Text Patterns and Compression Models for Semantic Class Learning

Chung-Yao Chuang
Institute of Information Science
Academia Sinica, Taiwan
cychuang@iis.sinica.edu.tw

Yi-Hsun Lee
Institute of Information Science
Academia Sinica, Taiwan
rog@iis.sinica.edu.tw

Wen-Lian Hsu
Institute of Information Science
Academia Sinica, Taiwan
hsu@iis.sinica.edu.tw

Abstract

This paper proposes a weakly-supervised approach for extracting instances of semantic classes. This method constructs simple wrappers automatically based on specified seed instances and uses a compression model to assess the contextual evidence of its extraction. By adopting this compression model, our approach can better avoid erroneous extractions in a noisy corpus such as the Web. The empirical results show that our system performs quite consistently even when operating on a noisy text with a lot of possibly irrelevant documents.

1 Introduction

Extracting instances of semantic classes is a fundamental task in natural language processing (NLP). Such a task aims to extract instances belonging to a specific category such as acquiring *Tom Hanks* and *Al Pacino* into a list containing other actors. This kind of information serves as building blocks for various NLP tasks. For example, major search engines such as Yahoo and Google gather large amount of such classes (Paşca, 2007; Chaudhuri et al., 2009) to better interpret queries and provide search suggestions. Other applications include ontology learning (Cimiano et al., 2004), co-reference resolution (McCarthy and Lehnert, 1995) and advertisement matching (Chang et al., 2009).

Most of the approaches for this task can be roughly classified into two categories: distributional and pattern-based. The distributional approaches use contextual similarity to model the instances of a given class. Following the distributional hypothesis (Harris, 1970), these methods take a small set of seed instances and generate new instances from noun phrases that are most similar to the seeds in terms of the distributions of surrounding words (Sarmento et al., 2007; Pantel et al., 2009).

The pattern-based approaches use text patterns to extract instances of a given semantic class (Riloff and Shepherd, 1997; Riloff and Jones, 1999; Banko et al., 2007; Paşca, 2007). The most representative study is the group of patterns proposed by Hearst (1992). For example, patterns like '*X such as Y*' and '*X including Y*' can be applied to extract instances from '*actors such as Tom Hanks*' and '*countries including Japan*'. In these approaches, semantic classes are specified by providing small sets of seed instances or seed patterns such as (Kozareva et al., 2008) which utilized a single hyponym pattern combined with graph structure to extract semantic lexicons from the Web. In addition to natural language patterns, Wang and Cohen (2007) demonstrated an approach that learns the pattern of specific meta-structure of the document (e.g., tags in HTML) automatically from seed instances.

In this paper, we propose a method similar to (Kozareva et al., 2008) and (Wang and Cohen, 2007) in that it also employs graph ranking algorithm to assess the reliability of the extracted candidates and uses the Web as the source of extraction. Different from them, the wrappers we induced from web pages are less specific and do not contain any structural cues such as HTML tags. In our approach, those wrappers serve primarily as a means to bracket candidate mentions. The main discriminative power resides in adopting a text compression model called Prediction by Partial Matching (PPM) (Cleary and Witten, 1984) to evaluate the contextual similarity between a mention and seed instances. The similarity is measured by compression ratio achieved when compressing the surrounding context of the mention using the PPM model loaded with context statistics of seed instances. In this work, we focused on

assessing the effectiveness of such measurement and this effort can potentially be extended to systems that adopt more elaborated text patterns.

The rest of this paper is organized as follows. Section 2 briefly describe the PPM compression model. In section 3, we detail the idea of using compression ratio as similarity measure. Section 4 outlines our approach. Section 5 shows the results of experiments and section 6 concludes this paper.

2 Prediction by Partial Matching

In this work, we use the Prediction by Partial Matching (PPM) compression scheme (Cleary and Witten, 1984) which has become a benchmark in the lossless text compression. It generates "predictions" for each input token in turn. Each prediction takes the form of a probability distribution that is provided to an encoder, which is usually an arithmetic coder. However, the details of actual coding technique are of no relevance to this paper.

