It is an open source package and has previously been used to process speech recognition output and newswire in English, Mandarin, and Arabic.

2 Approaches

Single-iteration adaptation was the idea behind two techniques we investigated: Viterbi training (Viterbi, 1967) and selecting word classes to better adapt the model to the test data. The third technique was to add external resources.

2.1 Viterbi Training

The *words* in the test set are fair game for use in adaptation. The named-entity and part-of-speech tags were stripped from the data and experiments were conducted utilizing the unannotated text to augment the HMM. An iteration of Viterbi training with the development test set was attempted, but the model did not change enough to make a difference in relabeling the test data. However, one model that was Viterbi-trained on a pool of extra "found" data concatenated to the development training data gave an improvement on the test sets.

2.2 Target-informed Smoothing

As mentioned in section 1.1, the smoothing technique in `phrag` allows several classes of unknown words, and these classes can be introduced prior to training the HMM. Each word in a class is treated by `phrag` as if it were the *same* word, e.g. every word in the class *English surnames* is treated as the word "unknown-english-surname". Smoothing is done with many classes of unknown words, but the classes are prioritized such that each previously unseen word in the test data[2] falls into one unique class. Test set named entities are often found only in these word lists and not in the training data.

As an alternative for introducing adaptation information from the test data, word lists were created by running the baseline system on the untagged test data. One word list was created for each target entity type in the resulting system output, and, aside from removing words containing non-alphabetic characters, no provision was made to avoid duplicates. We termed his technique *adaptation*.[3] When including the hapax legomena from the training data in these lists as well, we call the system a *bridge*: it bridges the statistical gap created by zerotons by clustering the zerotons with words from the training data.

`phrag` was also used to train a character bigram spelling model which recognizes sequences of characters as persons, organizations, locations, miscellanea, or non-named-entity words. While the spelling model was too impoverished to use as a NER system ($F \approx 26$ on the Dutch development set), it was used to further subdivide the bridging word lists. For each named-entity type, two lists were created. The first list contained all of the words for which the spelling model and the baseline system agreed, and the second list contained the rest of the words that had been placed into that list by the baseline `phrag` system. A system of this type is a *pair bridge* if the spelling model is trained according to the distribution of tokens in the dataset. When the spelling model is trained using the set of word types in the dataset, without regard to frequency of the type, we call this system a *type pair bridge*. These were the only two models to incorporate sub-word features.

2.3 Additional data

Two types of additional data were used. Table 1 shows the set of word lists used in the Spanish and Dutch systems. These lists were each added as unknown word classes into `phrag`.

The second type of data consisted of 100 million words from TREC Spanish and 400 million

[2] These are often referred to as *zerotons*.

[3] After *speaker adaptation* from speech recognition.

S	D	Size	Description
√	√	120146	English location words from the TIPSTER gazetteer
√	√	88799	English surnames (U.S. Census Bureau, 1995)
√	√	17576	All three-letter acronyms, AAA through ZZZ
√	√	5163	English given names (U.S. Census Bureau, 1995)
	√	1410	Dutch surnames (Dupon, 2000)
	√	1162	Dutch given names (Dupon, 2000)
	√	1410	Dutch province and city names (Kuyper, 1865)
√		639	Spanish surnames (Word and Perkins, 1996)
√		362	Spanish names of capitals (prominent global cities)
√		203	Spanish geographic adjectives, e.g. "norteamericano"
√		138	Spanish country words e.g. "Estatos" and "Unidos"

Table 1: External word lists introduced to **S**panish and **D**utch *wdlist* systems.

words of Spanish Newswire, both distributed by the LDC,[4] and 2.3 million words of Dutch text, harvested from the Dutch news site *Planet Internet*.[5] For the experiments below, a randomly drawn subsample of 200,000 words of each language was used.

3 Results

Table 2 presents the results of our experiments. The left column gives an experiment label, and the subsequent columns indicate the overall F-measure as given by the CoNLL scoring software for the Spanish and Dutch development and test sets. The rows are sorted by performance on the Spanish development set.

	Spanish		Dutch	
System	F_{dev}	F_{test}	F_{dev}	F_{test}
phrag*	69.13	74.01	66.60	71.23
rand	68.39	73.08	63.75	67.45
rviterbi	69.43	73.61	67.57	70.53
wdlst	70.49	74.37	70.20	72.60
adapt*	70.92	75.13	66.68	71.73
bridge*	71.69	75.51	70.25	**73.51**
pbridge*	72.00	**75.77**	**70.61**	72.57
tpbridge*	**72.25**	**75.78**	69.63	72.86

Table 2: Summary of experiment results. (*indicates a system built using only development data, i.e. excluding external resources.)

The first line, labeled *phrag*, gives the performance of the baseline system using standard maximum likelihood training.

