Spoken Language Translation through Confusion Network decoding

Nicola Bertoldi
FBK-irst,Trento, Italy

Edinburgh, 20 April 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

Outline

- Spoken Language Translation
  - task
  - specific issues
  - formal definition
  - common approaches
- SLT by Confusion Network decoding
  - definition of Confusion Network
  - CN decoding algorithm
  - efficiency
  - advanced features of Moses and CN
  - evaluation
- Other applications of CN decoding

Credits: R. Zens (RWTH, Aachen), M. Federico (FBK-irst, Trento)

Spoken Language Translation

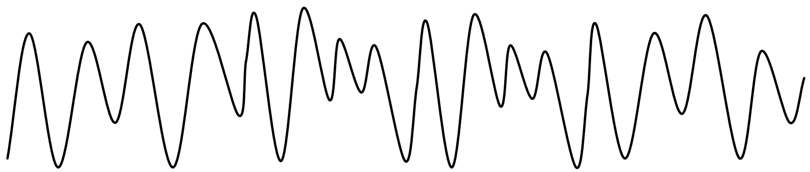
- **Translation from speech input**
  - recent and challenging task of Machine Translation

- **Combination of ASR and MT**:
  - *cascade* of ASR and MT systems
  - different *interfaces*, different approaches

- **Harder** than text translation
  - input genre is more *spontaneous*
  - ASR is far from being a solved problem
    - *transcription errors* are generated
    - *punctuation* is missing (or post-added)
    - *case information* is (often) missing

SLT issues

"and ... then ... here we have seen success"

Speech Signal: 

Correct Transcription:	and	@ehm	then	@mh	here	we	have	seen		success
Best ASR Transcription:	and	me		@mh	there	we	have	seen	a	success

- transcription errors: substitution, insertion, deletion
- spontaneous speech phenomena: hesitation, repetition

SLT issues

- spontaneous speech phenomena can cause
  - *transcription errors*:
    and @ehm then here we have seen ⟶ and me there we have seen
    @uh I see ⟶ you see
  - *bad-formed* sentence
    mister mister @ehm mister maaten

- transcription errors modify both *meaning* and *syntax*:
  - *semantic errors*:
    mister maaten has the floor ⟶ mister martin has the floor
    market ⟶ mark at   ate ⟶ eight   you ⟶ e.u.
  - *syntactic errors*:
    I move on to the committee ⟶ I'll move onto the committee
    @uh I see ⟶ you see

SLT issues

- transcription and translation quality *strongly correlate*
  – the better transcription, the better translation

- ASR quality increases in a set of transcription hypotheses

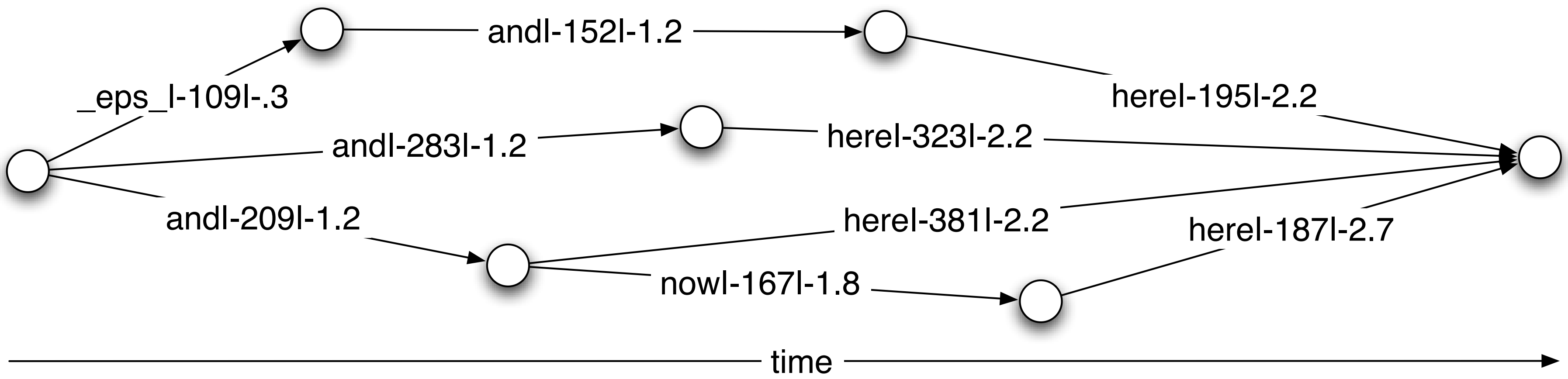
- but unfortunately the *oracle* is unknown

$\Longrightarrow$ **translation of as many alternative transcriptions** as possible

- In principle:
  – all transcriptions in the *Word Graph* generated by the ASR system

Word Graph

- large amount of transcription hyps produced by the ASR system
- arcs are labelled with words and ASR scores
- nodes are labelled with starting and ending times of words
- *redundancy* is high (from the point of view of MT):
  - many paths represent the same hyp differing just in timestamps
- topology is *complex* (from the point of view of MT):
  - word-coverage and word-reordering are hard to handle

