Machine Learning For Machine Translation

An Introduction to Statistical Machine Translation

Prof. Pushpak Bhattacharyya,
Anoop Kunchukuttan, Piyush Dungarwal, Shubham Gautam
{pb,anoopk,piyushdd,shubhamg}@cse.iitb.ac.in

Indian Institute of Technology Bombay
Center for Indian Language Technology
http://www.cfilt.iitb.ac.in

ICON-2013: 10th International Conference on Natural Language Processing
18th December 2013, C-DAC NOIDA

Motivation for MT

- MT: NLP Complete
- NLP: AI complete
- AI: CS complete
- How will the world be different when the language barrier disappears?
- Volume of text required to be translated currently exceeds translators' capacity (demand > supply).
  - *Solution*: automation

Roadmap (1/4)

- Introduction
  - MT Perspective
  - Vauquois Triangle
  - MT Paradigms
  - Indian language SMT
  - Comparable to Parallel Corpora
- Word based Models
  - Word Alignment
  - EM based training
  - IBM Models

Roadmap (2/4)

- Phrase Based SMT
  - Phrase Pair Extraction by Alignment Templates
  - Reordering Models
  - Discriminative SMT models
  - Overview of Moses
  - Decoding
- Factor Based SMT
  - Motivation
  - Data Sparsity
  - Case Study for Indian languages

Roadmap (3/4)

- Hybrid Approaches to SMT
  - Source Side reordering
  - Clause based constraints for reordering
  - Statistical Post-editing of ruled based output
- Syntax Based SMT
  - Synchronous Context Free Grammar
  - Hierarchical SMT
  - Parsing as Decoding

Roadmap (4/4)

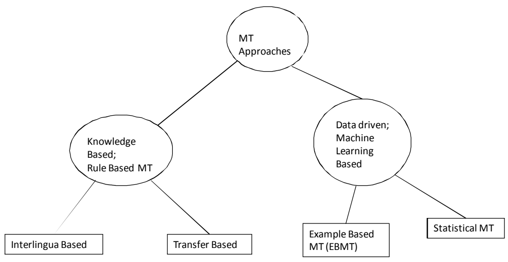
- MT Evaluation
  - Pros/Cons of automatic evaluation
  - BLEU evaluation metric
  - Quick glance at other metrics: NIST, METEOR, etc.
- Concluding Remarks

INTRODUCTION

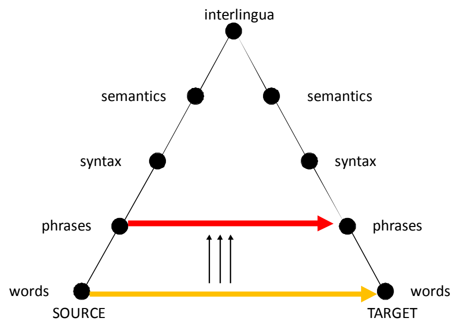
Set a perspective

- When to use ML and when not to
  - "Do not learn, when you know"/"Do not learn, when you can give a rule"
  - What is difficult about MT and what is easy
- Alternative approaches to MT (not based on ML)
  - What has preceded SMT
- SMT from Indian language perspective
- Foundation of SMT
  - Alignment

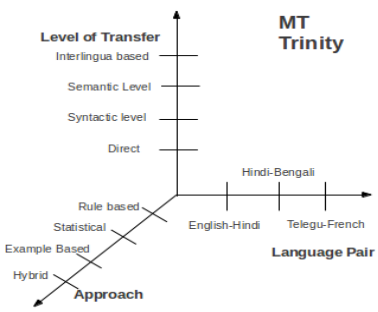
Taxonomy of MT systems



MT Approaches



MACHINE TRANSLATION TRINITY



Why is MT difficult?

Language divergence

Why is MT difficult: Language Divergence

- One of the main complexities of MT: *Language Divergence*
- Languages have different ways of expressing meaning
  - Lexico-Semantic Divergence
  - Structural Divergence

Our work on English-IL Language Divergence with illustrations from Hindi
(Dave, Parikh, Bhattacharyya, Journal of MT, 2002)

Languages differ in expressing thoughts: Agglutination

Finnish: "istahtaisinkohan"

English: "I wonder if I should sit down for a while"

Analysis:

- ist + "sit", verb stem
- ahta + verb derivation morpheme, "to do something for a while"
- isi + conditional affix
- n + 1st person singular suffix
- ko + question particle
- han a particle for things like reminder (with declaratives) or "softening" (with questions and imperatives)

Language Divergence Theory: *Lexico-Semantic Divergences* (few examples)

- Conflational divergence
  - F: vomir; E: to be sick
  - E: *stab;* H: *chure se maaranaa (knife-with hit)*
  - S: *Utrymningsplan;* E: *escape plan*
- Categorial divergence
  - Change is in POS category:
  - *The play is on_PREP (vs. The play is Sunday)*
  - *Khel chal_rahaa_haai_VM (vs. khel ravivaar ko haai)*

Language Divergence Theory: *Structural Divergences*

- SVO→SOV
  - E: *Peter plays basketball*
  - H: *piitar basketball kheltaa haai*
- Head swapping divergence
  - E: *Prime Minister of India*
  - H: *bhaarat ke pradhaan mantrii (India-of Prime Minister)*

Language Divergence Theory: *Syntactic Divergences* (few examples)

- Constituent Order divergence
  - E: *Singh, the PM of India, will address the nation today*
  - H: *bhaarat ke pradhaan mantrii, singh, … (India-of PM, Singh…)*
- Adjunction Divergence
  - E: *She will visit here in the summer*
  - H: *vah yahaa garmii meM aayegii (she here summer-in will come)*
- Preposition-Stranding divergence
  - E: *Who do you want to go with?*
  - H: *kisake saath aap jaanaa chaahate ho? (who with…)*

Vauquois Triangle

Kinds of MT Systems

(point of entry from source to the target text)

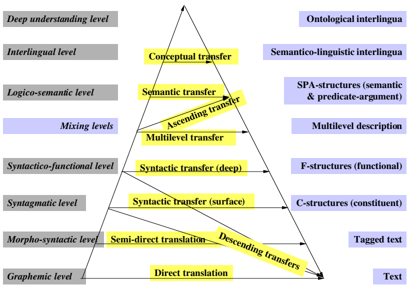
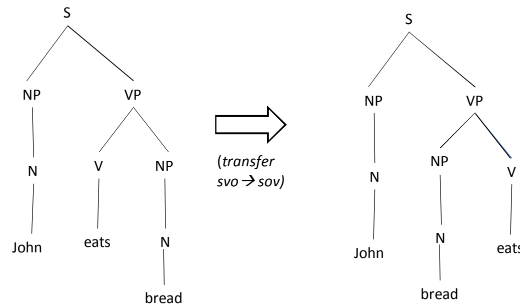


Illustration of transfer SVO→SOV



Universality hypothesis

Universality hypothesis: At the level of “deep meaning”, all texts are the “same”, whatever the language.

Understanding the Analysis-Transfer-Generation over Vauquois triangle (1/4)

H1.1: सरकार_ने चुनावो_के_बाद मुंबई में करों_के_माध्यम_से अपने राजस्व_को बढ़ाया।
T1.1: Sarkaar ne chunaawo ke baad Mumbai me karoM ke maadhyam se apne raajaswa ko badhaayaa
G1.1: Government_(ergative) elections_after Mumbai_in taxes_through its revenue_(accusative) increased
E1.1: The Government increased its revenue after the elections through taxes in Mumbai

Understanding the Analysis-Transfer-Generation over Vauquois triangle (2/4)

Entity	English	Hindi
Subject	The Government	सरकार (sarkaar)
Verb	Increased	बढ़ाया (badhaayaa)
Object	Its revenue	अपने राजस्व (apne raajaswa)

Understanding the Analysis-Transfer-Generation over Vauquois triangle (3/4)

Adjunct	English	Hindi
Instrumental	Through taxes in Mumbai	मुंबई_में करों_के_माध्यम_से (mumbai me karo ke maadhyam se)
Temporal	After the elections	चुनावो_के_बाद (chunaawo ke baad)

Understanding the Analysis-Transfer-Generation over Vauquois triangle (3/4)

The Government increased its revenue

P0 P1 P2 P3

E1.2: after the elections, the Government increased its revenue through taxes in Mumbai
E1.3: the Government increased its revenue through taxes in Mumbai after the elections

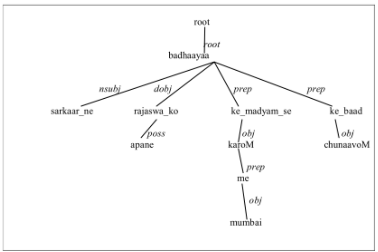
More flexibility in Hindi generation

Sarkaar_ne badhaayaa

P_0 (the govt) P_1 (increased) P_2

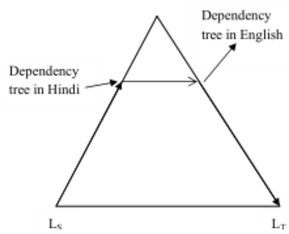
H1.2: चुनावो_के_बाद सरकार_ने मुंबई_में करों_के_माध्यम_से अपने राजस्व_को बढ़ाया।
T1.2: elections_after government_(erg) Mumbai_in taxes_through its revenue increased.
H1.3: चुनावो_के_बाद मुंबई_में करों_के_माध्यम_से सरकार_ने अपने राजस्व_को बढ़ाया।
T1.3: elections_after Mumbai_in taxes_through government_(erg) its revenue increased.
H1.4: चुनावो_के_बाद मुंबई_में करों_के_माध्यम_से अपने राजस्व_को सरकार_ने बढ़ाया।
T1.4: elections_after Mumbai_in taxes_through its revenue government_(erg) increased.
H1.5: मुंबई_में करों_के_माध्यम_से चुनावो_के_बाद सरकार_ने अपने राजस्व_को बढ़ाया।
T1.5: Mumbai_in taxes_through elections_after government_(erg) its revenue increased.

Dependency tree of the Hindi sentence



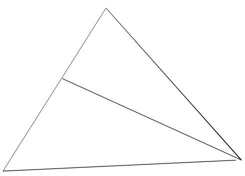
H1.1: सरकार_ने चुनावो_के_बाद मुंबई में करों_के_माध्यम_से अपने राजस्व_को बढ़ाया

Transfer over dependency tree



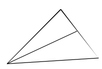
Descending transfer

- नृपायते सिंहासनासीनो वानरः
- Behaves-like-king sitting-on-throne monkey
- A monkey sitting on the throne (of a king) behaves like a king



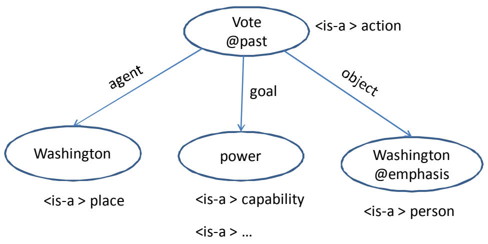
Ascending transfer: Finnish→English

- *istahtaisinkohan* "I wonder if I should sit down for a while"
- ist + "sit", verb stem
- ahta + verb derivation morpheme, "to do something for a while"
- isi + conditional affix
- n + 1st person singular suffix
- ko + question particle
- han a particle for things like reminder (with declaratives) or "softening" (with questions and imperatives)



Interlingual representation: complete disambiguation

- Washington voted Washington to power

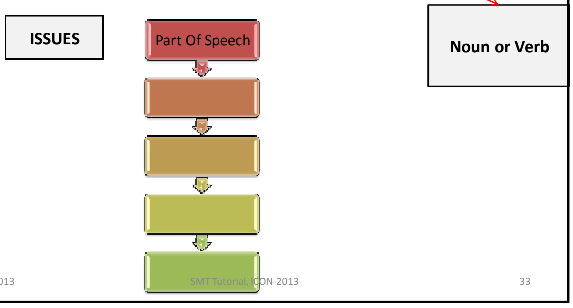


Kinds of disambiguation needed for a complete and correct interlingua graph

- N: Name
- P: POS
- A: Attachment
- S: Sense
- C: Co-reference
- R: Semantic Role

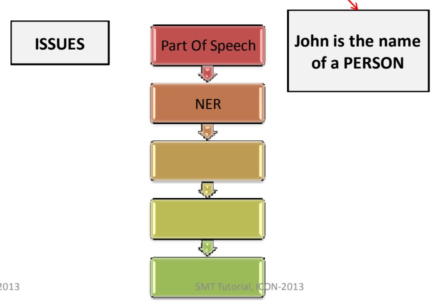
Issues to handle

Sentence: *I went with my friend, John, to the bank to withdraw some money but was disappointed to find it closed.*



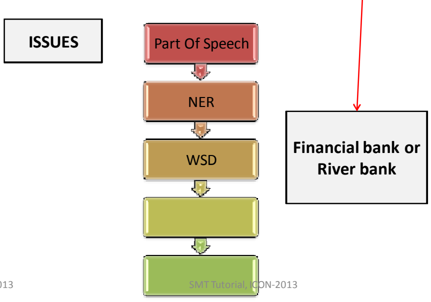
Issues to handle

Sentence: *I went with my friend, John, to the bank to withdraw some money but was disappointed to find it closed.*



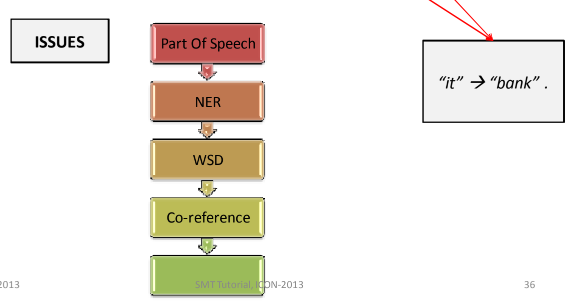
Issues to handle

Sentence: *I went with my friend, John, to the bank to withdraw some money but was disappointed to find it closed.*

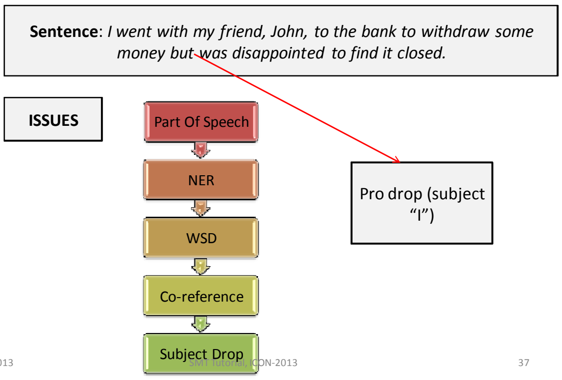


Issues to handle

Sentence: *I went with my friend, John, to the bank to withdraw some money but was disappointed to find it closed.*



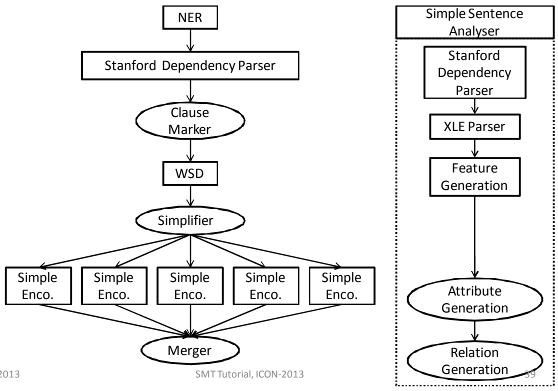
Issues to handle



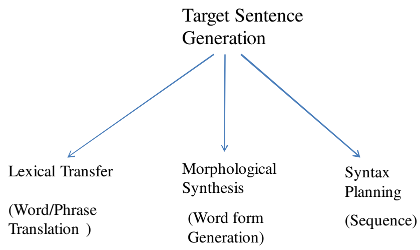
Typical NLP tools used

- POS tagger
- Stanford Named Entity Recognizer
- Stanford Dependency Parser
- XLE Dependency Parser
- Lexical Resource
  - WordNet
  - Universal Word Dictionary (UW++)

