FBK @ IWSLT 2007

N. Bertoldi, M. Cettolo, R. Cattoni, M. Federico
FBK - Fondazione B. Kessler, Trento, Italy

Trento, 15 October 2007

HERMES
Cross-Language Information Processing

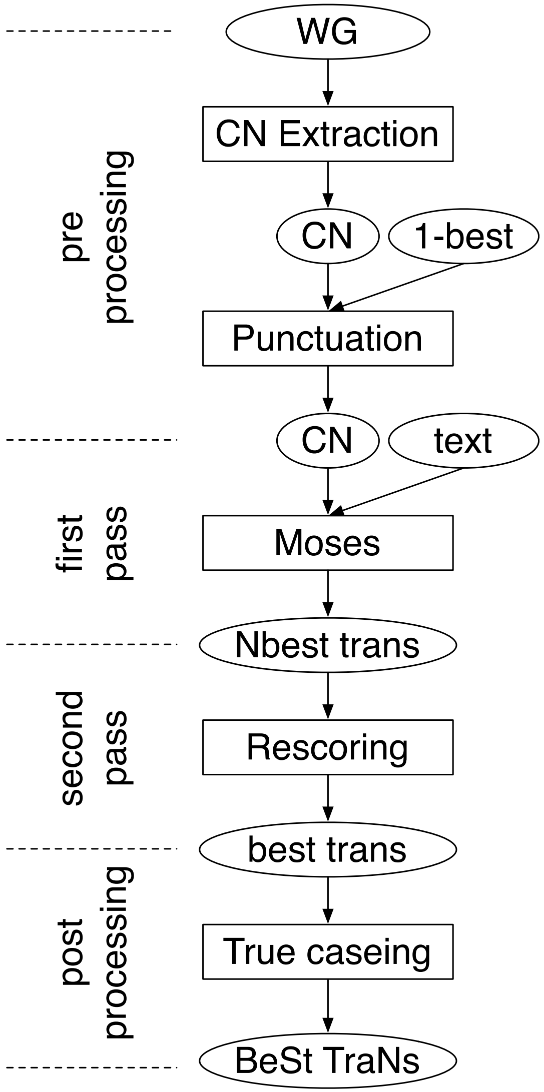
ITC ISTITUTO TRENTINO DI CULTURA
Dal 1° marzo 2007 FONDAZIONE BRUNO KESSLER
irst CENTRO PER LA RICERCA SCIENTIFICA E TECNOLOGICA

- system architecture
- confusion network
- punctuation insertion
- improvement of lexicon
- use of multiple lexicons and language models
- system evaluation

Acknowledgments

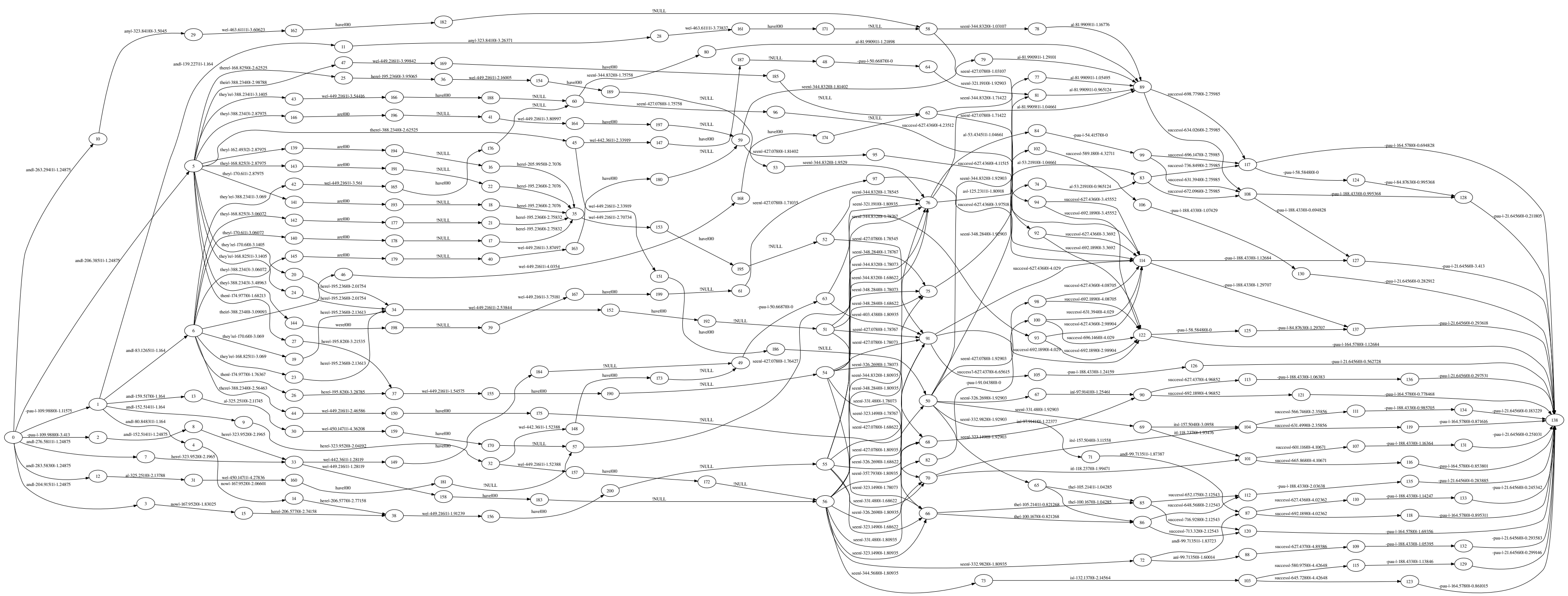
- Hermes people: Marcello, Mauro, Roldano

