

Orthographic Tries in Language Independent Named Entity Recognition

Casey Whitelaw and Jon Patrick

Language Technology Research Group

Capital Markets Co-operative Research Centre

University of Sydney

{casey, jonpat}@it.usyd.edu.au

Abstract

Orthographic tries are an efficient way of storing information for strings with shared prefixes. By storing probabilistic occurrence data, orthographic tries have been shown to be well-suited to the task of named entity recognition. In this paper, we examine the use of tries in language-independent named entity recognition. We show that the performance of tries is greatly affected by trie depth and the weighting functions used, and that superior results can be obtained through the application of machine learning techniques. Also, by selective construction we make very deep partial tries with good performance and low cost. We explore the relative merits of word-internal and contextual features, and combine them to increase overall performance.

1 Introduction

Named entity (NE) recognition is an important preliminary task in many areas of language technology, including information extraction and machine translation. There has been a move away from hand-coded systems towards machine learning systems, the main advantages being in reduced development and maintenance costs, and the ease in which a system can be moved between languages. A variety of machine learning techniques have been applied to the problem, including supervised approaches such

as Hidden Markov models (Bikel et al., 1997) and Maximum Entropy theory (Borthwick et al., 1998), as well as systems learning from seed lists and unannotated data (Buchholz and van den Bosch, 2000).

While there has been much work on language-specific named entity recognition, the problem of developing a language-independent system has recently gained prominence. Language independence forces a move away from a specific language model towards a more general meta-linguistic model, which is tailored to each target language. To achieve portability and robustness in a language-independent strategy, it is desirable to make as few assumptions as possible but instead allow these differences to be exposed through machine learning.

The task generally referred to as “named entity recognition” can be divided into separate sub-tasks: the first, named entity *recognition*, is the identification of named entity phrases. Once recognition has occurred, the named entity *classification* stage determines the type of each NE phrase - person, location, organisation, or other categories as required. In this paper, we focus on the recognition process, but the techniques are equally applicable to classification.

In Sections 2 and 3, we state our aim of language independence, and briefly introduce the datasets used. In Sections 4 and 5, we expand on the notion of using weighting functions in trie-based classifications, and show that genetic algorithms can be used to obtain high quality language-specific weighting functions. In Section 6, we propose a new method for constructing tries that achieve the performance of very

deep tries with smaller resource costs. To capture both word-internal and contextual data we combine multiple tries, and in Section 7 show that their combination can be significantly more effective. We examine the effect that the size of the training corpora has on performance, and perform an analysis of the strengths and weaknesses of a trie-based approach.

2 Language Independence

When a system is designed to be language independent, it is important to make as few linguistic assumptions as possible. Despite the move towards machine learning approaches, many NER systems are still built around an assumed language model. These assumptions can be made in forms such as rules (Zhou and Su, 2002), rule-ordering, and feature extraction (Collins and Singer, 1999).

As will be seen, our trie-based system is well suited to language independent named entity recognition. The only assumption made is that the word boundaries have been identified. This brings not only language-independence, but problem-independence, so our approach is easily applicable to other contextual word-classification tasks.

External lists of known NEs, also known as gazetteers (Mikheev et al., 1999), are commonly used in NER systems, either explicitly or as the basis for rules such as “common business suffixes” (Chieu and Ng, 2002). While such lists can be useful, particularly in a single language and domain, the associated maintenance costs can be high. Through avoiding the use of gazetteers, orthographic tries remain highly language-independent.

3 The Data

To test the language-independence of a trie-based approach, datasets from Dutch and Spanish were used. These datasets were made available as part of the shared task of CoNLL 2002 (Tjong Kim Sang, 2002), and contain tokenised text divided into the categories of non-entity (O), person (PER), location (LOC), organisation (ORG), and miscellaneous entity (MISC).

An entity consists of a headword (eg. B-PER) and zero or more continuation words (eg. I-PER). Both datasets consist of a number of newspaper articles, totalling over 250,000 and 200,000 tokens for Spanish and Dutch respectively. The Dutch dataset also contained part-of-speech tags, which were not used in this test.