System Architecture

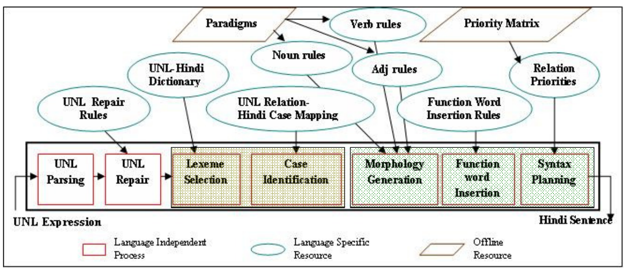


Target Sentence Generation from interlingua



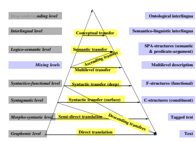
Generation Architecture

Deconversion = Transfer + Generation



Transfer Based MT

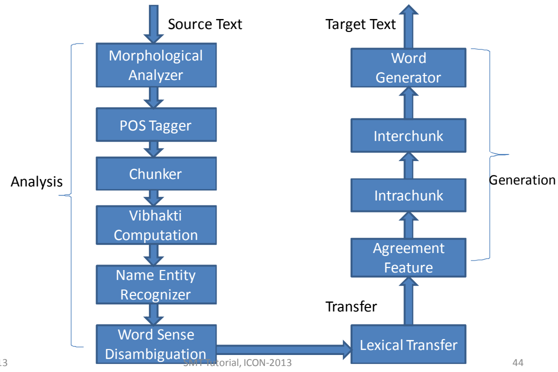
Marathi-Hindi



Indian Language to Indian Language Machine Translation (ILILMT)

- Bidirectional Machine Translation System
- Developed for nine Indian language pairs
- Approach:
  - Transfer based
  - Modules developed using both rule based and statistical approach

Architecture of ILILMT System



M-H MT system: Evaluation

- Subjective evaluation based on machine translation quality
- Accuracy calculated based on score given by linguists

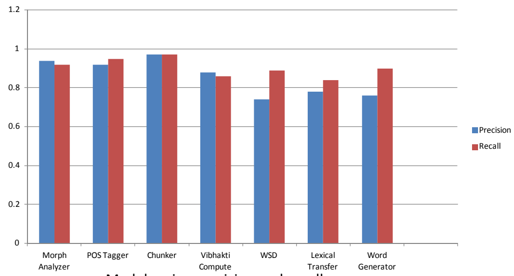
Score : 5	Correct Translation
Score : 4	Understandable with minor errors
Score : 3	Understandable with major errors
Score : 2	Not Understandable
Score : 1	Non sense translation

S5: Number of score 5 Sentences,
S4: Number of score 4 sentences,
S3: Number of score 3 sentences,
N: Total Number of sentences

$$\text{Accuracy} = \frac{1 * S5 + 0.8 * S4 + 0.6 * S3}{N}$$

Evaluation of Marathi to Hindi MT System

- Module-wise evaluation
  - Evaluated on 500 web sentences



Module-wise precision and recall

Evaluation of Marathi to Hindi MT System (cont..)

- Subjective evaluation on translation quality
  - Evaluated on 500 web sentences
  - Accuracy calculated based on score given according to the translation quality.
  - Accuracy: **65.32 %**
- Result analysis:
  - Morph, POS tagger, chunker gives more than 90% precision but Transfer, WSD, generator modules are below 80% hence degrades MT quality.
  - Also, morph disambiguation, parsing, transfer grammar and FW disambiguation modules are required to improve accuracy.

Important challenge of M-H Translation- Morphology processing: kridanta

Ganesh Bhosale, Subodh Kembhavi, Archana Amberkar, Supriya Mhatre, Lata Popale and Pushpak Bhattacharyya, *Processing of Participle (Krudanta) in Marathi*, International Conference on Natural Language Processing (ICON 2011), Chennai, December, 2011.

Kridantas can be in multiple POS categories

- **Nouns**

Verb	Noun
वाच {vaach}{read}	वाचणे {vaachaNe}{reading}
उतर {utara}{climb down}	उतरण {utaraN}{downward slope}

- **Adjectives**

Verb	Adjective
चाव {chav}{bite}	चावणारा {chaavaNaara}{one who bites}
खा {khaa} {eat}	खाल्लेले {khallele} {something that is eaten}.

Kridantas derived from verbs (cont.)

- **Adverbs**

Verb	Adverb
पळ {paL}{run}	पळताना {paLataanaa}{while running}
बस {bas}{sit}	बसून {basun}{after sitting}

Kridanta Types

Kridanta Type	Example	Aspect
“णे” {*Ne-Kridanta*}	vaachNyaasaaThee pustak de. (Give me a book for reading.) For reading book give	Perfective
“ला” {*laa-Kridanta*}	Lekh vaachalyaavar saaMgen. (I will tell you that after reading the article.) Article after reading will tell	Perfective
“ताना” {*Taanaa-Kridanta*}	Pustak vaachtaanaa te lakShaat aale. (I noticed it while reading the book.) Book while reading it in mind came	Durative
“लेला” {*Lela-Kridanta*}	kaal vaachlele pustak de. (Give me the book that (I/you) read yesterday.) Yesterday read book give	Perfective
“ऊन”{*Un-Kridanta*}	pustak vaachun parat kar. (Return the book after reading it.) Book after reading back do	Completive
“णारा”{*Nara-Kridanta*}	pustake vaachNaaRyaalaa dnyaan miLte. (The one who reads books, gets knowledge.) Books to the one who reads knowledge gets	Stative
“वे” {*ve-Kridanta*}	he pustak pratyekaane vaachaave. (Everyone should read this book.) This book everyone should read	Inceptive
“ता” {*taa-Kridanta*}	to pustak vaachtaa vaachtaa zopee gelaa. (He fell asleep while reading a book.) He book while reading to sleep went	Stative

Participial Suffixes in Other Agglutinative Languages

- **Kannada:**

  muridiruwaa kombe jennu esee

  Broken to branch throw

  Throw away the broken branch.

  \- similar to the *lela* form frequently used in Marathi.

Participial Suffixes in Other Agglutinative Languages (cont.)

- **Telugu:**

  ame padutunnappudoo nenoo panichesanoo

  she singing I work

  I worked while she was singing.

  -similar to the *taanaa* form frequently used in Marathi.

Participial Suffixes in Other Agglutinative Languages (cont.)

- **Turkish:**

  hazirlanmis plan

  prepare-past plan

  The plan which has been prepared

  Eqv Marathi: *lelaa*

Morphological Processing of *Kridanta* forms (cont.)

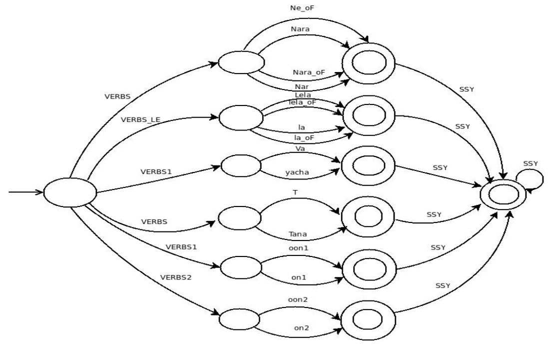
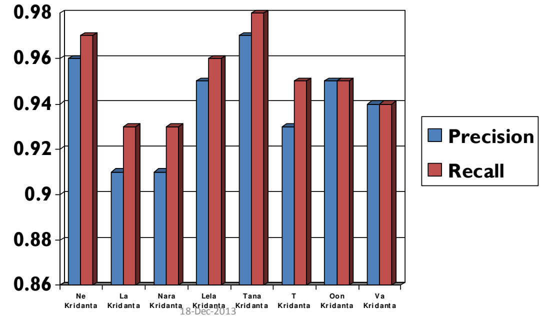


Fig. Morphotactics FSM for *Kridanta* Processing

Accuracy of Kridanta Processing: Direct Evaluation



Summary of M-H transfer based MT

- Marathi and Hindi are close cousins
- Relatively easier problem to solve
- Will interlingua be better?
- Web sentences being used to test the performance
- Rule governed
- Needs high level of linguistic expertise
- Will be an important contribution to IL MT

Indian Language SMT

Recent study: Anoop, Abhijit

Pan-Indian Language SMT

http://www.cfilt.iitb.ac.in/indic-translator

- SMT systems between 11 languages
  - 7 Indo-Aryan: Hindi, Gujarati, Bengali, Oriya, Punjabi, Marathi, Konkani
  - 3 Dravidian languages: Malayalam, Tamil, Telugu
  - English
- Corpus
  - Indian Language Corpora Initiative (ILCI) Corpus
  - Tourism and Health Domains
  - 50,000 parallel sentences
- Evaluation with BLEU
  - METEOR scores also show high correlation with BLEU

Natural Partitioning of SMT systems

High accuracy between Indo-Aryan languages

Low accuracy between Dravidian languages

Structural Divergence between English-IL results in low accuracy

[illegible]	[illegible]	[illegible]	pa	bn	gu	mr	kK	ta	te	ml	en
[illegible]	[illegible]	[illegible]	68.21	34.96	51.31	39.12	37.81	14.43	21.38	10.98	29.23
[illegible]	[illegible]	[illegible]	52.02	29.59	39.00	27.57	28.29	11.95	16.61	8.65	22.46
[illegible]	[illegible]	[illegible]		29.89	43.85	30.87	30.72	10.75	18.81	9.11	[illegible]
bn	[illegible]	32.08	31.38		28.14	22.09	23.47	10.94	13.40	8.10	[illegible]
gu	55.66	[illegible]	45.14	28.50		32.06	30.48	12.57	17.22	8.01	[illegible]
[illegible]	[illegible]	32.60	33.28	23.73	32.42		27.81	10.74	12.89	[illegible]	[illegible]
[illegible]	[illegible]	34.00	34.31	24.59	31.07	27.52		10.36	[illegible]	7.89	17.07
[illegible]	[illegible]	18.12	15.57	13.21	16.53	11.60	11.87		8.48	6.31	11.79
[illegible]	[illegible]	25.07	25.56	16.57	20.96	14.94	17.27	8.68		6.68	12.34
[illegible]	[illegible]	13.39	12.97	10.67	9.76	8.39	9.18	5.90	5.94		8.61
en	28.94	22.96	22.33	15.33	15.44	12.11	13.66	6.43	6.55	4.65	

Baseline PBSMT - % BLEU scores (S1)

- **Clear partitioning of translation pairs by language family pairs**, based on translation accuracy.
  - Shared characteristics within language families make translation simpler
  - Divergences among language families make translation difficult
- ***Language families are the right level of generalization*** for building SMT systems in continuum from totally language independent systems to per language pair system continuum

The Requirement of Hybridization for Marathi – Hindi MT

Sreelekha, Dabre, Bhattaccharyya, ICON 2013

Challenges in Marathi – Hindi Translation

- Ambiguity within language
  - Lexical
  - Structural
- Differences in structure between languages
- Vocabulary differences

Lexical Ambiguity

- Marathi- मी फोटो काढला *{me photo kadhla}*
- Hindi- मैने फोटो निकाला *{maenne photo nikala}*
- *English- I took the photo*
- "काढला"{kadhla}, "निकाला"{nikala}, and "*took"* have ambiguity in meaning.
- Not clear that whether the word "काढला"{kadhla} is used as the "clicked the photo" ("निकाला" *{'nikala'}* in Hindi) sense or the "took" (*nikala*) sense.
- Both in source language and target language ambiguity is present for the same word.
- Usually be clear from the context.
- Disambiguation is generally non-trivial.

Structural Ambiguity

- Marathi – तिथे उंच मुली आणि मुले होती.
  - {tithe oonch muli aani mulen hoti}
  - *{There were tall girls and boys}*
  - Not clear whether उंच applies to both boys and girls or only one of them.
- Hindi equivalent – *वहाँ* लंबी लड़कियाँ और लड़के थे.
  - *{vahan lambi ladkiyam our ladkem the }*
  - *OR*
  - *वहाँ लंबी लड़कियाँ और लंबे लड़के थे*
  - *{vahan lambi ladkiyam our lambe ladkem the}*
  - *{There were tall girls and tall boys}*
- In some cases free rides are possible.

Constructions in Hindi having Participials in Marathi

- **Example 1**:
  - जो लड़का गा रहा था वह चला गया
  - jo ladkaa gaa rahaa thaa wah chalaa gayaa
  - rel. boy sing stay+perf.+cont. be+past walk go+perf.
  - The boy who was singing, has left.
- **Example 2**:
  - जब मैं गा रहा था तब वह चला गया
  - jab main gaa rahaa thaa tab wah chalaa gayaa
  - rel. I sing stay+perf. be+past he walk go+perf.
  - He left when (while) I was singing.

Marathi (Direct Translations)

- **Example 1**:
  - जो मुलगा गात होता तो निघून गेला
  - jo mulgaa gaat hotaa to nighoon gelaa
  - rel. boy sing+imperf. be+past leave+CP go+perf.
  - The boy who was singing, has left.
- **Example 2**:
  - जेव्हा मी गात होतो तेव्हा तो निघून गेला
  - jevhaa mee gaat hoto tevhaa to nighoon gelaa
  - rel. I sing+imperf. be+past he leave+CP go+perf.
  - He left when (while) I was singing.

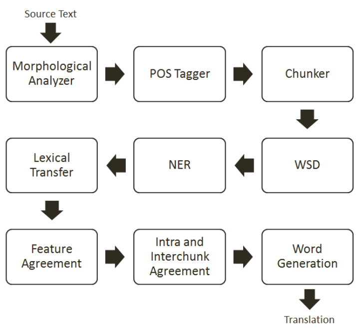
Participial Constructions in Marathi (Actual Translations)

- **Example 1**:
  - गाणारा मुलगा निघून गेला
  - gaaNaaraa mulgaa nighoon gelaa
  - sing+part. boy leave+CP go+perf.
  - The boy who was singing left
- **Example 2**:
  - मी गात असताना तो निघून गेला
  - mee gaat asataanaa to nighoon gelaa
  - I sing+imperf. be+part. he leave+CP go+perf.
  - He left while I was singing.

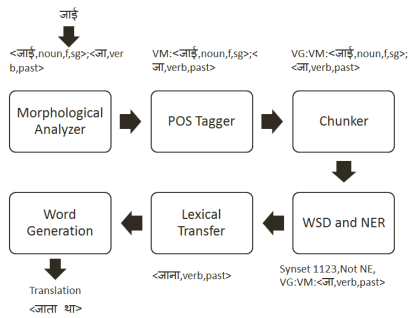
Vocabulary Differences

- Marathi : " काल आनंदीचे केळवण होते . "
  - **{kaal anandiche kelvan hote}**
  - **{yesterday was held Anandi's kelvan ceremony which is a lunch given by relatives after engagement and before marriage}**
- *Here* " केळवण " as a verb has no equivalent in Hindi (or English), and this sentence has to be translated as,
  - "कल आनंदी का सगाई होने के बाद एवं शादि के पहले लड़का या लडकी को संबंधीयों द्वारा दिया जाने वाला भोज था ।"
  - *{"Kal aanandii ka sagaayi hone ke baad evam shaadi ke pahle ladka ya ladki ko sambandhiyon dwara diya jaane wala bhoj tha . " }*

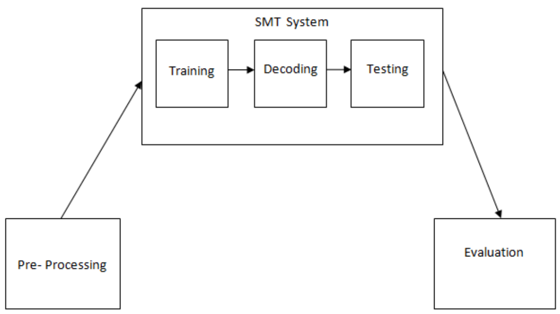
RBMT System



Working



SMT System



Evaluation

- Bleu for direct/objective evaluation

MT System	BLEU Score
Rule Based	5.9
Statistical	9.31

- Adequacy and Fluency for Subjective Evaluation
  - $A/F = 100 * \frac{(S5 + 0.8 * S4 + 0.6 * S3)}{N}$