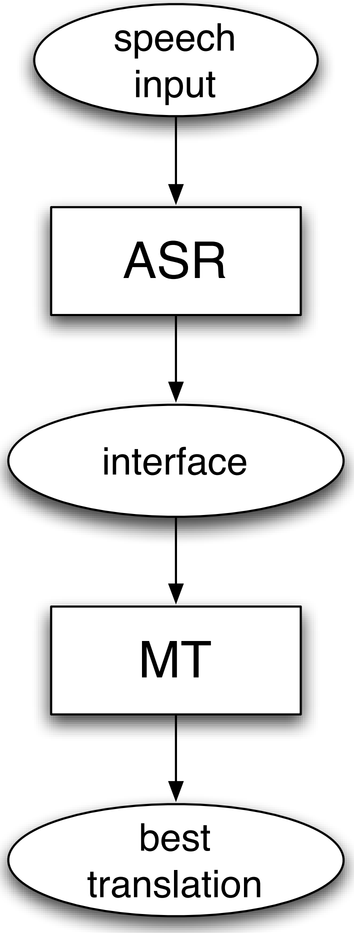


Approaches to SLT

- different *approximations* of a WG
- different *interfaces*:
  - 1-best, N-best, **confusion network**
  - full word graph
- *dedicated* MT decoder

- Finite State Transducer:
  - ASR and MT models merged into one finite-state network
  - a transducer decodes the input speech in one shot
  - difficult scaling up to very large domains

- [Casacuberta et al., CSL, 2004]



Statistical Spoken Language Translation

Given a *speech input* $\mathbf{o}$ in the source language,
find the *best translation* through the following approximate criterion:

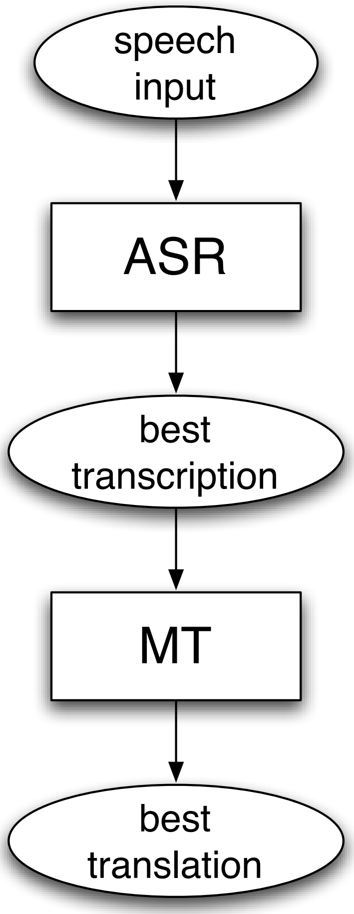
$$\begin{aligned}\mathbf{e}^* = \arg\max_{\mathbf{e}} \Pr(\mathbf{e} \mid \mathbf{o}) &= \arg\max_{\mathbf{e}} \sum_{\mathbf{f} \in \mathcal{F}(\mathbf{o})} \Pr(\mathbf{e}, \mathbf{f} \mid \mathbf{o}) \\ &\approx \arg\max_{\mathbf{e}} \max_{\mathbf{f} \in \mathcal{F}(\mathbf{o})} \Pr(\mathbf{e}, \mathbf{f} \mid \mathbf{o})\end{aligned}$$

- $\mathcal{F}(\mathbf{o})$ is any **set of possible transcriptions** of $\mathbf{o}$
  – interface between ASR and MT
- $\Pr(\mathbf{e}, \mathbf{f} \mid \mathbf{o})$ is any **phrase-based speech translation model**
- the actual transcription $\mathbf{f}$ is regarded as a hidden variable
- approximation simplifies the search algorithm

1-best Decoder

- translation of the *first best* transcription only
- use of a *standard MT system* of text

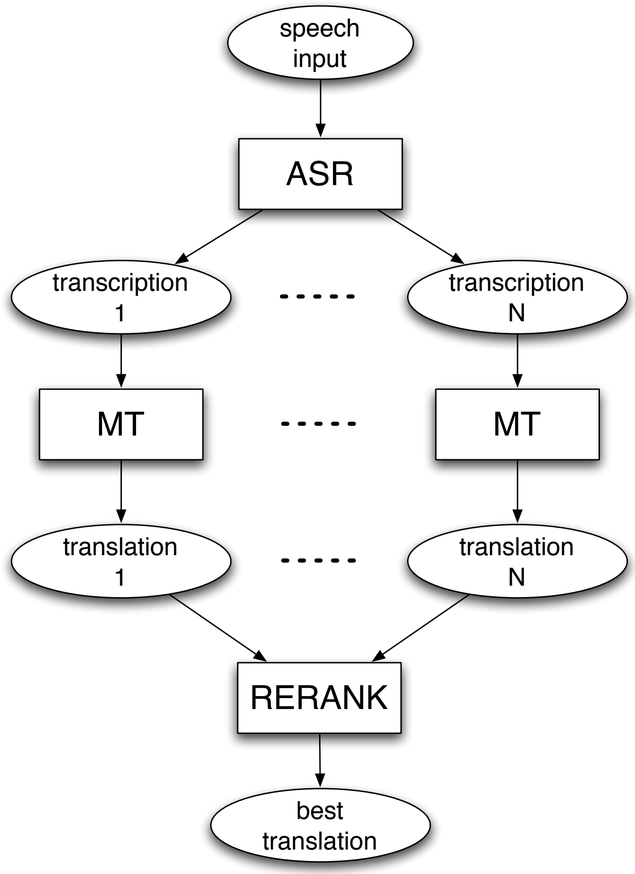
- no multiple transcriptions
- impossible recover from ASR errors



N-best Decoder

- translation of *N-best* transcription hypotheses
- *rerank* with additional ASR scores
  – acoustic likelihood and source LM

1	and there we have seen a success	-217	-12
2	and there we have seen success	-198	-9
....			
8	and then here we have seen success	-215	-21
9	and now here we have seen a success	-265	-3
....			