The FBK SLT System



- input from speech (word-graph or 1-best) or text
- pre and post processing (optional)
  - use of the SRILM toolkit
  - **CN extraction**: `lattice-tool`
  - **punctuation insertion**: `hidden-ngram`
  - case restoring: `disambig`
- **Moses** is a text/CN decoder
- rescoring of N-best translations (optional)

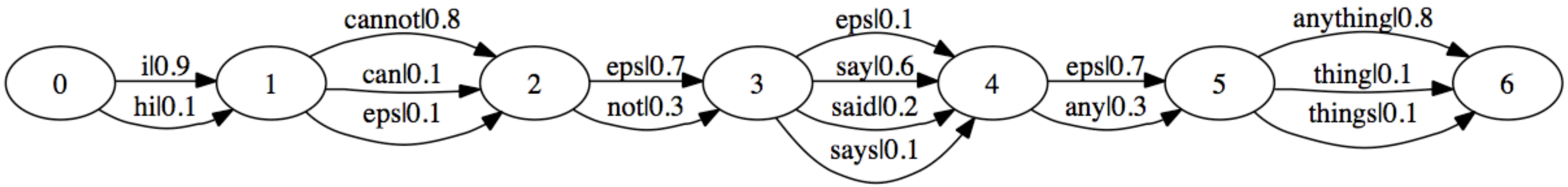
Step 1: take the ASR *word lattice*



- arcs are labeled with *words* and *acoustic and LM scores*
- arcs have start and end *timestamps*
- any path is a *transcription hypothesis*

Confusion Network Extraction

Step 2: approximate the word lattice into a *Confusion Network*



- a CN is a linear word graph
- arcs are labeled with *words* or with the *empty word* (ϵ-word)
- arcs are weighted with word *posterior probabilities*
- paths are a *superset* of those in the word lattice
- paths can have different lengths
- algorithm proposed by [Mangu, 2000]
  - exploit start and end timestamps of the lattice arcs
  - collapse/cluster close words
  - `lattice-tool`

Step 3: represent the CN as a *table*

$\text{i}_{.9}$	$\text{cannot}_{.8}$	$\epsilon_{.7}$	$\text{say}_{.6}$	$\epsilon_{.7}$	$\text{anything}_{.8}$
$\text{hi}_{.1}$	$\text{can}_{.1}$	$\text{not}_{.3}$	$\text{said}_{.2}$	$\text{any}_{.3}$	$\text{thing}_{.1}$
	$\epsilon_{.1}$		$\text{says}_{.1}$		$\text{things}_{.1}$
			$\epsilon_{.1}$		

Step 3: represent the CN as a *table*

$\text{i}_{.9}$	$\text{cannot}_{.8}$	$\epsilon_{.7}$	$\text{say}_{.6}$	$\epsilon_{.7}$	$\text{anything}_{.8}$
$\text{hi}_{.1}$	$\text{can}_{.1}$	$\text{not}_{.3}$	$\text{said}_{.2}$	$\text{any}_{.3}$	$\text{thing}_{.1}$
	$\epsilon_{.1}$		$\text{says}_{.1}$		$\text{things}_{.1}$
			$\epsilon_{.1}$		