MT System	Adequacy	Fluency
Rule Based	69.6%	58%
Statistical	62.8%	73.4%

Error Analysis

Source Sentence	केंद्रीय सरकारी संग्रहालय १८७६ मध्ये प्रिन्स औफ वेल्सच्या भारतभेटीच्या वेळी उभारण्यात आले व १८८६ साली ते जनतेसाठी खुले करण्यात आले.	In the rule based system since each word was morphologically analyzed the overall meaning is conveyed however "1886 सालें" {1886 saale} {year (plural) 1886} is not a grammatically good construction. This is overcome in the SMT system by replacing it by a more fluent form "1886 में" {1886 mein}. Moreover the proper form of वह {waha} {it} is picked in the SMT system but not in the rule based system namely "वे" {wey} {they}. However, the content words are not translated in the SMT system due to lack of learned word forms.
Meaning	In 1986 the national central museum was established during the visit of the Prince of Wales and in 1886 was opened for the public.	
Rule based system	केंद्रीय सरकारी संग्रहालय 1876 में प्रिन्स औफ वेल्स के भारतभेट का बार में उठाया गया व 1886 सालें वे जनता के लिए खुला किया गया ।	
Statistical System	केंद्रीय सरकारी संग्रहालय १८७६ मध्ये प्रिंस औफ वेल्सच्या भारतभेटीच्या के शेड डाला गया व १८८६ में वह जनता के लिए खोल दिया गया ।	

Error Analysis

Source Sentence	दीग पॅलेस भक्कम व प्रचंड किल्ला आहे, जो भरतपूरच्या शासकांचे ग्रीष्मकालीन निवासस्थान होता.	The RB system makes a mistake in sense disambiguation of the word "प्रचंड"{prachand}{huge} which also has the sense of many, which the SMT system does not. SMT is also able to overcome the number agreement between "का" and "ग्रीष्मकालीन" leading to a more fluent translation. Due to the morphological richness of Marathi "भरतपूरच्या" is translated correctly as "भरतपूर के" by RB system but not by SMT system (it gives "भरतपूरच्या के").
Meaning	Deeg palace, which was the summer residence of the rulers of Bharatpur, is tough and huge.	
Rule based system	दीग पैलेस मजबूत व बहुत किला है , जो भरतपूर के शासकों के ग्रीष्मकालीन आवास हो ।	
Statistical System	दीग पैलेस मजबूत व विशाल किला है , जो भरतपूरच्या के शासकों का ग्रीष्मकालीन निवास था ।	

Error Analysis

Source Sentence	मारवाड हा राजस्थानमधील मुख्य उत्सव, ऑक्टोबर महिन्यामध्ये संप्पन्न होतो.	Since "मारवाड" was not present in the training corpus and the input dictionary the SMT system made a wrong translation. However function word translation of "मधील" {madhil} {of} is better done by the SMT system. Overall the RB translation is clear but not as fluent as the SMT system.
Meaning	Marwad, a major festival in Rajasthan, takes place in the month of October.	
Rule based system	मारवाड हा राजस्थान में के मुख्य उत्सव ऑक्टोबर महीने में संप्पन्न हो ।	
Statistical System	राजस्थान का यह राजस्थान का प्रमुख त्योहार अक्टूबर के महीने में संप्पन्न होता है ।	

Observations

- Surprising!
  - RBMT does well on Nominals
  - SMT better or verbals
- Points to hybridization between RBMT and SMT

SMT

Czeck-English data

- [nesu] "I carry"
- [ponese] "He will carry"
- [nese] "He carries"
- [nesou] "They carry"
- [yedu] "I drive"
- [plavou] "They swim"

To translate …

- I will carry.
- They drive.
- He swims.
- They will drive.

Hindi-English data

• [DhotA huM]	"I carry"
• [DhoegA]	"He will carry"
• [DhotA hAi]	"He carries"
• [Dhote hAi]	"They carry"
• [chalAtA huM]	"I drive"
• [tErte hEM]	"They swim"

Bangla-English data

• [bai]	"I carry"
• [baibe]	"He will carry"
• [bay]	"He carries"
• [bay]	"They carry"
• [chAlAi]	"I drive"
• [sAMtrAy]	"They swim"

To translate … (repeated)

- I will carry.
- They drive.
- He swims.
- They will drive.

Foundation

- Data driven approach
- Goal is to find out the English sentence e given foreign language sentence f whose $p(e|f)$ is maximum.

$$\tilde{e} = \underset{e \in e^*}{\operatorname{argmax}}\, p(e|f) = \underset{e \in e^*}{\operatorname{argmax}}\, p(f|e)p(e)$$

- Translations are generated on the basis of statistical model
- Parameters are estimated using bilingual parallel corpora

SMT: Language Model

- To detect *good* English sentences
- Probability of an English sentence $w_1 w_2 \ldots\ldots w_n$ can be written as

$$Pr(w_1 w_2 \ldots\ldots w_n) = Pr(w_1) * Pr(w_2|w_1) * \ldots * Pr(w_n|w_1\, w_2 \ldots w_{n-1})$$

- Here $Pr(w_n|w_1\, w_2 \ldots w_{n-1})$ is the probability that word w_n follows word string $w_1\, w_2 \ldots w_{n-1}$.
  - N-gram model probability
- Trigram model probability calculation

$$p(w_3|w_1 w_2) = \frac{count(w_1 w_2 w_3)}{count(w_1 w_2)}$$

SMT: Translation Model

- $P(f|e)$: Probability of some f given hypothesis English translation e
- How to assign the values to $p(e|f)$?
  - Sentences $p(f|e) = \frac{count(f,e)}{count(e)}$ ← Sentence level
    :o find pair(e,f) for all sentences
- Introduce a hidden variable $\boldsymbol{a}$, that represents alignments between the individual words in the sentence pair

$$\Pr(\boldsymbol{f}|\boldsymbol{e}) = \sum_{\boldsymbol{a}} \Pr(\boldsymbol{f}, \boldsymbol{a}|\boldsymbol{e})$$ ← Word level

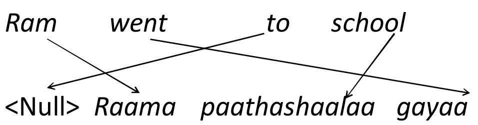
Alignment

- If the string, $e = e_1^l = e_1 e_2 \dots e_l$, has l words, and the string, $f = f_1^m = f_1 f_2 \dots f_m$, has m words,
- then the alignment, a, can be represented by a series, $\boldsymbol{a_1^m = a_1 a_2 \dots a_m}$, of m values, each between 0 and l such that if the word in position j of the f-string is connected to the word in position i of the e-string, then
  - $\boldsymbol{a_j = i}$, and
  - if it is not connected to any English word, then $\boldsymbol{a_j = 0}$

Example of alignment

English: *Ram went to school*

Hindi: *Raama paathashaalaa gayaa*



Translation Model: Exact expression

$$\Pr(\boldsymbol{f}, \boldsymbol{a}|\boldsymbol{e}) = \Pr(m|\boldsymbol{e}) \prod_{j=1}^{m} \Pr(a_j | a_1^{j-1}, f_1^{j-1}, m, \boldsymbol{e}) \Pr(f_j | a_1^j, f_1^{j-1}, m, \boldsymbol{e})$$

Choose the length of foreign language string given *e*

Choose alignment given *e* and *m*

Choose the identity of foreign word given *e, m, a*

- Five models for estimating parameters in the expression [2]
- Model-1, Model-2, Model-3, Model-4, Model-5

Proof of Translation Model: Exact expression

$$\Pr(f|e) = \sum_a \Pr(f, a|e)$$; marginalization

$$\Pr(f, a|e) = \sum_m \Pr(f, a, m|e)$$; marginalization

$$\Pr(f, a, m|e) = \sum_m \Pr(m|e) \Pr(f, a|m, e)$$

$$= \sum_m \Pr(m|e) \Pr(f, a|m, e)$$

$$= \sum_m \Pr(m|e) \prod_{j=1}^{m} \Pr(f_j, a_j | a_1^{j-1}, f_1^{j-1}, m, e)$$

$$= \sum_m \Pr(m|e) \prod_{j=1}^{m} \Pr(a_j | a_1^{j-1}, f_1^{j-1}, m, e) \Pr(f_j | a_1^j, f_1^{j-1}, m, e)$$

m is fixed for a particular *f*, hence

$$\Pr(f, a, m|e) = \Pr(m|e) \prod_{j=1}^{m} \Pr(a_j | a_1^{j-1}, f_1^{j-1}, m, e) \Pr(f_j | a_1^j, f_1^{j-1}, m, e)$$

Alignment

Fundamental and ubiquitous

- Spell checking
- Translation
- Transliteration
- Speech to text
- Text to speeh

EM for word alignment from sentence alignment: example

English	French
(1) three rabbits a b	(1) trois lapins w x
(2) rabbits of Grenoble b c d	(2) lapins de Grenoble x y z

Initial Probabilities: each cell denotes *t(a←→ w), t(a←→ x) etc.*

	a	b	c	d
w	1/4	1/4	1/4	1/4
x	1/4	1/4	1/4	1/4
y	1/4	1/4	1/4	1/4
z	1/4	1/4	1/4	1/4

The counts in IBM Model 1

Works by maximizing *P(f|e)* over the entire corpus

For IBM Model 1, we get the following relationship:

$$c(w^f \mid w^e; f, e) = \frac{t(w^f \mid w^e)}{t(w^f \mid w^{e_0}) + \square + t(w^f \mid w^{e_l})} \square . \square$$

$c(w^f \mid w^e; f, e)$ is the fractional count of the alignment of w^f with w^e in f and e

$t(w^f \mid w^e)$ is the probability of w^f being the translation of w^e

$\square$ is the count of w^f in f

$\square$ is the count of w^e in e

Example of expected count

C[a←→w; (a b)←→(w x)]

= t(a←→w) / (t(a←→w)+t(a←→x)) X #(a in 'a b') X #(w in 'w x')

= 1/4 / (1/4+1/4) X 1 X 1= 1/2

"counts"

a b ←→ w x	a	b	c	d
w	1/2	1/2	0	0
x	1/2	1/2	0	0
y	0	0	0	0
z	0	0	0	0

b c d ←→ x y z	a	b	c	d
w	0	0	0	0
x	0	1/3	1/3	1/3
y	0	1/3	1/3	1/3
z	0	1/3	1/3	1/3

Revised probability: example

$$t_{revised}(a \leftarrow\rightarrow w) = \frac{1/2}{(1/2+1/2+0+0)_{(a\,b)\leftarrow\rightarrow(w\,x)} + (0+0+0+0)_{(b\,c\,d)\leftarrow\rightarrow(x\,y\,z)}}$$

Revised probabilities table

	a	b	c	d
w	1/2	1/4	0	0
x	1/2	5/12	1/3	1/3
y	0	1/6	1/3	1/3
z	0	1/6	1/3	1/3

"revised counts"

a b ←→ w x	a	b	c	d
w	1/2	3/8	0	0
x	1/2	5/8	0	0
y	0	0	0	0
z	0	0	0	0

b c d ←→ x y z	a	b	c	d
w	0	0	0	0
x	0	5/9	1/3	1/3
y	0	2/9	1/3	1/3
z	0	2/9	1/3	1/3

Re-Revised probabilities table

	a	b	c	d
w	1/2	3/16	0	0
x	1/2	85/144	1/3	1/3
y	0	1/9	1/3	1/3
z	0	1/9	1/3	1/3

Continue until convergence; notice that (b,x) binding gets progressively stronger; b=rabbits, x=lapins

Derivation of EM based Alignment Expressions

V_E = vocalbulary of language L_1 (Say English)

V_F = vocabulary of language L_2 (Say Hindi)

E1 *what is in a name ?*
F1 *नाम में क्या है ?*
naam meM kya hai ?
name in what is ?
what is in a name ?

E2 *That which we call rose, by any other name will smell as sweet.*
F2 *जिसे हम गुलाब कहते हैं, और भी किसी नाम से उसकी कुशबू सामान मीठा होगी*
Jise hum gulab kahte hai, aur bhi kisi naam se uski khushbu samaan mitha hogii
That which we rose say , any other name by its smell as sweet
That which we call rose, by any other name will smell as sweet.

Vocabulary mapping

Vocabulary

V_E	V_F
what , is , in, a , name , that, which, we , call ,rose, by, any, other, will, smell, as, sweet	naam, meM, kya, hai, jise, hum, gulab, kahte, hai, aur, bhi, kisi, bhi, uski, khushbu, saman, mitha, hogii

Key Notations

English vocabulary : V_E
French vocabulary : V_F
No. of observations / sentence pairs : S
Data D which consists of S observations looks like,

$$e^1{}_1, e^1{}_2, \dots, e^1{}_{l^1} \Leftrightarrow f^1{}_1, f^1{}_2, \dots, f^1{}_{m^1}$$
$$e^2{}_1, e^2{}_2, \dots, e^2{}_{l^2} \Leftrightarrow f^2{}_1, f^2{}_2, \dots, f^2{}_{m^2}$$
.....
$$e^s{}_1, e^s{}_2, \dots, e^s{}_{l^s} \Leftrightarrow f^s{}_1, f^s{}_2, \dots, f^s{}_{m^s}$$
.....
$$e^S{}_1, e^S{}_2, \dots, e^S{}_{l^S} \Leftrightarrow f^S{}_1, f^S{}_2, \dots, f^S{}_{m^S}$$

No. words on English side in s^{th} sentence : l^s
No. words on French side in s^{th} sentence : m^s
$index_E(e^s{}_p)$ =Index of English word $e^s{}_p$ in English vocabulary/dictionary
$index_F(f^s{}_q)$ =Index of French word $f^s{}_q$ in French vocabulary/dictionary

(Thanks to Sachin Pawar for helping with the maths formulae processing)

Hidden variables and parameters

Hidden Variables (Z) :
Total no. of hidden variables = $\sum_{s=1}^{S} l^s\, m^s$ where each hidden variable is as follows:
$z_{pq}^s = 1$, if in s^{th} sentence, p^{th} English word is mapped to q^{th} French word.
$z_{pq}^s = 0$, otherwise

Parameters (Θ) :
Total no. of parameters = $|V_E| \times |V_F|$, where each parameter is as follows:
$P_{i,j}$ = Probability that i^{th} word in English vocabulary is mapped to j^{th} word in French vocabulary

Likelihoods

Data Likelihood *L(D; Θ)* :

$$L(D;\Theta) = \prod_{s=1}^{S}\prod_{p=1}^{l^s}\prod_{q=1}^{m^s}\left(P_{index_E(e_p^s),index_F(f_q^s)}\right)^{z_{pq}^s}$$

Data Log-Likelihood LL(D; Θ) :

$$LL(D;\Theta) = \sum_{s=1}^{S}\sum_{p=1}^{l^s}\sum_{q=1}^{m^s} z_{pq}^s \log\left(P_{index_E(e_p^s),index_F(f_q^s)}\right)$$

Expected value of Data Log-Likelihood E(LL(D; Θ)) :

$$E(LL(D;\Theta)) = \sum_{s=1}^{S}\sum_{p=1}^{l^s}\sum_{q=1}^{m^s} E(z_{pq}^s) \log\left(P_{index_E(e_p^s),index_F(f_q^s)}\right)$$

Constraint and Lagrangian

$$\sum_{j=1}^{|V_F|} P_{i,j} = 1 \ , \forall i$$

$$\sum_{s=1}^{S}\sum_{p=1}^{l^s}\sum_{q=1}^{m^s} E(z_{pq}^s) \log\left(P_{index_E(e_p^s),index_F(f_q^s)}\right) - \sum_{i=1}^{|V_E|} \lambda_i \left(\sum_{j=1}^{|V_F|} P_{i,j} - 1\right)$$

Differentiating wrt P_{ij}

$$\sum_{s=1}^{S}\sum_{p=1}^{l^s}\sum_{q=1}^{m^s} \delta_{index_E(e_p^s),i}\, \delta_{index_F(f_q^s),j} \left(\frac{E(z_{pq}^s)}{P_{i,j}}\right) - \lambda_i = 0$$

$$P_{i,j} = \frac{1}{\lambda_i} \sum_{s=1}^{S}\sum_{p=1}^{l^s}\sum_{q=1}^{m^s} \delta_{index_E(e_p^s),i}\, \delta_{index_F(f_q^s),j} E(z_{pq}^s)$$

$$\sum_{j=1}^{|V_F|} P_{i,j} = 1 = \sum_{j=1}^{|V_F|} \frac{1}{\lambda_i} \sum_{s=1}^{S}\sum_{p=1}^{l^s}\sum_{q=1}^{m^s} \delta_{index_E(e_p^s),i}\, \delta_{index_F(f_q^s),j} E(z_{pq}^s)$$

Final E and M steps

M-step

$$P_{i,j} = \frac{\sum_{s=1}^{S}\sum_{p=1}^{l^s}\sum_{q=1}^{m^s} \delta_{index_E(e_p^s),i}\, \delta_{index_F(f_q^s),j} E(z_{pq}^s)}{\sum_{j=1}^{|V_F|}\sum_{s=1}^{S}\sum_{p=1}^{l^s}\sum_{q=1}^{m^s} \delta_{index_E(e_p^s),i}\, \delta_{index_F(f_q^s),j} E(z_{pq}^s)}, \forall i,j$$

E-step

$$E(z_{pq}^s) = \frac{P_{index_E(e_p^s),index_F(f_q^s)}}{\sum_{q'=1}^{m^s} P_{index_E(e_p^s),index_F(f_{q'}^s)}}, \forall s,p,q$$

Combinatorial considerations

Example

E2.1: Peter went to school early

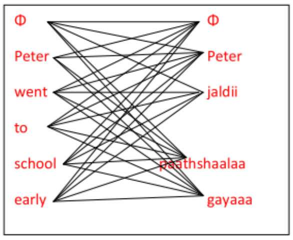
H2.1:पीटर जल्दी पाठशाला गया

T2.1: piitar jaldii paathshaalaa gayaa

G2.1: Peter early school went

Non English text

All possible alignments



First fundamental requirement of SMT

Alignment requires evidence of:

- firstly, a translation pair to introduce the POSSIBILITY of a mapping.
- then, another pair to establish with CERTAINTY the mapping

For the "certainty"

- We have a translation pair containing alignment candidates and **none** of the other words in the translation pair

OR

- We have a translation pair containing **all** words in the translation pair, except the alignment candidates

Therefore…

- *If M valid bilingual mappings exist in a translation pair then an additional M-1 pairs of translations will decide these mappings with certainty.*

Rough estimate of data requirement

- SMT system between two languages L_1 and L_2
- Assume no a-priori linguistic or world knowledge, *i.e.*, no meanings or grammatical properties of any words, phrases or sentences
- Each language has a vocabulary of 100,000 words
- can give rise to about 500,000 word forms, through various morphological processes, assuming, each word appearing in 5 different forms, on the average
  – For example, the word 'go' appearing in 'go', 'going', 'went' and 'gone'.