When testing named entity recognition, all entity categories were combined to form B-ENT and I-ENT categories. The ENT category that appears in results in this paper is the combination of the B-ENT and I-ENT categories to form a single “entity” category. Unless noted, figures for precision and recall are calculated on a token-by-token basis, as opposed to the phrase-by-phrase basis used in the CoNLL shared task.

4 Orthographic Tries and Trie Weighting Functions

Tries are an efficient data structure for capturing statistical differences between categories. In an orthographic trie, a path from the root through n nodes represents a string $a_1a_2\dots a_n$. The n -th node in the path stores the occurrences (frequency) of the string $a_1a_2\dots a_n$ in each category. These frequencies can be normalised to give relative probabilities $P(c | a_1a_2\dots a_n)$ for each category c . Our method of using tries has much in common with the concept of memory-based learning (Daelemans et al., 1999).

Given a string $a_1a_2\dots a_n$ and a category c an orthographic trie yields a set of relative probabilities $P(c | a_1)$, $P(c | a_1a_2)$, ..., $P(c | a_1a_2\dots a_n)$. The probability that a string indicates a particular class is computed along the whole trie path, which helps to smooth scores for rare strings. A general weighting function in which all probabilities are considered independently is given by:

$$P(c | a_1a_2\dots a_n) = \sum_{i=1}^n \lambda_i P(c | a_1a_2\dots a_i)$$

$$\text{where } \lambda_i \in [0, 1] \quad \text{and} \quad \sum_{i=1}^n \lambda_i = 1$$

Cucerzan and Yarowsky claim that “it is reasonable to expect that smaller lambdas should

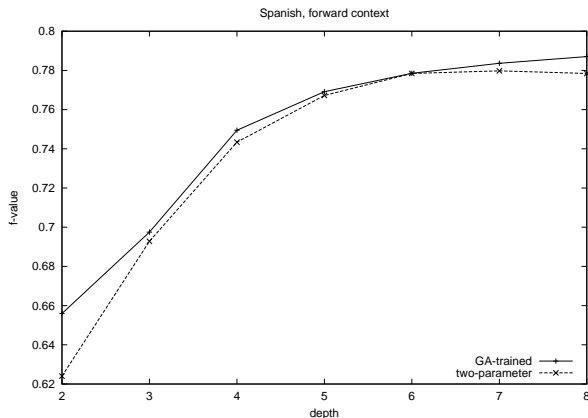


Figure 1: Trained general weighting functions versus two-parameter weighting functions

correspond to smaller indices, or even that $\lambda_1 < \lambda_2 < \dots < \lambda_n$ ”, and use this simplified two-parameter weighting function:

$$P(c \mid a_1 a_2 \dots a_n) = \beta P(c \mid a_1) + \sum_{i=2}^n \alpha^{n-i} P(c \mid a_1 a_2 \dots a_i)$$

where $\alpha, \beta \in (0, 1)$ and β is small

This function has the advantage of requiring less parameters, but risks decreasing performance through its assumptions. The general weighting function allows a much greater level of versatility, but it is difficult to estimate effective sets of parameters, especially as the trie depth (and hence number of parameters) increases. It is also unclear whether a single set of weights is effective for multiple languages, or if using different weights achieves better performance.

To evaluate the performance of the general weighting function, genetic algorithms were used to obtain sets of weights. Genetic algorithms were chosen because we had no knowledge of the smoothness of the parameter space. Selection criteria were chosen to maximise $F_{\beta=1}$, and were tested using four-fold cross-validation on the training set.