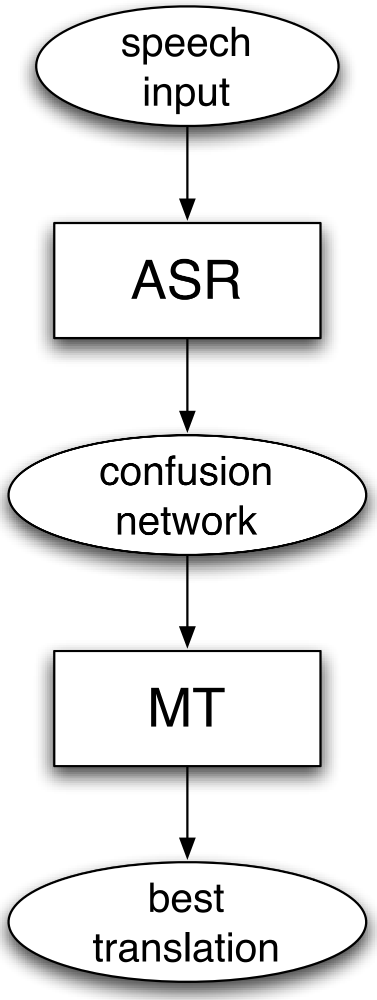
- possible recover from ASR errors
- no exploitation of overlaps among N-best

Confusion Network Decoder

- translation of a **confusion network**,
  a *compact structure* approximating a WG

- exploitation of multiple transcription hypotheses
- exploitation of overlaps among hypotheses
- extension of a standard text decoder

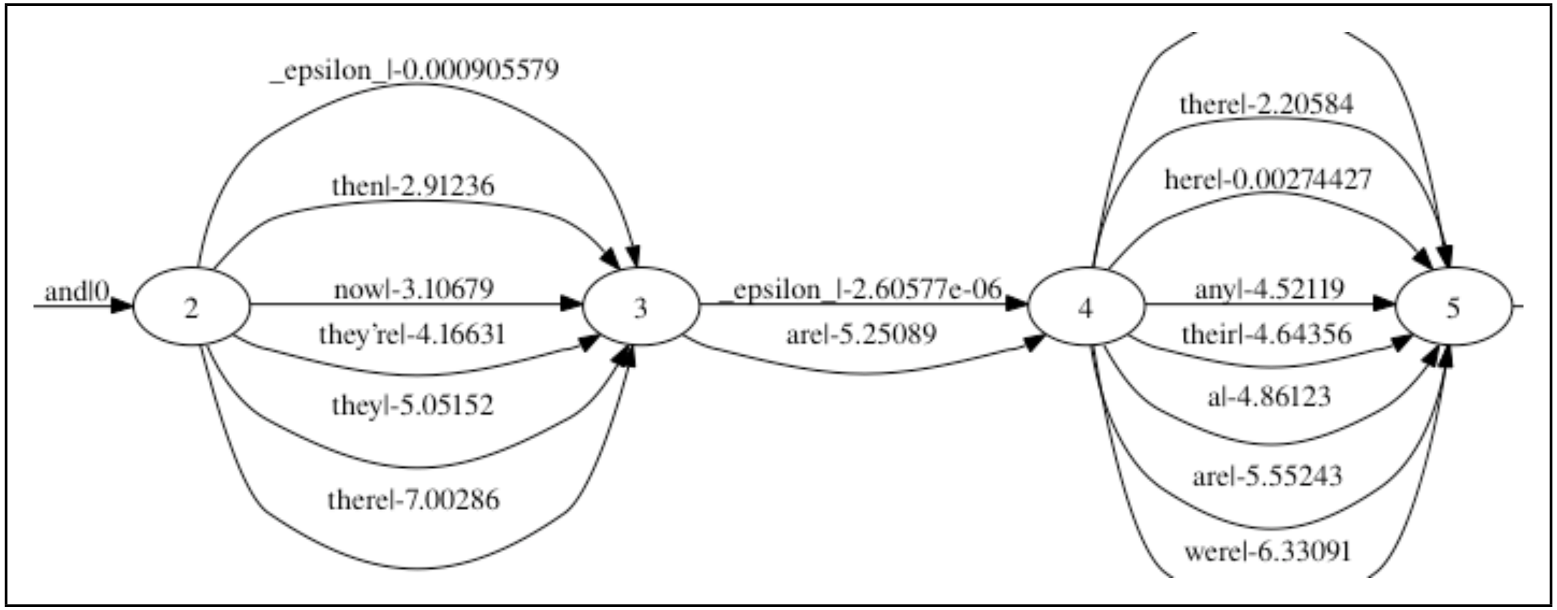
- [ASRU,2005], [ICASSP, 2007], Moses' doc



