Building a Bilingual Dictionary with Scarce Resources: A Genetic Algorithm Approach

Benjamin Han
Language Technologies Institute, Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3702, USA
benhdj@cs.cmu.edu

Abstract

Current corpus-based machine translation systems usually require significant amount of parallel text to build a useful bilingual dictionary for translation. To alleviate this data dependency I propose a novel approach based on genetic algorithms to improve translations by fusing different linguistic hypotheses. A preliminary evaluation is also reported.

Introduction

Most of the current corpus-based machine translation systems rely on statistical methods to extract bilingual dictionaries. While these methods, such as [Kay and Röscheisen 1993], K-vec and DK-vec [Fung and McKeown 1994, 1997], and [Brown 1997], presume little or no prior knowledge of source languages, they require significant amount of parallel text to build an accurate bilingual dictionary [Jones and Somers 1995] [Somers and Ward 1996] [Haruno and Yamazaki 1996]. The requirement makes these approaches less desirable when little data of the source languages could be obtained.

The lack of parallel text presents one aspect of the data scarcity problem [Al-Onaizan 2000]. Being able to solve the problem has both theoretical and practical values. On the one hand an effective approach could shed light on the theoretical framework required for building general lexical acquisition systems, on the other it could facilitate accessing information expressed in minority or indigenous languages.

Obviously the problem cannot be remedied without help. As proposed in [Kumano and Hirakawa 1994], [Utsuro et al. 1994], [Haruno and Yamazaki 1996] and [Brown 1999], incorporating more linguistic knowledge could be a promising solution. In this paper it has motivated the proposal of a novel approach based on *genetic algorithms* (GA) [Holland 1975, Goldberg 1989] as a way to fuse different linguistic hypotheses. The paper is organized as follows: in Section 1 a view of translating by optimization together with the two working hypotheses are introduced. Section 2 presents the details of the algorithm, and Section 3 reports the preliminary evaluation. The paper is then concluded with Future Works.

1 Translating by Optimization

Let f: $W_1 \times W_2$ be a translation mapping, where W_1/W_2 represents the words of the source/target language (L1/L2), a good translation f should maximize an objective score $S(f)$. Viewing translation as an optimization problem allows a straightforward incorporation of different information. In this paper the score function S is determined based on the two proposed linguistic hypotheses, namely, the similarity of *locality* and *part-of-speech* (POS) distributions across the two languages. More specifically, the score function over a translation f is defined as

$$S(f) = \sum_i \lambda \cdot S_{\mathrm{L}}(f, s_i) + (1-\lambda) \cdot S_{\mathrm{P}}(f, s_i) \ldots (1)$$

which is the sum of a linear combination of $S_{\mathrm{L}}(f, s_i)$ and $S_{\mathrm{P}}(f, s_i)$ – the scores contributed by the locality and the POS hypotheses, respectively, over an L1 sentence s_i[1]. It is worth noting that for different language pairs, e.g., from a case-marking language (e.g. Russian) to a configurational language (e.g. English), the

[1] λ is a constant set to 0.5 in the evaluation.

hypotheses used here might not be appropriate and new hypotheses of different types must be proposed due to the distinct natures of the linguistic mappings between the languages.

The two hypotheses are described below.

1.1 Locality Hypothesis

The locality hypothesis stipulates that a good translation f should map two words in an L1 sentence into a pair of words with similar word distance in the L2 sentence. Let w_i, $w_j \in W_1$, a preferable f has the following property:

$$\frac{Dist(w_i, w_j)}{Dist(f(w_i), f(w_j))} \approx 1 \qquad \ldots (2)$$

where $Dist(.,.)$ denotes the distance between two words. Note it does not presume a strict word order similarity since only the relative distances of words are taken into account.

For an L1 sentence with n words Equ. (2) implies a $O(n^2)$ time for evaluating f over the sentence. To save time an approximation is adopted where only every k+1-th word in s_i is considered. The score S_L is then formulated as[2]

$$S_L(f, s_i) = \frac{\sum_{j=1}^{|s_i|-k} \frac{\kappa_i - |k - Dist_{\min}(f(w_j), f(w_{j+k}))|}{\kappa_i}}{|s_i| - k} \qquad \ldots (3)$$

where $Dist_{\min}(.,.)$ returns the minimum distance between a pair of words, $\kappa_i = \max(k, |t_i|-k)$ is a normalizing constant, and t_i is the translated sentence in L2. This is simply an average of the relative differences of the word distances between the L1 sentence s_i and its translation t_i in L2.