Notes

- text is a trivial CN
- CN can be used for representing ambiguity of the input
  - transcription alternatives
  - punctuation
  - upper/lower case

The Problem

- *punctuation* improves *readability* and *comprehension* of texts
- *punctuation marks* are important clues for the translation process
- most ASR systems generate output *without* punctuation

The Problem

- *punctuation* improves *readability* and *comprehension* of texts
- *punctuation marks* are important clues for the translation process
- most ASR systems generate output *without* punctuation

Our approach [Cattoni, Interspeech 2007]

- insert punctuation as a *pre-processing* step
- exploit *multiple* hypotheses of punctuation
- use *punctuated models* (i.e. trained on texts with punctuation)
- let the decoder choose the best punctuation (and translation)

Step 1: take the input *not-punctuated CN*

i$_{.9}$	cannot$_{.8}$	$\epsilon_{.7}$	say$_{.6}$	$\epsilon_{.7}$	anything$_{.8}$	at$_{.9}$	this$_{.8}$	point$_{.7}$	are$_{1}$	there$_{.8}$	$\epsilon_{.8}$	any$_{.7}$	comments$_{.7}$
hi$_{.1}$	can$_{.1}$	not$_{.3}$	said$_{.2}$	any$_{.3}$	thing$_{.1}$	$\epsilon_{.1}$	these$_{.1}$	points$_{.1}$		the$_{.1}$	a$_{.1}$	new$_{.1}$	comment$_{.2}$
	$\epsilon_{.1}$		say$_{.1}$		things$_{.1}$		those$_{.1}$	$\epsilon_{.1}$		their$_{.1}$	air$_{.1}$	a$_{.1}$	commit$_{.1}$
			$\epsilon_{.1}$					pint$_{.1}$				$\epsilon_{.1}$	

Step 2: extract the not-punctuated *consensus decoding*

i cannot say anything at this point are there any comments

Step 3: compute the *N-best* hypotheses of punctuation (with `hidden-ngram`)

NBEST_0	-15.270	i cannot say anything		at this point	.	are there any comments	
NBEST_1	-15.317	i cannot say anything		at this point	.	are there any comments	?
NBEST_2	-16.275	i cannot say anything		at this point		are there any comments	?
NBEST_3	-16.322	i cannot say anything		at this point	?	are there any comments	?
NBEST_4	-17.829	i cannot say anything		at this point		are there any comments	.
NBEST_5	-18.284	i cannot say anything		at this point	?	are there any comments	
NBEST_6	-18.331	i cannot say anything		at this point		are there any comments	
NBEST_7	-18.473	i cannot say anything	.	at this point		are there any comments	
NBEST_8	-18.521	i cannot say anything	.	at this point		are there any comments	?
NBEST_9	-18.834	i cannot say anything		at this point	.	are there any comments	.

Step 4: compute the *punctuating CN* with *posterior probs* of multiple marks

i_1	cannot_1	say_1	anything_1	$\epsilon_{.9}$	at_1	this_1	point_1	$\cdot_{.7}$	are_1	there_1	any_1	comments_1	$?_{.6}$
				$\cdot_{.1}$				$\epsilon_{.2}$					$\epsilon_{.3}$
								$?_{.1}$					$\cdot_{.1}$

Step 5: *merge* the input CN and the punctuating CN

i$_{.9}$	cannot$_{.8}$	$\epsilon_{.7}$	say$_{.6}$	$\epsilon_{.7}$	anything$_{.8}$	at$_{.9}$	this$_{.8}$	point$_{.7}$	are$_{1}$	there$_{.8}$	$\epsilon_{.8}$	any$_{.7}$	comments$_{.7}$
hi$_{.1}$	can$_{.1}$	not$_{.3}$	said$_{.2}$	any$_{.3}$	thing$_{.1}$	$\epsilon_{.1}$	these$_{.1}$	points$_{.1}$		the$_{.1}$	a$_{.1}$	new$_{.1}$	comment$_{.2}$
	$\epsilon_{.1}$		say$_{.1}$		things$_{.1}$		those$_{.1}$	$\epsilon_{.1}$		their$_{.1}$	air$_{.1}$	a$_{.1}$	commit$_{.1}$
			$\epsilon_{.1}$					pint$_{.1}$				$\epsilon_{.1}$	

