Not All Seeds Are Equal: Measuring the Quality of Text Mining Seeds

Zornitsa Kozareva and Eduard Hovy
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292-6695
{kozareva,hovy}@isi.edu

Abstract

Open-class semantic lexicon induction is of great interest for current knowledge harvesting algorithms. We propose a general framework that uses patterns in bootstrapping fashion to learn open-class semantic lexicons for different kinds of relations. These patterns require seeds. To estimate the *goodness* (the potential yield) of new seeds, we introduce a regression model that considers the connectivity behavior of the seed during bootstrapping. The generalized regression model is evaluated on six different kinds of relations with over 10000 different seeds for English and Spanish patterns. Our approach reaches robust performance of 90% correlation coefficient with 15% error rate for any of the patterns when predicting the *goodness* of seeds.

1 Introduction: What is a Good Seed?

The automated construction of semantically typed lexicons (terms classified into their appropriate semantic class) from unstructured text is of great importance for various kinds of information extraction (Grishman and Sundheim, 1996), question answering (Moldovan et al., 1999), and ontology population (Suchanek et al., 2007). Maintaining large semantic lexicons is a time-consuming and tedious task, because open classes (such as: all singers, all types of insects) are hard to cover completely, and even closed classes (such as: all countries, all large software companies) change over time. Since it is practically impossible for a human to collect such knowledge adequately, many supervised, unsupervised, and semi-supervised techniques have been developed.

All these techniques employ some sort of context to specify the appearance in text of the desired information. This approach is based on the general intuition, dating back at least to the distributional similarity idea of (Harris, 1954), that certain contexts are specific enough to constrain terms or expressions within them to be specific classes or types. Often, the context is a string of words with an empty slot for the desired term(s); sometimes, it is a regular expression-like pattern that includes word classes (syntactic or semantic); sometimes, it is a more abstract set of features, including orthographic features like capitalization, words, syntactic relations, semantic types, and other characteristics, which is the more complete version of the distributional similarity approach.

In early information extraction work, these contexts were constructed manually, and resembled regular expressions (Appelt et al., 1995). More recently, researchers have focused on learning them automatically. Since unsupervised algorithms require large training data and may or may not produce the types and granularities of the semantic class desired by the user, and supervised algorithms may require a lot of manual oversight, semi-supervised algorithms have become more popular. They require only a couple of seeds (examples filling the desired semantic context) to enable the learning mechanism to learn patterns that extract from unlabeled texts additional instances of the same class (Riloff and Jones, 1999; Etzioni et al., 2005; Pasca, 2004).

Sometimes, the pattern(s) learned are satisfactory

enough to need no further elaboration. They are applied to harvest as many additional terms of the desired type as possible (for example, the instance-learning pattern '*<type> such as ?*' introduced in (Hearst, 1992)). More often, the method is applied recursively: once some pattern(s) have been learned, they are used to find additional terms, which are then used as new seeds in the patterns to search for additional new patterns, etc., until no further patterns are found. At that point, the satisfactory patterns are selected and large-scale harvesting proceeds as usual. In an interesting variation of this method, (Kozareva et al., 2008) describe the 'doubly-anchored pattern' (DAP) that includes a seed term in conjunction with the open slot for the desired terms to be learned, making the pattern itself recursive by allowing learned terms to replace the initial seed terms directly: '*<type> such as <seed> and ?*'.

Context-based information harvesting is well understood and has been the focus of extensive research. The core unsolved problem is the selection of seeds. In current knowledge harvesting algorithms, seeds are chosen either at random (Davidov et al., 2007; Kozareva et al., 2008), by picking the top N most frequent terms of the desired class (Riloff and Jones, 1999; Igo and Riloff, 2009), or by asking experts (Pantel et al., 2009). None of these methods is quite satisfactory. (Etzioni et al., 2005) report on the impact of seed set noise on the final performance of semantic class learning, and Pantel et al. observe a tremendous variation in the entity set expansion depending on the initial seed set composition. These studies show that the selection of 'good' seeds is very important. Recently, (Vyas et al., 2009) proposed an automatic system for improving the seeds generated by editors (Pantel et al., 2009). The results show 34% improvement in final performance using the appropriate seed set. However, using editors to select seeds or to guide their seed selection process is expensive and therefore not always possible. Because of this, we address in this paper two questions: "*What is a good seed?*" and "*How can the goodness of seeds be automatically measured without human intervention?*".