1.2 Part-of-speech (POS) Similarity Hypothesis

The second hypothesis adopted is that a good translation should preserve the POS distribution. Let the POS distribution of a word w, $D(w)$, be a vector of POS confidence values, i.e.,

$D(w) = \{c_t \mid 0 \le c_t \le 1$ is the confidence that w is of POS $t\}$

The POS distribution of a sentence s can then be defined as

$$D(s) = \sum_{w_i \in s} D(w_i)$$

And the score S_P is defined as

$$S_P(f, s_i) = 1 - Dist_P(D(f(s_i)), D(t_i)) \qquad \ldots (4)$$

where $Dist_P(D_1, D_2) = \text{acos}\left(\frac{D_1 \cdot D_2}{|D_1||D_2|}\right) \Big/ \frac{\pi}{2}$ is the normalized angle between the two distribution vectors. Note since both $f(s_i)$ and t_i are in L2, only the POS information of L2 is required. For *translation for assimilation* this is usually not a problem since in these scenarios the POS information of L2 is usually available[3].

2 A Genetic Algorithm Approach

The GA-based approach is proposed to solve the translation optimization problem based on the following observations: (a) the search space is huge and not well understood, and (b) the problem satisfies the "building block hypothesis" in that a good translation possesses many useful smaller "building blocks" which can be exchanged with the others to potentially yield a better translation [Goldberg 1989].

Until recently GA has not been widely used in the field of computational linguistics. Several works have been reported for grammar induction, robust parsing, anaphora resolution and morphological analysis [Losee 1995] [Rosé 1998] [Orasan *et al* 2000] [Kazakov and Manandhar 2000]. To the author's knowledge there has not been any report on using GA-based techniques to extract bilingual dictionaries.

2.1 Algorithm Outline

The algorithm applies various GA operators on a population of *solutions* to maximize the objective function $S(f)$. A solution (translation mapping) is encoded by a vector v where $v_i = j$ denotes the translation $f(w_i) = x_j$. A sketch of the

[2] For the current implementation k=1.

[3] For the current implementation Brill's Transformation-based POS Tagger [Brill 1992] is used for L2 (English).

proposed *steady-state genetic algorithm* with *least-fit-deletion* strategy is described below:

1. For each $w_i \in W_1$, *iteratively* compute a *candidate set* $CS(w_i)$ containing the possible translations with their respective confidence values (described below).
2. Initialize a population of solutions by randomly picking a candidate translation for each w_i according to the confidence distributions[4], and adding a solution containing the candidates with the highest confidence values[5]. Evaluate all of the solutions according to Equ. (1), (3) and (4) and linearly scale the scores into the fitness values.
3. Randomly pick a GA operator according to the operator fitness distribution. One or two solutions are then randomly selected according to the solution fitness distribution, and the GA operator is applied on them to generate new solutions.
4. The new solutions are evaluated again by Equ. (1), (3) and (4), and the operator fitness values are adapted according to the performances of the new solutions. The worst solutions are then replaced by the new ones.
5. A complete run of Step 3 and 4 is called an *epoch*. Run a certain number of epochs or stop when the score of the best solution exceeds a preset threshold.

The following sub-sections give the rest of the details of the algorithm.

2.2 Iterative Candidate Set Computation

For a word $w_i \in W_1$ and $x_j \in W_2$, we first define S_{w_i} and S_{x_j} to be the set of sentences where w_i and x_j occurs respectively. The *coverage* of x_j with respect to w_i and vice versa are then computed by

$$C(w_i, x_j) = \frac{\left| S_{w_i} \cap S_{x_j} \right|}{\left| S_{w_i} \right|}$$
$$C(x_j, w_i) = \frac{\left| S_{x_j} \cap S_{w_i} \right|}{\left| S_{x_j} \right|} \quad \text{... (5)}$$

The *n-th iterative confidence* that x_j is the correct translation of w_i is then defined recursively as[6]

$$c_n(w_i, x_j) = \frac{c_{n-1}(w_i, x_j) \cdot c_{n-1}(x_j, w_i)}{\sum_{c_{n-1}(w_i, x_k) \neq 0} c_{n-1}(w_i, x_k) \cdot c_{n-1}(x_k, w_i)},$$
$$c_1(w_i, x_j) = C(w_i, x_j) \cdot C(x_j, w_i) \quad \text{... (6)}$$

Finally for w_i we rank x_j in descending order according to $c_n(w_i,x_j)$, and form the candidate set $CS(w_i)$ by only taking the top k (*cutoff value*) L2 words in the ranking list[7].

