Constructing a Named Entity Ontology from Web Corpora

Ming-Shun Lin and Hsin-Hsi Chen

Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan
E-mail: mslin@nlg.csie.ntu.edu.tw; hhchen@csie.ntu.edu.tw

Abstract

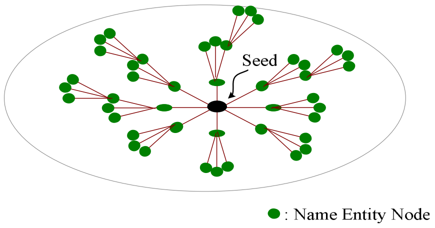
This paper proposes a named entity (NE) ontology generation engine, called X_{NE}-*Tree* engine, which produces relational named entities by given a seed. The engine incrementally extracts high co-occurring named entities with the seed by using a common search engine. In each iterative step, the seed will be replaced by its siblings or descendants, which form new seeds. In this way, X_{NE}-*Tree* engine will build a tree structure with the original seed as a root incrementally. Two seeds, Chinese transliteration names of Nicole Kidman (a famous actress) and Ernest Hemingway (a famous writer), are experimented to evaluate the performance of the X_{NE}-*Tree*. For test the applicability of the ontology, we employ it to a phoneme-character conversion system, which convert input phoneme syllable sequences to text strings. Total 100 Chinese transliteration names, including 50 person names and 50 location names are used as test data. We derive an ontology composed of 7,642 named entities. The results of phoneme-character conversion show that both the recall rate and the MRR are improved from 0.79 and 0.50 to 0.84 to 0.55, respectively.

1. Introduction

Named entities are common foci of searchers. Thompson & Dozier (1997) showed that named entity recognition (NER) could improve the performance of information retrieval systems. Named entity ontology is an important language resource for NER, however, collecting named entities is challenging due to their flexible formulation and up-to-date use. For some emerging applications like personal name disambiguation (Fleischman & Hovy, 2004; Mann & Yarowsky, 2003), social chain finding (Bekkerman & McCallum, 2005; Culotta et al, 2004; Raghavan et al, 2004), *etc*., glossary-based representation of named entities is not enough. How to distinguish the relationships among named entities of the same relation type, e.g., Nicole Kidman and Tom Cruise are two persons, and they are actress and actor, is indispensable.

The web, which provides huge, dynamic, and rich information, is considered as a very large scale live corpus for many natural language applications. Named entities are important objects in web documents. Google sets, http://labs.google.com/sets, extract named entity items from web pages, when inputting a few named entities in languages other than Chinese. Matsuo et al (2004) based on the information of related web pages to find web of trust. Besides, how to get the word counts and the word association counts from the web pages without scanning over the whole collections is essential. Keller & Lapata (2003) show that bigram statistics for English language is correlated between corpus and web counts. Directly managing the web pages is not an easy task when the Web grows very fast. How to utilize the huge volume of web data for training a language model and how to measure the similarity among named entities are important issues to be resolved. In the past, various measures have been proposed to compute the similarity score of objects of different granularity (Li et al, 2003; Rodríguez & Egenhofer, 2003). However, they compute the semantic similarity based on WordNet rather than Web information. In this paper, we propose a *Co-Occurrence Double-Check score* (*CODC*) to measure the similarity of named entities by using any common search engine.

We focus on Chinese named entities and implement a named entity ontology generation, called X_{NE}-*Tree* engine, which produces relational named entities by given a seed. This engine incrementally extracts high co-occurring named entities from the related web pages by using Google. Based on PageRank algorithm, the extracted named entities have similar relational property. In each iterative step, the seed will be replaced by its siblings or descendants, which form new seeds. In this way, X_{NE}-*Tree* engine will build a tree structure as follows with the original seed as a root incrementally.



Section 2 presents the *Co-Occurrence Double-Check score* and the overall flow of the X_{NE}-*Tree* engine. Section 3 discusses its applications to the generation of an NE ontology. Section 4 shows the utilization of the NE ontology on phoneme-character conversion. Section 5 concludes the remarks.

