Experimenting with Phrase-Based Statistical Translation within the IWSLT 2004 Chinese-to-English Shared Translation Task

Philippe Langlais
RALI/DIRO
University of Montreal
Canada
felipe@iro.umontreal.ca

Michael Carl
IAI
Saarbrücken
Germany
carl@iai.uni-sb.de

Oliver Streiter
NUK
National Univ. of Kaohsiung
Taiwan
ostreiter@nuk.edu.tw

Motivations

- How far can we go in one month of work, starting from (almost) scratch and relying intensively on available packages ?

- Interested by the perspective taken by the organizers: validation of existing evaluation methodologies. See also the CESTA project (TECHNOLANGUE):

  http://www.technolangue.net/

We participated in:

- The Chinese-to-English track using only the 20K sentences provided

Plan

- Few words on the core engine
- Our phrase-based extractors
- Experiments with phrase-based models (PBMs)
- Conclusions

The core engine

We used an off-the-shelf decoder: `Pharaoh` (Koehn,2004). It requires:

- a flat PBM ($e.g$ small boats $\leftrightarrow$ bateau de plaisance 0.82)

  details are coming soon

- we used `SRILM` (Stolcke,2002) to produce a 3-gram

  ngram-count -interpolate -kndiscount1 -kndiscount2 -kndiscount3

- a set a parameters (one for the PBM, one for the language model, one for the length penalty and one for the built-in distorsion model)

  details are coming soon

`Pharaoh` is a noisy channel phrase-based statistical engine.

Our phrase-based extractors

We tried two different methods of extraction:

WABE: relying on viterbi alignments computed from IBM model 3

We used `Giza++` (Och and Ney, 2000) to get them out of an IBM model 3

SBE: One capitalizing on redundancies in the training corpus at the sentence level

- WABE = Word-Alignment Based Extractor
- SBE = String-Based Extractor

WABE: Word-alignment based extractor

Yet another version of (Koehn et al.,2003; Tillmann,2003) and others. Basically:

- Considering the intersection of the word links obtained by viterbi alignment in both directions (C-E, E-C)
- (more or less) carefully extending this set of links with links belonging to the union of both sets (C-E,E-C)

Few meta-parameters are controlling the phrases acquired in this way:

length-ratio : `ratio = 2`

min-max src/tgt length : `min=1, max=8`

SBE: String-based extractor

If two strings are in relation of translation and if part of them also are, then we can induce a specific translation relation between the other parts.

$res \leftarrow \mathcal{T} = \{(E_i, C_i), i \in [1, |\mathcal{T}|]\}$ (the training corpus)
repeat
  for all $\langle (E_i, C_i), (E_j, C_j) \rangle \in res$ **do**
    if $C_j = C_i\alpha$ or $C_i = C_j\alpha$ **then**
      if $E_j = E_i\beta$ or $E_i = E_j\beta$ **then**
        $res \leftarrow res \cup (\beta, \alpha)$
until convergence of res

54 461 parameters out of 20K sentences

Experiments with PBMs: setting

corpus	|pair|	Chinese tokens	Chinese words	English tokens	English words
TRAIN	20 000	182 904	7 643	188 935	7 181
TRAIN-A	11 884	112 000	6 456	116 343	6 008
TRAIN-Q	8 116	70 904	4 024	72 592	3 900
CSTAR	506	3 515	870	–	–
TEST	500	3 794	893	–	–

- the **tokenization** was the one provided, English material was **lowerized**,
- **punctuation** marks where removed from the translations in accordance to the specifications (`s/ \.//g`, `s/ ?//g`, `s/ ,//g` , `s/ "//g`, `s/ \!//g`, `s/-/ /g`, `s/ */ /g`)
- source **OOV** appearing in the translations were replaced afterward by the most likely word according to our 3g model (in a left-to-right manner). Uppercased OOV were left unmodified.