The intuition behind Equ (5) & (6) is that a preferable candidate x_j for w_i should have a higher ranking in $CS(w_i)$ and *vice versa*. This prevents a high-frequency L2 word from being a preferable translation for too many L1 words.

2.3 Genetic Operators

Three GA operators are adopted: *crossover*, *mutation* and *creep*. Each of them has its own fitness value, which is then used in selecting one operator in each epoch so that an operator with higher fitness value will on average be picked more frequently. To achieve greater autonomy and more dynamic system response the fitness values are adapted based on the idea of [Davis 1989], although the realization is somewhat different. In each epoch after evaluating the new solutions, a reward proportional to their improvements over the best solution in the population is credited to the responsible operator, and a proportion of the reward is propagated back to the operator generating the parent(s), and to the operator generating the grandparent(s), etc., until we reach out of a preset *history window*.

[4] The same random selection operation, *roulette-wheel* selection, which picks a selectee randomly according to a given distribution, is adopted in the entire experiment.

[5] This is to ensure that the GA improves upon the initial best solution.

[6] For the experiment results reported here n=2.

[7] For the experiment results reported here k=10.

The crossover operator takes two solutions, randomly picks a crossover point and swaps the sub-solutions between the two up to the point. The mutation operator randomly picks an L1 word and changes the current translation to another L2 candidate according to the candidate confidence distribution. These two operators are adopted in most of GA implementations, and contribute to the search process by combining the potentially useful building blocks and randomly exploring the search space, respectively.

The third operator, creep, locally optimizes a solution over a randomly chosen sentence. The new solution has the local alteration incorporated if it improves the objective score over the sentence, and the result of this perturbation is subsequently measured by Equ. (1), (3) and (4).

The introduction of the creep operator is based on observing how a human tries to 'decode' words using a small parallel corpus [Al-Onaizan 2000]. It is usually done by coming up with a set of translation hypotheses upon observing the correspondences between an L1 sentence and its corresponding L2 sentence. The hypotheses are then tested against the rest of the corpus.

The objective function to be optimized by the creep operator is the translation score over a particular sentence s_i, namely

$$S(f, s_i) = \lambda \cdot S_{\mathrm{L}}(f, s_i) + (1-\lambda) \cdot S_{\mathrm{P}}(f, s_i).$$

To make the search problem tractable this is done by a *limited depth-first beam search*: for each word w_i, only 3 possible translations randomly picked from $CS(w_i)$ are searched[8], and only the first 500 or 50% of the total different sentence translations are searched[9].

2.4 Fitness Scaling

To avoid premature population convergence the fitness values are computed by linearly scaling the solution scores following the suggestion in [Goldberg 1989]. The fitness value $F(f)$ of a solution f is computed by $F(f) = a \cdot S(f) + b$, where the scaling coefficients a and b are computed by solving the linear equations[10]

$$\overline{S} = a \cdot \overline{S} + b$$
$$c \cdot \overline{S} = a \cdot S_{\max} + b$$

Should they fail the coefficients are then obtained from solving the following equations:

$$\overline{S} = a \cdot \overline{S} + b$$
$$F_{\min} = a \cdot S_{\min} + b$$

where $F_{\min}$ is a parameter[11]. If these fail again the population is fully converged and we give up scaling by setting $a = 1.0$ and $b = 0.0$.

3 Experiments and Results[12]

In order to evaluate the effectiveness of the approach with limited data, the experiments were conducted on three small corpora of different sizes (corpus C1 – C3), taken from the Spanish and English portions of the UN Multilingual Corpus [Graff and Finch 1994]. The statistics of each corpus together with the population size used in the training session is shown below.