Figure 1 shows the performance of the best weighting sets produced through the use of genetic algorithms, and their performance compared to the simpler two-parameter weighting

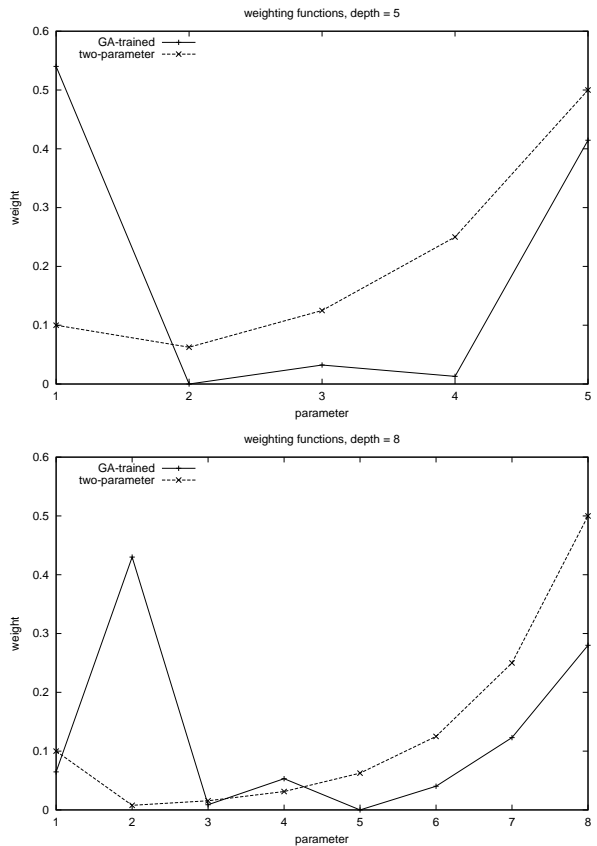


Figure 2: Differences in trained and standard weighting function parameters for tries of depth (a) 5, and (b) 8

function. The simpler weighting function is consistently outperformed, showing that its implicit assumptions hinder its performance.

An inspection of the trained parameter sets shows the assumption “smaller lambdas should correspond to smaller indices” does not hold. Figure 2 shows the best obtained weight sets for tries of depth 5 and 8, and compares them to a two-parameter model for the same depth. There are marked differences in the solutions obtained, both in comparison to the two-parameter model and to each other. While we have not explored the reason for these differences, it is clear that the combination of a more general weighting model with a parameter optimisation technique allows for the exploitation of language-specific phenomena, resulting in better performance.

5 Variable-Depth Weighting Approaches

When classifying a token, one cannot always expect the entire context string to be present in the training corpus, and hence the trie. If the trie is constructed from a very similar corpus to the test documents, the average matching length will be longer, but there will always be unseen tokens whose matched paths are shorter than the full trie depth. In this situation, only a subset of the parameters of the weighting function are used, resulting in inferior performance. An alternative approach is to use a weighting function with the number of parameters equal to the matched depth of the path in the trie. This requires the use of a set of weighting functions, rather than just one.

Through training weighting functions exclusively on strings that matched to a particular depth, the $F_{\beta=1}$ value increased by up to 0.4%. While not a large improvement, this shows that a variable-depth weighting approach can yield superior results to a single weighting function. Variable-depth weighting becomes more important as the test data differs more from the training data, since unseen tokens will generally yield shorter substrings. In the Spanish dataset used in this test, the test corpus is highly similar to the training corpus, with over 60% of tokens having at least their 8-letter-prefix present in the training set.

6 Minimum-Depth Orthographic Tries

Each node in an orthographic trie stores the cumulative frequency information for each category in which a given string of characters occurs. A heterogeneous node represents a string that occurs as more than one category, while a homogeneous node represents a string that occurs as only one category. If a string $a_1a_2 \dots a_n$ occurs in only one category, all longer strings $a_1a_2 \dots a_n \dots a_{n+k}$ are also of the same category. This redundancy can be exploited when constructing a trie.