+

i$_{1}$	cannot$_{1}$	say$_{1}$	anything$_{1}$	$\epsilon_{.9}$	at$_{1}$	this$_{1}$	point$_{1}$	$._{.7}$	are$_{1}$	there$_{1}$	any$_{1}$	comments$_{1}$	$?_{.6}$
				$._{.1}$				$\epsilon_{.2}$					$\epsilon_{.3}$
								$?_{.1}$					$._{.1}$

Step 6: get the final *punctuated CN*

i$_{.9}$	cannot$_{.8}$	$\epsilon_{.7}$	say$_{.6}$	$\epsilon_{.7}$	anything$_{.8}$	$\epsilon_{.9}$	at$_{.9}$	this$_{.8}$	point$_{.7}$	$._{.7}$	are$_{1}$	there$_{.8}$	$\epsilon_{.8}$	any$_{.7}$	comments$_{.7}$	?$_{.6}$
hi$_{.1}$	can$_{.1}$	not$_{.3}$	said$_{.2}$	any$_{.3}$	thing$_{.1}$	$._{.1}$	$\epsilon_{.1}$	these$_{.1}$	points$_{.1}$	$\epsilon_{.2}$		the$_{.1}$	a$_{.1}$	new$_{.1}$	comment$_{.2}$	$\epsilon_{.3}$
	$\epsilon_{.1}$		say$_{.1}$		things$_{.1}$			those$_{.1}$	$\epsilon_{.1}$	?$_{.1}$		their$_{.1}$	air$_{.1}$	a$_{.1}$	commit$_{.1}$	$._{.1}$
			$\epsilon_{.1}$						pint$_{.1}$					$\epsilon_{.1}$		

Step 6: get the final *punctuated CN*

i$_{.9}$	cannot$_{.8}$	$\epsilon_{.7}$	say$_{.6}$	$\epsilon_{.7}$	anything$_{.8}$	$\epsilon_{.9}$	at$_{.9}$	this$_{.8}$	point$_{.7}$	.$_{.7}$	are$_{1}$	there$_{.8}$	$\epsilon_{.8}$	any$_{.7}$	comments$_{.7}$	?$_{.6}$
hi$_{.1}$	can$_{.1}$	not$_{.3}$	said$_{.2}$	any$_{.3}$	thing$_{.1}$	.$_{.1}$	$\epsilon_{.1}$	these$_{.1}$	points$_{.1}$	$\epsilon_{.2}$		the$_{.1}$	a$_{.1}$	new$_{.1}$	comment$_{.2}$	$\epsilon_{.3}$
	$\epsilon_{.1}$		say$_{.1}$		things$_{.1}$			those$_{.1}$	$\epsilon_{.1}$	?$_{.1}$		their$_{.1}$	air$_{.1}$	a$_{.1}$	commit$_{.1}$	.$_{.1}$
			$\epsilon_{.1}$						pint$_{.1}$					$\epsilon_{.1}$		

Notes

- this approach works with any speech input (1-best and CN) without punctuation and with partially punctuated input

Step 6: get the final *punctuated CN*

i$_{.9}$	cannot$_{.8}$	$\epsilon_{.7}$	say$_{.6}$	$\epsilon_{.7}$	anything$_{.8}$	$\epsilon_{.9}$	at$_{.9}$	this$_{.8}$	point$_{.7}$	.$_{.7}$	are$_{1}$	there$_{.8}$	$\epsilon_{.8}$	any$_{.7}$	comments$_{.7}$	?$_{.6}$
hi$_{.1}$	can$_{.1}$	not$_{.3}$	said$_{.2}$	any$_{.3}$	thing$_{.1}$	.$_{.1}$	$\epsilon_{.1}$	these$_{.1}$	points$_{.1}$	$\epsilon_{.2}$		the$_{.1}$	a$_{.1}$	new$_{.1}$	comment$_{.2}$	$\epsilon_{.3}$
	$\epsilon_{.1}$		say$_{.1}$		things$_{.1}$			those$_{.1}$	$\epsilon_{.1}$	?$_{.1}$		their$_{.1}$	air$_{.1}$	a$_{.1}$	commit$_{.1}$	.$_{.1}$
			$\epsilon_{.1}$						pint$_{.1}$					$\epsilon_{.1}$		

Notes

- this approach works with any speech input (1-best and CN) without punctuation and with partially punctuated input
- one system (with punctuated models) translates any input (text and speech)

Which is the better approach to add punctuation marks?