	# of sentence pairs	Spanish lexicon size	English lexicon size	Pop. size
C1	2000	6533	5335	150
C2	4000	9910	8010	200
C3	6000	12526	10034	250

Table 1. Statistics for corpus C1, C2 and C3

In all of the three training sessions the initial fitness values for the three GA operators were set equal (1/3), and 1,000,000 epochs were run for each corpus. After the training sessions 400 Spanish words were randomly picked from C3 and their translations were given according to the best solution before and after training using

[8] Again they are selected according to the candidate confidence distribution.

[9] Whichever smaller.

[10] Constant c is usually set in [1.2,2]. For the current implementation c=1.2.

[11] For the the current implementation $F_{\min} = 0.1$.

[12] All of the experiments were run on a Linux machine with AMD 700Mhz CPU and 256MB RAM.

	Init. score	Init. score per sentence	Final score	Final score per sentence	Init. Word Precision	Init. POS Precision	Final Word Precision	Final POS Precision
C1	853.961	0.4270	1051.63	0.5258	46%	60%	**49%**	**62%**
C2	1689.97	0.4225	1987.23	0.4968	49%	65%	**50%**	**67%**
C3	2566.68	0.4278	2966.91	0.4945	51%	63%	**54%**	**68%**

Table 2. Translation accuracies before and after training, with recall=100%; the max. precision is shown in bold typeface

each corpus. The results were then graded by native Spanish speakers, who were given the translations and the actual parallel sentence pairs to judge the correctness of the translations. The results are summarized in Table 2.

The initial best objective score and the per-sentence score for each corpus are shown in the 1st and the 2nd column, while the final cores are shown in the 3rd and 4th column. The initial scores represent the pre-GA, pure statistical translation precisions achieved by the iterative candidate set computation (see Section 2.2). Comparing the initial and the final scores indicates that the GA-based approach did perform better, although the advantages seemed to keep decreasing with the corpus size. This might be due to the insufficient running epochs when the corpus size grows larger.

The rest of Table 2 shows the evaluation results given by the native speaker graders. At recall level 100% for all corpora, both of the final word and POS precisions are consistently higher than the pre-GA ones – although the improvements are not significant (the most significant improvement is about 6.5%).

In addition to the premature termination of the training processes mentioned earlier, one possible explanation for the relatively modest improvements is that the random selection of a candidate in the mutation and creep operators is based on the candidate confidence distribution, which biases strongly toward the initial best solution. Another possible reason might be that the hypotheses were not reflecting perfectly the real linguistic similarities between these two languages. It has become apparent during the evaluation process that the system could take advantage of the observed linguistic constraints between these two languages, e.g., reinforcing the patterns where a Spanish adjective is translated behind a Spanish nominal head. Obviously any more specific hypotheses would risk the generality of the approach.

As a comparison with a conventional statistical approach, the experiment reported in [Brown 1997] used the same Spanish-English UN corpus, but instead of using only a small portion the author used the entire corpus for training (total 685,000 sentence pairs with 96,793 unique Spanish words). The best precision achieved at recall = 14.92% was 71%, but at the highest recall level (38.07%) the precision was 54%.

Future Works

In this paper I addressed the problem of translating with scarce resources, and demonstrated that by viewing translation as optimization, useful information could be fused to yield better translations. In particular, two linguistic hypotheses – locality and POS similarity – were postulated, and a GA-based technique was developed to solve the optimization problem. A preliminary evaluation based on the three small Spanish-English corpora is also reported

However there are several problems that need to be addressed in order to make the technique fully practical. Since the approach optimizes translation mappings sentence by sentence, as the size of the corpus grows the running time becomes unacceptable. The problem could be alleviated by only optimizing over the most discriminating sentences. Another limitation of the approach is the overly simplified representation used to represent lexical entries. There are no phrasal terms, no polysemous words, and no linguistic constraints between

words are learned (e.g., verb subcategorization). The problem must be overcome in order to make the machine-generated lexicon more useful within a complete translation system.

Finally, as cued in the previous section, it remains to be seen if a more accurate fitness function and a set of more robust GA operators can be discovered. In the realm of machine translation this might translate into finding the commonalities between a specific pair, or even any pair of languages, and a set of more universal operators for transforming word tokens of L1 into those of L2.

