Experiments with Czech-English Phrase-Based Translation

Ondřej Bojar
obo@cuni.cz

November 25, 2005

Outline

- Data Overview
- Alignment Type (intersection/union/. . .)
- Language Model / Unprocessed Dictionary / Parallel Data
- Simple Rule-Based Handling of Numbers
- Dependency-Based Corpus Expansion
- Simple Hacks
- Comments on Corpus Preparation

Corpus Statistics

		Czech	English
Train:	Sentences	21106	
	Running Words	474452	493462
	Running Words without Punct. Marks	403503	438533
	Vocabulary	56970	30739
	Singletons	31394	14618
Dev:	Sentences	259	
	Running Words	6386	6522
	Running Words without Punct. Marks	5422	5848
	Vocabulary	2528	1884
	OOVs (running words)	466	232
	OOVs (in voc.)	385	172
Test:	Sentences	256	
	Running Words	5815	6175
	Running Words without Punct. Marks	4974	5445
	Vocabulary	2363	1799
	OOVs (running words)	478	263
	OOVs (in voc.)	407	225

Treatment of Unknowns
Fixing Reference Tokenization

	DEVFIX	TESTFIX	DEVORIG	TESTORIG
confess unknown words	30.2	25.9	20.8	17.6
drop unknown words	31	26.5	22.5	19.1
keep unknown words	32.4	27.3	21.9	18.4

- ORIG – the original reference translations
- FIX – reference translations automatically altered (added splits) to mimic tokenization rules of training data
  $\Rightarrow\sim$10% improvement absolute!

Intersection Alignment Worst for MT

	DEV-std	TEST-optbleu	TEST-std
intersect sing	27.9	26.6	23.4
intersect	28	25.9	23.5
sym	28.4	27.4	23.8
sym sing	28.5	27.2	24.7
sym intersect	28.5	26.9	23.8
iu	29.1	27.2	24.3
IU	29.2	27.3	24.4
union sing	29.3	27.1	24.9
sym union	29.3	27.3	24.4
union	29.8	27.3	24.6

Alignment was always based on lemmas.
sing=singletons replaced with POS, sym=symmetric, iu/IU=refined

Full Lemmatization > Simple Stemming

		DEV-std	TEST-optbleu	TEST-std
	stem42	28.5	26.1	23.5
baseline →	forms	28.6	25.8	23.6
	lemmas sing	29.3	27.1	24.9
	stem4	29.6	26.7	23.9
	lemmas	29.8	27.3	24.6

		Vocab		Singl/Vocab	
Type of input for the alignment		CZ	EN	CZ	EN
Forms	Produkce malých vozů se více než ztrojnásobila .	57k	31k	55.1%	47.6%
Stem4	Prod malý vozů se více než ztro .	17k	14k	36.5%	35.8%
Stem42	Prod/ce malých vozů se více než ztro/la .	52k	28k	51.2%	45.3%
Lem+Sing	produkce malý vůz se hodně než-2 UNK-verb .	15k	13k	0.1%	0.0%
Lemmas	produkce malý vůz se hodně než-2 ztrojnásobit .	28k	25k	46.4%	47.5%

More Data? LM>parallel>dictionary

		DEV-std	TEST-optbleu	TEST-std
	pcedt5k ali-lemmas	22.7	21.5	19.1
	pcedt5k lmpcedt ali-lemmas	25.6	24	21.2
	pcedt10k ali-lemmas	26.6	23.7	21.2
baseline →	pcedt20k ali-lemmas	29.8	27.3	24.6
dict worse →	pcedt20k+dict ali-stem4	29.8	27.5	24.6
than parallel →	pcedt20k+stories ali-stem4	31.6	28	25.9
	pcedt20k+dict lmpcedt ali-stem4	32.7	29.6	26.9
and than LM →	pcedt20k lmpcedt ali-lemmas	33.2	29.4	26.4
	pcedt20k lm600M4grKN ali-lemmas	33.4	31.9	27.3
	pcedt20k+stories lmpcedt ali-stem4	35.9	32.3	29.7

pcedt 5k 10k 20k	the core parallel corpus, various number of sentences
dict	Czech-English Web Dict, 116k entries, 198/202k, tokens 20k/30k vocab
stories	more parallel texts, 85k sents, 1.5/1.7M tokens, 118/44k vocab
lmpcedt	in-domain LM provided by (Čmejrek, Cuřín, and Havelka, 2003), n-gram vocab 0.4:5:7M
lm600M4grKN	"generic" LM provided by Richard Zens, 600M tokens, n-gram vocab 1.7:26:38:63M

Circumventing Proper Names and Numbers?

