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

Benjamin Han (benhdj@cs.cmu.edu)
Language Technologies Institute
Carnegie Mellon University

SRW, NAACL, June 4, 2001

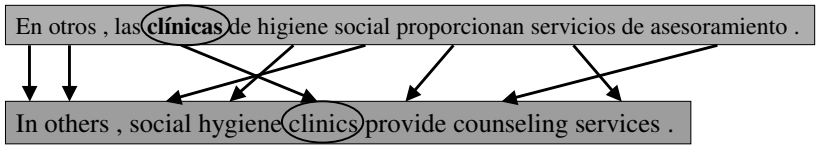
Translation with Scarce Resources

- Problem: limited amount of parallel texts
  - Statistical methods requires large amount of parallel texts
- Possible solution: incorporating linguistic knowledge

Bilingual Data

- Sentence-aligned bilingual texts

L_1 (source language): Spanish



L_2 (target language): English

Question: What is the English translation of the Spanish word '*clínicas*'?

What Do We Have?

- Assuming you know nothing about Spanish (like me), but you know English pretty well, you have
  - *statistics* (χ^2, mutual information, KL distance, etc.): doesn't work for limited amount of data, inaccurate for low-frequency words, etc;
  - *linguistic knowledge*: only limited to English.

Proposal: Combining the Two

- Forming hypothetical mappings by using statistics.
- Refining the mappings by using the limited linguistic knowledge.
- Advantage: fusing additional knowledge is straightforward.

Translation by Optimization

- A simple word-for-word translation mapping

$$f : W_1 \times W_2$$

  $W_{1/2}$ are words in $L_{1/2}$.
  - Neither polysemy nor 'null' translation is allowed.
- For each sentence s_i we give a translation score $S_j(f, s_i)$ based on linguistic knowledge j.
- Find a better mapping by maximizing

$$S(f) = \sum_{i,j} \lambda_j \cdot S_j(f, s_i); \quad \sum_j \lambda_j = 1$$

What to Optimize?

- The translation scores are defined based on the two hypotheses:
  - *Locality score* $S_L(f, s_i)$: Assuming good mappings reserve the relative distances between words.
  - *Part-of-speech similarity* (POS) *score* $S_P(f, s_i)$: Assuming good mappings reserve the POS distributions.

Locality Hypothesis

- Good mappings reserve the relative distances between words:

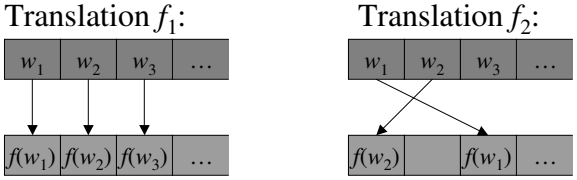
$$\frac{Dist(w_i, w_j)}{Dist(f(w_i), f(w_j))} \approx 1; \quad w_i, w_j \in L_1$$

- It doesn't presume any word orderings.
- But it requires $O(n^2)$ time to compute (n is the length of the sentence).

Locality Hypothesis

- To cut down computation we only consider every *k*-th word in a sentence.

k=1:



$$S_{\mathrm{L}}(f_1, s_i) \quad > \quad S_{\mathrm{L}}(f_2, s_i)$$

Locality Hypothesis

- $O(n)$ approximation

$$S_{\mathrm{L}}(f, s_i) = \mathrm{Average}[\frac{\kappa_i - |k - Dist_{\min}(f(w_j), f(w_{j+k}))|}{\kappa_i}]$$

(κ_i is a normalizing constant so that the score is between 0 and 1)

- The equation gives a higher score to the mapping which yields sentence pairs with more similar relative word distances.

POS Similarity Hypothesis

- Define POS distance
  - Each word has a POS vector (distribution)

'bank':

NN	VB	JJ	...
0.8	0.2	0	…

  - Each sentence s_i forms its POS vector D_i by summing up the individual word POS vectors
  - Distance between two sentence POS vectors (normalized angle)

$$Dist_{\mathrm{P}}(D_1, D_2) = \mathrm{acos}\left(\frac{D_1 \cdot D_2}{|D_1||D_2|}\right) / \frac{\pi}{2}$$

POS Similarity Hypothesis

- Good mappings reserve the POS distributions
  - But we only know the POS of L_2!
- Approximate POS of L_1 by the POS of L_2
  - For the sentence s_i in L_1

$$D(s_i) \approx D(f(s_i))$$

POS Similarity Hypothesis

- POS similarity score for the *i*-th sentence pair (s_i,t_i)

$$S_{\mathrm{P}}(f, s_i) = 1 - Dist_{\mathrm{P}}(D(f(s_i)), D(t_i))$$

- The equation gives a higher score to the mapping which yields sentence pairs with more similar POS.