A full trie of depth n contains every n -letter context that occurs in the training set. As

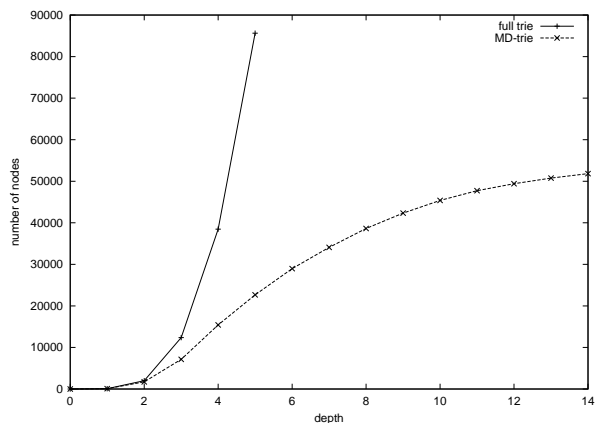


Figure 3: trie growth, full tries versus minimum-depth tries

trie depth increases, the number of strings (and hence the number of nodes) increases exponentially (see Figure 3). Although it is tempting to construct deeper tries to increase performance, it becomes less feasible to work with depths greater than 5 or 6.

As depths increase, so too does the number of homogeneous nodes. In practice, many NEs differ from all non-entities within the first few characters, while words with broader usage may require many more characters to be fully disambiguated. To build only the most useful sections of the trie, nodes are added only if their parent is heterogeneous. As soon as a string $a_1a_2 \dots a_n$ occurs in only a single category, longer strings $a_1a_2 \dots a_n \dots a_{n+k}$ are not necessary. We refer to a trie in which all non-terminal nodes are heterogeneous as a *minimum-depth* or *MD-trie*. This incremental creation of a trie also speeds construction time significantly. Figure 3 shows the difference in trie size between full and MD-tries. The number of nodes added at each depth decreases after depth = 4, and by depth = 15 only a few tokens remain ambiguous. Overall, an MD-trie of depth = 15 is smaller than a full trie of depth = 5.

To use a minimum-depth trie effectively, a variable-depth weighting strategy must be adopted, in which an independent weighting function is used for each depth. As is to be expected, tokens matched more fully are generally

trie	nodes	$F_{\beta=1}$ (B-ENT)
full, depth = 4	38477	75.0
full, depth = 8	375101	78.7
MD, depth = 15	51832	78.6

Table 1: MD-trie performance, Spanish

classified more successfully (Figure 4 (a)). Figure 4 (b) shows the number of matches at each depth, as compared to the full trie.

Overall performance of MD-tries is excellent, as seen in Table 1. The MD-trie performs as well as a full trie of depth = 8, and contains seven times fewer nodes. MD-tries give many of the practical advantages of deeper tries while requiring less resources.

7 Recognition using contextual features

The results reported thus far are from constructing a single trie by taking characters from the beginning of the token to be classified. There are four separate orthographic features that have been useful in NE recognition, identified in (Cucerzan and Yarowsky, 1999). These are:

- **forward**: left-to-right, from the beginning of current token. Captures common prefixes.
- **backward**: right-to-left, from the end of the current token. Captures common suffixes.
- **left context**: right-to-left, from the end of the previous token. Captures preceding contextual information.
- **right context**: left-to-right, from the beginning of the next token. Captures following contextual information.

Each feature may extend beyond its initial token, in which case the tokens are concatenated directly upon each other. Table 2 shows the relative performance of each of the features when used independently. As may be expected, the tries that start on the current token (forward, backward) are the most effective, but the contextual tries perform surprisingly well. The classifications made by each trie differ; Table 3 gives a breakdown of how many tokens were correctly classified by each trie independently. For example, in the Spanish dataset the right-context