Reasons for mapping to multiple words

- Synonymy on the target side (e.g., "*to go*" in English translating to "*jaanaa*", "*gaman karnaa*", "*chalnaa*" *etc.* in Hindi), a phenomenon called lexical choice or register
- polysemy on the source side (e.g., "*to go*" translating to "*ho jaanaa*" as in "*her face went red in anger*"→"*usakaa cheharaa gusse se laal ho gayaa*")
- syncretism ("went" translating to "*gayaa*", "*gayii*", or "*gaye*"). Masculine Gender, 1st or 3rd person, singular number, past tense, non-progressive aspect, declarative mood

Estimate of corpora requirement

- Assume that on an average a sentence is 10 words long.
- → an additional 9 translation pairs for getting at one of the 5 mappings
- → 10 sentences per mapping per word
- → a first approximation puts the data requirement at 5 X 10 X 500000= 25 million parallel sentences
- Estimate is not wide off the mark
- Successful SMT systems like Google and Bing reportedly use 100s of millions of translation pairs.

WORD BASED MODELS

Acknowledgements: Piyush, Ankit, Ankur, Mandar; M.Tech, CSE, IIT Bombay

Noisy channel model

$\text{argmax}_e \Pr(e|f) = \text{argmax}_e \Pr(e).\Pr(f|e)$

$\Pr(f|e) = \Sigma_a \Pr(f, a|e)$

$\Pr(f,a|e)$

$= \Pr(m|e).\prod_{j=1}^{m} \Pr(f_j, a_j | a_1^{j-1}, f_1^{j-1}, m, e)$

$= \Pr(m|e).\prod_{j=1}^{m} \Pr(a_j | a_1^{j-1}, f_1^{j-1}, m, e).\Pr(f_j | a_1^{j}, f_1^{j-1}, m, e)$

IBM Model-1

- Focuses on lexical translation
- Assumptions
  1. $\Pr(m|e)$ is independent of e & m
     - New parameter $\varepsilon = \Pr(m|e)$
  2. Uniform distribution of alignment probability over (l+1) (null included)
     - Alignment probability is $1/(l+1)$
  3. $\Pr(f_j | a_1^j, f_1^{j-1}, m, e)$ depends only on f_j and e_{aj}
     - Translation probability, $t(f_j|e_{aj}) = \Pr(f_j | a_1^j, f_1^{j-1}, m, e)$

Derivation

- Final Derivation

$\Pr(f, a|e)$
$= \Pr(m|e).\prod_{j=1}^{m} \Pr(a_j|a_1^{j-1}, f_1^{j-1}, m, e).\Pr(f_j|a_1^{j}, f_1^{j-1}, m, e)$

$\Downarrow$

$\Pr(f, a|e) = \epsilon /(l+1)^m.\prod_{j=1}^{m} t(f_j|e_{aj})$

Learning Parameters

•EM Algorithm consists of two steps:

- Expectation-Step: Apply model to the data
  - parts of the model are hidden (here: alignments)
  - using the model, assign probabilities to possible values
- Maximization-Step: Estimate model from data
  - take assigned values as fact
  - collect counts (weighted by probabilities)
  - estimate model from counts
- Iterate these steps until convergence

IBM Model-2

- Why model 2 when we have 1?

<NULL> राम पाठशाला गया

Ram went to school

<NULL> राम पाठशाला गया

school Ram to went

IBM Model 2: expressions

- Focuses on absolute alignment
- Assumptions
  1. Pr(m|e) is independent of e & m
     - New parameter ϵ = Pr(m|e)
  2. ~~Uniform distribution over l+1 (null included)~~
     - Alignment probability is $\Pr(a_j|j, m, l)$
  3. $\Pr(f_j|a_1^j, f_1^{j-1}, m, e)$ depends only on f_j and e_{aj}
     - Translation probability, t(fj|eaj) = Pr(fj|a1-j, f1-j-1, m, e)
- Number of new parameters: m (a_j for j=1 to m)
- Training

IBM Model-3

- Adds fertility model
  - Fertility probability
    - Eg. n(2|house) = prob. of generating 2 words for the word ‘house’
  - Translation probability: same as model 1
    - Eg. t(maison|house) = prob. of ‘maison’ being translation of ‘house’
  - Distortion probability
    - Eg. d(5|2) = prob. that word at position 2 goes to position 5

Derivation from Noisy Channel

$$\Pr(f, a|e) = \Pr(m|e) \prod_j \Pr\big(a_j | a_{1-(j-1)}, f_{1-(j-1)}, m, e\big) \Pr(f_j | a_{1-j} f_{1-(j-1)}, m, e)$$

$$\Downarrow$$

$$\Pr(f, a|e) = \prod_i n(\phi_i|e_i) \prod_j t(f_j|e_{a_j}) \prod_j d(j|a_j, l, m)$$

Example

- This city is famous for its flora.
- This city is famous for its flora flora

fertility step

- This city is famous NULL for its flora flora null

insertion step

- यह शहर है मशहूर के लिए अपने पेड़ पौधों

lexical translation

-
- यह शहर अपने पेड़ पौधों के लिए मशहूर है

distortion step

Deficiency

- Distortion probabilities do not depend on the earlier words
- Model 3 wastes some of its probability on “useless” strings
  - Strings that have some positions with several words and others with none.
- When a model has this property of not concentrating all of its probability on events of interest, it is said to be *deficient.*

Example

<Null> राम पाठशाला गया

- Ram
- <Null> <Null> went school
- to

Comparison of Statistical Models

	Alignment Model	Fertility Model	E-Step	Deficient
Model 1	Uniform	No	Exact	No
Model 2	Zero order	No	Exact	No
Model 3	Zero order	Yes	Approximate	Yes

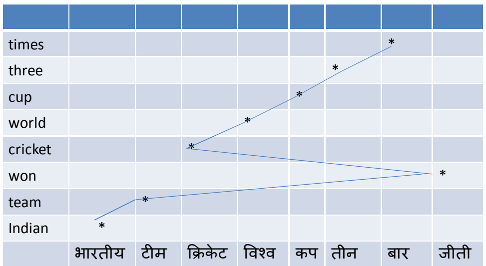
Hidden Markov Alignment Model

Motivation

- In the translation process, large phrases tend to move together.
- Words that are adjacent in the source language tend to be next to each other in the target language.
- Strong localization effect is observed in alignment.

Motivation

- Hindi-English Alignment Example



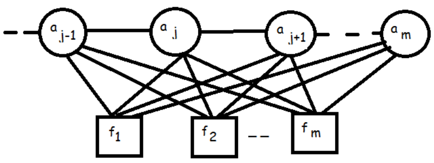
What is Hidden?

- In HMM, states are hidden, outputs are visible
- Alignment is hidden, translation is visible.
- $\Pr(f|e) = \Sigma_a \Pr(m|e) . \prod_{j=1}^{m} \Pr(a_j|a_1^{j-1}, f_1^{j-1}, m, e) . \Pr(f_j|a_1^{j}, f_1^{j-1}, m, e)$

$$= \Sigma_a \Pr(m|e) . \underbrace{\prod_{j=1}^{m} a(a_j|a_{j-1}, m)}_{\text{state transition}} \quad . \quad \underbrace{t(f_j|e_{aj})}_{\text{output generation}}$$

Capturing Locality

- HMM captures the locality of English sentence.



Homogenous HMM

- To make the alignment parameters independent of absolute word positions, we assume that the alignment probabilities $p(i \mid i', m)$ depend only on the jump width $(i - i')$.

$$p(i \mid i', I) = \frac{c(i - i')}{\sum_{i''=1}^{I} c(i'' - i')}$$

Comparison of Statistical Models

	Alignment Model	Fertility Model	E-Step	Deficient
Model 1	Uniform	No	Exact	No
Model 2	Zero order	No	Exact	No
HMM	First-order	No	Exact	No
Model 3	Zero order	Yes	Approximate	Yes
Model 4	First-order	Yes	Approximate	Yes
Model 5	First-order	Yes	Approximate	No
Model 6	First-order	Yes	Approximate	Yes

References: Word Based Models

- Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Association of Computational Linguistics* . March 2003. pages 19-51.
- Peter F. Brown , Vincent J.Della Pietra , Stephen A. Della Pietra , Robert. L. Mercer.The Mathematics of Statistical Machine Translation: Parameter Estimation. *Association of Computational Linguistics* .1993.
- Philipp Koehn. Statistical Machine Translation.2010. *Chapter-4*(*Word-Based Models*)
- Vogel, Stephan, Hermann Ney, and Christoph Tillmann. HMM-based word alignment in statistical translation. *COLING. 1996*, pages 836–841.

PARALLEL AND COMPARABLE CORPORA

Acknowledgement: Rucha Kulkarni, M.Tech student, CSE, IITB

Parallel Corpus

- An SMT system is trained on a parallel corpus.
- Parallel corpus consists of sentence aligned, bilingual text.
- The aligned sentences are perfect translations of each other.

English	Hindi
So far there is no evidence that there is a limit to the Universe .	ब्रम्हांड की कोई सीमा होने का अब तक कोई सबूत नहीं है।
The limit is rather on what we can see and how much we can understand .	सीमा बल्कि यही है कि हम क्या देख सकते हैं और हम कितना समझ पाते हैं ।

Challenge

- Scarce availability of bilingual corpora
- Manual creation of a large parallel corpus very costly

Proposed Solution

- Comparable corpora and non-parallel corpora are largely available for all language pairs.
- We can devise methods for automatic extraction of parallel corpora from such resources.

Potential Sources for Extraction

- Comparable corpora
- Quasi-comparable corpora
- Wikipedia
- The Internet Archive

Comparable Corpora

- Bilingual Documents that are not sentence aligned.
- Many sentences are rough translations of each other, or convey the same information.
- Sometimes, documents may be on the same topic, but may have very different information.
- Lexical and structural differences in the sentences make the problem of "parallel sentence selection", non-trivial.
- e.g. multilingual news feeds provided by news agencies like Agence France Presse, Xinhua News, Reuters, CNN, BBC, etc

Comparable Corpora

English		Hindi
Jagdish Tytler is accused of leading a mob during the 1984 riots.		दिल्ली की एक अदालत ने हुक्म दिया है कि कांग्रेस नेता और पूर्व मंत्री जगदीश टाइटलर के ख़िलाफ़ 1984 सिख विरोधी दंगा मामले में फिर से जांच शुरू की जाए.
The court has ordered the reopening of a case against this Congress Party leader for his involvement in anti-Sikh riots in 1984.		केंद्रीय जांच एजेंसी सीबीआई की सिफारिश पर दिल्ली की एक कोर्ट ने पहले जगदीश टाइटलर के खिलाफ़ मामले को बंद करने की इजाज़त दे दी थी.
Jagdish Tytler was originally cleared by the Central Bureau of Investigation (CBI).		दिल्ली से सांसद रह चुके जगदीश टाइटलर पर आरोप लगते रहे हैं कि उन्होंने 1984 में लोगों को सिख विरोधी दंगो के दौड़ान भड़काया था.
The 1984 riots began following the assassination of Mrs Gandhi.		जगदीश टाइलर कांग्रेस के तीन अहम नेताओं में से एक हैं जिनके खिलाफ़ सिख विरोधी दंगों को लेकर आरोप लगते रहे हैं.

Quasi-Comparable Corpora

- A quasi-comparable corpus (Fung and Cheung, 2004b) contains non-parallel bilingual documents.
- These documents may be on the same topic or may be of very different topics.
- So, a small number of the bilingual sentences can be translations of each other, while some others may be bilingual paraphrases.
- e.g. TDT3 Corpus, which consists of transcriptions of radio broadcasts and TV news reports.

Wikipedia

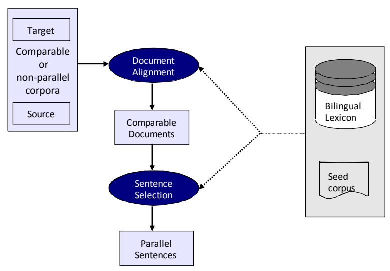
- Wikipedia is a collection of noisy parallel and comparable document pairs.
- Articles are on a large variety of topics and in various languages.
- So, it is rich in information from various domains and in many different languages.
- Some of the characteristics like Interwiki Links, Markup, Image Captions, Lists and Section Headings, etc. can be very useful.

Wikipedia Characteristics

The Internet Archive

- The Internet Archive attempts to archive the entire Web.
- It preserves content in web pages and makes them freely and publicly available through a Wayback Machine Web Interface.
- The data of the Archive is freely accessible by signing up for an account on their cluster.
- Web pages can be searched for finding multilingual translations, or multilingual pages giving same or similar content.

General Architecture: Parallel Sentence Extraction



- Select resources
- Document Alignment
- Sentence Selection

Document Alignment

- A comparable or non-parallel corpus is likely to be huge. It is not possible to examine every sentence pair in the entire corpus.
- So, focus should be on sentence pairs belonging to documents having similar or overlapping content.
- Document Alignment finds comparable or similar documents from the set of all documents.
- Techniques that can be used are:
  - TFIDF Retrieval
  - Cosine Similarity
  - Topic Alignment
  - Content Based Alignment

TFIDF Retrieval and Cosine Similarity

- TFIDF = Term frequency * Inverse Document Frequency
  - This is a metric to show how important the given word is to a document.
- TFIDF is used to compute a ranking function to rank documents according to their relevance to a given query of words.
- Cosine similarity is a measure of similarity between two documents.
- The documents should be represented as a TFIDF vector of the words they contain.
- Cosine similarity is the dot product of these vectors. It is the similarity score of the pair of documents

Content Based Alignment

- The method uses a translational similarity score based on a word-to-word translation lexicon (Resnik and Smith, 2003).
- Link: It is defined as a pair (x,y) where x is a word in foreign language and y is a word in English language.
- A generative, symmetric model based on a bilingual dictionary gives a probability distribution 'p' over all possible link types in the corpus.
- In two documents X and Y, the most probable link sequence is found using

  $Pr(link\text{-}sequence) = \Pi_l\, Pr(x,y)$

  where, $l = (x,y)$

Content Based Alignment

- Tsim: this is defined to be a cross-language similarity score between two documents based on the link sequences.