References

Al-Onaizan, Y., Germann, U., Hermjakob, U., Knight, K., Koehn, P., Marcu, D. and Yamada, K. (2000) Translating with Scarce Resources. *The 17th National Conference of the American Association for Artificial Intelligence (AAAI-2000),* Austin, Texas.

Brill, E. (1992) A simple rule-based part of speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing, ACL*..

Brown, R. (1997) Automated Dictionary Extraction for "Knowledge-Free" Example-Based Translation". In *Proceedings of the Seventh International Conference on Theoretical and Methodological Issues in Machine Translation*, pp. 111-118. Santa Fe.

Brown, R. (1999) Adding Linguistic Knowledge to a Lexical Example-Based Translation System. In *Proceedings of the Eighth International Conference on Theoretical and Methodological Issues in Machine Translation*, pp. 22-32. Chester, UK.

Davis, L. (1989) Adapting Operator Probabilities in Genetic Algorithms. In *Proceedings of the Third International Conference on Genetic Algorithms*, pp. 61 - 69, Morgan Kaufmann.

Fung, P. and McKeown, K. (1994) Aligning Noisy Parallel Corpora Across Language Groups: Word Pair Feature Matching by Dynamic Time Warping. In *Proceedings of the Conference on Theoretical and Methodological Issues in Machine Translation*, pp. 81-88.

Fung, P. and KcKeown, K. (1997) A Technical Word- and Term-Translation Aid Using Noisy Parallel Corpora Across Language Groups. *Machine Translation*, Vol. 12, Nos. 1-2, pp. 53-87.

Goldberg, D. E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.

Graff, D. and Finch, R. (1994) *Multilingual Text Resources at the Linguistic Data Consortium*. In *Proceedings of the 1994 ARPA Human Language Technology Workshop*. Morgan Kaufmann.

Holland, J. H. (1975) *Adaptation in Natural and Artificial Systems*, University of Michigan Press.

Haruno, M. and Yamazaki, T. (1996) High-Precision Bilingual Text Alignment Using Statistical and Dictionary Information. In *Proceedings of Annual Conference of the Association for Computational Linguistics*, pp. 131 -138.

Jones, D. and Somers, H. (1995) Bilingual Vocabulary Estimation from Noisy Parallel Corpora Using Variable Bag Estimation. In *JADT III Giornale Internazionale di Analsi Statistica dei Dati Testuali*, pp. 255-262, Rome.

Kay, M. and Röscheisen, M. (1993) Text-Translation Alignment. *Computational Linguistics*, Vol. 19, No 1, pp 121-142.

Kazakov, K. and Manandhar, S. (2000) Unsupervised Learning of Word Segmentation Rules with Genetic Algorithms and Inductive Logic Programming. To appear in *Journal of Machine Learning*.

Kumano, A. and Hirakawa, H. (1994) Building an MT Dictionary from Parallel Texts Based on Linguistic and Statistic Information. In *Proceedings of International Conference on Computational Linguistics*, pp. 76-81, Kyoto.

Losee, R. M. (1995) Learning Syntactic Rules and Tags with Genetic Algorithms for Information Retrieval and Filtering: An Empirical Basis for Grammatical Rules. In *Information Processing and Management.*

Orasan, C., Evans, R. and Mitkov, R. (2000) Enhancing Preference-Based Anaphora Resolution with Genetic Algorithms. In Christodoulakis (ed.) *Proceedings of Natural Language Processing (NLP 2000)*, pp. 185 – 195.

Rosé, C. P. and A. Lavie (1998) A Domain Independent Approach for Efficiently Interpreting Extragrammatical Utterances. In *Journal of Natural Language Engineering*, 1 (1) pp. 1-57.

Somers, H. and Ward, A. (1996) Some More Experiments in Bilingual Text Alignments. In Oflazer, K. and Somers, H. (eds) *Proceedings of the SecondInternational Conference on New Methods in Language Methods in Language Processing*, pp. 66-78, Ankara.

Utsuro, T. et al. (1994) Bilingual Text Matching Using Bilingual Dictionary and Statistics. In *Proceedings of International Conference on Computational Linguistics*, pp. 1076-1082, Kyoto.