Confusion Network

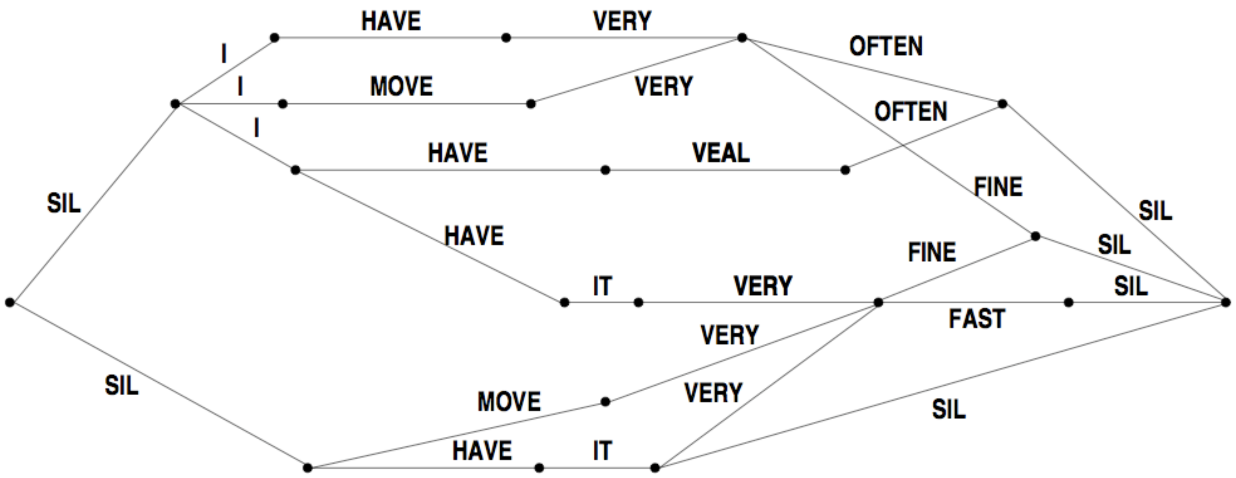
A **Confusion Network** approximates a WG by a linear network, s.t.:

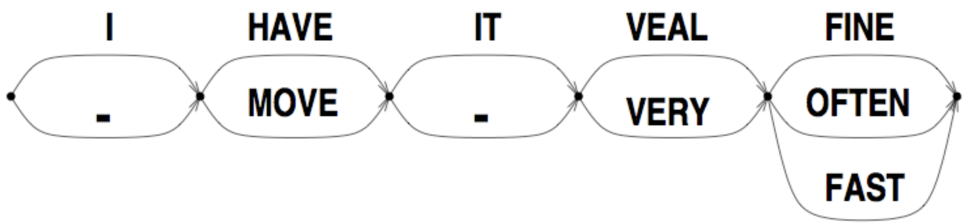
- arcs are labeled with words or with the *empty word* (ϵ-word)
- arcs are weighted with word *posterior probabilities*
- paths are a superset of those in the word graph
- paths can have different lengths



Extraction of CN from WG

- *cluster nodes* with close timestamps
- possibly *introduce special arcs* for empty-words
- *compute word posterior probabilities* exploiting ASR scores



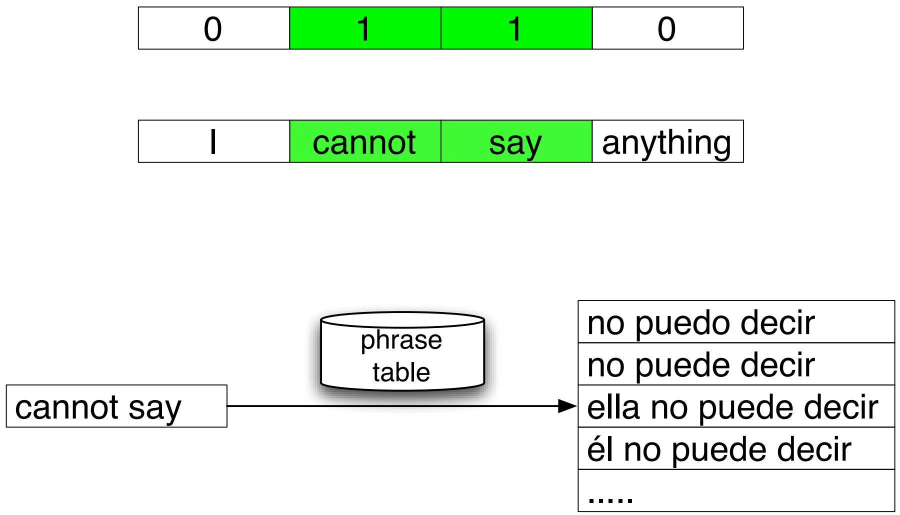


[Mangu's PhD. thesis, 2000]

Statistical model for CN decoding

- Translation Model is a *log-linear* combination of features
- Features are defined in terms of *phrases*
- Standard feature functions for text decoder:
  - Language Models
  - Distortion Model
  - Lexicon Model (LexM)
  - Phrase and Word Penalties
- *Specific feature functions for Confusion Network* (CM)
  - **likelihood of the path** into the source CN: product of word posterior probs
  - **number of words** in the path (optional)
- *LexM* and *CM* depend on the source phrase:
  - different paths in a span give different scores