Tsim=

$$\frac{\Sigma(\log(Pr(\text{two-word links in best matching})))}{\Sigma(\log(Pr(\text{number of links in best matching})))}$$

- The document pairs with highest Tsim score can be considered as relevant or similar documents.

Parallel Sentence Selection

- After document alignment, parallel sentences are extracted from them.
- A reliable way of finding parallel sentence pairs such document pairs is needed.
- Some techniques that can be used for classifying parallel sentence pairs from all sentence the pairs are
  - Word Overlap
  - Maximum Entropy Binary Classifier
  - ME Ranking Model
  - Sentence Similarity

Word Overlap

- It can be used only as "candidate" sentence pair selection step, not the final sentence alignment or extraction step.
- All possible sentence pairs are generated from the document pairs; then, following conditions are verified for each sentence pair:
  - Ratio of lengths of the two sentences is not greater than 2.
  - At least half the words in each sentence of the sentence pair, have a translation in the other sentence according to a dictionary.
- Sentence pairs that do not fulfil these conditions are discarded.
- This step is useful for further reducing noisy pairs and also for reducing the number of candidate sentence pairs to be given for classification.
- Improves efficiency.

ME Classifier and Ranking Model

- An ME classifier can be used to classify parallel sentence pairs from non-parallel.
- The model can be a log linear combination of feature functions.

$$P(ci|sp) = \frac{1}{Z(sp)} \cdot \prod_{j=1}^{k} \lambda_j^{f_{ij}(c,sp)}$$

where c_i *is the class,* c_0 = *parallel and* c_1 = *non-parallel*
$Z(sp)$ *is the normalization factor*
f_{ij} *are the feature functions.*

- Also, a Ranking Approach, based on the same model can be used.
- In this approach, for each source language sentence, we find the target language sentence that is most parallel to it.

Features for Classification

- Features for this particular classification problem should help the classifier distinguish between parallel and non-parallel sentence pairs.
- Following features are used:

Sr. No.			
1.	Sentence Length and ratio	Number of Aligned Words	Distortion of sentences in Document.
2.	Word overlap	Length of contiguous connected spans	
3.	Relative position of sentences in Documents	Largest fertilities	
4.		Length of contiguous unconnected spans	

Classification Features

Sentence Similarity

- Sentence similarity technique is similar to the document similarity techniques.
- Instead of documents, each sentence is represented as a word vector.
- Then, pairwise sentence similarity is calculated for all possible sentence pairs in the aligned document pairs.
- Sentence pairs yielding a similarity score beyond a threshold, are considered to be parallel.
- Similarity score may be computed using TFIDF (in this case, document is a sentence) and cosine similarity.

Parallel phrase extraction

LLR based Parallel Phrase Extraction

- Using Log-Likelihood ratio (Munteanu and Marcu, 2006):
  - Identify which consecutive words in source sentence have translation in target sentence
  - A lexicon obtained by GIZA++ is not very useful because such a lexicon contains entries for even unrelated word pairs.
  - Incorrect correspondences can adversely affect the results that we obtain from this step.
  - Precision is of utmost importance in this step.

LLR based Parallel Phrase Extraction

(Munteanu and Marcu, 2006)

- LLR is Measure of the likelihood that two samples are not independent
  - If source word f and target word e are independent, then p(e|f) = p(e|~f) = p(e)
  - If the words are independent, i.e., these distributions are very similar, the LLR score of this word pair is low. If the words are strongly associated, then the LLR score is high.
- But, a high LLR score implies either positive correspondence (p(e|f) > p(e|~f)) or a negative correspondence (p(e|f) < p(e|~f)) between the words.
  - the set of co-occurring word pairs in the parallel corpus, is split into two sets: positively associated and negatively associated word pairs.
  - co-occurring words are those that are linked together in the word-aligned parallel corpus.
- LLR(e,f) is computed for each of the linked word pairs and then, two conditional probability distributions are computed:
  - P + (e|f) is probability that source word f gets translated to target word e
  - P − (e|f) is probability that source word f does not get translated to target word e

Detecting Parallel Fragments

- The target sentence is considered as a numeric signal.
  - The translated words give positive signals (from P + distribution) and untranslated words give negative signals (from P −) distribution.
  - only that part which is positive, is retained as the parallel fragment of the sentence.
- For each linked target word, the value of the signal is the probability of its alignment link P + (e|f).
  - All the remaining unaligned target words have signal value P − (e|f). This forms the initial signal.
  - Then, a filtering signal is obtained by averaging the signal values of nearby points.
  - The number of points to be used for averaging is decided empirically.
- Then, the "positive signal fragment" of the sentence is retained.
  - This approach tends to produce very short fragments.
  - So, fragments less than 3 words in length can be discarded.
- The procedure can be repeated in the opposite direction and the results can be symmetrized.

Chunking Based Approach

- The comparable sentences were broken into fragments and then, we check which of the fragments have a translation on the target side.
- Instead of segmenting the source sentence into N-grams, chunking is used to obtain linguistic phrases from the source sentences.
  - According to linguistic theory, the tokens within a chunk do not contribute towards long distance reordering, when translated.
  - ad-hoc N-gram segments may not be linguistic phrases, and are always of constant length.
  - Chunks are are variable length and chunks can be merged to form larger chunks or even sentences.

Chunking Source Sentences and Merging Chunks

- CRF-based chunking algorithm is used to chunk the source side sentences.
- Chunks are further merged into bigger chunks, because sometimes, even merged bigger chunks can have a translation on the target side.
- Merging is done in two ways:
  - Strict Merging: Merge two consecutive chunks only if they together form a bigger chunk of length <= 'V' words. 'V' can be an empirically decided value.
  - Window Merging: In this type of merging, not just two, but as many smaller chunks are merged together, as possible, unless the number of tokens in the merged chunk does not exceed 'V'. Then, an imaginary window is slided over to the next chunk and the process is repeated.

Finding Parallel Chunks

- The source side chunks from the previous step are first translated to the target language using the baseline SMT system.
- each of these translated chunks is compared with all the target side chunks of that document pair.
- The overlap between two target side chunks (one translated from source side chunk and the other is a chunk from the target side document) is found out.
  - Overlap(T1 , T2) = Number of tokens in T1 which are aligned in T2
- The overlap of chunk is found both ways symmetrically.
- If at least 70% overlap is found both ways, then the source side chunk corresponding to the translated chunk and the target side chunk are considered as parallel.
- Comparison of tokens for finding the overlap of two chunks is based on orthographic similarities like Levenshtein distance, longest common subsequence ratio and length of the two strings.

Refining the Extracted Parallel Chunks

- From the extracted chunks, it is often observed that ordering of tokens in the source side is different to that of target side.
- Also, there could be some unaligned tokens on either side.
- So, the parallel chunk pairs are refined by reordering source side chunks according to its corresponding target side chunk and the unaligned tokens from either side are discarded.

References: Parallel Corpora Extraction

- Munteanu, D. S. and Marcu, D. (2005). **Improving machine translation performance by exploiting non-parallel corpora.** Computational Linguistics, 31(4):477–504.
- Munteanu, D. S. and Marcu, D. (2006). **Extracting parallel sub-sentential fragments from non-parallel corpora.** In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, pages 81–88. Association for Computational Linguistics.
- Och, F. J. and Ney, H. (2003). **A systematic comparison of various statistical alignment models.** Computational linguistics, 29(1):19–51.
- Resnik, P. and Smith, N. A. (2003). **The web as a parallel corpus.** Computational Linguistics, 29(3):349–380.
- Smith, J. R., Quirk, C., and Toutanova, K. (2010). **Extracting parallel sentences from comparable corpora using document level alignment**. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 403–411. Association for Computational Linguistics.
- Vogel, S. (2005). **Pesa: Phrase pair extraction as sentence splitting**. In Proc. of the Machine Translation Summit, pages 251–258.

PHRASE BASED SMT (PB-SMT)

Acknowledgement: Kashyap Popat (M.Tech student, CSE, IITB)

Outline

- **Motivation**
- Mathematical Model
- Learning Phrase Translations
- Learning Reordering Models
- Discriminative PB-SMT Models
- Decoding
- Overview of Moses
- Summary

Key ideas

- Why stop at learning word correspondences?
- Basic Unit of Translation:
  - "Phrase" (Sequence of Words)
  - Could be 'non-linguistic' phrases

The Prime Minister of India	भारत के प्रधान मंत्री bhaarat ke pradhaan maMtrI India of Prime Minister
is running fast	तेज भाग रहा है tej bhaag raha hai fast run -continuous is
honoured with	से सम्मानित किया se sammanit kiya with honoured did
Rahul lost the match	राहुल मुकाबला हार गया rahul mukaabalaa haar gayaa Rahul match lost

Benefits of PB-SMT

- Local Reordering
  - Intra-phrase re-ordering can be memorized

The Prime Minister of India	भारत के प्रधान मंत्री bhaarat ke pradhaan maMtrI India of Prime Minister

- Sense disambiguation based on local context
  - Neighbouring words help do the right translation

heads towards Pune	पुणे की ओर जा रहे है pune ki or jaa rahe hai Pune towards go –continuous is
heads the committee	समिति की अध्यक्षता करते है Samiti kii adhyakshata karte hai committee of leading -verbalizer is

Benefits of PB-SMT (2)

- Handling institutionalized expressions
  - Institutionalized expressions, idioms can be learnt as a single unit

hung assembly	त्रिशंकु विधानसभा trishanku vidhaansabha
Home Minister	गृह मंत्री gruh mantrii
Exit poll	चुनाव बाद सर्वेक्षण chunav baad sarvekshana

- Improved Fluency
  - The phrases can be arbitrarily long (even entire sentences)

Outline

- Motivation
- **Mathematical Model**
- Learning Phrase Translations
- Learning Reordering Models
- Discriminative PB-SMT Models
- Overview of Moses
- Summary

Mathematical Model

- Decision Rule for the source-channel model

$$\begin{aligned}\mathbf{e}_{\text{best}} &= \text{argmax}_{\mathbf{e}}\ p(\mathbf{e}|\mathbf{f})\\ &= \text{argmax}_{\mathbf{e}}\ p(\mathbf{f}|\mathbf{e})\ p_{\text{LM}}(\mathbf{e})\end{aligned}$$

- Source sentence can be segmented in **I** phrases
- Then, $p(\mathbf{f}|\mathbf{e})$ can be decomposed as:

$$p(\bar{f}_1^I|\bar{e}_1^I) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)\ d(\text{start}_i - \text{end}_{i-1} - 1)$$

Distortion probability

Phrase Translation Probability

start_i :start position in **f** of i^{th} phrase of **e**
end_i :end position in **f** of i^{th} phrase of **e**

Outline

- Motivation
- Mathematical Model
- **Learning Phrase Translations**
- Learning Reordering Models
- Discriminative PB-SMT Models
- Overview of Moses
- Summary

Learning The Phrase Translation Model

Involves Structure + Parameter Learning:

- Learn the **Phrase Table**: the central data structure in PB-SMT

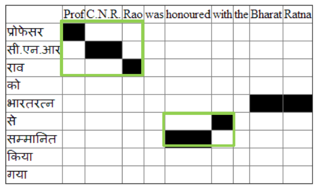
The Prime Minister of India	भारत के प्रधान मंत्री
is running fast	तेज भाग रहा है
the boy with the telescope	दूरबीन से लड़के को
Rahul lost the match	राहुल मुकाबला हार गया

- Learn the **Phrase Translation Probabilities**

Prime Minister of India	भारत के प्रधान मंत्री India of Prime Minister	*0.75*
Prime Minister of India	भारत के भूतपूर्व प्रधान मंत्री India of former Prime Minister	*0.02*
Prime Minister of India	प्रधान मंत्री Prime Minister	*0.23*

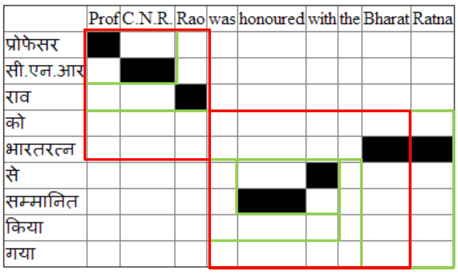
Learning Phrase Tables from Word Alignments

- Leverages word alignments learnt from IBM models
- Word Alignment : reliable input for phrase table learning
  - high accuracy reported for many language pairs
- Central Idea: A consecutive sequence of aligned words constitutes a “phrase pair”

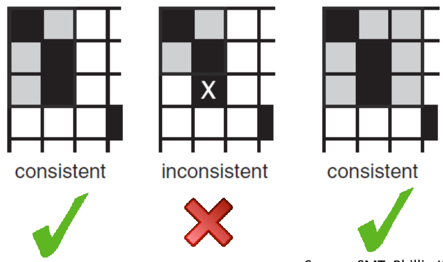


Which phrase pairs to include in the phrase table?

Extracting Phrase Pairs



Phrase Pairs “consistent” with word alignment



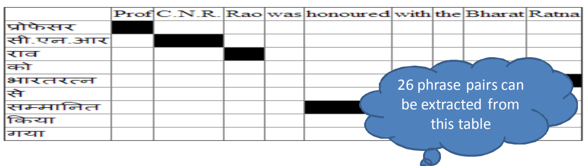
Source: SMT, Phillip Koehn

Phrase Pairs “consistent” with word alignment

$(\bar{e}, \bar{f})$ consistent with $A \Leftrightarrow$

$$\forall e_i \in \bar{e} : (e_i, f_j) \in A \Rightarrow f_j \in \bar{f}$$
$$\text{AND } \forall f_j \in \bar{f} : (e_i, f_j) \in A \Rightarrow e_i \in \bar{e}$$
$$\text{AND } \exists e_i \in \bar{e}, f_j \in \bar{f} : (e_i, f_j) \in A$$

Examples



Professor CNR	प्रोफेसर सी.एन.आर
Professor CNR Rao	प्रोफेसर सी.एन.आर राव
Professor CNR Rao was	प्रोफेसर सी.एन.आर राव
Professor CNR Rao was	प्रोफेसर सी.एन.आर राव को
honoured with the Bharat Ratna	भारतरत्न से सम्मानित
honoured with the Bharat Ratna	भारतरत्न से सम्मानित किया
honoured with the Bharat Ratna	भारतरत्न से सम्मानित किया गया
honoured with the Bharat Ratna	को भारतरत्न से सम्मानित किया गया

Computing Phrase Translation Probabilities

- Estimated from the relative frequency:

$$\boxed{\phi(\bar{f}|\bar{e}) = \frac{\text{count}(\bar{e},\bar{f})}{\sum_{\bar{f}_i}\text{count}(\bar{e},\bar{f}_i)}}$$

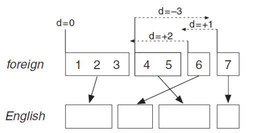
Prime Minister of India	भारत के प्रधान मंत्री India of Prime Minister	*0.75*
Prime Minister of India	भारत के भूतपूर्व प्रधान मंत्री India of former Prime Minister	*0.02*
Prime Minister of India	प्रधान मंत्री Prime Minister	*0.23*

Outline

- Motivation
- Mathematical Model
- Learning Phrase Translations
- **Learning Distortion Models**
- Discriminative PB-SMT Models
- Overview of Moses
- Summary

Distortion Models for PB-SMT

- Model the relative order of phrases
- The distortion models learnt during word-alignment no longer useful for PB-SMT
- Distance based reordering model:
  - Reordering distance: Number of words skipped when taking foreign words out of sequence
    $start_i - end_{i-1} - 1$
  - Distortion probability:
    $d(start_i - end_{i-1} - 1)$



Phrase	Translates	Movement	Distance
1	1–3	start at beginning	0
2	6	skip over 4–5	+2
3	4–5	move back over 4–6	−3
4	7	skip over 6	+1

Source: SMT, Phillip Koehn

Monotone Reordering Distortion Model

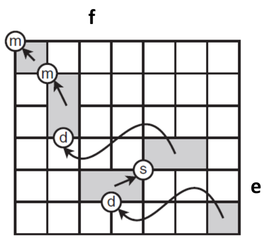
- Penalizes for larger out of sequence movements of phrases
- Naïve reordering model, which can work for language with roughly the same word order

$$\boxed{d(x) = \alpha^{|x|}}$$

$$\boxed{\alpha \in [0,1]}$$

Lexicalized Reordering

- Reordering is conditioned on actual phrase pairs
- However, model will be sparse
- To reduce sparsity, only 3 reordering orientations (O) considered:
  - monotone (m)
  - swap (s)
  - Disjoint (d)