The contributions of this paper are as follows:

- First, we use recursive patterns to automatically learn seeds for open-class semantic lexicons.
- Second, we define what the 'goodness' of a seed term is. Then we introduce a regression model of seed quality measurement that, after a certain amount of training, automatically estimates the goodness of new seeds with above 90% accuracy for bootstrapping with the given relation.
- Next, importantly, we discover that training a regression model on certain relations enables one to predict the goodness of a seed even for other relations that have never been seen before, with an accuracy rate of over 80%.
- We conduct experiments with six kinds of relations and more than 10000 automatically harvested seed examples in both English and Spanish.

The rest of the paper is organized as follows. In the next section, we review related work. Section 3 describes the recursive pattern bootstrapping (Kozareva et al., 2008). Section 4 presents our seed quality measurement regression model. Section 5 discusses experiments and results. Finally, we conclude in Section 6.

2 Related Work

Seeds are used in automatic pattern extraction from text corpora (Riloff and Jones, 1999) and from the Web (Banko, 2009). Seeds are used to harvest instances (Pasca, 2004; Etzioni et al., 2005; Kozareva et al., 2008) or attributes of a given class (Paşca and Van Durme, 2008), or to learn concept-specific relations (Davidov et al., 2007), or to expand already existing entity sets (Pantel et al., 2009). As mentioned above, (Etzioni et al., 2005) report that seed set composition affects the correctness of the harvested instances, and (Pantel et al., 2009) observe an increment of 42% precision and 39% recall between the best and worst performing seed sets for the task of entity set expansion.

Because of the large diversity of the usage of seeds, there has been no general agreement regarding exactly how many seeds are necessary for a given task. According to (Pantel et al., 2009) 10 to 20 seeds are a sufficient starting set in a distributional similarity model to discover as many new correct instances as may ever be found. This observation differs from the claim of (Paşca and Van Durme, 2008) that 1 or 2 instances are sufficient to discover thousands of instance attributes. For some

pattern-based algorithms one to two seeds are sufficient (Davidov et al., 2007; Kozareva et al., 2008), some require ten seeds (Riloff and Jones, 1999; Igo and Riloff, 2009), and others use a variation of 1, 5, 10 to 25 seeds (Talukdar et al., 2008).

As mentioned, seed selection is not yet well understood. Seeds may be chosen at random (Davidov et al., 2007; Kozareva et al., 2008), by picking the most frequent terms of the desired class (Riloff and Jones, 1999; Igo and Riloff, 2009), or by asking humans (Pantel et al., 2009). The intuitions for seed selection that experts develop over time seem to prefer instances that are neither ambiguous nor too frequent, but that at the same time are prolific and quickly lead to the discovery of a diverse set of instances. These criteria are vague and do not always lead to the discovery of good seeds. For some approaches, infrequent and ambiguous seeds are acceptable while for others they lead to deterioration in performance. For instance, the DAP (Kozareva et al., 2008) performance is not affected by the ambiguity of the seed, because the class and the seed in the pattern mutually disambiguate each other, while for the distributional similarity model of (Pantel et al., 2009), starting with an ambiguous seed leads to 'leakage' and the harvesting of non-true class instances. (Kozareva et al., 2008) show that for the closed class *country*, both high-frequency seeds like *USA* and low-frequency seeds like *Burkina Faso* can equally well yield all remaining instances. An open question to which no-one provides an answer is whether and which high/low frequency seeds can yield all instances of large, open classes like people or singers.

3 Bootstrapping Recursive Patterns

There are many algorithms for harvesting information from the Web. The main objective of our work is not the creation of a new algorithm, but rather determining the effect of seed selection on the general class of recursive bootstrapping harvesting algorithms for the acquisition of semantic lexicons for open class relations. For our experiments, since it is time-consuming and difficult for humans to provide large sets of seeds to start the bootstrapping process, we employ the recursive DAP mechanism introduced by (Kozareva et al., 2008) that produces seeds on its own.