PPM can be seen as an n-gram approach that uses finite-context models of tokens, where the previous few tokens predict the upcoming one. The conditional probability distribution of tokens, conditioned on the preceding context, is maintained and updated as each token of input is processed. This distribution, along with the the preceding few input tokens, is used to predict each upcoming token. Exactly the same distributions are maintained by the decoder, which updates the appropriate distribution as each token is received.

Rather than using a fixed context length, the PPM chooses a maximum context length, say ℓ, and maintains statistics for this and all shorter contexts. To combine these statistics, for each upcoming token, the PPM starts with the order ℓ model. If that model contains a prediction for the token, the token is encoded according to the order ℓ model. Otherwise, both encoder and decoder *escape* down to order $\ell - 1$. There are two possible situations. If the order ℓ context has not been encountered before, then escaping to order $\ell - 1$ is inevitable, and both encoder and decoder can arrive at that fact without any communication. On the other hand, if the preceding ℓ tokens have been encountered in sequence before but not followed by the upcoming character, then only the encoder knows that an escape is necessary. In this case, it must signal the decoder by transmitting an *escape event*. Thus, space must be reserved for this event in every probability distribution that encoder and decoder maintain.

Once any necessary escape event has been transmitted and received, both encoder and decoder agree that the upcoming token will be coded by $\ell - 1$ order model. Of course, this may not be possible either, and further escapes may take place. Ultimately, the order 0 model may be reached; in this case, the token can be encoded if it has occurred before. Otherwise, there is one further escape, and both encoder and decoder will agree that the token itself will be literally transmitted.

There is one remaining question regarding this backoff strategy: how much space to preserve for the escape probability. In this work, we assign the escape probability in particular context as

$$\frac{\frac{1}{2}d}{n}$$

where n is the number of times that context has appeared and d is the number of different tokens that have directly followed it. And the probability of a token that has occurred c times in that context before is

$$\frac{c - \frac{1}{2}}{n}$$

This allocation strategy is called PPMD (Howard, 1993) and has shown great performance in text compression. Once the token has been processed, the model will be updated to include this context-to-token prediction.

Most of the discourses of PPM were on character-based compression, although the above backoff strategy can be equally applied to other class of symbols such as words. Previous experiments with a wide range of English text has shown that word-based models consistently outperform the character-based counterpart (Teahan and Cleary, 1997). In this work, we adopt the word-based model for our task. A more comprehensive description of the PPM algorithm can be found in (Bell et al., 1990).

3 Compression Ratio as Similarity

In this work, we use the compression ratio as a measure of similarity between the context of a mention and the contexts of seed instances. More specifically, this ratio is defined as

$$\frac{\lambda_M(\mathbf{x})}{\lambda_B(\mathbf{x})}$$

where $\mathbf{x}$ is the sequence of words surrounding the mention, $\lambda_B(\mathbf{x})$ is the code length of $\mathbf{x}$ encoded by a blank PPM, i.e., the PPM without any context statistics pre-loaded. And $\lambda_M(\mathbf{x})$ is the code length of $\mathbf{x}$ compressed by the model M which loaded with the context statistics of seed instances, i.e., the model that has run through the contexts collected from the vicinity of seed instances observed in the corpus.