Which is the better approach to add punctuation marks?

- in the *source* as a *pre-processing* step

Which is the better approach to add punctuation marks?

- in the *source* as a *pre-processing* step
- in the *target* as a *post-processing* step
  - translate with not-punctuated models
  - add punctuation to the best translation (with `hidden-ngram`)

Which is the better approach to add punctuation marks?

- in the *source* as a *pre-processing* step
- in the *target* as a *post-processing* step
  - translate with not-punctuated models
  - add punctuation to the best translation (with `hidden-ngram`)

- evaluation
  - task: eval set 2006, TC-STAR English-to-Spanish
  - training data: FTE transcriptions of EPPS (36Mw English, 38Mw Spanish)
  - verbatim input (w/o punctuation), case-insensitive

approach	BLEU	NIST	WER	PER
target	42,23	9.72	46.12	34.38
source	**44.92**	**9.84**	**42.84**	**31.77**

Do multiple punctuation hypotheses help to improve translation quality?

Do multiple punctuation hypotheses help to improve translation quality?

- evaluation
  - verbatim (w/o punctuation)
  - case-insensitive

input	type	# punctuation hyps	BLEU	NIST	WER	PER
vrb		1	44.92	**9.84**	42.84	31.77
		1000	**45.33**	9.83	**42.58**	**31.59**

Do multiple punctuation hypotheses help to improve translation quality?

- evaluation
  - verbatim (w/o punctuation), 1-best
  - case-insensitive

input	type	# punctuation hyps	BLEU	NIST	WER	PER
vrb		1	44.92	**9.84**	42.84	31.77
		1000	**45.33**	9.83	**42.58**	**31.59**
asr	1-best	1	35.62	8.37	57.15	44.56
		1000	**36.01**	**8.41**	**56.78**	**44.39**

Do multiple punctuation hypotheses help to improve translation quality?

- evaluation
  - verbatim (w/o punctuation), 1-best, and CN
  - case-insensitive

input	type	# punctuation hyps	BLEU	NIST	WER	PER
vrb		1	44.92	**9.84**	42.84	31.77
		1000	**45.33**	9.83	**42.58**	**31.59**
asr	1-best	1	35.62	8.37	57.15	44.56
		1000	**36.01**	**8.41**	**56.78**	**44.39**
	CN	1	36.22	8.46	56.39	44.37
		1000	**36.45**	**8.49**	**56.17**	**44.19**

Create a phrase-pair lexicon

- take a case-sensitive parallel corpus
- word-align the corpus in direct and inverse directions (`GIZA++`)
- combine both word-alignments in one symmetric way:
  - `grow-diag-final`, `union`, and `intersection`
- extract phrase pairs from a symmetrized word-alignment
- add single word translation from direct alignment
- score phrase pairs according to word and phrase frequencies

Create a phrase-pair lexicon

- take a case-sensitive parallel corpus
- word-align the corpus in direct and inverse directions (`GIZA++`)
- combine both word-alignments in one symmetric way:
  – `grow-diag-final`, `union`, and `intersection`
- extract phrase pairs from a symmetrized word-alignment
- add single word translation from direct alignment
- score phrase pairs according to word and phrase frequencies

Ideas for improving the lexicon:

- use *case-insensitive* corpus for word-alignment, but case-sensitive extraction

Create a phrase-pair lexicon

- take a case-sensitive parallel corpus
- word-align the corpus in direct and inverse directions (`GIZA++`)
- combine both word-alignments in one symmetric way:
  - `grow-diag-final`, `union`, and `intersection`
- extract phrase pairs from a symmetrized word-alignment
- add single word translation from direct alignment
- score phrase pairs according to word and phrase frequencies

Ideas for improving the lexicon:

- use *case-insensitive* corpus for word-alignment, but case-sensitive extraction
- extract phrase pairs separately from more symmetrized word-alignments, concatenate them and compute their scores

How much improvement do we get?

How much improvement do we get?