2. An NE Ontology Generation Engine

How to recognize a named entity and to calculate the relational property score with a seed are two crucial issues. Firstly, we submit a given seed to a search engine, and select the top N returned snippets. Then, we use suffix tree to extract possible patterns automatically. The patterns, which are extracted on the basis of the global statistic, may be impacted by the frequency variance of pattern with the same substrings. Because our target is to generate named entities, most of the max-duplicated strings can be filtered out by using an NER system. The NER system will re-segment a candidate pattern to some substrings and give each substring a part of speech (POS) and a possible name tag. If any substring is tagged as a location, an organization, or a person, the candidate pattern is considered as a named entity. Because preposition is a high frequent function word, i.e., it often occurs before/after a named entity, the suffix tree approach may introduce the wrong boundary. We filter out those substrings having a preposition tag.

Secondly, we calculate a relational property score, called *Co-Occurrence Double-Check score* (*CODC*), of each extracted name entity (denoted *Y*) with a seed (denoted *X*). We postulate that *X* and *Y* have strong relationship if we can find *Y* from *X* (a forward process) and find *X* from *Y* (a backward process). The forward and the backward processes form a double check operation. *CODC*(*X*,*Y*) is defined as follows.

$$CODC(X,Y) = \begin{cases} 0 & \text{if } f(Y@X)=0 \text{ or } f(X@Y)=0 \\ e^{\log\left(\frac{f(Y@X)}{f(X)} \times \frac{f(X@Y)}{f(Y)}\right)^{\alpha}} & \text{Otherwise} \end{cases} \quad (1)$$

Where *f*(*X*@*Y*) is total occurrences of *X* in the top *N* snippets when query *Y* is submitted to search engine; similarly, *f*(*Y*@*X*) is the total occurrences of *Y* in the top *N* snippets for query *X*; *f*(*X*) is the total occurrences of *X* in the top *N* snippets of query *X*, and, similarly, *f*(*Y*) is the total occurrences of *Y* in the top *N* snippets of query *Y*. In each iterative step, *Y* will be added into a queue when *CODC* (*X*,*Y*) is larger than a threshold *θ*. Then, we get a new seed *X* from queue. *CODC* measure achieves the best performance when α=0.15. The overall flow is shown in Figure 1.

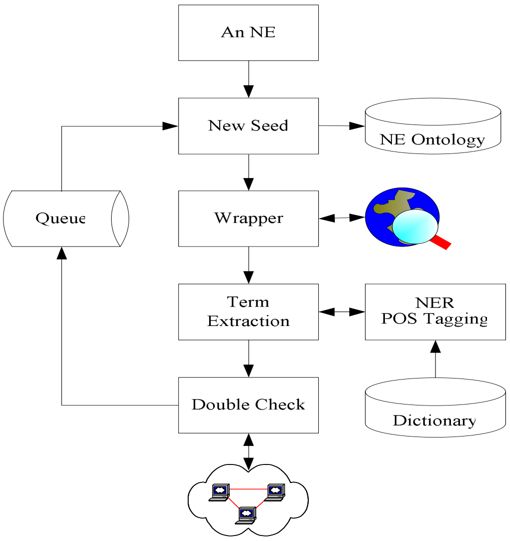


Figure 1. Flow of Named Entity Ontology Generation

3. Generating an NE Ontology

For control the generation of ontology, we set a condition as follows. Each initial seed can derive at most four layers and no more than *N* children are allowed in the first layer. The maximal number of children of a named entity at layer (*i*+1) is bounded by the number at the layer *i* multiplying by a decreasing rate, *γ*. In other words, each node at layer *i* can generate at most $N\times\gamma^i$ nodes. Those named entities with *CODC* scores larger than a predefined threshold are sorted and sufficient number of named entities is selected in sequence for expansion. If the size of ontology is larger than *M*, then we stop expansion. Here we employ Touch-Graph[1] to represent named entity ontology. Figure 2 shows an example by using "妮可基嫚" as a seed, which is a Mandarin transliteration name of a famous actress "Nicole Kidman", to build an ontology. In this example, we set *N*=15, *γ*=0.7, *θ*=0.1, and *M*=200.

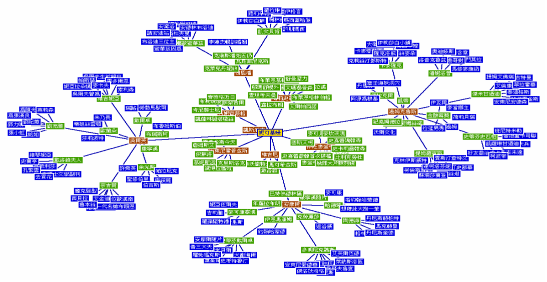


Figure 2 A Snapshot of Named Entity Ontology of "妮可基嫚" ("Nicole Kidman")

To evaluate the performance, we consider the following four types.