The algorithm starts with a *seed* of type *class* which is fed into the doubly-anchored pattern '$<$*class*$>$ *such as* $<$*seed*$>$ *and* *' and learns in the * position new instances of type *class*. The newly learned instances are then systematically placed into the position of the *seed* in the DAP pattern, and the harvesting process is repeated until no new instances are found. The general framework is as follows:

1. Given:
   a language L={English, Spanish}
   a pattern P_i={e.g., [verb prep, noun, verb]}
   a seed $seed$ for P_i
2. Build a query in DAP-like fashion for P_i using template T_i of the type 'class such as $seed$ and *', '* and $seed$ verb prep', '* and $seed$ noun', '* and $seed$ verb'
3. submit T_i to Yahoo! or another search engine
4. extract instances occupying the * position
5. take instances from 4. and go to 2.
6. repeat steps 2–5 until no new instances are found

At the end of bootstrapping, the harvested instances can be considered to be seeds with which the bootstrapping procedure could have been initiated. We can now compare any of them to study their relative 'goodness' as bootstrapping seeds.

4 Seed Quality Measurement

4.1 Problem Formulation

We define our task as:

Task Definition: Given a seed and a pattern in a language (say English or Spanish), (1) use the bootstrapping procedure to learn instances from the Web; (2) build a predictive model to estimate the 'goodness' of seeds (whether generated by a human or learned) .

Given a desired semantic class, a recursive harvesting pattern expressing its context, and a seed term for use in this pattern, we define the 'goodness' of the seed as consisting of two measures:

- the *yield*: the total number of instances learned, not counting duplicates, until the bootstrapping procedure has run to exhaustion;
- the *distance*: the number of iterations required by the process to reach exhaustion.

Our approach is to build a model of the behavior of many seeds for the given pattern. Any new seed can then be compared against this model, once its basic characteristics have been determined, and its yield and distance estimates produced. In order to determine the characteristics of the new seed, it first has to be employed in the pattern for a small number of iterations. The next subsection describes the regression model we employ in our approach.

4.2 Regression Model

Given a seed s, we seek to predict the yield g of s as defined above. We do this via a parametrized function f:$\hat{g} = f(s; w)$, where $w \in R^d$ are the weights. Our approach is to learn w from a collection of N training examples $\{< s_i, g_i >\}_{i=1}^{N}$, where each s_i is a seed and each $g_i \in R$.

Support vector regression (Drucker et al., 1996) is a well-known method for training a regression model by solving the following optimization problem:

$$\min_{w \in R^s} \frac{1}{2}||w||^2 + \frac{C}{N}\sum_{i=1}^{N} \underbrace{\max(0, |g_i - f(s_i; w)| - \epsilon)}_{\epsilon\text{-insensitive loss function}}$$

where C is a regularization constant and ϵ controls the training error. The training algorithm finds weights w that define a function f minimizing the empirical risk.

Let h be a function from seeds into some vector-space representation $\subseteq R^d$, then the function f takes the form: $f(s; w) = h(s)^T w = \sum_{i=1}^{N} \alpha_i K(s, s_i)$, where f is re-parameterized in terms of a polynomial kernel function K with dual weights α_i. K measures the similarity between two seeds. Full details of the regression model and its implementation are beyond the scope of this paper; for more details see (Schölkopf and Smola, 2001; Smola et al., 2003). In our experimental study, we use the freely available implementation of SVM in Weka (Witten and Frank, 2005).

To evaluate the quality of our prediction model, we compare the actual yield of a seed with the predicted value obtained, and compute the correlation coefficient and the relative absolute error.

5 Experiments and Results

5.1 Data Collection

We conducted an exhaustive evaluation study with the open semantic classes *people* and *city*, initiated with the seeds *John* and *London*. For each class, we submitted the DAP patterns as web queries to Yahoo!Boss and retrieved the top 1000 web snippets for each query, keeping only unique instances. In total, we collected 1.5GB of snippets for people and 1.9GB of snippets for cities. The algorithm ran until complete exhaustion, requiring 19 iterations for people and 12 for cities. The total number of unique harvested instances was 3798 for people and 5090 for cities. We used all instances as seeds and instantiated for each seed the bootstrapping process from the very beginning. This resulted in 3798 and 5090 separate bootstrapping runs for people and cities respectively. For each seed, we recorded the total number of instances learned at the end of bootstrapping, the number of iterations, and the number of unique instances extracted on each iteration. After the harvesting part terminated, we analyzed the connectivity / bootstrapping behavior of the seeds, and produced the regression model.