How to Search?

- Genetic Algorithm (GA)
  - Maintain and evolve a population of solutions ('chromosomes')
  - Only well performers reproduce
  - Reproduction involves *crossover* to combine potentially good "building blocks"
  - Individual solutions can be operated on additional GA 'operators', such as *mutation*.

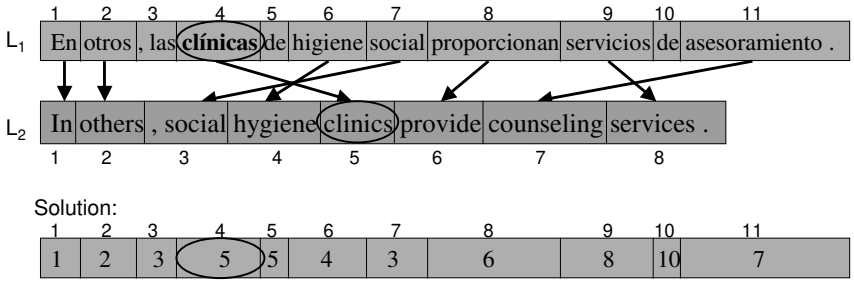
Why GA?

- Good for huge and ill-defined search space
- Optimize upon the baseline population – can't hurt

Encoding a Mapping

- Each solution encodes a mapping, a simple vector $v[1 \ldots m]$ (m is the vocabulary size)

$v[\mathrm{i}] = j$ means $f(w_i)=x_j$; $w_i \in L_1$ and $x_j \in L_2$



Algorithm

- Outline of a *steady-state*, *least-fit-deletion* GA:
  - Initialize a population by *candidate set computation*.
  - an epoch:
    - Randomly pick a GA operator and one/two solutions, and apply the operator to the solutions.
    - Evaluate the altered solutions, and replace the one/two worst solutions in the population with them.
  - Run for a certain number of epochs, or until the performance of the best solution exceeds a threshold.

Population Initialization

- For each word w_i in L_1 we collect the co-occurred words in L_2 into its *candidate set* CS(w_i).

En otros , las **clínicas** de higiene social proporcionan servicios de asesoramiento .

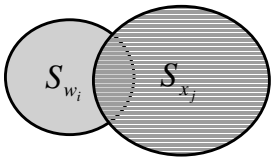
In others , social hygiene clinics provide counseling services .

Candidate set CS(*clínicas*) =
{in, others, social, hygiene, clinics, provide, counseling, services}

Population Initialization

- We then order the candidates x_j for w_i by computing their confidences

$$c(w_i, x_j) = \text{normalized}[\frac{|S_{w_i} \cap S_{x_j}|}{|S_{w_i}|} \cdot \frac{|S_{x_j} \cap S_{w_i}|}{|S_{x_j}|}]$$



S_{w_i}, S_{x_j} denote the set of sentence pairs containing word w_i and x_j

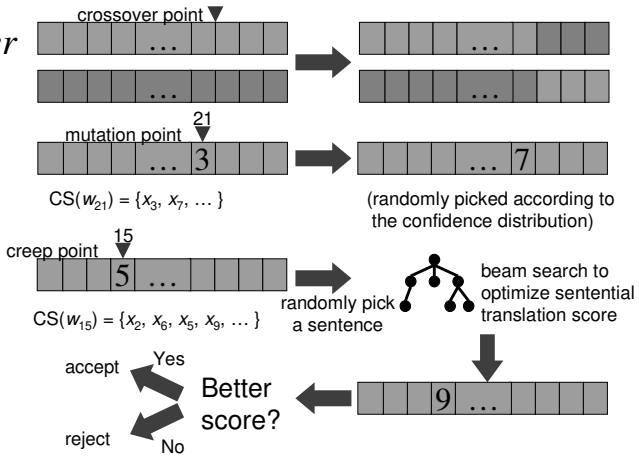
- The intuition: a preferable candidate x_j for w_i should have a high ranking in CS(w_i), and *vice versa*. This addresses the problem of low-frequency words.