The encoding of a token x_i in $\mathbf{x} = x_1 x_2 \cdots x_n$ is based on the probability predicted by the PPM, which conditioned on the preceding tokens $\mathbf{y_i} = \cdots x_{i-2} x_{i-1}$. Let p_M and p_B be the probabilities predicted by those two models, from an information theoretic perspective, the above ratio can be interpreted as follows,

$$\begin{aligned}\frac{\lambda_M(\mathbf{x})}{\lambda_B(\mathbf{x})} &= \frac{\sum_{i=1}^{n} -log_2 p_M(x_i|\mathbf{y_i})}{\sum_{i=1}^{n} -log_2 p_B(x_i|\mathbf{y_i})} \\ &= \frac{-log_2 \prod_{i=1}^{n} p_M(x_i|\mathbf{y_i})}{-log_2 \prod_{i=1}^{n} p_B(x_i|\mathbf{y_i})}\end{aligned}$$

which is the log-likelihood ratio. Intuitively, this ratio gives an estimate of how much better M predicts $\mathbf{x}$ compared to the prediction without any prior assumptions. And the more effective the prediction, the more similar $\mathbf{x}$ and the contexts of seed instances, which M was built upon.

Leveraging this concept, we can filter possibly irrelevant mentions by comparing the ratio to a threshold θ. More specifically, we test whether

$$\frac{\lambda_M(\mathbf{x})}{\lambda_B(\mathbf{x})} > \theta \quad (1)$$

If this inequality holds, we skip the corresponding mention. In this work, we adopt $\theta = 0.3$, which empirically gives a good performance.

4 Proposed Approach

Our approach is outlined in Algorithm 1. The procedure starts at collecting the contexts of seed instances observed in the corpus and making wrappers based on these occurrences. In this work, a wrapper is constructed as a regular expression of two tokens[1] preceding and one or two tokens following a seed occurrence depending on what next to the occurrence is a punctuation or word. The collected contexts are then fed into a PPM. Utilizing this PPM, we filter the mentions according to Inequality 1. Following that, a graph G is build based on the wrappers and remaining mentions. In this graph, a vertex represents either a wrapper or a mention, and an edge denotes a captured-by or produced-by relationship. The vertices in G are then ranked by graph ranking algorithm. In this work, we adopt RageRank with Prior (White and Smyth, 2003). Finally, we obtain a list of mentions according to the vertices ranking in G.

[1] A token means a word or a punctuation.

Algorithm 1 The Proposed Approach

Input: A set of seed instances S and corpus C
Output: A ranked list of extracted instances

```
W = φ, T = φ
for each s in S do
    for each occurrence of s in C do
        Make a wrapper and add it into W.
        Collect the surrounding text into T.
    end for
end for
Build a PPM M based on T.
for each w in W do
    for each mention in C captured by w do
        x ← text surrounding this mention
        if λ_M(x)/λ_B(x) > θ then
            Skip this mention.
        end if
    end for
end for
Construct a wrapper-mention graph G.
Rank the vertices in G by graph random walk.
return ranked list of mentions.
```

5 Experiment Settings and Results

In this work, the experiments are designed to evaluate the effectiveness of the proposed filtering strategy in learning semantic classes. The proposed approach is compared with a baseline counterpart which runs *without* the filtering mechanism. A noted impediment (McIntosh and Curran, 2009) to a fair evaluation is that the same seeds used to initiate the algorithms can cause different algorithms to generate diverse lexicons which vary greatly in precision. This makes evaluation unreliable — seeds which perform well on one algorithm can perform poorly on another.

To conduct a fair comparison, we adopt a bagging approach which resembles to the one used by McIntosh and Curran (2009) to aggregate the results over 30 runs for each algorithm. In each run, our system uses three seeds randomly selected from a set of ten prepared instances. The resulting 30 lexicons are then merged by a weighted vot-

ing function which is based on two hypotheses of the ranked lexicons: firstly, the candidates ranked higher in lexicons are considered more reliable; secondly, this ranking evidence can be better supported if a run extracted more candidates. More specifically, for each run, the candidate extracted with the r-th rank is assigned with a score by

$$S_c(r) = \frac{m}{log(r)}$$

in which m is the number of candidates extracted in this run. This score is averaged over all lexicons in which the candidate is listed.