- evaluation
  - task: IWSLT Chinese-to-English, 2006 eval set
  - training data: BTEC and dev sets ('03-'05)
  - weight optimization on 2006 dev set
  - verbatim input, case-sensitive

symmetrization	text for word-alignment	# phrase pairs	BLEU	NIST
grow-diag-final	case-sensitive	496K	20.50	5.57

How much improvement do we get?

- evaluation
  - task: IWSLT Chinese-to-English, 2006 eval set
  - training data: BTEC and dev sets ('03-'05)
  - weight optimization on 2006 dev set
  - verbatim input, case-sensitive

symmetrization	text for word-alignment	# phrase pairs	BLEU	NIST
grow-diag-final	case-sensitive	496K	20.50	5.57
"	case-insensitive	507K	21.86	5.59

How much improvement do we get?

- evaluation
  - task: IWSLT Chinese-to-English, 2006 eval set
  - training data: BTEC and dev sets ('03-'05)
  - weight optimization on 2006 dev set
  - verbatim input, case-sensitive

symmetrization	text for word-alignment	# phrase pairs	BLEU	NIST
grow-diag-final	case-sensitive	496K	20.50	5.57
"	case-insensitive	507K	21.86	5.59
+union	"	507K	22.35	6.20

How much improvement do we get?

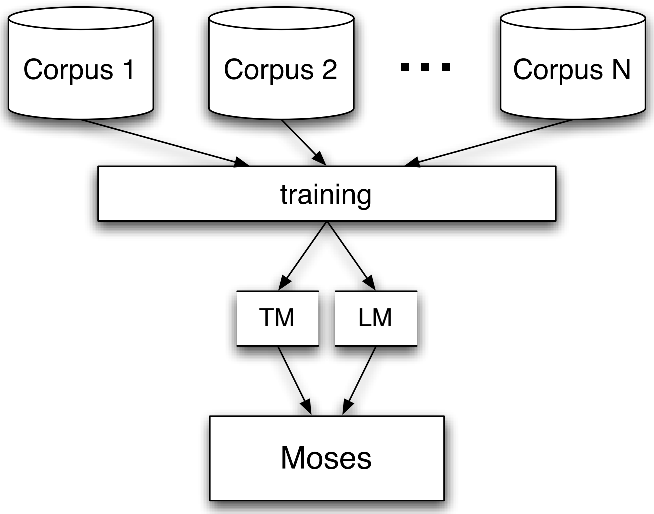
- evaluation
  - task: IWSLT Chinese-to-English, 2006 eval set
  - training data: BTEC and dev sets ('03-'05)
  - weight optimization on 2006 dev set
  - verbatim input, case-sensitive

symmetrization	text for word-alignment	# phrase pairs	BLEU	NIST
grow-diag-final	case-sensitive	496K	20.50	5.57
"	case-insensitive	507K	21.86	5.59
+union	"	507K	22.35	6.20
+intersection	"	5.2M	**22.71**	**6.31**

Multiple TMs and LMs

- *multiple training corpora*
  - non-homogeneous data (size, domain)
  - small corpus for domain adaptation

Multiple TMs and LMs

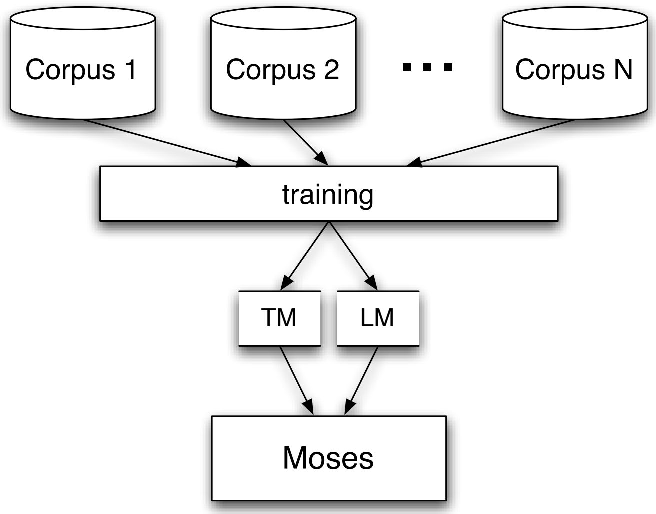


- *multiple training corpora*
  - non-homogeneous data (size, domain)
  - small corpus for domain adaptation