Population Initialization

- We then form the initial population by
  - randomly generating mappings based on the candidate confidence distribution;
  - including the best solution, which consists of the top words from all candidate sets.
- Each candidate set is pruned to only contain the top k (cutoff value) candidates for later use.

GA Operators

- Three operators
  - *Crossover*
  - *Mutation*
  - *Creep*



GA Operator Adaptation

- Operators are chosen randomly according to their fitness distribution
  - Each well-performing operator receives a reward
    - Reward ∝ (The fitness of the produced solution – the fitness of the best solution in the population)
  - The reward propagates back in time to the operators responsible for the predecessor solutions.

Fitness Scaling

- Solutions are picked for an operator randomly according to their fitness distribution.
- Fitness values are computed by linear scaling the translation scores $S(f)$.
  - Prevent premature population convergence.

Experiment Setup

- Goal: Investigating the performance of the approach with small corpora.
- Data: Subsets of UN Spanish-English Corpus.

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

Experiment Setup

- Parameters settings
  - Both locality and POS scores receive the same weights: $\lambda_1 = \lambda_2 = 0.5$.
  - Running 1 million epochs.
  - Operator adaptation window size = 5.

Evaluation

- For each corpus
  - the same 400 Spanish words were randomly chosen;
  - the translations were evaluated by 3 native speakers via a web interface.
- The graders evaluated both the word translations and the POS accuracies of L_1.

Results (recall 100%)

pre-GA performance

	Init. score per sentence	Final score per sentence	Init. word precision	Init. POS precision	Final word precision	Final POS precision
C1	0.4270	0.5258	46%	60%	49%	62%
C2	0.4225	0.4968	49%	65%	50%	67%
C3	0.4278	0.4945	51%	63%	54%	68%

- Improvement keeps decreasing as the corpus size increases.
- The most significant improvements are 6.5% (word precision) and 7.9% (POS precision), relative.

Results (recall 100%)

pre-GA performance

	Init. score per sentence	Final score per sentence	Init. word precision	Init. POS precision	Final word precision	Final POS precision
C1	0.4270	0.5258	46%	60%	49%	62%
C2	0.4225	0.4968	49%	65%	50%	67%
C3	0.4278	0.4945	51%	63%	54%	68%

- Comparing to [Brown 1997]
  - Training corpus size 685,000 sentence pairs with 96,793 unique Spanish words.
  - The best precision 71% was achieved at recall 14.92%.
  - At highest recall 38.07% the precision was 54%.

Postmortem

- Premature termination of the training for the larger corpora.
  - Increase the # of epochs.
- Random selection of alternative candidates in *mutate* and *creep* operators are constrained by the candidate confidence distributions.
  - Relax the constraints.
- Hypotheses are not reflecting the real linguistic similarities between Spanish and English.
  - Incorporating language-pair specific features, e.g., adjectives follow nominal heads in Spanish.

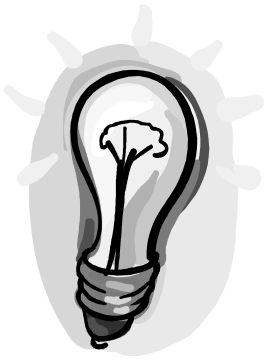
Summary

- We addressed the problem of translating with scarce resources
  - Translation by optimization
  - A GA-based approach is proposed and tested

Future Works

- Improve the training efficiency by considering only the *most discriminating* sentences.
- Enrich the representation to accommodate phrasal terms, polysemous words, null translation, etc.
- Search for a better set of linguistic correspondences between a specific language pair, or *even* any language pair.

Acknowledgements

- I'd like to thank
  - **Ariadna Font Llitjos** and her friends: for grading the translations.
  - **Ralf Brown**: for providing the tokenized training corpus and insightful discussions.
  - **Joy Zhang**: for discussions and supports.

Feedback?