Word-based translation versus PB translation

engine	NIST	BLEU%	mWER	mSER
ibm2+3g	5.0726	26.57	60.56	94.47
`Pharaoh`	5.5646	26.16	59.70	94.27
wbm by `Pharaoh`	4.8417	15.54	64.95	97.63

- *ibm2+3g* is an extension of the decoder described by (Niessen et al., 1998)
- `Pharaoh` was run with its default setting; each parameter of the FPBM was scored by relative frequency

Tuning the decoder

λ_d	λ_ϕ	λ_w	λ_{lm}	NIST	BLEU%	mWER	mSER
1	1	0	1	5.5646	26.16	59.70	94.27
1	1	-1.5	1	6.3470	25.63	58.93	94.27
.2	.9	-1.5	.8	6.8401	28.44	56.25	94.07

λ_d, distorsion weight ($[0, 1]$) λ_ϕ, transfer weight ($[0, 1]$)
λ_w, word penalty ($[-3, 3]$) λ_{lm}, language model weight ($[0, 1]$)

We applied a poor man's strategy (sampling uniformly the parameter ranges)

$\hookrightarrow$ a relative gain over the default configuration (line 1) of 23%

$\hookrightarrow$ 61% of this gain obtained by tuning only the word penalty parameter

Merging different FPBMs

config	$\|p\|$	NIST	BLEU%	mWER	mSER
WABE		6.8401	28.44	56.25	94.07
+ WBM		7.0766	31.38	54.88	93.28
+ SBE		7.0926	31.78	54.56	92.69

Merging 2 models was done harshly by:

- copying $p_i(s|t), \forall s$ whenever t has not been seen in one model,
- averaging them in case both $p_1(s|t)$ and $p_2(s|t)$ exist,
- normalizing

$\hookrightarrow$ a relative gain of 3.7%

The weakness of relative frequency

min	max	$\|model\|$	%f1	%f2	%f3+	$\%p = 1$
1	8	166 481	90.6	4.9	4.5	74.6
2	8	153 512	92.7	4.3	3.0	78.5
2	4	73 369	87.0	7.1	5.9	68.7

- %f1, %f2 and %f3+ stand for the percentage of parameters (pairs of phrases) seen 1, 2 or at least 3 times in the TRAIN corpus.

- $\%p = 1$ stands for the percentage of parameters that have a relative frequency of 1.

Scoring phrases with IBM model 1

model	NIST	BLEU%	mWER	mSER
`relfreq`	7.0926	31.78	54.56	92.69
`ibm`	7.3067	32.98	53.86	92.49
`relfreq&ibm`	7.3118	34.48	52.73	91.90
`relfreq&pn-ibm`	7.4219	34.6	53.02	91.70

- baseline model (line 1) = merged FPBM of 306 585 parameters trained by relative frequency.
- rating these parameters by IBM model 1 yields a relative improvement in the NIST score of 3%
- `pn-ibm`: do not normalize parameters where $|\{s : p(s|t)\exists\}| = 1$ holds

Specific models

config	NIST	BLEU%	mWER	mSER
relfreq&ibm	**7.3118**	**34.48**	**52.73**	**91.90**
A	7.1862	34.21	53.12	91.18
Q	6.4995	34.92	52.12	93.00
specific-lm	**7.4702**	**33.64**	**53.27**	**91.90**
A	7.3229	33.66	53.08	90.85
Q	6.7010	33.58	53.55	93.50

- around 40% of the training sentences were interrogatives ones

$\Rightarrow$ specific language model combined to the general one (specific tuning over 6 parameters)

(we did not observe improvements by modelling specific FPBMs)

Translations we submitted before the deadline

ibm2+3g word-based translation engine,

straight a WABE FPBM

merge the best model obtained by merging word and phrase associations

QA the one submitted for manual evaluation

manual to measure the usefulness of the automatic translations for human post-editing

Task: selecting one translation among the generated ones and enhancing its quality though slight modifications

The *manual* experiment

- 423 (84.6%) were just selections of one of the automatic translations.
- Out of these 423 translations, 85 (20%) were produced by the word-based engine ($ibm2{+}3g$).

trans1	take a bath for a twin room .
trans2	please take a bath for a double .
trans3	take a bath of double .
trans4	take one twin room with bath .
trans5	have a bath for double .
trans6	have a twin room with bath , please .
trans7	have a double room with bath , please .
manual	**please, a twin room with bath .**

Translations we submitted before the deadline

config	BLEU%	NIST	GTM	WER	PER
ibm2+3g	27.27	6.55	62.49	58.12	48.82
straight	30.92	7.52	66.93	56.05	47.90
merge	35.32	8.00	68.60	51.74	43.86
QA	33.89	7.85	68.55	53.24	45.14
manual	36.93	8.13	68.42	49.62	42.53

↪ the ordering of the variants was (almost) consistent with the one observed on the CSTAR corpus

Conclusions

- Is phrase-based translation $\equiv$ $\texttt{Pharaoh}(\texttt{Giza++}^{\lambda_g} \times \texttt{SRILM}^{\lambda_s})$?