(1) Named Entity (NE) type: In this case, the proposed candidate should be a named entity and does not have wrong boundary. A personal name with a title or first name of more than 3 characters is regarded as correct. In contrast, patterns with last name only are considered as error.

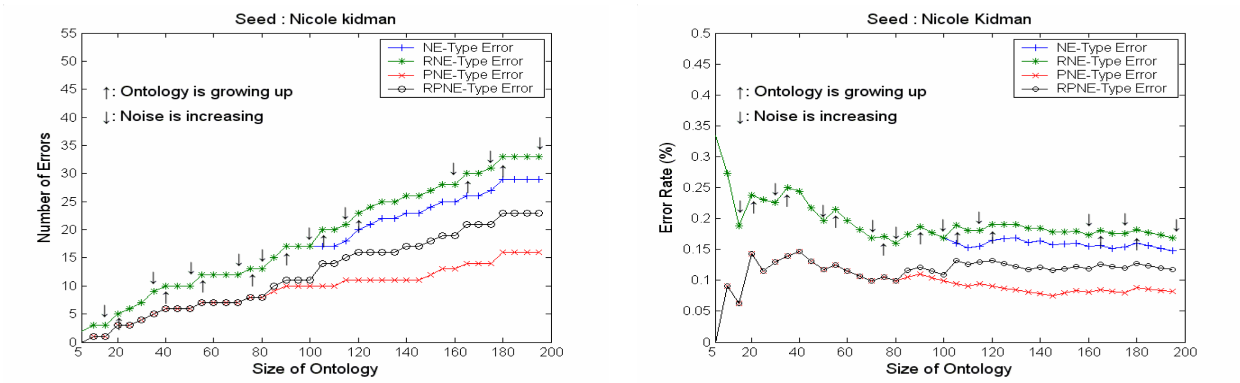
(2) Relational property of NE (RNE) type: The acceptable strings in (1) which have the same relational property with the initial seed or its parent are considered as correct. The remaining nodes are wrong.

(3) Partial Named Entity (PNE) type: We relax the restriction of boundary errors specified in (1). Patterns consisting of partial named entities are regarded as correct. The remaining nodes are wrong.

(4) Relational property of PNE (RPNE) type: The acceptable strings in (3) which have the same relational property with the initial seed or its parents are considered as correct. The remaining nodes are wrong.

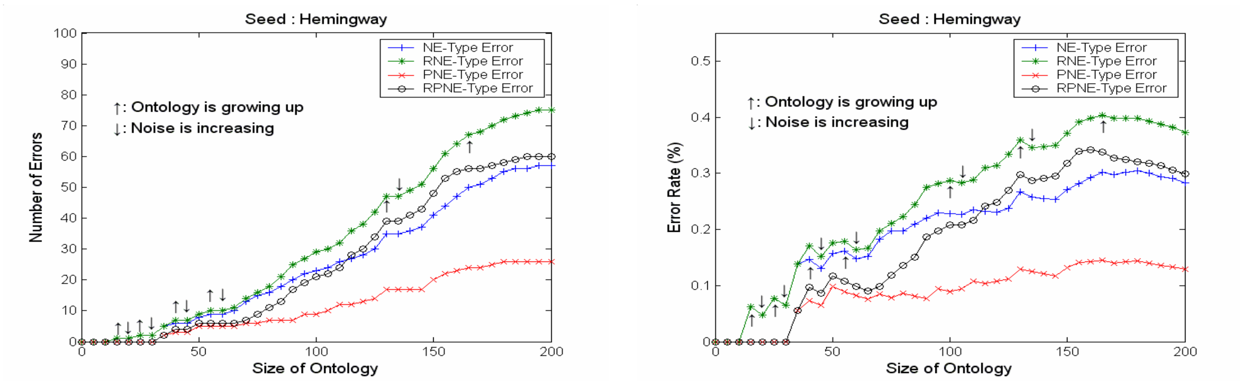
Figure 3(a) shows the number of errors of the four types in above example, i.e., Nicole Kidman. The figure depicts that there are some points in which NE ontology is growing up without introducing too many errors and there are some points in which NE ontology is growing up accompanying with noise. We use the symbol '↑' to represent when ontology is growing up and symbol '↓' to represent when noise is increasing, respectively. Figure 3(b) shows the error rates of the four types, where the error rate is number of errors divided by ontology size. The error rates of the NE type, the RNE type, the PNE type and the PRNE type are 14.58%, 16.59%, 8.05% and 11.56%, respectively.