Translation from text

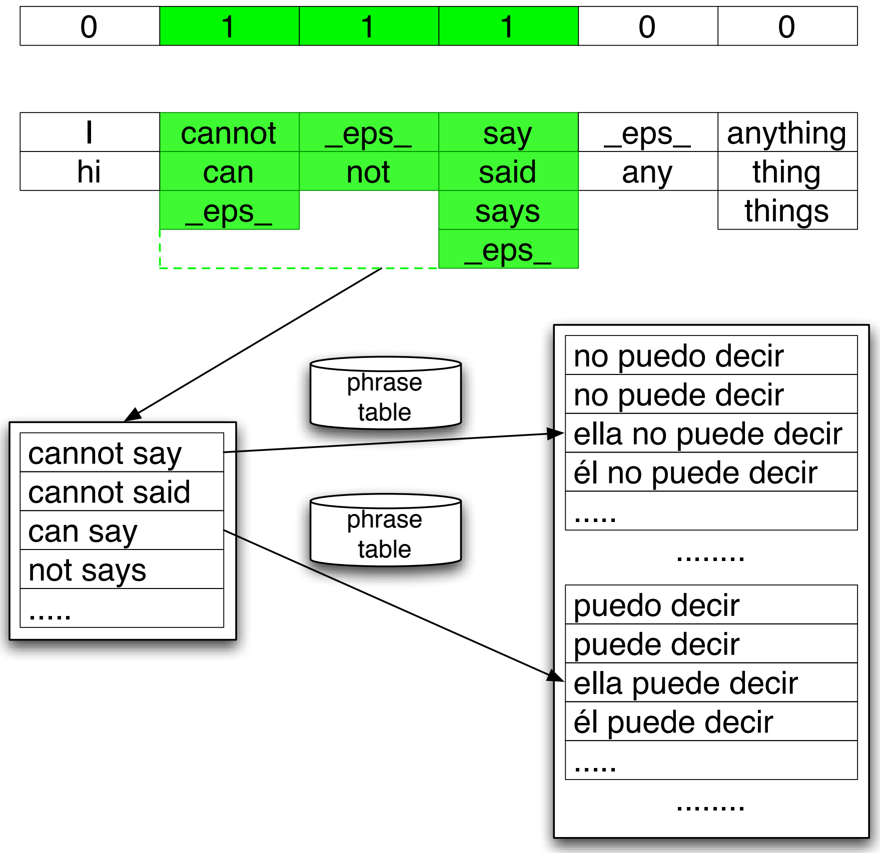


- **cover** a not yet covered *span*
  - *one source phrase*
- **retrieve** all translation options
  - looking up into the phrase table
- **compute** feature scores
- **recombine** hypotheses
- ...

Translation from Confusion Network

Extension of the translation from text

- **cover** a not yet covered *span*
  - *many source phrases*
- **retrieve** all translation options
  - for all source phrases in the span
  - looking up into the phrase table
- **compute** scores
- **recombine** hypotheses
- ...



Issues of CN Decoding

- Number of paths grows **exponentially** with span length
- Look-up of translations for a huge number of source phrases
- *Enumeration* of all alternatives is *unfeasible*
- and *dummy*!

Indeed:

- Paths can correspond to phrases without translations

those$_{0.92}$	$\epsilon_{0.99}$	were$_{0.99}$
$\epsilon_{0.07}$	was$_{6e-5}$	well$_{7e-5}$
as$_{6e-4}$	is$_{1e-5}$	ϵ_{1e-5}
there$_{5e-5}$	who$_{2e-6}$	who$_{1e-5}$
who$_{1-5}$		was$_{8e-6}$
who's$_{5e-6}$		

Issues of CN Decoding

- different paths into a span can correspond to the same phrase (who was)
  - different CM score

those$_{0.92}$	$\epsilon_{0.99}$	were$_{0.99}$
$\epsilon_{0.07}$	was$_{6e-5}$	well$_{7e-5}$
as$_{6e-4}$	is$_{1e-5}$	ϵ_{1e-5}
there$_{5e-5}$	who$_{2e-6}$	who$_{1e-5}$
who$_{1e-5}$		was$_{8e-6}$
who's$_{5e-6}$		

those	ϵ	were
ϵ	was	well
as	is	ϵ
there	who	who
who		was
who's		

those	ϵ	were
ϵ	was	well
as	is	ϵ
there	who	who
who		was
who's		

- different phrases into the same span can have equal translation
  - who's who and who is who translates into quién es quién
  - different CM and LexM scores