In subsequent lines, we see how several types of additional data and adaptation techniques improve system performance. The alterations from the base system are *not* cumulative, unless where obvious or indicated. Choosing the best combination of experimental systems is left as a mechanical exercise.

rand corresponds to adding words drawn from a baseline-tagged randomly-selected 1-million word subset of the large corpus to the adaptation word lists. While this is a negative result, performing one iteration of Viterbi training on that randomly drawn set improved over the baseline for the two development sets (as shown in the line labeled *rviterbi*).

wdlist corresponds to incorporating the full set of word lists described in Table 1, and the experiment labeled *adapt* corresponds to producing word lists (smoothing classes as described in section 2.2) from phrases in the test data that were recognized by the baseline system.

In *bridge*, smoothing word lists were created from both the training and baseline-tagged test data. In the pair bridge system, *pbridge*, priority was given to lists of words that were agreed upon by both the bigram spelling model and the baseline system. The type pair bridge system, *tpbridge*, was the same as the pair bridge, except the spelling model was built on the word types, disregarding the distribution of the words in the data.

Table 3 gives a breakdown of best system performance (those entries bolded in Table 2) by named-entity type.

[4] LDC catalog numbers LDC2000T51, LDC95T9, and LDC99T41.

[5] `http://www.planet.nl/`.

4 Concluding Remarks

These results show that the simple HMM adaptation technique *bridging* can give more gain than incorporating found word lists or performing Viterbi training on the test set.

References

M. Banko and E. Brill. 2001. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 26–33, Toulouse, France.

T. C. Bell, J. G. Cleary, and I. H. Witten. 1990. *Text Compression*. Prentice Hally.

A. Borthwick. 1999. *A Maximum Entropy Approach to Named Entity Recognition*. Ph.D. thesis, New York University.

E. Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, 21(4):543–565.

M. Collins and Y. Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of the 1999 Joint SIGDAT Conference on EMNLP and VLC*, University of Maryland, MD.

S. Cucerzan and D. Yarowsky. 1999. Language independent named entity recognition combining morphological and contextual evidence. In *Proceedings of 1999 Joint SIGDAT Conference on EMNLP and VLC*.

J. Dupon. 2000. Dutch genealogy. `http://www.geocities.com/jwdupon/Dutch.html`.

J. Kuyper. 1865. *Gemeente Atlas van Nederland*. Hugo Suringar Leeuwarden. `http://dutchgenealogy.com/Dutch_Maps/index.html`.

M. Meteer, R. Schwartz, and R. Weischedel. 1991. Empirical studies in part of speech labelling. In *Proceedings of the DARPA Speech and Natural Language Workshop*. Morgan Kaufmann.

D. Miller, S. Boisen, R. Schwartz, R. Stone, and R. Weischedel. 2000. Named entity extraction from noisy input: speech and OCR. In *Proceedings of Conference for Applied Natural Language Processing*, pages 316–324, Seattle, WA.

D. D. Palmer and D. S. Day. 1997. A statistical profile of the named entity task. In *Proceedings of Fifth ACL Conference for Applied Natural Language Processing*, Washington D.C.

D. D. Palmer, J. D. Burger, and M. Ostendorf. 1999. Information extraction from broadcast news speech data. In *Proceedings of the DARPA Broadcast News Workshop*, pages 41–46.

L. Ramshaw and M. Marcus, 1999. *Natural Language Processing Using Very Large Corpora*, chapter Text Chunking Using Transformation-based Learning. Kluwer.

B. Sundheim. 1995. Overview of results of the MUC-6 evaluation. In *Proceedings of the Sixth Message Understand Conference*, November.

E. F. Tjong Kim Sang. 2002. Memory-based shallow parsing. *Journal of Machine Learning Research*, 2(559–594), March.

U.S. Census Bureau. 1995. Frequently occurring first names and surnames from the 1990 census. `http://www.census.gov/genealogy/www/freqnames.html`.

A. Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, IT-63.

D. L. Word and R. C. Perkins, Jr. 1996. Building a spanish surname list for the 1990's—a new approach to an old problem. Technical Working Paper 13, U.S. Census Bureau. `http://www.census.gov/genealogy/www/spanname.html`.

Spanish dev.	precision	recall	$F_{\beta=1}$
LOC	62.46%	81.42%	70.69
MISC	48.07%	47.64%	47.86
ORG	75.88%	72.53%	74.17
PER	86.75%	75.04%	80.47
overall	71.79%	72.70%	72.25

Spanish test	precision	recall	$F_{\beta=1}$
LOC	75.52%	77.40%	76.45
MISC	50.74%	50.29%	50.52
ORG	74.63%	79.43%	76.96
PER	81.60%	86.26%	83.86
overall	74.19%	77.44%	75.78

Dutch dev.	precision	recall	$F_{\beta=1}$
LOC	77.73%	77.73%	77.73
MISC	68.30%	63.54%	65.83
ORG	78.29%	62.15%	69.29
PER	66.83%	77.40%	71.73
overall	71.73%	69.53%	70.61

Dutch test	precision	recall	$F_{\beta=1}$
LOC	81.08%	76.13%	78.53
MISC	67.87%	63.35%	65.53
ORG	73.27%	66.25%	69.58
PER	71.79%	84.60%	77.67
overall	72.69%	72.45%	72.57

Table 3: Best results obtained for the development and the test data sets for the two languages used in this shared task.