[1] http://www.touchgraph.com



(a) (b)

Figure 3. (a) Number of Errors and (b) Error Rates for NE Ontology Generated by Using “Nicole Kidman”



(a) (b)

Figure 4. (a) Number of Errors and (b) Error Rates for NE Ontology Generated by Using “Hemingway”

We also consider another seed “Ernest Hemingway (a famous writer)”. Here we set parameters N=30, γ=0.2, θ=0.01, and M=200. The reason of using larger N and smaller γ and θ for this seed is that the ontology quickly converges on front nodes in this case. Because “Ernest Hemingway” is relatively not active, the seed may generate more noise easily. Figure 4 shows the results. The error rates become larger when the size of the ontology is increased. The error rates of the NE type, the RNE type, the PNE type and the PRNE type are 27.95%, 36.77%, 12.75% and 29.41%, respectively. The experiments show that NE ontology generated by using “Nicole Kidman” is more stable than that generated by using “Hemingway”.

4. An Application of Constructed Ontology on Phoneme-Character Conversion

Lin et al (2005) use the Web as a live corpus for spoken transliteration name access. Their speech recognition system converts speech signals to text strings. In the experiments, we assume that the phoneme syllable is correctly identified. For example, the input of phoneme syllables for “妮可基嫚” are “ni ko ki man“. The architecture is shown in Figure 3.

We use recall rate and MRR (Mean Reciprocal Rank) (Voorhees 1999) to evaluate the performance. Recall rate means how many transliteration names are correctly recognized. MRR defined below means the average ranks of the correctly identified transliteration names in the proposed candidates.

$$MRR = \frac{1}{M}\sum_{i=1}^{M} r_i \qquad (2)$$

where r_i = 1/$rank_i$ if $rank_i$ > 0; and r_i is 0 if no answer is found, and M is total number of test cases. The $rank_i$ is the rank of the first right answer of the i^{th} test case. That is, if the first right answer is rank 1, the score is 1/1; if it is at rank 2, the score is 1/2, and so on. The value of MRR is between 0 and 1. The inverse of MRR denotes the average position of the correct answer in the proposed candidate list. The higher the MRR is, the better the performance is.

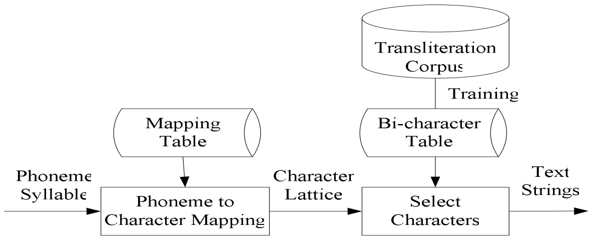


Figure 3. Flow Chart of Phoneme-Character Conversion

In Figure 3, a transliteration name corpus is needed to train a bi-character table. In the experiments, we compare a given transliteration names corpus and an automatically constructed ontology to demonstrate the performance of the X_{NE} engine. Total 100 transliteration named entities are used for testing. The test data include 50 American state names, 29 movie star names and 21 NBA star names (Lin et al, 2005). First, we employed 51,111 transliteration names (BaselineTN) to train the bi-character language model. Nevertheless, some transliteration names might not be active on the web. We submitted all transliteration names to a search engine. If the search engine returned no web pages for a name, we filter it out. Finally, we got 36,178 transliteration names (FilterTN) in this step. Table 1 shows that FilterTN is a little better than BaselineTN.