In this work, we evaluated the algorithms on nine semantic classes as listed in Table 1. In each run, the systems operate on 50 web pages retrieved from Google by submitting a query containing three seed instances. To further test the ability of the filtering mechanism, we also conducted experiments which added another 500 web pages gathered from `reddit.com`[2] into the original retrieval to simulate a noisy source.

It can be observed from the results that the proposed approach consistently outperforms the baseline system and maintains the precision better as the evaluation includes more instances. Moreover, the results of experiments that includes noise web pages further demonstrate the effectiveness of the filtering mechanism. In those experiments, the baseline system often exhibits a performance drop compared to the results obtained without the additional noise web pages. On the other hand, the proposed approach displays a consistent performance regardless whether there is additional noise or not. This shows that the filtering mechanism does help the overall performance and is better in preventing erroneous extractions.

Class	Method	Noise	Precision @			
			25	50	75	100
Actor Actress	Baseline	no	0.76	0.84	0.87	0.84
		with	0.76	0.68	0.79	0.83
	Proposed	no	1.0	0.98	0.99	0.97
		with	1.0	0.98	0.99	0.97
Animal	Baseline	no	0.84	0.72	0.69	0.66
		with	0.68	0.42	0.36	0.27
	Proposed	no	0.88	0.86	0.77	0.77
		with	0.84	0.82	0.76	0.75
Kitchen Item	Baseline	no	0.76	0.68	0.56	0.48
		with	0.68	0.56	0.47	0.47
	Proposed	no	0.88	0.72	0.72	0.72
		with	0.88	0.76	0.68	0.70
Outdoor Activity	Baseline	no	0.92	0.70	0.56	0.47
		with	0.80	0.66	0.56	0.44
	Proposed	no	0.96	0.94	0.87	-
		with	0.96	0.96	0.87	-
Philosopher	Baseline	no	0.76	0.68	0.60	0.63
		with	0.60	0.42	0.41	0.42
	Proposed	no	1.0	0.94	0.91	0.86
		with	1.0	0.94	0.91	0.86
Portland Attraction	Baseline	no	0.76	0.58	0.55	0.54
		with	0.76	0.58	0.47	0.43
	Proposed	no	0.88	0.94	0.87	0.83
		with	0.88	0.94	0.87	0.83
Software Dev. Tool	Baseline	no	0.88	0.82	0.73	0.73
		with	0.84	0.74	0.71	0.69
	Proposed	no	0.88	0.84	0.84	0.82
		with	0.88	0.84	0.83	0.83
Shape	Baseline	no	0.84	0.6	0.51	0.4
		with	0.56	0.42	0.35	0.32
	Proposed	no	0.88	0.82	0.77	0.69
		with	0.88	0.84	0.79	0.69
Politician	Baseline	no	0.8	0.78	0.81	0.85
		with	0.72	0.72	0.69	0.74
	Proposed	no	0.96	0.96	0.92	0.84
		with	0.96	0.96	0.92	0.85

Table 1: Empirical results compared with the baseline system on the precision of the instances extracted. The experiments include settings with and without the addition of 500 noise web pages collected randomly from `reddit.com`.

6 Conclusion and Future Works

In this work, we have proposed a weakly-supervised approach for extracting instances of semantic classes. The proposed approach utilizes a compression model for filtering possibly irrelevant mentions, and uses a graph ranking algorithm for sorting the extraction.

This study focused primarily on assessing the effectiveness of using PPM model for evaluating the contextual evidence, and thus we use only very simple wrappers. Our approach can potentially be extended to adopt more elaborated patterns such as Kozareva et al. (2008) and Xu et al. (2007). In addition, as our future work, we plan to apply this method to other languages such as Japanese and Chinese.

Acknowledgments

This research was supported in part by the National Science Council under grant NSC99-3112-B-001-005, the Academia Sinica Investigator Award 95-02 and the research center for Humanities and Social Sciences under grant IIS-50-23.

[2] `www.reddit.com` , a social bookmark website. 500 URLs gathered in May, 25th, 2011.