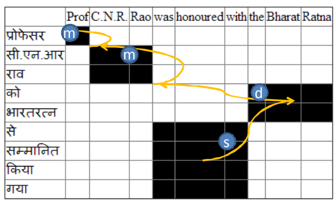


Source: SMT, Phillip Koehn

<u>**Reordering Probability – a smoothed version also exists**</u>

$$\boxed{p_o(\text{orientation}|\bar{f},\bar{e}) = \frac{\text{count}(\text{orientation},\bar{e},\bar{f})}{\sum_o \text{count}(o,\bar{e},\bar{f})}}$$

Example: Lexicalized Reordering



- o(Prof, प्रोफेसर)=m
- o(CNR Rao, सी एन आर राव)=m
- o(the Bharat Ratna, को भारतरत्न)=d
- o(was honoured with, से सम्मानित किया गया)=s

Outline

- Motivation
- Mathematical Model
- Learning Phrase Translations
- Learning Distortion Models
- **Discriminative PB-SMT Models**
- Overview of Moses
- Summary

Generative vs. Discriminative models in Machine Learning

Generative Model

- Noisy channel model of translation from sentence f to sentence e.
- Task is to recover e from noisy f.

$$\hat{\mathbf{e}} = \underset{\mathbf{e}}{\operatorname{argmax}} \Pr(\mathbf{e}) \Pr(\mathbf{f}|\mathbf{e})$$

P(f|e): Translation model, addresses adequacy

P(e): Language model, addresses fluency

- Joint modeling of entire parameter space
- The generative story is too simplistic, not reflective of translation process

Discriminative Model

- Maximum Entropy based model, incorporating arbitrary features

$$\hat{\mathbf{e}} = \underset{e}{\operatorname{argmax}} \exp \sum_i \lambda_i h_i(f, e)$$

- h_i - features functions (phrase/lexical direct/inverse translation probability, LM probability, distortion score)
- λ_i are weights of the features
- No need to model source, reduces parameter space
- Arbitrary features can better capture translation process
- Why exponential function form? – maximizing entropy w.r.t data constraints

Discriminative Training of PB-SMT

- Directly model the posterior probability p**(f|e)**
- Use the Maximum Entropy framework

$$P(\mathbf{e}|\mathbf{f}) = \exp\left(\sum_i \lambda_i h_i(f_1^I, e_1^J)\right)$$

$$e^* = \arg\max_{e_i} \sum_i \lambda_i h_i(f_1^I, e_1^J)$$

  - h_i**(f,e)** are feature functions
  - λ_i's are feature weights
- Benefits:
  - Can add arbitrary features to score the translations
  - Can assign different weight for each features

Generative Model as a special case

Generative model

$$\mathbf{e}_{\text{best}} = \text{argmax}_{\mathbf{e}}\, p(\mathbf{e}|\mathbf{f}) = \text{argmax}_{\mathbf{e}}\, p(\mathbf{f}|\mathbf{e})\, p_{\text{LM}}(\mathbf{e})$$

$$p(\bar{f}_1^I|\bar{e}_1^I) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)\, d(\text{start}_i - \text{end}_{i-1} - 1)$$

Feature function mappings for corresponding discriminative model

$$h_1 = \prod_{i=1}^{I} \phi(f_i, e_i) \quad , \quad \lambda_1 = 1$$ translation model

$$h_2 = \prod_{i=1}^{I} d(start_i - end_{i-1} - 1) \quad , \quad \lambda_2 = 1$$ distortion model

$$h_3 = p_{LM}(\mathbf{e}) \quad , \quad \lambda_3 = 1$$ language model

More features for PB-SMT

- Inverse phrase translation probability ($\phi(\bar{f}|\bar{e})$)
- Lexical Weighting

$$\text{lex}(\bar{e}|\bar{f}, a) = \prod_{i=1}^{\text{length}(\bar{e})} \frac{1}{|\{j|(i,j) \in a\}|} \sum_{\forall (i,j) \in a} w(e_i|f_j)$$

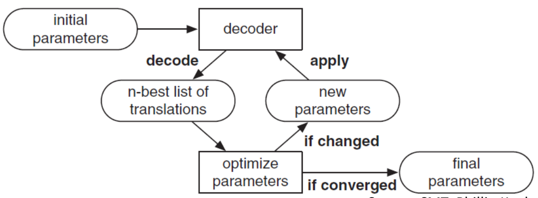
  - *a:* alignment between words in phrase pair (ē, f)
  - *w(x|y):* word translation probability
- Inverse Lexical Weighting
  - Same as above, in the other direction

More features for PB-SMT (2)

- Word Penalty (ω)
  - Control number of words in output
  - ω < 1: output shorter than input sentence
  - ω > 1: output longer than input sentence
- Phrase Penalty (ρ)
  - Control number of phrases
  - ρ < 1: fewer phrases
  - ρ > 1: more phrases

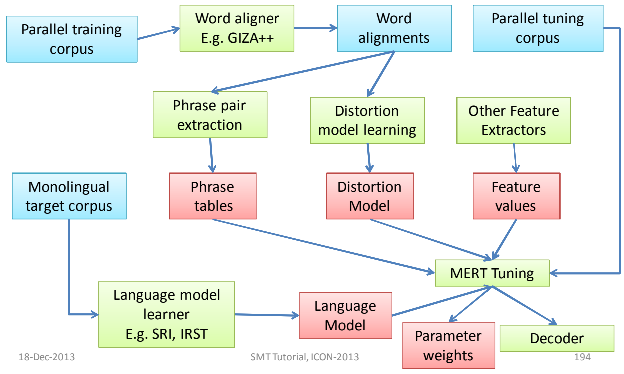
Tuning

- Learning feature weights from data – λ_i
- Minimum Error Rate Training (MERT)
- Search for weights which minimize the translation error on a held-out set (tuning set)
  - Translation error metric : (1 – *BLEU)*



Source: SMT, Phillip Koehn

Overall Training Process for PB-SMT



Moses phrase table

($workspace_dir/model/phrase-table.tgz)

```
' 'Twas he that ||| निखरे माती ||| 0.2 1.39907e-05 1 0.0834042 2.718 ||| 0-0 1-0 2-0 1-1 ||| 5 1 1
' 'Twas he ||| निखरे माती ||| 0.2 0.00209263 1 0.0834042 2.718 ||| 0-0 1-0 2-0 1-1 ||| 5 1 1
' 'Very good. ||| --ठीक तो है। ||| 1 0.0123742 1 7.53276e-05 2.718 ||| 0-0 1-0 2-0 2-1 2-2 ||| 1 1 1
' 'Very well, sir. ||| हाँ सर! ||| 0.5 9.46519e-06 1 0.0063612 2.718 ||| 0-0 1-0 2-0 3-0 3-1 ||| 2 1 1
' 'Very well, then. ||| ठीक ही है। ||| 1 2.77816e-12 1 9.01339e-06 2.718 ||| 0-0 1-0 2-0 3-0 2-1 2-2 ||| 1 1 1
' 'Very well. ||| अच्छा! ||| 0.25 0.00115741 1 0.0434682 2.718 ||| 0-0 1-0 2-0 ||| 8 2 2
' 'Watching me, of all persons. ||| --मुझको? ||| 1 2.14335e-05 1 0.169273 2.718 ||| 0-0 1-0 2-0 3-0 4-0 5-0 ||| 1 1 1
' 'We have heard that you have ||| " "हमने सुना है कि आपने ||| 1 0.000316347 1 7.88927e-08 2.718 ||| 0-0 1-1 2-1 3-2 4-3 4-4 5-5 6-5 ||| 1 1 1
' 'We have heard that ||| " "हमने सुना है कि ||| 1 0.00391593 1 2.99769e-06 2.718 ||| 0-0 1-1 2-1 3-2 4-3 4-4 ||| 1 1 1
' 'We have heard ||| " "हमने सुना ||| 1 0.0118525 1 4.3827e-05 2.718 ||| 0-0 1-1 2-1 3-2 ||| 1 1 1
' 'We have ||| " "हमने ||| 1 0.0282705 1 0.00021881 2.718 ||| 0-0 1-1 2-1 ||| 1 1 1
' 'Well, I do take rest, father. ||| मृत्यु क्या है? ||| 1 5.60474e-20 1 1.34553e-05 2.718 ||| 0-0 1-0 2-0 4-0 5-0 6-0 1-1 3-1 1-2 ||| 1 1 1
' 'Well, it happens. ||| --जरुर होता है। ||| 1 0.00130446 1 0.000452107 2.718 ||| 0-0 1-0 2-0 3-0 3-1 3-2 ||| 1 1 1
' 'Well, people who are good at ||| जो ||| 7.19321e-05 7.11023e-21 1 0.299537 2.718 ||| 3-0 ||| 13902 1 1
```

- inverse phrase translation probability
- inverse lexical weighting
- direct phrase translation probability
- direct lexical weighting
- phrase penalty (always exp(1) = 2.718)
- Within-phrase alignment information

Moses model file ($workspace_dir/model/moses.ini)

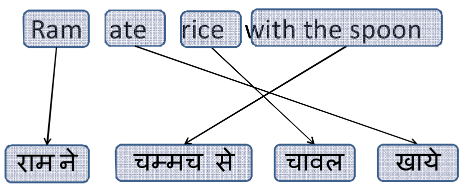
```
#########################
### MOSES CONFIG FILE ###
#########################

# input factors
[input-factors]
0

# mapping steps
[mapping]
0 T 0

# translation tables: table type (hierarchical(0), textual (0), binary (1)), source-factors, target-factors, number of scores, file
# OLD FORMAT is still handled for back-compatibility
# OLD FORMAT translation tables: source-factors, target-factors, number of scores, file
# OLD FORMAT a binary table type (1) is assumed
[ttable-file]
0 0 0 5 /home/anoop/tmp/sample_data/workspace/moses_data/model/phrase-table.gz

# no generation models, no generation-file section

# language models: type(srilm/irstlm), factors, order, file
[lmodel-file]
0 0 3 /home/anoop/tmp/sample_data/sample_monolingual.en.lm


# limit on how many phrase translations e for each phrase f are loaded
# 0 = all elements loaded
[ttable-limit]
20

# distortion (reordering) files
[distortion-file]
0-0 wbe-msd-bidirectional-fe-allff 6 /home/anoop/tmp/sample_data/workspace/moses_data/model/reordering-table.wbe-msd-bidirectional-fe.gz

# distortion (reordering) weight
[weight-d]
0.3
0.3
0.3
0.3
0.3
0.3
0.3

# language model weights
[weight-l]
0.5000


# translation model weights
[weight-t]
0.20
0.20
0.20
0.20
0.20

# no generation models, no weight-generation section

# word penalty
[weight-w]
1
```


Decoding

Searching for the best translations in the space of all translations

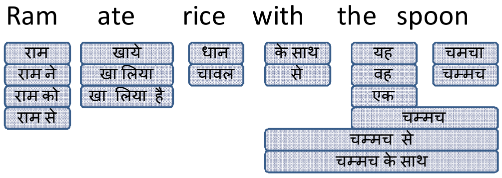
$$e^* = \arg\max_{e_i} \sum_i \lambda_i h_i(f_1^J, e_1^I)$$

An Example of Translation



Reality

- We picked the phrase translation that made sense to us
- The computer has less intuition
- Phrase table may give many options to translate the input sentence



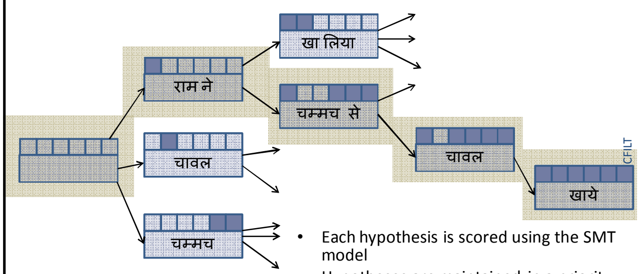
Decoding

- The task of decoding in machine translation is to find the best scoring translation according to translation models
- Hard problem, since there is a exponential number of choices, given a specific input sentence
- Shown as an NP complete problem
- Need to come up with heuristic search methods
- No guarantee of finding the best translation

Incremental Construction

- **Hypotheses:** partial translations
  - Which input words have been translated?
  - The chosen translations for these words
  - Hypotheses are constructed in target language order, source words may be chosen out of sequence.
- **Expansion:** when we pick one of the translation options and construct a new hypothesis
- Start with the empty hypothesis
- Expansion is carried out recursively until all the hypotheses get expanded
- A hypothesis that covers all input words forms an end point

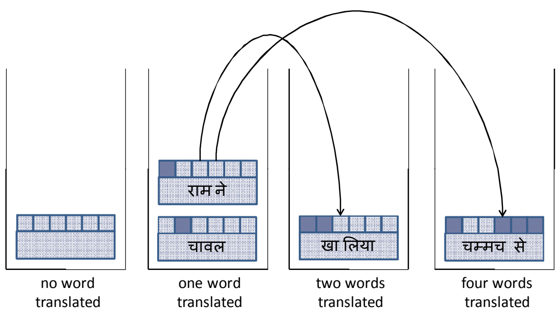
Search Space and Search Organization



- Each hypothesis is scored using the SMT model
- Hypotheses are maintained in a priority queue (called stack decoding historically)
- Limit to the reordering window for efficiency

Multi-Beam Search

- Shorter hypothesis will have higher score. Solution:
  - Organize hypotheses into hypothesis stacks (pile)
  - Based on the number of input word translated
- When a word is translated, hypothesis is transferred to a different stack
- Are hypotheses that have the **same number of words translated** comparable?
- Priority queue size is bounded
- If the stack gets full, we prune out the worst hypotheses from the stack – Beam search



Pseudo-code[6]

```
place empty hypothesis into stack 0
for all stacks 0...n−1 do
  for all hypotheses in stack do
    for all translation options do
      if applicable then
        create new hypothesis
        place in stack
        recombine with existing hypothesis if possible
        prune stack if too big
      end if
    end for
  end for
end for
```


Pruning

- To remove the bad hypotheses from the stacks
- Uses partial score of the translation
- Two types:

1. Histogram pruning
   - Keep a maximum number *'n'* of hypotheses in the stack
   - Inconsistent in pruning out bad hypotheses
2. Threshold pruning
   - Proposes a fixed *'α'*, by which a hypothesis is allowed to be worse than the best one in stack

Problem with the approach

- Comparing hypotheses with the **same number of foreign words translated** and pruning out the ones that have the worst probability score!
- Some parts of the sentence may be easier to translate than others
- Hypotheses that translate the easy part first are unfairly preferred to ones that do not
- *e.g.*, the translation of unusual nouns and names is usually more expensive than the translation of common function words

Future cost

- The expected cost of translating the rest of the sentence
- Base pruning decision not only on the hypotheses score but also on future cost
- Computationally too expensive to compute the expected cost

Future cost estimation

- Translation model
  - Phrase translation table look up
- Language Model
  - Can not compute the probability without knowing the preceding words
  - Unigram probability for the first word of the output phrase, bigram probability for the second word and so on
- *"the partial score + the future score"* : better measure of the quality of a hypothesis – A* search
- Lower search error than using just the probability score

Phrase based SMT systems for Indian languages

Work with Abhijit Mishra, Rajen Chatterjee and Ritesh Shah

Pan-Indian Language SMT

http://www.cfilt.iitb.ac.in/indic-translator

- SMT systems between 11 languages
  - 7 Indo-Aryan: Hindi, Gujarati, Bengali, Oriya, Punjabi, Marathi, Konkani
  - 3 Dravidian languages: Malayalam, Tamil, Telugu
  - English
- Corpus
  - Indian Language Corpora Initiative (ILCI) Corpus
  - Tourism and Health Domains
  - 50,000 parallel sentences
- Evaluation with BLEU
  - METEOR scores also show high correlation with BLEU

SMT Systems Trained (PBSMT+extensions)

- **Phrase-based** (PBSMT) baseline system (S1)
- E-IL PBSMT with **Source side reordering rules** *(Ramanathan et al., 2008)* (S2)
- E-IL PBSMT with **Source side reordering rules** *(Patel et al., 2013)* (S3)
- IL-IL PBSMT with **transliteration post-editing** (S4)