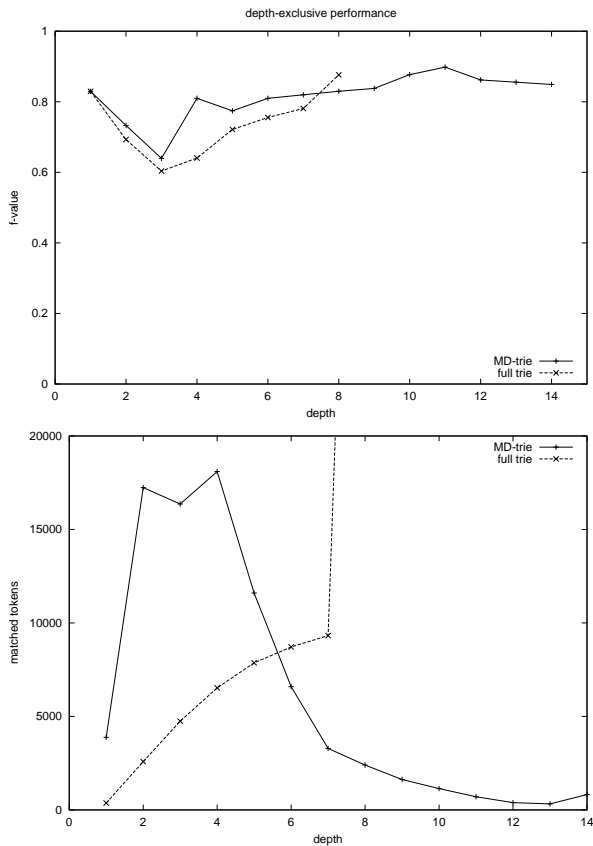


Figure 4: (a) Performance of each trie depth, (b) number of matches at each trie depth

classification (\overline{FB} , \overline{LR}) correctly identified 34 B-ENT tokens that were not identified using any other context, and only 58 B-ENT tokens were not identified by any of the tries (\overline{FB} , \overline{LR}). Each context has distinctive strengths and weaknesses, as will be discussed in Section 8.

To utilise these strengths, the scores for each category can be combined using a linear weighting function. For a set of n features, the combined score of a string s for a category c is given by

$$sc(c, s) = \sum_{i=1}^n \alpha_i sc_i(c, s)$$

where $sc_i(c, s)$ is the score given by an individual feature. Using the same genetic algorithm approach used for trie weighting functions (Section 4), effective trie combination functions were derived experimentally. For the Spanish

	B-ENT	I-ENT	ENT
forward	78.6	57.2	87.8
backward	77.4	59.1	82.1
left	41.9	57.5	51.7
right	44.5	36.1	57.6
combined	83.9	72.6	92.5
DGRAPH-GP	83.5	73.9	91.2
combined-ph	86.9	82.3	92.6
Carreras et al	94.1	91.3	95.6

Table 2: $F_{\beta=1}$ value for each feature, Spanish test

dataset, tries were found to be ranked in the following order (most significant to least significant): forward, left, right, backward. This ordering does not reflect the individual performance of each trie. The combination of tries performed markedly better than any individual trie, achieving $F_{\beta=1}$ values of 83.9, 72.6, and 92.5 for B-ENT, I-ENT, and ENT (combined) categories (see Table 2). Also shown is results from correcting phrase boundaries through simple post-processing (combined-ph), and the best results from CoNLL 2002 (Carreras et al., 2002). Considering the relative complexities of the approaches, the comparison is encouraging.

Weighting functions are not necessarily the most appropriate method for combining classifications from multiple tries. For each token, the scores for each category (O, B-ENT, I-ENT) for each feature (forward, backward, left-context, right-context) can be used as attributes for use with a machine learner. To test this approach, we used DGRAPH-GP, a decision-graph based classifier featuring boosting (Patrick and Goyal, 2001). Performance was very good, with $F_{\beta=1}$ values of 83.5, 73.9, and 91.2 for B-ENT, I-ENT and ENT (combined) categories. A disadvantage of using an external classifier is that, depending on the classifier used, the relative ranking of classifications for each token may not be available, whereas such information can be useful in further post-processing.

By using multiple tries to capture both word-internal and contextual information, performance can be increased significantly. While this

all	\overline{LR}	\overline{LR}	$L\overline{R}$	LR
\overline{FB}	858	376	483	348
\overline{FB}	222	256	475	439
FB	3272	11717	10586	71907
$F\overline{B}$	617	944	630	1326

B-ENT	\overline{LR}	\overline{LR}	$L\overline{R}$	LR
\overline{FB}	58	34	27	42
\overline{FB}	26	18	32	37
FB	900	669	1792	3417
$F\overline{B}$	243	143	263	208

Table 3: (a) All tokens, (b) B-ENT tokens correctly identified by four features, Spanish test

is evident in NE recognition, greater benefits will be realised when classifying NEs into finer-grained categories (person, place, etc), as it is expected that patterns of usage vary greatly between NE types. Through the use of trained weighting functions, or machine learners, tries can be combined effectively for each target language, making this technique very suitable for language-independent systems.