Simplistic circumventing of proper names hurts (declination, tokenization).
Circumventing numbers helps a bit.

	DEV-std	TEST-optbleu	TEST-std
propnames numbers	25.1	23.4	21.3
propnames numbers reform	25.5	24.9	22.9
propnames	25.8	-	21.4
numbers	29.2	27.1	24.2
numbers reform	29.7	28.6	25.8
baseline	29.8	27.3	24.6

	input	fed in pbt	output
baseline	na 57,375 dolarech	na 57,375 dolarech	at UNK_57,375 $
numbers	na 57,375 dolarech	na _NUM dolarech	at $ 57,375
numbers reform	na 57,375 dolarech	na _NUM dolarech	**at $ 57.375**

Dependency-based Corpus Expansion

Create new training sentences (with new n-grams) by deleting aligned leaves of dependency structures ("reducing sentences").

- Off-line: print all possible reduced sentences given the training corpus
  $\Rightarrow$unbearable, explosion of data.
- On-line: given the test source data (the set of "needed" n-grams)
  - Scan training corpus for sentences with sample *non-contiguous* occurrences of the needed n-grams.
  - Mark the source nodes, aligned nodes and all dependency neighbours needed for a certain level of grammaticality.
  - Print marked nodes.

Flavors of Corpus Expansion

	DEV-std	TEST-optbleu	TEST-std
ali-intersect exp-intersect	28.1	26.3	23.5
ali-intersect exp-intersect nohooker	28.1	25.7	23.5
ali-union exp-union nohooker	29.7	27.8	24.4
ali-union exp-union	29.7	27.4	24.4
ali-union exp-intersect	29.7	27	24.5
ali-union exp-union nolimit	29.8	26.8	24.5

263 test sentences contain 5146 bigrams. For 60% at least one non-contiguous sample is found, for 33% no sample is found, 7% have only contiguous samples.

Out of 440k test corpus samples, 20% are ignored (contiguous), 60% are rather random co-occurrences (too distant in the dependency tree), 93k (20%) seem useful.

However, after "ensuring grammaticality", 92% of 93k not useful any more (became non-contiguous). 7.8k are finally used, boosting the corpus size from 21k to 29k sentences (only 2k of them are unique).

However: Corpus Expansion Helps Too Little

	pcedt 20k	pcedt 10k	pcedt 5k
Baseline	27.3	23.7	21.5
Expanded	27.4	23.4	21.2
Expanded without "L.J. Hooker"	27.8	-	-

Alignment (union) based on lemmas; optimized BLEU results.

Impression: dependency based expansion is a bit useful, if:

- grammaticality of samples is ensured (language-dependent rules)
- obtained samples are carefully filtered

Causes of BLEU Errors (worst translation)

Top missing bigrams:

19	, "	12	" said
12	of the	10	Free Europe
10	Radio Free	7	. "
6	L.J. Hooker	6	United States
6	in the	6	the United
6	the strike	5	" We
5	, a	5	is a
5	margin calls		
4	28 tokens, 7 types		
3	54 tokens, 18 types		
2	94 tokens, 47 types		
1	698 tokens, 698 types		

Top superfluous bigrams:

26	, ''	18	'' .
14	" said	12	, which
11	Svobodná Evropa	8	, when
8	the state	7	, who
7	J. Hooker	7	L. J.
7	company GM	7	firm Hooker
7	radio Svobodná	7	spokesman for
7	the company		
6	18 tokens, 3 types		
5	35 tokens, 7 types		
4	40 tokens, 10 types		
3	117 tokens, 39 types		
2	342 tokens, 171 types		
1	3214 tokens, 3214 types		