  $\hookrightarrow$ at least a decent system can be obtained this way

- Things we tried that did not work better:
  - splitting the training sentences into shorter ones
  - replacing proper names by `NAME`

- Many factors to be tried:
  - word alignment procedure (Simard and Langlais, 2003)
  - other scoring functions (Zao et al., 2004)

- Not clear whether the best settings we found here would be appropriate for another translation task

References

Koehn P., "Pharaoh: a Beam Search Decoder for Phrase-Based SMT", To appear in Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA), 2004

Stolcke A., "SRILM - An Extensible Language Modeling Toolkit", In Proceedings of the International Conference for Speech and Language Processing (ICSLP), Denver, Colorado, September 2002

Och F.J. and Ney H., "Improved Statistical Alignment Models", in Proceedings of the Conference of the Association for Computational Linguistic (ACL), Hongkong, China, pp. 440–447, 2000

Koehn P., Och F.J. and Marcu D., "Statistical Phrase-Based Translation", In Proceedings of the Human Language Technology Conference (HLT), pp. 127–133, 2003

Tillmann C., "A Projection Extension Algorithm for Statistical Machine Translation", In

Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2003

Niessen S., Vogel S. Ney H. and Tillmann C., "A DP based Search Algorithm for Statistical Machine Translation", in Proceedings of the International Conference On Computational Linguistics (COLING), pp. 960–966, 1998

Simard M. and Langlais P., "Statistical Translation alignment with Compositionnality Constraints", HLT-NAACL Workshop: Building and Using Parallel Texts: Data Driven Machine Translation and Beyond, Edmonton, Canada, May 31, pp.19–22, 2003

Zhao B., Vogel S. and Waibel A., "Phrase Pair Rescoring with Term Weightings for Statistical Machine Translation", In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Barcelona, Spain, July 2004

WABE

Require: $\mathcal{P}, \mathcal{R}, minLength, maxLength, ratio$
Ensure: res contains all the pairs of phrases

1: *Initialization*
2: $res \leftarrow \{\}$
3: **for all** $x : 1 \rightarrow |S|$ **do** $T[x] \leftarrow \{\}$
4: **for all** $y : 1 \rightarrow |T|$ **do** $T[y] \leftarrow \{\}$
5:
6: *Step1: $\mathcal{P}$-projection*
7: **for all** $(x, y) \in \mathcal{P}$ **do** $add(x, y)$
8:
9: *Step2: Extension*
10: **for** $p : 1 \rightarrow 2$ **do**
11: **repeat**
12: $a \leftarrow \{\}$
13: **for** $s : 1 \rightarrow |S|$ **do**

for all $t \in T[s]$ **do**
if $p = 2$ **then**
neighbor(x-1,y-1); neighbor(x+1,y-1);
neighbor(x-1,y+1); neighbor(x+1,y-1);
else
neighbor(x-1,y); neighbor(x+1,y);
neighbor(x,y-1); neighbor(x,y+1);
for all $(x, y) \in a$ **do** $add(x, y)$
until $|a| = 0$

Step3: Collect independent boxes
$b \leftarrow \{\}$
for $x : 1 \rightarrow |S|$ **do**
$X \leftarrow \{x\}$; $Y \leftarrow \{\}$
repeat
$X_m \leftarrow X$; $Y_m \leftarrow Y$

30: **for all** $x \in X$ **do** $Y \leftarrow Y \cup T[x]$
31: **if** Y != Y_m **then**
32: **for all** $y \in Y$ **do** $X \leftarrow X \cup T[y]$
33: **until** $X = X_m$ and $Y = Y_m$
34: $b \leftarrow b \cup \left\{ \begin{array}{l} (min\{x : x \in X\}, max\{x : x \in X\}), \\ (min\{y : y \in Y\}, max\{y : y \in Y\}) \end{array} \right\}$
35: $x \leftarrow max\{x : x \in X\} + 1$
36:
37: *Step4: Combine boxes*
38: **for** $i : 1 \rightarrow |b|$ **do**
39: **let** $((x_{m_i}, x_{M_i}), (y_{m_i}, y_{M_i})) = b_i$
40: $add(x_{m_i}, x_{M_i}, y_{m_i}, y_{M_i})$
41: **for** $j : i + 1 \rightarrow |b|$ **do**
42: **let** $((x_{m_j}, x_{M_j}), (y_{m_j}, y_{M_j})) = b_j$
43: **if** $x_{M_i} + 1 = x_{m_j}$ **then**
44: $add(x_{m_i}, x_{M_j}, y_{m_i}, y_{M_j})$