8 Error Analysis

Tries provide a simple way to capture statistical orthographic differences between token types. Through an examination of the types of systematic errors made using this technique, the relative strengths and weaknesses can be assessed. Each of the four features (forward, backward, left, right) are dealt with individually before looking at their combined performance. Table 4 gives a breakdown of the classifications made by each feature.

The forward feature performed best of all tries in every tested language. All these languages (Spanish, Dutch) use capitalisation, both to mark proper nouns and the beginning of a sentence, so reasonable performance is expected. The forward trie produced the best results for B-ENT and O categories. Of the 527 non-entities misclassified as B-ENTs by the forward trie, 482 (91%) were either capital-initial (49%) or all-caps (51%) tokens. This represents only 17% of capitalised non-entities, showing that per-

formance is much higher than a simple “entities have initial capitals” rule. Of the 220 B-ENT tokens falsely classified as non-entities, 80 (37%) were capitalised. Of the remaining non-capitalised tokens, most occurred directly before NEs, and were present in the training set with ambiguous classifications. For example, ‘restaurante’ appears in the training set only as a non-entity, but occurs once in the test set as a B-ENT. Problematic words included ‘plaza’, ‘hotel’, and ‘estadio’.

The backward feature aims to classify tokens based on their suffix. This leaves more room for errors on tokens broadly used suffixes. From inspection of the 1126 false positives, 76% were not capitalised, in contrast to the errors made by the forward feature. The most commonly misclassified tokens had suffixes that were used commonly by both NEs and non-entities, such as ‘ria’ (37% NE), ‘cia’ (33% NE) and ‘ado’ (10% NE).

The left feature ignores the token it is classifying, focusing instead on the previous token. This approach should work well when the NE does not appear to be an NE, but it is used in the same way as other NEs. The left trie was the most successful classifier of I-ENTs, and made the least misclassifications between B- and I-ENTs. There were 4093 tokens correctly classified by the left token that were wrongly classified using the forward trie, including 99 NEs. Of these, 29 were non-capitalised tokens, typically the most problematic.

Like the left-context trie, the right-context trie looks beyond the current token, focusing on the following tokens. Overall, this performed the worst, but correctly identified 2905 tokens missed by the forward trie, including 58 NEs.

The four tries were combined using a trained linear weighting function (see Section 4). When used together, many of the individual shortcomings of each feature were overcome. Greatest improvements were in the I-ENT category, with around half the error rate of the forward trie. Misclassifications between O and B-ENT categories actually increased in both directions compared to the forward trie, which indicates that the combination of tries can be detrimental.

	trie classification			
forward	B-ENT	I-ENT	O	total
B-ENT	6433	1256	220	7909
I-ENT	1507	3745	584	5836
O	527	2265	87919	90711
total	8467	7266	88723	104456

backward	B-ENT	I-ENT	O	total
B-ENT	6299	896	714	7909
I-ENT	953	3753	1130	5836
O	1126	2221	87364	90711
total	8378	6870	89208	104456

left	B-ENT	I-ENT	O	total
B-ENT	4997	324	2588	7909
I-ENT	396	4365	1075	5836
O	10550	4651	75510	90711
total	15943	9340	79173	104456

right	B-ENT	I-ENT	O	total
B-ENT	4157	1581	2171	7909
I-ENT	1083	3143	1610	5836
O	5520	6851	78340	90711
total	10760	11575	82121	104456

combined	B-ENT	I-ENT	O	total
B-ENT	6853	758	298	7909
I-ENT	876	4407	553	5836
O	684	1142	88885	90711
total	8413	6307	89736	104456

Table 4: Classification and error frequencies for each feature, Spanish test

9 Size of Training Corpus

As used in this paper, an annotated training corpus is required both to construct the trie and for use in obtaining suitable weighting functions through machine learning. Only orthographic phenomena present in the training corpus can be captured by the trie; given less data, the average matched length of tokens will decrease, giving a less accurate classification.