Natural Partitioning of SMT systems

	hi	ur	pa	bn	gu	mr	kK	ta	te	ml	en
[illegible]	[illegible]	[illegible]	68.21	34.96	51.31	39.12	37.81	14.43	21.38	10.98	29.23
[illegible]	[illegible]	[illegible]	52.02	29.59	39.00	27.57	28.29	11.95	16.61	8.65	22.46
[illegible]	[illegible]	[illegible]		29.89	43.85	30.87	30.72	10.75	18.81	9.11	[illegible]
bn	[illegible]	32.08	31.38		28.14	22.09	23.47	10.94	13.40	8.10	[illegible]
gu	55.66	[illegible]	45.14	28.50		32.06	30.48	12.57	17.22	8.01	[illegible]
[illegible]	[illegible]	32.60	33.28	23.73	32.42		27.81	10.74	12.89	[illegible]	[illegible]
[illegible]	[illegible]	34.00	34.31	24.59	31.07	27.52		10.36	[illegible]	7.89	17.07
[illegible]	[illegible]	18.12	15.57	13.21	16.53	11.60	11.87		8.48	6.31	11.79
[illegible]	[illegible]	25.07	25.56	16.57	20.96	14.94	17.27	8.68		6.68	12.34
ml	[illegible]	13.39	12.97	10.67	9.76	8.39	9.18	5.90	5.94		8.61
en	28.94	22.96	22.33	15.33	15.44	12.11	13.66	6.43	6.55	4.65	

High accuracy between Indo-Aryan languages

Low accuracy between Dravidian languages

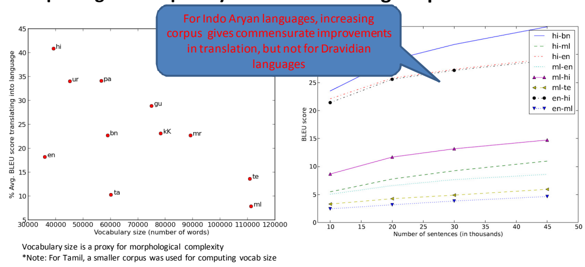
Structural Divergence between English-IL results in low accuracy

Baseline PBSMT - % BLEU scores (S1)

- **Clear partitioning of translation pairs by language family pairs**, based on translation accuracy.
  - Shared characteristics within language families make translation simpler
  - Divergences among language families make translation difficult
- ***Language families are the right level of generalization*** for building SMT systems in continuum from totally language independent systems to per language pair system continuum

The Challenge of Morphology

Morphological complexity vs BLEU **Training Corpus size vs BLEU**



Vocabulary size is a proxy for morphological complexity
*Note: For Tamil, a smaller corpus was used for computing vocab size

- Translation accuracy decreases with increasing morphology
- Even if training corpus is increased, commensurate im5provement in translation accuracy is not seen for morphologically rich languages
- ***Handling morphology in SMT is critical***

Common Divergences, Shared Solutions

System	hi	ur	pa	bn	gu	mr	kK	ta	te	ml
Baseline PBSMT	28.94	22.96	22.33	15.33	15.44	12.11	13.66	6.43	6.55	4.65
Source Reordering (Generic)	31.41	24.85	24.56	15.89	17.38	13.42	14.55	7.84	8.23	4.95
Source Reordering (Hindi-adapted)	33.54	26.67	26.23	17.86	19.06	14.15	15.56	7.96	8.37	5.30

Comparison of source reordering methods for E-IL SMT - % BLEU scores (S1,S2,S3)

- All Indian languages have similar word order
- The same structural divergence between English and Indian languages *SOV<->SVO, etc.*
- **Common source side reordering rules** improve E-IL translation by 11.4% (generic) and 18.6% (Hindi-adapted)
- ***Common divergences can be handled in a common framework in SMT systems*** (This idea has been used for knowledge based MT systems e.g. *Anglabharati*)

Harnessing Shared Characteristics

	hi	ur	pa	bn	gu	mr	kK	ta	te	ml
hi		61.28	64.85	35.49	52.98	39.12	37.81	14.52	21.68	11.07
ur	61.42		52.02	29.59	39.00	27.57	28.29	11.95	16.61	8.65
pa	74.14	56.00		30.05	44.39	31.46	30.99	10.77	18.96	9.12
bn	38.17	32.08	31.54		28.73	22.60	23.79	10.97	13.52	8.17
gu	57.22	44.12	45.55	28.90		33.22	31.55	12.64	17.46	8.05
mr	45.11	32.60	30.97	24.09	33.48		27.81	10.80	13.12	7.68
kK	41.92	34.00	32.04	24.91	32.05	27.52		10.40	14.92	7.96
ta	20.54	18.12	15.57	13.25	16.57	11.64	11.94		8.57	6.40
te	29.23	25.07	25.67	16.68	21.20	15.19	17.43	8.71		6.77
ml	14.81	13.39	12.98	10.73	9.84	8.42	9.25	5.99	6.02	

PBSMT+ transliteration post-editing for E-IL SMT - % BLEU scores (S4)

- Out of Vocabulary words are transliterated in a post-editing step
- Done using a simple transliteration scheme which harnesses the common phonetic organization of Indic scripts
- Accuracy Improvements of 0.5 BLEU points with this simple approach
- ***Harnessing common characteristics can improve SMT output***

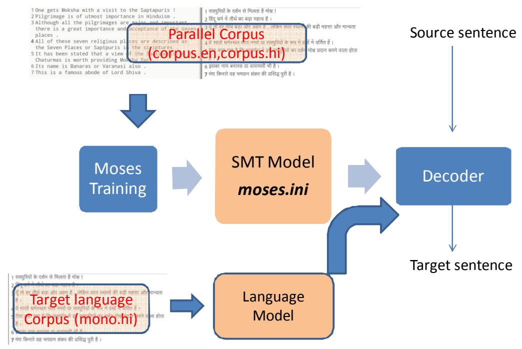
Outline

- Motivation
- Mathematical Model
- Learning Phrase Translations
- Learning Distortion Models
- Discriminative PB-SMT Models
- **Overview of Moses**
- Summary

What is Moses?

- Most widely used **phrase-based** SMT framework
  - *'Moses'* actually refers to the SMT decoder
  - However, includes training, tuning, pre-processing tools, etc.
  - Open-source, modular and extensible - developed primarily at the University of Edinburgh
- Written in C++ along with supporting scripts in various languages
  - https://github.com/moses-smt/mosesdecoder
- Also supports *factored, hierarchical phrase based, syntax based* MT systems
  - Other decoders of interest: cdec, Joshua, ISI ReWrite
- Visit: http://www.statmt.org/moses/

What does Moses do?



Installing Moses

- Compile and install the following:
  - Moses
  - GIZA++
  - Language Modelling toolkit (SRILM/IRSTLM)
- Installation Guides
  - From StatMT: http://www.statmt.org/moses_steps.html
  - Works best for Ubuntu: http://organize-information.blogspot.in/2012/01/yet-another-moses-installation-guide.html
  - A bit older guide: http://www.cfilt.iitb.ac.in/Moses-Tutorial.pdf
- Be ready for a few surprises !

Workflow for building a phrase based SMT system

- **Corpus Split:** Train, Tune and Test split
- **Pre-processing**: Normalization, tokenization, etc.
- **Training**: Learn Phrase tables from *Training* set
- **Tuning**: Learn weights of discriminative model on *Tuning* set
- **Testing**: Decode *Test* set using tuned data
- **Post-processing**: regenerating case, re-ranking
- **Evaluation**: Automated Metrics or human evaluation

Pre-processing -1 (Normalize the text) Case normalization

- Recasing method:
  - Convert training data to lowercase
  - Learn recasing model for target language

```
scripts/recaser/train-recaser.perl --dir MODEL --corpus CASED [--ngram-count NGRAM] [--train-script TRAIN]
```

  - Restore case in test output using recasing model

```
scripts/recaser/recase.perl --in IN --model MODEL/moses.ini --moses MOSES >OUT
```

- Truecasing method
  - Learnt via True casing model

```
scripts/recaser/train-truecaser.perl --model MODEL --corpus CASED
```

  - Convert words at start of sentence to lowercase (if they generally occur in lowercase in corpus)

```
scripts/recaser/truecase.perl --model MODEL < IN > OUT
```

  - Restore case in test output using truecasing model

```
scripts/recaser/detruecase.perl < in > out
```

Pre-processing -1 (Normalize the text) Character Normalization

Important for Indic scripts

- Multiple Unicode representations
  - e.g. ज़ can be represented as +u095B or +u091c (ज) +1093c (nukta)
- Control characters
  - Zero-Width Joiner/Zero-Width Non-Joiner
- Characters generally confused
  - Pipe character (|) with *poorna-virama* (।)
  - Colon(:) with *visarga* (◌ः)

https://bitbucket.org/anoopk/indic_nlp_library

Preprocessing-2 (Other steps)

- Sentence splitting
  - Stanford Sentence Splitter
  - Punkt Tokenizer (NLTK library)
- Tokenization
  - Scripts/tokenizer/tokenizer.perl
  - Stanford Tokenizer
  - Many tokenizers in the NLTK library

Train Language Model

- Supported LM tools:
  - KenLM comes with Moses
  - SRILM and IRSTLM are other supported language models
- Can train with one and test with another LM
  - All generate output in ARPA format
- **Training SRILM based language model**

```
ngram-count -order <n> -kndiscount -interpolate -text <corpus> -lm <lmfile>
```


Training Phrase based model

- The training script (train-model.perl) is a meta-script which does the following:
  - Run GIZA
  - Align words
  - Extract Phrases
  - Score Phrases
  - Learn Reordering model
- Run the following command

```
scripts/training/train-model.perl \
  -external-bin-dir <external_bin_dir>
  -root-dir <workspace_dir> \
  -corpus <train_path_without_ext> \
  -e <tgt_lang> -f <src_lang> \
  -alignment <phrase_extraction_strategy e.g. grow-diag-final-and> \
  -reordering <reordering_strategy e.g. msd-bidirectional-fe>
  -lm <lm_type, 0 for srilm>:<lm_order>:<lm_file>:0
```


More Training Options

- Configure maximum phrase length
  - -max-phrase-length
- Train the SMT system in parallel
  - -parallel
- Options for parallel training
  - -cores, -mgiza, -sort-buffer-size, -sort-parallel, etc.

Tuning the Model

- Tune the parameter weights to maximize translation accuracy on '*tuning set*'
- Different tuning algorithms are available:
  - MERT, PRO, MIRA, Batch MIRA
- Generally, a small tuning set is used (~500-1000 sentences)
- MERT (Minimum Error Rate Tuning) is most commonly used tuning algorithm:
  - Model can be tuned to various metrics (BLEU, PER, NIST)
  - Can handle only a small number of features

MERT Tuning

- Command:

```
scripts/training/mert-moses.pl <tun_src_file>
   <tun_tgt_file>  <decoder_binary_path>  \
   <untuned_model_file> --working-dir <workspace> --rootdir
   <moses_script_dir>
```

- Important Options
  - Maximum number of iterations. Default: 25
    `--maximum-iterations=ITERS`
  - How big nbestlist to generate
    `--nbest=100`
  - Run decoder in parallel
    `--jobs=N`

Decoding test data

- Decoder command
  `bin/moses  -config <moses_config> -input-file <input_file>`
- Other common decoder options
  - alignment-output-file <file>: output alignment information
  - n-best-list: generate n-best outputs
  - threads: number of threads
  - ttable-limit: number of translations for every phrase
  - xml-input: supply external translations (named entities, etc.)
  - minimum-bayes-risk: use MBR decoding to get best translation
  - Options to control stack size

Evaluation Metrics

- Argument for validation of automated metrics: correlation with human judgments
- Automatic Metrics:
  - BLEU (Bilingual Evaluation Understudy)
  - METEOR: More suitable for Indian languages since it allows synonym, stemmer integration
  - TER, NIST
- Commands
  - Bleu scoring tool:
    scripts/generic/multi-bleu.perl
  - Mteval scoring tool: official scoring tool at many workshops (BLEU and NIST)
    scripts/generic/mteval-v13a.pl

More Moses Goodies

- XML RPC server
- Binarize the phrase tables
- Load Phrase table on demand
- Experiment Management System (EMS)
- A simpler EMS
  - https://bitbucket.org/anoopk/moses_job_scripts
- … continue exploring

Outline

- Motivation
- Mathematical Model
- Learning Phrase Translations
- Learning Distortion Models
- Discriminative PB-SMT Models
- Overview of Moses
- **Summary**

Summary

- Basic Unit of Translation: word sequences "phrases"
- Learn phrase translation pairs from word alignments
  - There are methods of directly learning phrase translation pairs from corpora
- Basically, "memorizes" phrase translation pairs
  - Corpus provides confidence scores
- Reordering is difficult to model in PB-SMT
  - Looked at two simple models

Summary (2)

- Pros
  - Local reordering, some local sense disambiguation, fluency and institutionalized phrases
- Cons
  - Does not generalize well
- Discriminative learning
  - Ability to have arbitrary features to provide evidence
- Beam search based decoding: heuristic approach
- One of the most successful SMT approaches

Extensions to PBSMT

- Reordering
  - Source side reordering (rule based, learning)
  - Hierarchical Phrase based SMT
- Handling Morphological Complexity
  - Factor Based SMT
- Re-ranking of top-k best translation candidates

References: Phrase Based SMT

- Philipp Koehn. *Statistical Machine Translation.* Cambridge University Press. 2010.
- Philipp Koehn, Franz Josef Och, and Daniel Marcu. *Statistical phrase-based translation.* NAACL. 2003.
- Franz Josef Och and Hermann Ney. *The alignment template approach to statistical machine translation.* Computational Linguistics. 2004.
- Franz Josef Och, and Hermann Ney. *Discriminative training and maximum entropy models for statistical machine translation.* ACL. 2002.
- Och, Franz Josef. *Minimum error rate training in statistical machine translation.* ACL. 2003.
- Marcu, Daniel, and William Wong. *A phrase-based, joint probability model for statistical machine translation.* EMNLP. 2002.
- Koehn, Philipp, et al. *Moses: Open source toolkit for statistical machine translation.* ACL Demo Session. 2007.

References: Decoding in SMT

- Ye-Yi Wand and Alex Waibel. *Decoding Algorithm in Statistical Machine Translation*. EACL. 1997.
- Franz Josef Och, Nicola Ueffing, Hermann Ney. *An Efficient A* Search Algorithm for Statistical Machine Translation*. ACL. 2001.
- Knight, Kevin. *Decoding complexity in word-replacement translation models*. Computational Linguistics. 1999

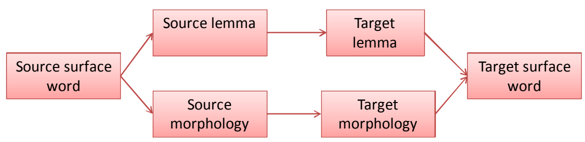
FACTOR BASED SMT

Motivation

- Phrase-based models translate the words based on their surface form only

Ex. Horse-Horses

- Even if 'horses' is present in the training data, we can not translate 'horse' or vice-versa
- To cover all such morphological forms of each word in phrase-based models, we require huge parallel corpora



Motivation

- Phrase-based models can not differentiate between various morphological forms of words

Ex. Boys -> लड़के (*ladake*), लड़कों (*ladakon*)

- These morphological forms require some extra information apart from surface word to be used while translating from English to Hindi
- Factored models support incorporation of such linguistic information

Generalization

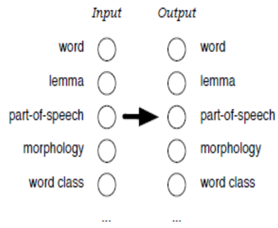
- Factored models are in-fact generalization of phrase-based models
- Phrase-based models are special case of factored models

Outline

- Motivation
- **What are factored models?**
- Decomposition of Factored translation
- Statistical modeling of Factored models
- Disadvantages of Factored models
- Case-studies

Factored translation models

- Extension of Phrase-based models to include linguistic information
- Word is not only a token, but a vector of factors that represent different levels of annotation

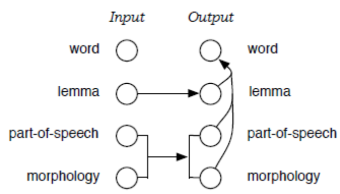


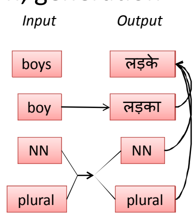
Outline

- Motivation
- What are factored models?
- **Decomposition of Factored translation**
- Statistical modeling of Factored models
- Disadvantages of Factored models
- Case-studies

Decomposition of Factored translation

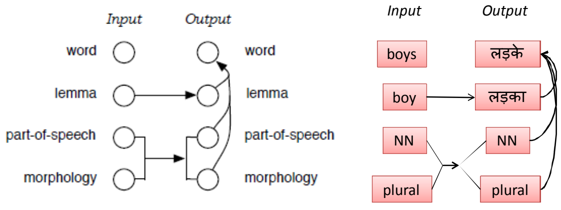
- A single translation is broken down into a sequence of mapping steps
- Types of mappings: Translation, generation





Decomposition of Factored translation

- Translation steps map factors in source phrases to factors in target phrases
- Generation steps map target factors within individual target words



Example (Generating translation options)

Parallel factored corpus:

boys|boy|NN|directCase|plural
लड़के|लड़का|NN|-e
युवक|युवक|NN|-e
...

boys|boy|NN|obliqueCase|plural
लड़को |लड़का|NN|-on
युवको|युवक|NN|-on
...