- *one TM* and *one LM*
  - concatenation of all corpora
  - corpus characteristics are (too?) smoothed

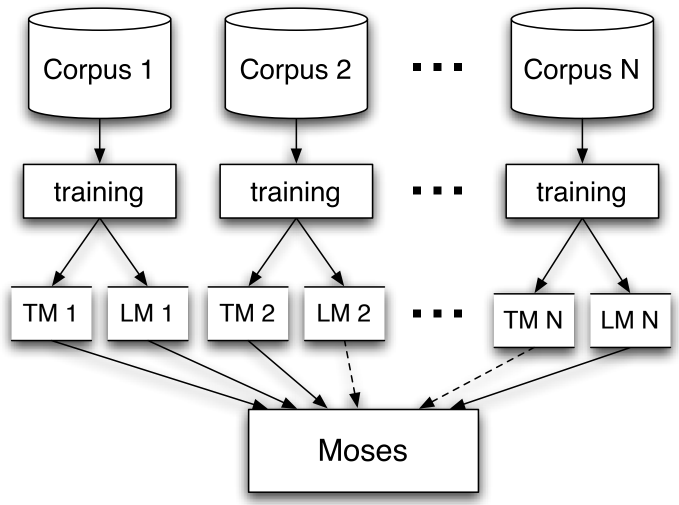
Multiple TMs and LMs

- *multiple training corpora*
  - non-homogeneous data (size, domain)
  - small corpus for domain adaptation



- *one TM* and *one LM*
  - concatenation of all corpora
  - corpus characteristics are smoothed

- *multiple TMs* and *multiple LMs*
  - *advantages*
    - ∗ more specialized models, more flexibility
    - ∗ easy combination/selection of models
    - ∗ effective (for TMs)
  - *drawbacks*
    - ∗ complexity of the model



How much improvement do we get?

How much improvement do we get?

- evaluation
  - task: IWSLT Italian-to-English, second half of 2007 dev set
  - training data:
    - baseline: BTEC, Named Entities, MultiWordNet and dev sets ('03-'06): 3.8M phrase pairs, 362K 4-grams
    - EU Proceedings (39M phrase pairs, 16M 4-grams)
    - Google Web 1T (336M 5-grams)
  - weight optimization on the first half of 2007 devset
  - verbatim input repunctuated with CN, case-insensitive

TM_1,LM_1	TM_2,LM_2	LM_3	OOV	BLEU	NIST
baseline	-	-	1.68	28.70	5.76

How much improvement do we get?

- evaluation
  - task: IWSLT Italian-to-English, second half of 2007 dev set
  - training data:
    - baseline: BTEC, Named Entities, MultiWordNet and dev sets ('03-'06): 3.8M phrase pairs, 362K 4-grams
    - EU Proceedings (39M phrase pairs, 16M 4-grams)
    - Google Web 1T (336M 5-grams)
  - weight optimization on the first half of 2007 devset
  - verbatim input repunctuated with CN, case-insensitive

TM_1,LM_1	TM_2,LM_2	LM_3	OOV	BLEU	NIST
baseline	-	-	1.68	28.70	5.76
"	-	web	"	29.66	5.83

How much improvement do we get?

- evaluation
  - task: IWSLT Italian-to-English, second half of 2007 dev set
  - training data:
    - baseline: BTEC, Named Entities, MultiWordNet and dev sets ('03-'06): 3.8M phrase pairs, 362K 4-grams
    - EU Proceedings (39M phrase pairs, 16M 4-grams)
    - Google Web 1T (336M 5-grams)
  - weight optimization on the first half of 2007 devset
  - verbatim input repunctuated with CN, case-insensitive

TM_1,LM_1	TM_2,LM_2	LM_3	OOV	BLEU	NIST
baseline	-	-	1.68	28.70	5.76
"	-	web	"	29.66	5.83
"	EP	"	**0.28**	**30.79**	**5.92**

1-best vs. Confusion Networks

1-best vs. Confusion Networks

task	input	BLEU
IE, ASR	`1bst`	41.51
	`cn`	**42.29***

* primary run

- CN outperforms 1-best

1-best vs. Confusion Networks

task	input	BLEU
IE, ASR	1bst	41.51
	cn	**42.29***
JE, ASR	1bst	39.46*
	cn	**39.69**