Missing bigram = all references contained it but not the hypothesis
Superfluous bigram = the hypothesis contained it but none of the references

A Simple Hack

	DEV-std	TEST-optbleu	TEST-std
pcedt5k	22.7	21.5	19.1
pcedt5k fix	24.5	22.2	20
pcedt20k	29.8	27.3	24.6
pcedt20k fix	31.6	28.2	25.6
pcedt20k lm600M4grKN	33.4	31.9	27.3
pcedt20k lm600M4grKN fix	35.1	32.9	28.4

The "fix" is just this:

’ ’ .	→	. "
’ ’	→	"
L. J. Hooker	→	L.J. Hooker
the U.S.	→	the United States

Summary of PBT Impressions

sym/union alignment instead of intersection	+1.5 to +2.0
stemming for alignment	+1.0
lemmatization for alignment	+1.5
raw dictionary	+0.2
out-of-domain parallel texts, also in LM	+0.7 to +1.7
in-domain LM	+2.1 to +3.4
bigger generic LM	+4.6
out-of-domain parallel for phrases, in-domain LM	+5.0 to +6.0
handling numbers	+0.5
dependency-based corpus expansion	+0.5
fixing clear translation problems	+1.0 to +1.5
similar tokenization of reference	+10.0

Comments on Corpus Preparation: Crep

crep – a new tool I implemented in the last two weeks:

- competing regular expressions
  crep script = set of regexp/replacement+priority
- operates on non-segmented input (unlike sed/straightforward Perl)
- reports all ambiguous replacements (unlike sed/Perl/flex)
- performance worse than sed/Perl/flex
  but usable for 1.5M corpora and a 5000 of exceptions

```
ALPHA=[a-zA-Z]
NONALPHA=[^\32\\10\a-zA-Z]
acronym={ALPHA}|Mr|Mrs|Dr|St

1       {ALPHA}(()){NONALPHA}   <nosp>\32\
2       \32\{acronym}((\.({NONALPHA})))       .\32\$1
```

Comments on Corpus Preparation: Tok/Hyp

Suspicious tokenization and hyphenation:

Splitted	Joined	Splitted occs	Joined occs
Mr .	Mr.	1	2151
it s	its	2	2063
Corp .	Corp.	86	705
Inc .	Inc.	132	601
in to	into	11	520
Co .	Co.	87	500
the re	there	1	381
a round	around	1	160

Lessons Learned

What helps research:

- qsub / Grid Engine
- shared source code in "fast" languages, a strong group of "programmers"
- fast experimental cycle ($\leftarrow$ automatic measures)

- CVS/subversion, make
- make clone+last_experiment+new_experiment;
  cd new_experiment; vim Makefile; make run;
  cd ..; make collect_results

A Big Thank You

. . . for the experience and for the friendly atmosphere

References

Čmejrek, Martin, Jan Cuřín, and Jiří Havelka. 2003. Czech-English Dependency-based Machine Translation. In *EACL 2003 Proceedings of the Conference*, pages 83–90. Association for Computational Linguistics, April.

Czech Data Relevant for Me/MT

Name and version	Sents.	Tokens	Vocab.	Lemmas	Notes
Czech National Corpus (SYN2000d)	6.8M	114M	1.7M	775k	
Prague Dep Tbk (PDT 1.0)	82k	1.3M	130k	55k	

Parallel Czech-English

Name and version	Sents.	Tokens	Vocab.	Lemmas	Notes
Prague Cz-En Dep Tbk (PCEDT 1.0)	22k/49k	0.5M/1.2M	57k/30k	28k/25k	
Reader's Digest (PCEDT 1.0)	44k/44k	658k/755k	84k/36k	?	stories
Kačenka	128k/105k	1.5M/1.5M	102k/47k	39k/22k	stories
OPUS EU Constitution	11k/10k	127k/164k	?	?	bad tok.
Kolovratník	107k/107k	1.3M/1.5M	190k/92k	?	not tok.!

BEAST: a compilation of web dictionaries (400k pairs, 235k cs, 225k en entries; if rejecting multi-word expressions: 138k pairs, 58k cs, 53k en)