those	ϵ	were
ϵ	was	well
as	is	ϵ
there	who	who
who		was
who's		

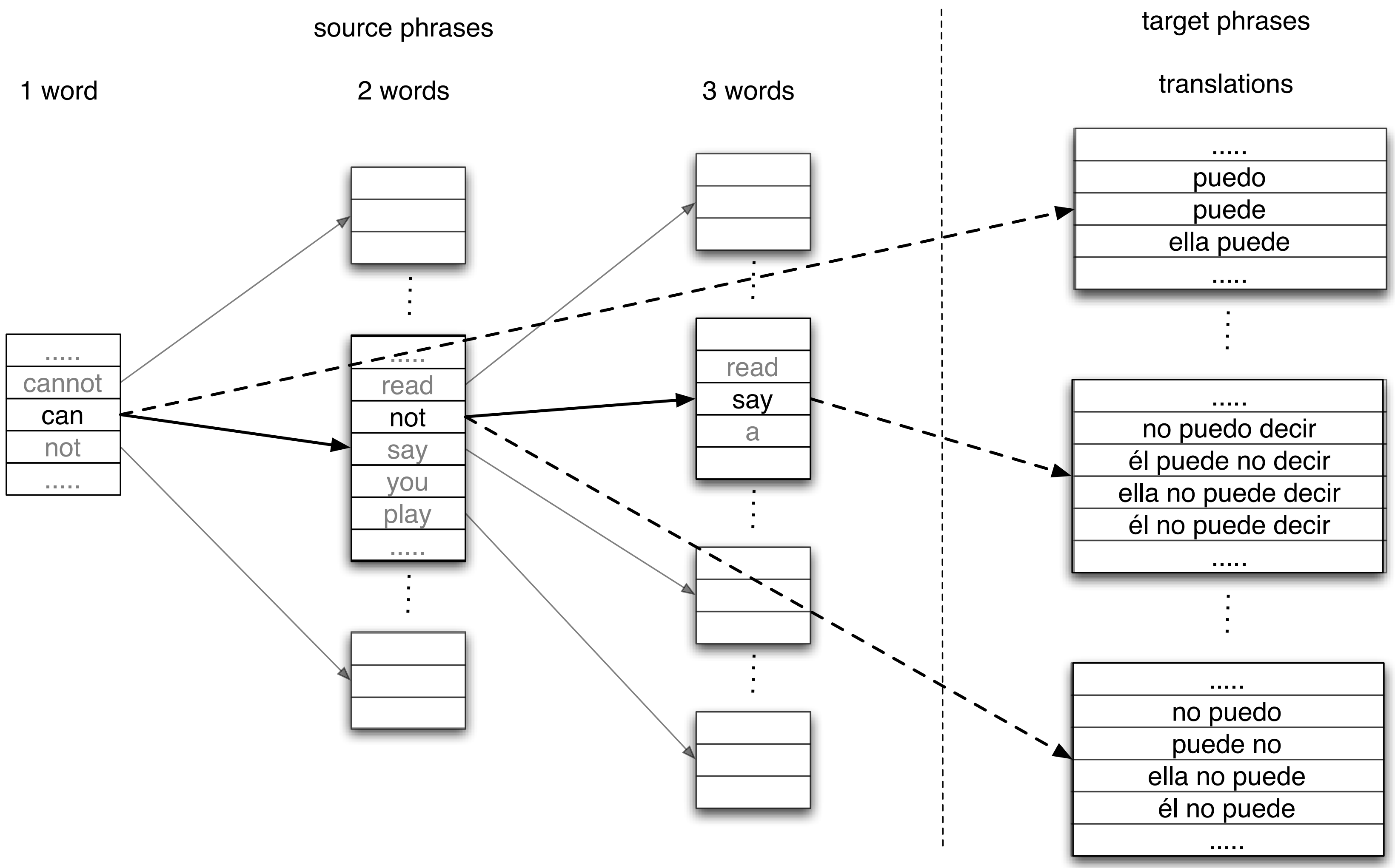
those	ϵ	were
ϵ	was	well
as	is	ϵ
there	who	who
who		was
who's		

Solution for an efficient CN decoding

- **Optimization of the retrieval of the translation options** by:
  - representing source entries of the phrase-table as *prefix-trees*
  - *incrementally pre-fetching* translation options
  - *early recombining* translation options

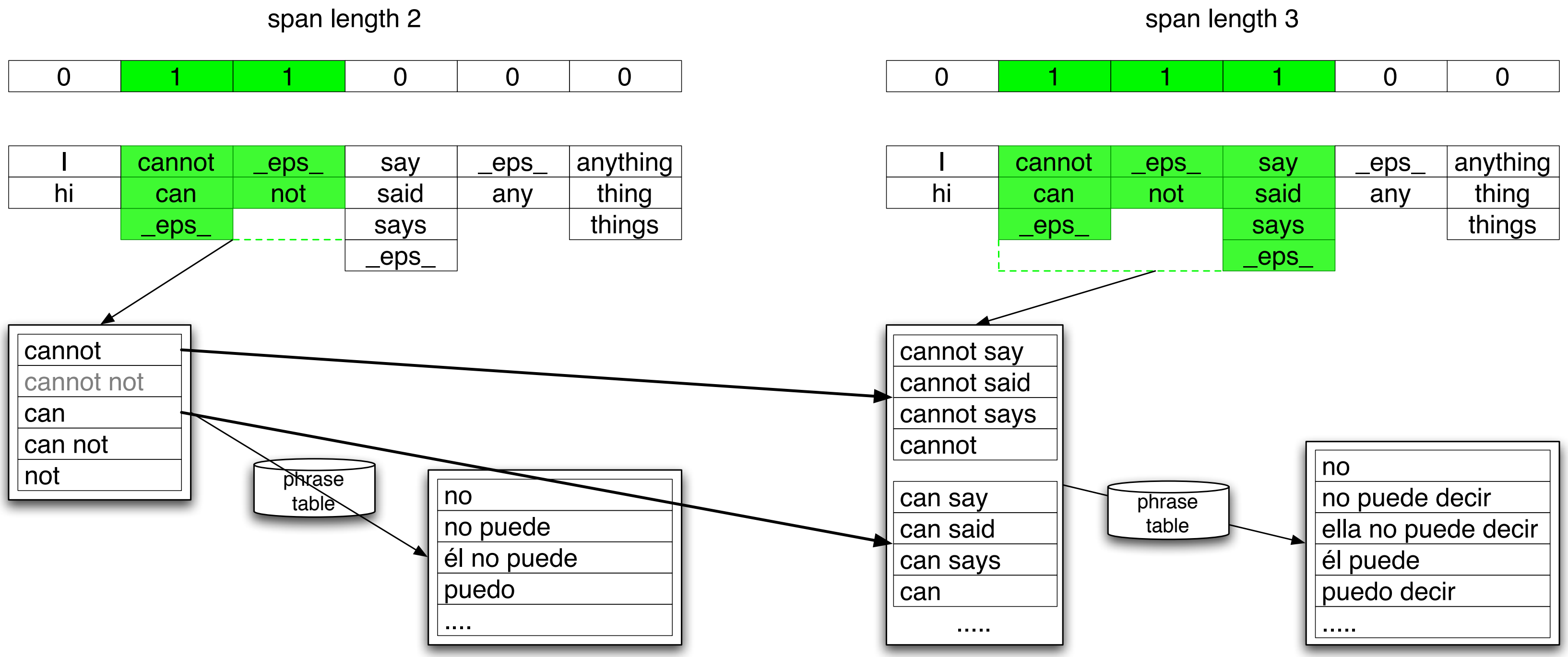
- **Once translation options are generated, usual decoding applies.**