* primary run

- CN outperforms 1-best
- no inspection on CN for JE

Multiple TMs and LMs

Multiple TMs and LMs

task	TMs	LMs	BLEU
IE, clean	baseline	baseline	43.41
	+EP	+EP+web	**44.32***

* primary run

Multiple TMs and LMs

task	TMs	LMs	BLEU
IE, clean	baseline	baseline	43.41
	+EP	+EP+web	**44.32***
IE, ASR, CN	baseline	baseline	40.74
	+EP	+EP+web	**41.51***

* primary run

Multiple TMs and LMs

task	TMs	LMs	BLEU
IE, clean	baseline	baseline	43.41
	+EP	+EP+web	**44.32***
IE, ASR, CN	baseline	baseline	40.74
	+EP	+EP+web	**41.51***
CE, clean	baseline	baseline	**35.08**
	"	+web	33.94
	+LDC	"	34.72*

* primary run

- additional TMs improves performance (+0.77 BLEU)
- Google Web LM severely affects performance on CE (-1.14 BLEU)

Future work

- punctuation insertion in other languages (Chinese, Japanese)
- use of *caseing* CN to for case restoring

Future work

- punctuation insertion in other languages (Chinese, Japanese)
- use of *caseing* CN to for case restoring

- automatic way of selecting corpora

- punctuation insertion in other languages (Chinese, Japanese)
- use of *caseing* CN to for case restoring

- automatic way of selecting corpora

- further inspection on the use of Google Web corpus

Thank you!

Chinese-to English

- word-alignment on ci texts, `grow-diag-final` + `union` + `inter`
- case sensitive models
- distortion models: distance-based and `orientation-bidirectional-fe`
- (stack size, translation option limit, reordering limit)=(2000,50,7)
- BTEC and dev sets ('03-'07) (TM_1: 5.9M phrase pairs, LM_1: 39K 6-grams)
  LDC: (TM_2: 27M phrase pairs)
  Google Web (LM_2: 336M 5-grams)
- 5 official runs

Japanese-to English

- word-alignment on ci texts, `grow-diag-final` + `union` + `inter`
- case sensitive models
- distortion models: distance-based and `orientation-bidirectional-fe`
- (stack size, translation option limit, reordering limit)=(2000,50,7)
- BTEC and dev sets ('03-'07) (TM_1: 9.1M phrase pairs, LM_1: 39K 6-grams)
  Reuters: (TM_2, 176K phrase pairs)
- 6 official runs

Italian-to English

- word-alignment on ci texts, `grow-diag-final` + `union`
- case insensitive TMs and LMs and case restoring
- distortion models: distance-based
- (stack size, translation option limit, reordering limit)=(200,20,6)
- BTEC NE, MWN, dev sets ('03-'07) (TM_1: 3.8M phrase pairs, LM_1: 362K 4-grams)
  EU Proceedings: (TM_2: 39M phrase pairs, LM_2: 16M 4-grams)
  Google Web (LM_3: 336M 5-grams)
- rescoring with 5K-best translations
- case-restoring with a 4-gram LM
- 12 official runs

- **Toolkit for SMT**:
  - translation of both text and CN inputs
  - incremental pre-fetching of translation options
  - handling multiple lexicons and LMs
  - handling of huge LMs and LexMs (up to Giga words)
  - on-demand and on-disk access to LMs and LexMs
  - factored translation model (surface forms, lemma, POS, word classes, ...)
- **Multi-stack DP-based decoder**:
  - theories stored according to the coverage size
  - synchronous on the coverage size
- **Beam search**:
  - deletion of less promising partial translations:
  - histogram and threshold pruning
- **Distortion limit**: reduction of possible alignments
- **Lexicon pruning**: limit the amount of translation options per span

- *log-linear statistical model*

- features of the *first* pass
  - (multiple) language models
  - direct and inverted word- and phrase-based (multiple) lexicons
  - word and phrase penalties
  - reordering model: distance-based and lexicalized (CE, JE)

- (additional) features of the *second* pass (IE)
  - direct and inverse IBM Model 1 lexicon scores
  - weighted sum of n-grams relative frequencies ($n = 1, ...4$) in N-best list
  - the reciprocal of the rank
  - counts of hypothesis duplicates
  - n-gram posterior probabilities in N-best list [Zens, 2006]
  - sentence length posterior probabilities [Zens, 2006]