Language Model	Size of TN	Performance	
		MRR	Recall
BaselineTN	51,111	0.50	0.79
FilterTN	36,178	0.50	0.80

Table 1. Two Basic Transliteration Name Corpora

Next, we employ the test data as 100 seeds to generate NE ontology. Let N=15, γ=0.7, θ=0.1, and M=200. Total 7,642 nodes are generated by the 100 seeds. Table 2 shows the performance of ontology generation. Of those 7,642 nodes, the error rates of the NE type, the RNE type, the PNE type and the PRNE type are 19.60%, 34.20%, 12.62% and 29.82%, respectively. Of 7,642 named entities (Total-Ontology) reported by X_{NE} engine, 6,146 named entities (NE-Ontology) belong to the correct NE type, and 5,023 named entities (RNE-Ontology) belong to the correct RNE type. In this way, we employ FilterTN+RNE-Ontology, FilterTN+NE-Ontology and FilterTN+Total-Ontology to build bi-character language models. Table 3 summarizes the experimental results of language models with NE ontology. The three models with NE ontology outperform the baseline models. In particular, the NE ontology improves the recall rate and the MRR from 0.79 and 0.50 (BaselineTN model) to 0.84 and 0.55 (FilterTN+RNE-Ontology model), respectively.

Total Seeds	Size of Ontology	NE	RNE	PNE	RPNE
100	7,642	19.60%	34.20%	12.62%	29.82%

Table 2. Performance of X_{NE} Engine with 100 Seeds

Language Model	Total TNs	Performance	
		MRR	Recall
FilterTN + RNE-Ontology	41,201	0.55	0.84
FilterTN + NE-Ontology	42,324	0.57	0.83
FilterTN + Total-Ontology	43,820	0.57	0.82

Table 3. Performance of Bi-character Language Models Trained with NE Ontology

5. Conclusion

This paper proposes an NE ontology generation engine, which automatically creates named entity ontology for a given seed. Such an ontology can be applied to information retrieval, social chain finding, language model training, personal name disambiguation, and so on. In the experiments, there are total 7,642 named entities in the ontology initiated by 100 seeds. Of those 7,642 nodes, the error rates of the NE type, the RNE type, the PNE type and the PRNE type are 19.60%, 34.20%, 12.62% and 29.82%, respectively. We employ the ontology to a phoneme-character conversion system, and the experimental results show that both the recall rate and the MRR are improved from 0.79 and 0.50 to 0.84 to 0.55, respectively. That demonstrates our NE ontology generator is effective indirectly.

Acknowledgments

Research of this paper was partially supported by National Science Council, Taiwan, under the contract NSC94-2752-E-001-001-PAE and NSC95-2752-E-001-001-PAE.

References

Thompson, P., Dozier, C. (1997). Name Searching and Information Retrieval. *Proceedings of Second Conference on Empirical Methods in Natural Language Processing*, pp.134-140.

Fleischman, M.B., Hovy, E. (2004). Multi-document Person Name Resolution. *Proceedings of the Workshop on Reference Resolution and its Applications: ACL'04*, Barcelona.

Culotta, A., Bekkerman, R., McCallum, A. (2004). Extracting Social Networks and Contact Information from Email and the Web. *Proceedings of the First Conference on Email and Anti-Spam*.

Mann, G., Yarowsky, D. (2003). Unsupervised Personal Name Disambiguation. *Proceedings of CoNLL-7.*

Bekkerman, R., McCallum, A. (2005). Disambiguating Web Appearances of People in a Social Network. *Proceedings of WWW'05*, pp. 463-470.

Raghavan, H., Allan, J., McCallum, A. (2004). An Exploration of Entity Models, Collective Classification and Relation Descriptions, *Proceedings of LinkKDD.*

Matsuo, Y., Tomobe, H., Hasida, K., Ishizuka, M. (2004). Finding Social Network for Trust Calculation. *Proceedings of 16th ECAI*, pp. 510-514.

Keller, F., Lapata, M. (2003). Using the Web to Obtain Frequencies for Unseen Bigrams. *Computational Linguistics*, pp. 459-484.

Li, Y., Bandar, Z.A., McLean, D. (2003). An Ap-proach for Measuring Semantic Similarity between Words Using Multiple Information Sources. *IEEE Transactions on Knowledge and Data Engineering*, pp. 871-882.

Rodríguez, M.A., Egenhofer, M.J. (2003). Determin-ing Semantic Similarity among Entity Classes from Different Ontologies. *IEEE Transactions on Knowledge and Data Engineering*, pp. 442-456.

Voorhees, E. (1999) "The TREC-8 Question Answering Track Evaluation", *Proceedings of the 8th TREC*, pp. 23-37.

Lin, M.S., Chen, C.P., Chen, H.H. (2005) An Approach of Using the Web as a Live Corpus for Spoken Transliteration Name Access. *Proceedings of 17th ROCLING Conference*, pp. 361-370.