Prefix-tree representation of phrase table



Incremental pre-fetching of translation options

- collect translation options *incrementally over the span length*
  – exploit knowledge about shorter span
- *once* and *before decoding*



- *worst case* (all phrases are present) is still exponential, but *never happens*

Early recombination

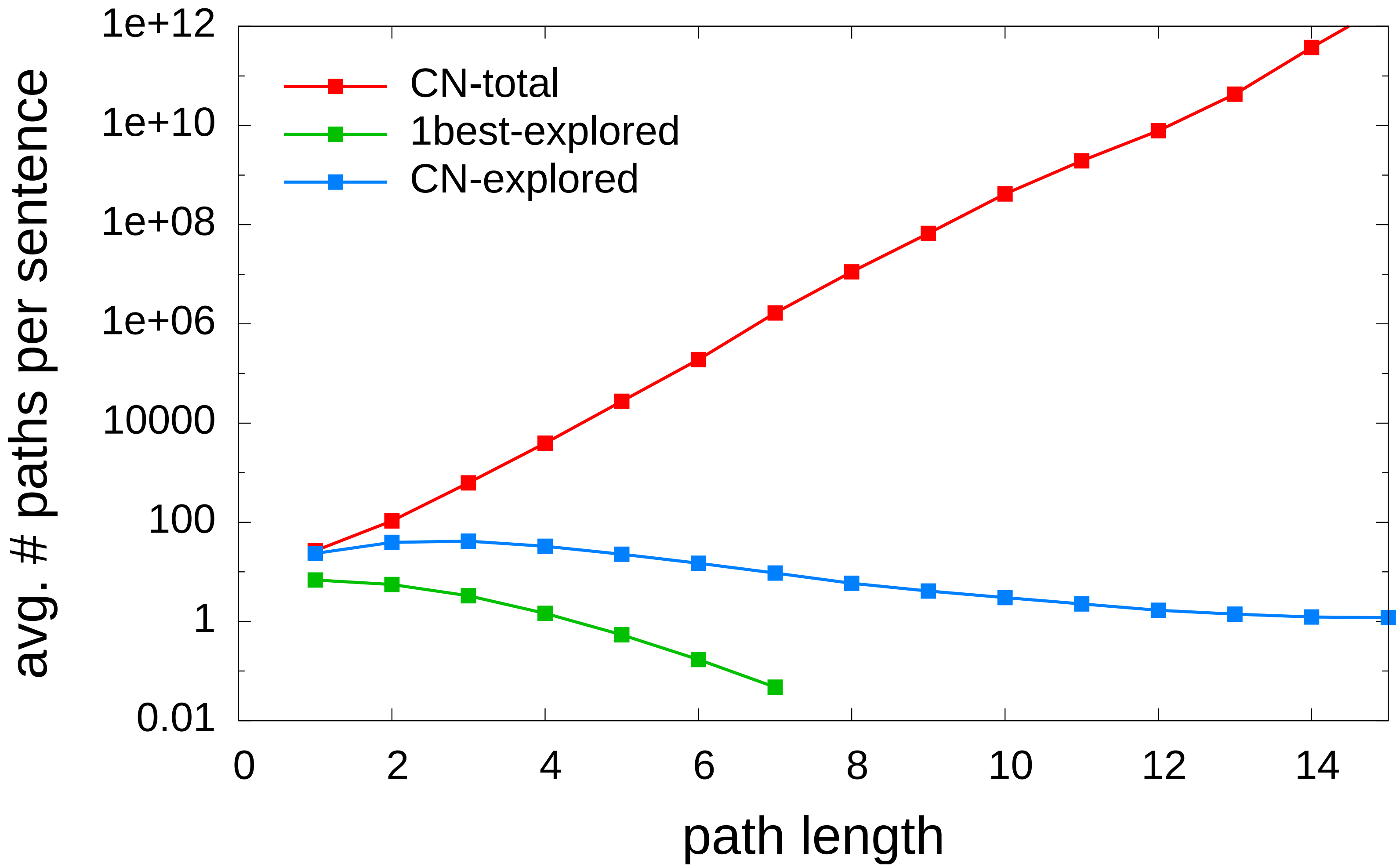
- *Different phrases* into the same span can have the *same translation*
- *Different LexM* and *CM* scores, the other are equal
- *Undistinguishable* from the decoder