5.2 Seed Characteristics

For many knowledge harvesting algorithms, the selection of a non-ambiguous seeds is of great importance. In the DAP bootstrapping framework, the ambiguity of the seed is eliminated as the *class* and the *seed* mutually disambiguate each other. Of great importance to the bootstrapping algorithm is the selection of a seed that can yield a large number of instances and can keep the bootstrapping process energized.

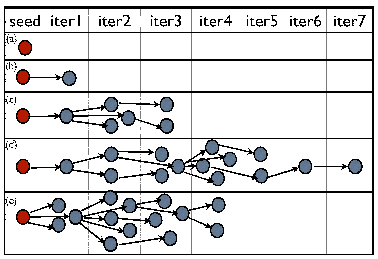

Figure 1: Seed Connectivity

Figure 1 shows the different kinds of seeds we found on analyzing the results of the bootstrapping process. Based on the yield learned on each iteration, we identify four major kinds of seeds: **hermit**, **one-step**, **mid**, and **high** connectors. In the figure, seed (a) is a hermit because it does not discover other instances. Seed (b) is a one-step connector as it discovers instances on the first iteration but

then becomes inactive. Seeds (d) and (e) are high connectors because they find a rich population of instances. Seed (c) is a mid connector because it has lower yield than (d) and (e), but higher than (a) and (b).

Table 1 shows the results of classifying the 3798 people and 5090 city seeds into the four kinds of seed. The majority of the seeds for both patterns are hermits, from 23 to 41% are high connectors, and the rest are one-step and mid connectors. For each kind of seed, we also show three examples.

people such as X and *		examples
#hermit	2271 (60%)	Leila, Anne Boleyn, Sophocles
#one-step	329 (9%)	Helene, Frida Kahlo, Cornelius
#mid	315 (8%)	Brent, Ferdinand, Olivia
#high	883 (23%)	Diana, Donald Trump, Christopher
cities such as X and *		examples
#hermit	2393 (47%)	Belfast, Najafabad, El Mirador
#one-step	406 (8%)	Durnstein, Wexford, Al-Qaim
#mid	207 (4%)	Bialystok, Gori, New Albany
#high	2084 (41%)	Vienna, Chicago, Marrakesh

Table 1: Connectivity-based Seed Classification.

This study shows that humans are very likely to choose non-productive seeds for bootstrapping: it is difficult for a human to know a priori that a name like Diana will be more productive than Leila, Helene, or Olivia.

Another interesting characteristic of a seed is the speed of learning. Some seeds, such as (e), extract large quantity of instances from the very beginning, resulting in fewer bootstrapping iterations, while others, such as (d), spike much later, resulting in more. In our analysis, we found that some high connector seeds of the people pattern can learn the whole population in 12 iterations, while others require from 15 to 20 iterations. Figure 2 shows the speed of learning of ten high connector seeds for the *people* pattern. The y axis shows the number of unique instances harvested on each iteration. Intuitively, a good seed is the one that produces a large *yield* of instances in short *distance*. Thus the 'goodness' of seed (e) is better than that of seed (d).

As shown in Figure 2, for each seed, we observe a single hump that corresponds to the point in which a seed generates the maximum number of instances. The peak occurs on different iterations because it is dependent both on the yield learned with each iteration and the total distance, for each seed. The occurrence of a single hump reveals regularity in the connectivity behavior of seeds, and is discussed in the Conclusion. We model this behavior as features in our regression model and use it to measure the quality of new seeds. The next subsection explains the features of the regression model and the experimental results obtained.