To evaluate the impact of changing training corpus size, the forward trie was constructed for the Spanish and Dutch datasets for training sizes of between 500 and 250,000 words. The

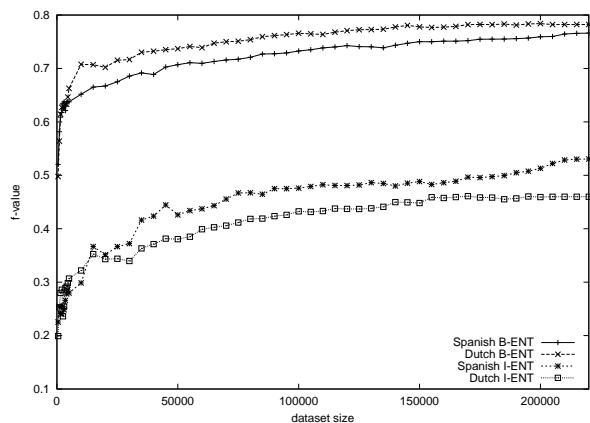


Figure 5: Effect of training corpus size

benefits of larger training corpora can be seen in Figure 5. Where a corpus of suitable size is unavailable, tries can be used with techniques such as bootstrapping (Cucerzan and Yarowsky, 1999), in which only a small seed list and a large unannotated corpus are needed.

10 Conclusion

Our method of using orthographic tries is highly language independent, and gives promising results for the languages tested, Spanish and Dutch. By using a meta-linguistic model with fewer assumptions, it requires less external input in the form of rules, feature-extraction, or gazetteers. We have increased the performance of trie-based classification through optimising a more general weighting function, and by using MD-tries to build deeper tries with very low cost.

Word-internal and contextual orthographic features effectively capture the differences between entities and non-entities. Combining these features, either through trained weighting functions or a machine learner, yields results considerably higher than any single feature.

While tries can be used as the sole basis for classification, they are also well-suited to larger NER systems. Trie-based features can be used in place of, or together with, traditional orthographic features. This has previously been shown to provide significant performance increases (Patrick et al., 2002).

References

- Daniel M. Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. 1997. Nymble: a high-performance learning name-finder. In *Proceedings of ANLP-97*, pages 194–201.
- A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. 1998. Exploiting diverse knowledge sources via maximum entropy in named entity recognition.
- S. Buchholz and A. van den Bosch. 2000. Integrating seed names and n-grams for a named entity list and classifier.
- Xavier Carreras, Lluís Màrques, and Lluís Padró. 2002. Named entity extraction using adaboost. In *Proceedings of CoNLL-2002*, pages 167–170. Taipei, Taiwan.
- Hai Leong Chieu and Hwee Tou Ng. 2002. Teaching a weaker classifier: Named entity recognition on upper case text. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 481–488.
- M. Collins and Y. Singer. 1999. Unsupervised models for named entity classification.
- S. Cucerzan and D. Yarowsky. 1999. Language independent named entity recognition combining morphological and contextual evidence.
- Walter Daelemans, Antal van den Bosch, and Jakub Zavrel. 1999. Forgetting exceptions is harmful in language learning. *Machine Learning*, 34(1-3):11–41.
- A. Mikheev, M. Moens, and C. Grover. 1999. Named entity recognition without gazetteers.
- Jon D. Patrick and Ishaan Goyal. 2001. Boosted decision graphs for nlp learning tasks. In Walter Daelemans and Rémi Zajac, editors, *Proceedings of CoNLL-2001*, pages 58–60. Toulouse, France.
- Jon Patrick, Casey Whitelaw, and Robert Munro. 2002. Slinerc: The sydney language-independent named entity recogniser and classifier. In *Proceedings of CoNLL-2002*, pages 199–202. Taipei, Taiwan.
- Erik F. Tjong Kim Sang. 2002. Introduction to the conll-2002 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2002*, pages 155–158. Taipei, Taiwan.
- GuoDong Zhou and Jian Su. 2002. Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 473–480.