- Take the *best path* only (and its scores)

- Use $LexM(span, e)$ and $CM(span)$, instead of $LexM(f, e)$ and $CM(f)$

$$\begin{aligned} LexM(span, e) &= LexM(\hat{f}, e) \\ CM(span, e) &= CM(\hat{f}, e) \\ \hat{f} &= \arg\max_{f \in span} \lambda_{LexM} LexM(f, e) + \lambda_{CM} CM(f) \end{aligned}$$

Efficiency of Search Algorithm



CN decoding in Moses

- Moses implements CN decoding

- *Factored models*
  – alternative over the full factor space

Haus|N	der|ART	Zeitung|N
aus|PREP	des|ART	$\epsilon \| \epsilon$
aus|ADV	$\epsilon \| \epsilon$	Zeitungs|N
$\epsilon \| \epsilon$	drei|N	Zeitungen|N

- *Lexicalized Distortion Models*
  – conditioned on the best path inside a span

CN decoding: results

- Spanish-English EPPS 2006 Evaluation

Input		Output			
type	WER	BLEU	NIST	PER	WER
`verbatim`	0.0	48.00	9.864	31.19	40.96
`cn-oracle`	8.45	44.12	9.356	34.37	44.95
`cons-dec`	23.30	36.98	8.550	39.17	49.98
`cn`	8.45	39.17	8.716	38.64	49.52
`1-best`	22.41	37.57	8.590	39.24	50.01
`5-best`	18.61	38.68	8.694	38.55	49.33
`10-best`	17.12	38.61	8.694	38.69	49.46

- Relative Improvement in BLEU: 30% (wrt to oracle)
- CN decoding speed is 2 times slower

CN decoding: results

- Moses vs. Irst-05 vs. Irst-06

Input		Output		
type	WER	BLEU		
		Irst-05	Irst-06	Moses
verbatim	0.0	40.84	44.64	48.00
1-best	14.61	36.64	39.67	42.84
cons-dec	14.46	36.54	39.65	42.92
cn	11.61	37.21	40.00	43.51

- Irst-06 was top system
- Irst-05 and Irst-06 translate pruned confusion networks
- Irst-05 translates CN 18 times slower than text

Other applications of CN decoder

- *CN represents ambiguity*
  - variations, alternatives, errors
- CN decoder *disambiguates* and *translates* in one shot:
  - **insertion of punctuation** and case restoring in translation
- CN decoder is also a *tagger*:
  - POS tagging, case restoring
  - Word Sense Disambiguation, NE Recognition, OCR, etc.
  - using monotone translation
  - using ad-hoc lexicon models and LMs

I@P	read@VP	a@R	book@N
	read@VPP		book@VP
	read@VI		book@VI

thank	you	mr.	bond
Thank	You	Mr.	Bond

Punctuating Confusion Networks

Confusion network without punctuation

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$_{.1}$				$\epsilon_{.1}$	

Consensus decoding

i cannot say anything at this point are there any comments

Punctuating confusion network

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$_{1}$	?$_{.6}$
				$\cdot_{.1}$				$\epsilon_{.2}$					$\epsilon_{.3}$
								?$_{.1}$					$\cdot_{.1}$

Punctuated confusion network

i$_{.9}$	cannot$_{.8}$	$\epsilon_{.7}$	say$_{.6}$	$\epsilon_{.7}$	anything$_{.8}$	$\epsilon_{.9}$	at$_{.9}$	this$_{.8}$	point$_{.7}$	$\cdot_{.7}$	are$_{1}$	there$_{.8}$	$\epsilon_{.8}$	any$_{.7}$	comments$_{.7}$	?$_{.6}$
hi$_{.1}$	can$_{.1}$	not$_{.3}$	said$_{.2}$	any$_{.3}$	thing$_{.1}$	$\cdot_{.1}$	$\epsilon_{.1}$	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$_{.1}$	$\cdot_{.1}$
			$\epsilon_{.1}$						pint$_{.1}$					$\epsilon_{.1}$		

Punctuating Confusion Networks: Results

- ASR 1-best output vs. confusion network
- 1-best punctuation vs. punctuating CN (from 1K-best)

Spanish-English EPPS Eval06					
ASR type	punctuation	BLEU	NIST	WER	PER
1-best	1-best	35.62	8.37	57.15	44.56
	CN	36.01	8.41	56.78	44.39
CN	1-best	36.22	8.46	56.39	44.37
	CN	36.45	8.49	56.17	44.19

Conclusion

- Spoken Language Translation
- SLT system:
  - combination of ASR and MT through Confusion Network
  - effective representation of a huge number of transcription hypotheses
- Efficient search algorithm for CN-based SMT:
  - prefix-tree representation and pre-fetching of lexicon models
  - early recombination of translation options
- Moses system:
  - CN decoding
  - state-of-the-art for SLT (translation performance and decoding speed)
  - slight improvement of CN decoder vs. 1-best decoder
- Moses for enriched translation
- Moses for tagging

Thank you!