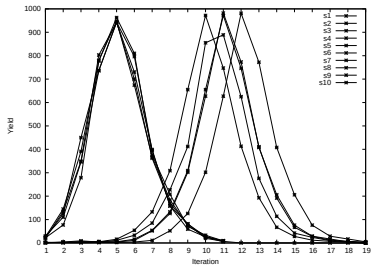

Figure 2: Seed Learning Speed

5.3 Predicting the Goodness of Seeds

Building a pattern specific model: For each pattern, we build N different regression models, where N corresponds to the total number of bootstrapping iterations of the pattern. For regression model R_i, we use the yield of a seed from iterations 1 to i as features. This information is used to model the activity of the seed in the bootstrapping process and later on to predict the extraction power of new seeds. For example, in Figure 1 on the first iteration seeds (b), (c), and (d) have the same low connectivity compared to seed (e). As bootstrapping progresses, seed (d) reaches productive neighbors that discover more instances, while seeds (b) and (c) become inactive. This example shows that the yield in the initial stage of bootstrapping is not sufficient to accurately predict the quality of the seeds. Since we do not know exactly how many iterations are necessary to accurately determine the 'goodness' of seeds, we model the yield learned on each iteration by each seed and subsequently include this information in the regression models.

The yield of a seed s_k at iteration i is computed as $yield(s_k)_i = \sum_{m=1}^{n}(s_m)$, where n is the total number of unique instances s_m harvested on iteration i. $Yield(s_k)_i$ is high when s_k discovers a large number of instances (new seeds), and small otherwise. For hermit seeds, $yield$=0 at any iteration, because the seeds are totally isolated and do not discover other

instances (seeds). For example, when building the second regression model R_2 using seeds (d) and (e) from Figure 1, the feature values corresponding to each seed in R_2 are: $yield(s_d)_1$=1 and $yield(s_d)_2$=2 for seed (d), and $yield(s_e)_1$=3 and $yield(s_e)_2$=5 for seed (e).

Results: Figure 3 shows the correlation coefficients (cc) and the relative absolute errors of each regression model R_i for the *people* and *city* patterns. The results are computed over ten-fold cross validation of the 3798 people and 5090 city seeds. The x axis shows the regression model R_i,. The y axis in the two upper graphs shows the correlation coefficient of the predicted and the actual total yield of the seeds using R_i, and in the two lower graphs, the y axis shows the error rate of each R_i.

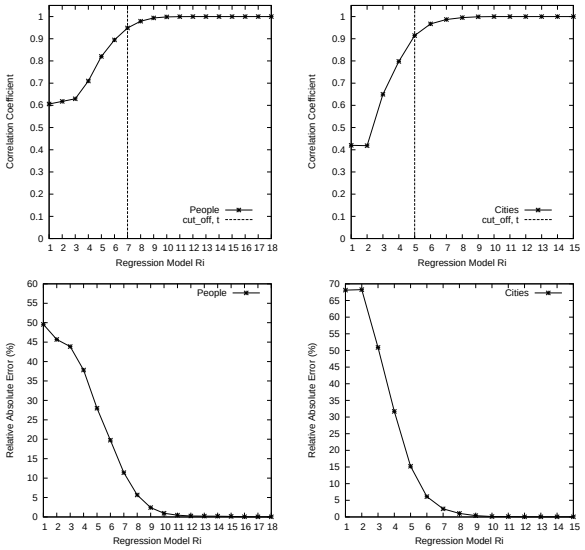

Figure 3: Regression for People and City.

We consider as a baseline model the regression R_1 which uses only the yield of the seeds on first iteration. The prediction of R_1 has cc=0.6 with 50% error for people and cc=0.4 with 70% error for cities. These results confirm our previous observation that the quality of the seeds cannot be accurately measured in the very beginning of bootstrapping. However, by the ninth iteration, the regression models for people and cities reach cc=1.0 with 5% error rate. To make such an accurate prediction, the model uses around one half of all bootstrapping iterations—generally, just past the hump in Figure 2, once the yield starts dropping.

Often in real applications or when under limited resources (e.g., a fixed amount of Web queries per day), running half the bootstrapping iterations is not feasible. This problem can be resolved by employing different stopping criteria, at the cost of lower cc and greater error. For example, one cut-off point can be the (averaged) iteration number of the hump for the given pattern. For people, the average hump occurs at the seventh iteration, and for the city at the fifth iteration. At this point, both patterns have a cc=0.9 with 15% error rate. An alternative stopping point can be the fourth iteration, where cc=0.7–0.8 with 35% error.