References

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, pages 2670–2676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Timothy C. Bell, John G. Cleary, and Ian H. Witten. 1990. *Text compression*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

William Chang, Patrick Pantel, Ana-Maria Popescu, and Evgeniy Gabrilovich. 2009. Towards intent-driven bidterm suggestion. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 1093–1094, New York, NY, USA. ACM.

Surajit Chaudhuri, Venkatesh Ganti, and Dong Xin. 2009. Exploiting web search to generate synonyms for entities. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 151–160, New York, NY, USA. ACM.

Philipp Cimiano, Aleksander Pivk, Lars S. Thieme, and Steffen Staab. 2004. Learning taxonomic relations from heterogeneous sources of evidence. In *Proceedings of the ECAI 2004 Ontology Learning and Population Workshop*.

John G. Cleary and Ian H. Witten. 1984. Data compression using adaptive coding and partial string matching. *IEEE Transactions on Communications*, 32:396–402.

Zelig Harris. 1970. Distributional structure. In *Papers in Structural and Transformational Linguistics*, pages 775–794. D. Reidel Publishing Company, Dordrecht, Holland.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Conference on Computational Linguistics*, volume 2 of *COLING '92*, pages 539–545, Stroudsburg, PA, USA. Association for Computational Linguistics.

Paul Glor Howard. 1993. *The design and analysis of efficient lossless data compression systems*. Ph.D. thesis, Providence, RI, USA. UMI Order No. GAX94-06956.

Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy. 2008. Semantic class learning from the web with hyponym pattern linkage graphs. In *Proceedings of the 46th Annual Meeting of the Association of Computational Linguistics: Human Language Technologies*, pages 1048–1056, Columbus, Ohio, June. Association for Computational Linguistics.

Joseph F McCarthy and Wendy G Lehnert. 1995. Using decision trees for coreference resolution. In *In Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1050–1055.

Tara McIntosh and James R. Curran. 2009. Reducing semantic drift with bagging and distributional similarity. In *Proceedings of the Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing*, volume 1 of *ACL-IJCNLP '09*, pages 396–404, Stroudsburg, PA, USA. Association for Computational Linguistics.

Marius Paşca. 2007. Weakly-supervised discovery of named entities using web search queries. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, CIKM '07, pages 683–690, New York, NY, USA. ACM.

Patrick Pantel, Eric Crestan, Arkady Borkovsky, Ana-Maria Popescu, and Vishnu Vyas. 2009. Web-scale distributional similarity and entity set expansion. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, volume 2 of *EMNLP '09*, pages 938–947, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of National Conference on Artificial Intelligence*, AAAI-99, pages 474–479.

Ellen Riloff and Jessica Shepherd. 1997. A corpus-based approach for building semantic lexicons. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing*, pages 117–124.

Luis Sarmento, Valentin Jijkuon, Maarten de Rijke, and Eugenio Oliveira. 2007. "more like these": growing entity classes from seeds. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, CIKM '07, pages 959–962, New York, NY, USA. ACM.

W. J. Teahan and John G. Cleary. 1997. Models of english text. In *Proceedings of the Conference on Data Compression*, pages 12–21, Washington, DC, USA. IEEE Computer Society.

Richard C. Wang and William W. Cohen. 2007. Language-independent set expansion of named entities using the web. In *Proceedings of the 7th IEEE International Conference on Data Mining*, volume 0 of *ICDM '07*, pages 342–350, Washington, DC, USA. IEEE Computer Society.

Scott White and Padhraic Smyth. 2003. Algorithms for estimating relative importance in networks. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 266–275, New York, NY, USA. ACM.

Feiyu Xu, Hans Uszkoreit, and Hong Li. 2007. A seed-driven bottom-up machine learning framework for extracting relations of various complexity. In

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 584–591, Prague, Czech Republic, June. Association for Computational Linguistics.