Overall, our study shows that it is possible to model the behavior of seeds and use it to accurately predict the 'goodness' of previously unseen seeds. The results obtained for both *people* and *city* patterns are very promising. However, a disadvantage of this regression is that it requires training over the whole extent of the given pattern. Also, each regression model is specific to the particular pattern it is trained over. Next, we propose a generalized regression model which surmounts the problem of training pattern-specific regression models.

5.4 Generalized Model for Goodness of Seeds

We built a generalized regression model (RG) combining evidence from the people and city patterns. We generated the features of each model as previously described in Section 5.3. From each pattern, we randomly picked 1000 examples which resulted in 30% of the people and 20% of the city seeds. We used these seed examples to train the RG_i models. In total, we built 15 RG_i, which is the maximum number of overlapping iterations between the two patterns. We tested our RG model with the remaining 2798 people and 4090 city seeds.

Figure 4 shows the results of the RG_i models for the people and city patterns. In the first two iterations, the predictions of the RG model are poorer compared to the pattern-specific regression. On the fourth iteration, both models have cc=0.7 and 0.8 for the people and city patterns respectively. The error rates of the generalized model are 41% and 35% for people and city, while for the pattern-specific model the errors are 37% and 32%. The early iterations show a difference of around 4% in the error rate of the two models, but around the ninth iteration both models have comparable results.

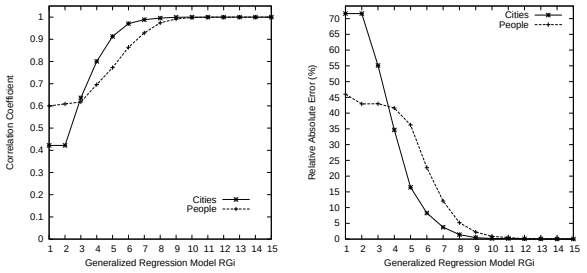

Figure 4: Generalized Regression for People and City.

This study shows that it is possible to combine evidence from two patterns harvesting different semantic information to predict accurately the behavior of unseen seed examples for either of the two patterns.

5.5 Evaluating the Generalized Model on Different Languages and Kinds of Patterns

So far, we have studied the performance of the generalized seed quality prediction method for specific patterns in English. However, the connectivity behavior of the seeds might change for other languages and kinds of patterns, making the generalized model impractical to use in such cases. To verify this, we evaluated the generalized model (RG) from Section 5.4 with the people and city patterns in Spanish (‘ *gente como X y* *’ and ‘ *ciudades como X y* *’), as well as with two new kinds of patterns (‘ ** and X fly to*’ and ‘ ** and X work for*’[1]). For each pattern, we ran the bootstrapping process from Section 3 until exhaustion and collected all seeds.

First, for each pattern we studied the connectivity behavior of the seeds. Table 2 shows the obtained results. The distribution is similar to the seed distribution for the English people and cities patterns. Although the total number of harvested instances (i.e., seeds) is different for each pattern, the proportion of hermits to other seeds remains larger. From 20% to 37% of the seeds are high connectors, and the rest are one-step and mid connectors. This analysis shows that the connectivity behavior of seeds across different languages and patterns is similar, at least for the examples studied. In addition to the seed analysis, we show in the table the total number of bootstrapping iterations for each pattern. The ‘*work for*’ and ‘*fly to*’ patterns run for a longer distance compared to the other patterns. While for the majority of the patterns the hump is observed on the fifth or seventh iteration, for these two patterns the average peak is observed on the fifteenth.

[1]The X indicates the position of the seed and (*) corresponds to the instances learned during bootstrapping.

	gente como X y	ciudades como X y
#hermit	318 (56%)	1061 (51%)
#one-step	58 (10%)	150 (8%)
#mid	79 (14%)	79 (4%)
#high	117 (20%)	795 (38%)
tot#iter	20	16
	and X fly to	and X work for
#hermit	389 (45%)	1262 (48%)
#one-step	87 (9%)	238 (9%)
#mid	75 (8%)	214 (8%)
#high	322 (37%)	922 (35%)
tot#iter	26	33

Table 2: Seed Classification for Spanish and Verb-Prep Patterns.

Second, we test the RG_i models from Section 5.4, which were trained on people and cities, to predict the total yield of the seeds in the new patterns. Figure 5 shows the correlation coefficient and the relative absolute error results of each pattern for RG_i.

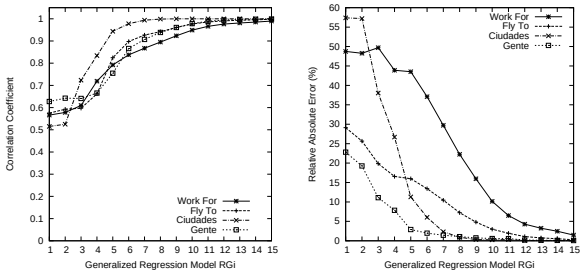

Figure 5: Generalized Regression for Different Languages and Patterns.

Interestingly, we found that our generalized method has consistent performance across the different languages and patterns. On the twelfth iteration, the model is able to predict the ‘goodness’ of seeds with cc=1.0 and from 0.4% to 8.0% error rate. Around the fifth and sixth iterations, all patterns reach cc=0.8 with error of 5% to 15%. The higher error bound is for patterns like ‘work for’ and ‘fly to’ which run for a longer distance. This experimental study confirms the robustness of our generalized model which is trained on the behavior of seeds from one kind of pattern and tested with seeds in different languages and on completely different kinds of patterns.

6 Conclusions and Future Work

It would, a fortiori, seem impossible to estimate the goodness of a seed term used in a recursive bootstrapping pattern for harvesting information from the web. After all, its eventual total yield and distance depend on the cumulation of the terms produced in each iteration of the bootstrapping, and there are no external constraints or known web language structure to be exploited.

We have shown that it is possible to create, using regression, a model of the grown behavior of seeds for a given pattern, and fitting it with an indication of a new seed's growth (considering its grown behavior in a limited number of bootstrapping iterations) in order to obtain a quite reliable estimate of the new seed's eventual yield and distance.

Going further, we are delighted to observe that the regularity of the single-hump harvesting behavior makes it possible to learn a regression model that enables one to predict, with some accuracy, both the yield and the distance of a new seed, even when the pattern being considered is not yet seen. All that is required is the indication of the seed's growth behavior, obtained through a number of iterations using the pattern of interest.

Our ongoing analysis takes the following approach. Let T_i be the set of all new terms (terms not yet found) harvested during iteration i. Then $T_0 = \{t_{0,1}\}$, just the initial seed term. Let $NY(t_{i,j})$ be the novel yield of term $t_{i,j}$, that is, the number of as yet undiscovered terms produced by a single application of the pattern using the term $t_{i,j}$. Notice that bootstrapping ceases when for some $i = d$ (the distance), $\sum_j NY(t_{d,j}) = 0$. Since the total number of terms that can be learned, $\sum_{i=0}^{d} \sum_j NY(t_{i,j}) = N$, is finite and fixed, there are exactly three alternatives for the growth of the NY curve when it is shown summed over each iteration: (i) either $\sum_j NY(t_{i,j}) = \sum_j NY(t_{i+1,j})$ and there is no larger NY sum for any iteration; or (ii) $\sum_j NY(t_{i,j})$ grows to a maximal value for some iteration $i = m$ and then decreases again; or (iii) $\sum_j NY(t_{i,j})$ reaches more than one locally maximal value at different iterations. The first case, in which exactly the same number of new terms is harvested every iteration for several or all iterations, would require that each new term once learned yields precisely and only one subsequent new term, or that the number of hermits is exactly balanced by the NY of one or more of the other terms in that iteration. This situation is so unlikely as to be dismissed outright. Case (ii), in which there is a single hump, appears to be how text is written on the web, as shown in Figure 2. Case (iii), the multi-hump case, would require that the terms be linked in semi-disconnected 'islands', with a relatively much smaller inter-island connectivity than intra-island one. Given our studies, it appears that language on the web is not organized this way, at least not for the patterns we studied. However, it is not impossible: this two-hump case would have to have occurred in (Kozareva et al., 2008) when the ambiguous seed term *Georgia* was used in the DAP '*states such as Georgia and* *', where initially the US states were harvested but, at some point, the learned term Georgia also initiated harvesting of the ex-USSR states. Such 'leakage' into a new semantic domain requires not only ambiguity of the seed but also parallel ambiguity of the class term, which is highly unlikely as well.

Accepting case (ii), therefore, we postulate that for any (or all regular) patterns there is some iteration m in which $\sum_j NY(t_{m,j})$ is maximal. The question is how rapidly the summed NY curve approaches it and then abates again. This depends on the out-degree connectivity of terms overall. In the population of N terms for a given semantic pattern, is the distribution of out-degrees Poisson (or Zipfian), or is it normal (Gaussian)? In the former case, there will be a few high-degree connector terms and a large number (the long tail) of one-step and hermit terms; in the latter, there will be a small but equal number of low-end and high-end connector terms, with the bulk of terms falling in the mid-connector range. One direction of our ongoing work is to determine this distribution, and to empirically derive its parameters. It might be possible to discover some interesting regularities about the (preferential) uses of terms within semantic domains, as reflected in term network connectivity.

Although not all seeds are equal, it appears to be possible to treat them with a single regression model, regardless of pattern, to predict their 'goodness'.

Acknowledgments: This research was supported by NSF grant IIS-0705091.

References

Douglas E. Appelt, Jerry R. Hobbs, John Bear, David Israel, Megumi Kameyama, Andy Kehler, David Martin, Karen Myers, and Mabry Tyson. 1995. SRI International FASTUS system MUC-6 test results and analysis. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 237–248.

Michele Banko. 2009. Open information extraction from the web. In *Ph.D. Dissertation from University of Washington*.

Dmitry Davidov, Ari Rappoport, and Moshel Koppel. 2007. Fully unsupervised discovery of concept-specific relationships by web mining. In *Proc. of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 232–239, June.

Harris Drucker, Chris J.C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. 1996. Support vector regression machines. In *Advances in NIPS*, pages 155–161.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: an experimental study. *Artificial Intelligence*, 165(1):91–134, June.

Ralph Grishman and Beth Sundheim. 1996. Message understanding conference-6: a brief history. In *Proceedings of the 16th conference on Computational linguistics*, pages 466–471.

Zellig S. Harris. 1954. Distributional structure. *Word*, 10:140–162.

Marti Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. of the 14th conference on Computational linguistics*, pages 539–545.

Sean Igo and Ellen Riloff. 2009. Corpus-based semantic lexicon induction with web-based corroboration. In *Proceedings of the Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics*.

Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy. 2008. Semantic class learning from the web with hyponym pattern linkage graphs. In *Proceedings of ACL-08: HLT*, pages 1048–1056.

Dan I. Moldovan, Sanda M. Harabagiu, Marius Pasca, Rada Mihalcea, Richard Goodrum, Roxana Girju, and Vasile Rus. 1999. Lasso: A tool for surfing the answer net. In *TREC*.

Marius Paşca and Benjamin Van Durme. 2008. Weakly-supervised acquisition of open-domain classes and class attributes from web documents and query logs. In *Proceedings of ACL-08: HLT*.

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*, pages 938–947, August.

Marius Pasca. 2004. Acquisition of categorized named entities for web search. In *Proc. of the thirteenth ACM international conference on Information and knowledge management*, pages 137–145.

Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI '99/IAAI '99: Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*.

Bernhard Schölkopf and Alexander J. Smola. 2001. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning)*. The MIT Press.

Alex J. Smola, Bernhard Schlkopf, and Bernhard Sch Olkopf. 2003. A tutorial on support vector regression. Technical report, Statistics and Computing.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 697–706.

Partha Pratim Talukdar, Joseph Reisinger, Marius Pasca, Deepak Ravichandran, Rahul Bhagat, and Fernando Pereira. 2008. Weakly-supervised acquisition of labeled class instances using graph random walks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2008*, pages 582–590.

Vishnu Vyas, Patrick Pantel, and Eric Crestan. 2009. Helping editors choose better seed sets for entity set expansion. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM*, pages 225–234.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, second edition.