- Factored model:
  - Translation step 1: Map lemma
  - Translation step 2: Map morphology
  - Generation step 1: Generate surface from lemma and morphology

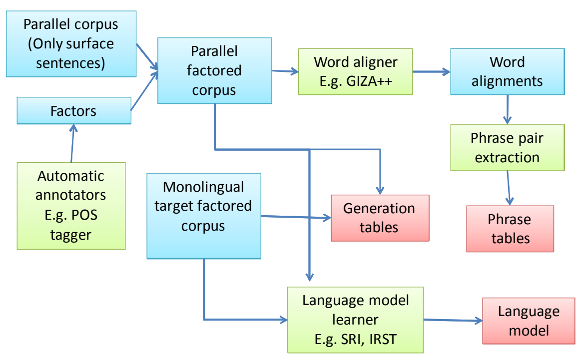
Example (Generating translation options)

- Source phrase: boys|boy|NN|directCase|plural
- Translation step 1: Mapping lemmas
  boy → लड़का (ladka), युवक (yuvak), etc.
- Translation step 2: Mapping morphology
  NN|directCase|plural → NN|-e, NN|-on, etc.
- Generation step 1: Generating surface forms
  लड़का|NN|-e → लड़के (ladke)
  लड़का|NN|-on → लड़को (ladkon)
  युवक|NN|-e → युवक (yuvak)
  युवक|NN|-on → युवको (yuvakon)
- Translation options:
  लड़के|लड़का|NN|-e
  लड़को |लड़का|NN|-on
  युवक|युवक|NN|-e
  युवको|युवक|NN|-on

Outline

- Motivation
- What are factored models?
- Decomposition of Factored translation
- **Statistical modeling of Factored models**
  - Training
  - Combination of components (Log-linear model)
  - Decoding
- Disadvantages of Factored models
- Case-studies

Training factored models



Factored parallel corpus

- Source sentences (English):

ram|ram|NN eats|eat|VBZ mango|mango|NN .|.|NA
sita|sita|NN is|be|VBZ playing|play|VBG cricket|cricket|NN .|.|NA
laxman|laxman|NN ate|eat|VBD an|an|DT apple|apple|NN .|.|NA

- Target sentences (Hindi):

राम|राम|NN आम|आम|NN खाता|खा|VBZ है|है|VAUX ।|।|।
सीता|सीता|NN क्रिकेट|क्रिकेट|NN खेल|खेल|VBG रही|रह|VAUX है|है|VAUX ।|।|।
लक्ष्मण|लक्ष्मण|NN ने|ने|CM सेब|सेब|NN खाया|खा|VBD ।|।|।

Sample factored model

- Translation step 1: Map lemmas
- Translation step 2: Map POS tag
- Generation step: Generate target surface from lemma and POS tag

Phrase-tables

- Lemma-lemma phrase-table

```
. ||| । ||| 1 1 1 1 2.718 ||| 0-0 ||| 3 3 3
be play cricket . ||| क्रिकेट खेल रह है । ||| 1 0.5 1 0.25 2.718 ||| 0-0 0-1 1-2 2-3 3-4 ||| 1 1 1
be play cricket ||| क्रिकेट खेल रह है ||| 1 0.5 1 0.25 2.718 ||| 0-0 0-1 1-2 2-3 ||| 1 1 1
be play ||| क्रिकेट खेल रह ||| 1 1 1 0.25 2.718 ||| 0-0 0-1 1-2 ||| 1 1 1
be ||| क्रिकेट खेल ||| 1 1 1 0.25 2.718 ||| 0-0 0-1 ||| 1 1 1
cricket . ||| है । ||| 1 0.5 1 1 2.718 ||| 0-0 1-1 ||| 1 1 1
cricket ||| है ||| 1 0.5 1 1 2.718 ||| 0-0 ||| 1 1 1
eat ||| खा ||| 1 0.333333 1 0.333333 2.718 ||| 0-0 ||| 1 1 1
laxman eat an apple . ||| लक्ष्मण ने सेब खा । ||| 1 0.0520833 1 0.0651042 2.718 ||| 0-0 0-1 1-1 1-2 2-2 2-3 3-3 4-4 ||| 1 1 1
laxman eat an apple ||| लक्ष्मण ने सेब खा ||| 1 0.0520833 1 0.0651042 2.718 ||| 0-0 0-1 1-1 1-2 2-2 2-3 3-3 ||| 1 1 1
play cricket . ||| रह है । ||| 1 0.5 1 1 2.718 ||| 0-0 1-1 2-2 ||| 1 1 1
play cricket ||| रह है ||| 1 0.5 1 1 2.718 ||| 0-0 1-1 ||| 1 1 1
play ||| रह ||| 1 1 1 1 2.718 ||| 0-0 ||| 1 1 1
```

Phrase-tables

- POS-POS phrase-table

```
NA ||| I ||| 1 1 1 1 2.718 ||| 0-0 ||| 3 3 3
NN NA ||| VAUX I ||| 1 0.666667 1 0.222222 2.718 ||| 0-0 1-1 ||| 1 1 1
NN VBD DT NN NA ||| NN CM NN VBD I ||| 1 0.0274658 1 0.0259345 2.718 ||| 0-0 0-1 1-1 1-2 2-2 2-3 3-3 4-4 ||| 1 1 1
NN VBD DT NN ||| NN CM NN VBD ||| 1 0.0274658 1 0.0259345 2.718 ||| 0-0 0-1 1-1 1-2 2-2 2-3 3-3 ||| 1 1 1
NN VBZ NN NA ||| NN NN VBZ VAUX I ||| 1 0.403646 1 0.0228624 2.718 ||| 0-0 0-1 2-1 1-2 2-3 3-4 ||| 1 1 1
NN VBZ NN ||| NN NN VBZ VAUX ||| 1 0.403646 1 0.0228624 2.718 ||| 0-0 0-1 2-1 1-2 2-3 ||| 1 1 1
NN VBZ VBG NN NA ||| NN NN VBG VAUX VAUX I ||| 1 0.078125 1 0.0137174 2.718 ||| 0-0 1-1 1-2 2-3 3-4 4-5 ||| 1 1 1
NN VBZ VBG NN ||| NN NN VBG VAUX VAUX ||| 1 0.078125 1 0.0137174 2.718 ||| 0-0 1-1 1-2 2-3 3-4 ||| 1 1 1
NN VBZ VBG ||| NN NN VBG VAUX ||| 1 0.117187 1 0.0617284 2.718 ||| 0-0 1-1 1-2 2-3 ||| 1 1 1
```

Generation tables

- Lemma, POS -> Surface

```
खा|VBD खाया 1.0000000 1.0000000
ने|CM ने 1.0000000 1.0000000
क्रिकेट|NN क्रिकेट 1.0000000 1.0000000
।|I । 1.0000000 1.0000000
सेब|NN सेब 1.0000000 1.0000000
आम|NN आम 1.0000000 1.0000000
खा|VBZ खाता 1.0000000 1.0000000
खेल|VBG खेल 1.0000000 1.0000000
रह|VAUX रही 1.0000000 1.0000000
लक्ष्मण|NN लक्ष्मण 1.0000000 1.0000000
राम|NN राम 1.0000000 1.0000000
है|VAUX है 1.0000000 1.0000000
सीता|NN सीता 1.0000000 1.0000000
```

Outline

- Motivation
- What are factored models?
- Decomposition of Factored translation
- **Statistical modeling of Factored models**
  - Training
  - Combination of components (Log-linear model)
  - Decoding
- Disadvantages of Factored models
- Case-studies

Combination of components

- Log-linear model:

$$p(e|f) = 1/Z \exp \sum_i \lambda_i h_i(e, f)$$

- Models and Feature functions:

Models	Feature functions
Language model	$h_{LM}(e, f) = p(e_1).p(e_2 \mid e_1).p(e_3 \mid e_2)...p(e_m \mid e_{m-1})$
Translation model	$h_T(e, f) = \sum_j \tau (f_j, e_j)$
Generation model	$h_G(e, f) = \sum_k \gamma (e_k)$

Understanding the factored model

Source sentence: *F*

Target sentence: *E*

Number of phrases: *1 ... k*

(Note: no. of phrases should be same on source and target side)

Objective function:

$$E^* = \text{argmax}_E \{Pr(E|F)\}$$

Understanding the factored model

Objective function:

$$E^* = \text{argmax}_E \{Pr(E|F)\}$$

Phrase-based model:

$$Pr(E|F) = \text{argmax}_E \{p(F|E).p(E)\}$$

Factored model: Combination of independent feature functions

$$Pr(E|F) = exp(\sum_{m=1}^{M} \lambda_m h_m (E, F)) / Z$$

Feature functions

Source factors: 1...S

Target factors: 1...T

Translation step: Mapping $s \subseteq \{1...S\}$ to $t \subseteq \{1...T\}$

$$h_{s\text{->}t}(E, F) \triangleq \sum_k \tau_{s\text{->}t} (f_k, e_k) = \sum_k \log p(f_k^s | e_k^t)$$

Ex. s={lemma, POS}, t={lemma, POS}

Ram|Ram|NN *eats|eat|VB* *mango|mango|NN*

f_1^s f_3^s f_2^s

राम|राम|NN *आम|आम|NN* *खाता|खा|VBZ* *है|है|VAUX*

e_1^t e_2^t

e_3^t

Feature functions

Source factors: 1...S

Target factors: 1...T

Generation step: Mapping $t_1 \subseteq \{1...T\}$ to $t_2 \subseteq \{1...T\}$

$$h_{t1\text{->}t2}(E, F) \triangleq \sum_k \gamma_{t1\text{->}t2} (e_k) = \sum_k \log \{\pi_{i=1}^{len(ek)} p(e_{k,i}^{t1} | e_{k,i}^{t2})\}$$

Ex. t_1*={surface},* t_2*={lemma, POS }*

राम|राम|NN *आम|आम|NN* *खाता|खा|VBZ* *है|है|VAUX*

$e_{3,1}^{t1}$ $e_{3,1}^{t2}$ $e_{3,2}^{t1}$ $e_{3,2}^{t2}$

Feature functions

Source factors: 1...S

Target factors: 1...T

Language model: over $t \subseteq \{1...T\}$

$$h_t(E, F) \triangleq L_t(E) = \log \{\prod_{i=1}^{l} p(e_i^t | e_{i-1}^t, e_{i-2}^t, e_{i-3}^t, ...)\}$$

*e_i is i^{th} word in the sentence E

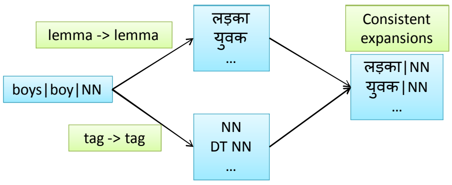
Note: There can be multiple translation, generation and language models

Outline

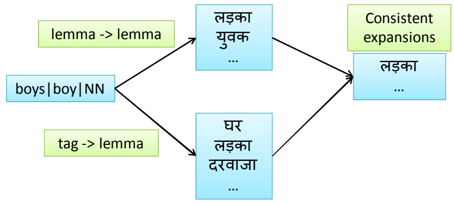
- Motivation
- What are factored models?
- Decomposition of Factored translation
- **Statistical modeling of Factored models**
  - Training
  - Combination of components (Log-linear model)
  - Decoding
- Disadvantages of Factored models
- Case-studies

Consistent expansion

- If the target side has the same length for each target factor and if the shared factors among the mapping steps match
- During decoding, consistency is used to prune out the unlikely translation options



Consistent expansion (2)



Note: Order of application of mapping steps plays important role in this case

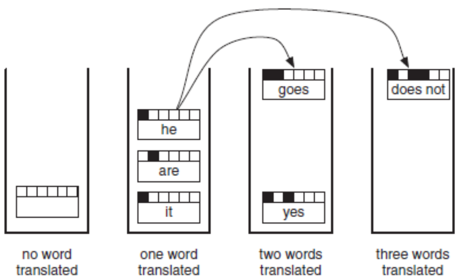
Decoding

- Entries in the phrase table that may be potentially used for a specific input sentence are called *Translation options*
- The decomposition of phrase translation into several mapping steps leads to additional computational complexity
- Multiple tables have to be searched instead of a single table look-up

Decoding

- Decoding algorithm is similar to that of a Phrase-based model (Stack based Beam search)
  - Start with an empty hypothesis
  - New hypotheses are generated by using all applicable translation options
  - Hypotheses are created until we get the hypotheses that covers the full input sentence
  - The highest scoring complete hypothesis indicates the best translation according to the model

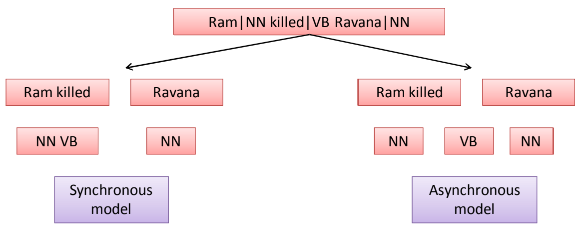
Stack decoding



Source: SMT by Koehn

Synchronous Factored model

- All mapping steps operate on the same phrase segmentation of the input and output sentence
- These models are called *Synchronous factored models*
- Synchronous models help reduce decoding complexity



Efficient decoding

- All mapping steps operate on the same phrase segmentation of input sentence
- The expansions can be efficiently pre-computed prior to the heuristic beam search and stored as translation options

Example

- Source phrase: boys|boy|NN|directCase|plural
- Translation: Mapping lemmas
  - boy → लड़का (ladka), युवक (yuvak), etc.
- Translation: Mapping morphology
  - NN|directCase|plural → NN|-e, NN|-o, etc.
- Generation: Generating surface forms
  - लड़का|NN|-e → लड़के (ladke)
  - लड़का|NN|-o → लड़को (ladkon)
  - युवक|NN|-e → युवक (yuvak)
  - युवक|NN|-o → युवको (yuvako)
- Translation options:
  - लड़के|लड़का|NN|-e
  - लड़को |लड़का|NN|-o
  - युवक|युवक|NN|-e
  - युवको|युवक|NN|-o

Efficient decoding (2)

- But we face a problem of combinatorial explosion of the number of translation options
- The problem is currently solved by heavy pruning of expansions
- Number of translation options per input phrase are limited to a maximum number, by default 50
- This is, however, not a perfect solution and results in degradation of translation output

Outline

- Motivation
- What are factored models?
- Decomposition of Factored translation
- Statistical modeling of Factored models
  - Training
  - Combination of components (Log-linear model)
  - Decoding
- **Disadvantages of Factored models**
  - Sparseness
  - High decoding complexity
  - Finding optimal factor settings
- Case-studies

Disadvantages of Factored models: Data sparseness

- Sparseness in translation step:
  - Combination of factors does not exist in the source side training data while translating
- Sparseness in generation step:
  - Combination of target factors does not exist in the training data while generating surface form

Disadvantages of Factored models: Data sparseness

- Sparseness in translation:

	Factored model T: (surface, gender -> surface)	
Training data	Ram|. eats|+musc food|.	राम खाना खाता है raam khana khata hai Ram food eats
Test input	eats|-musc	
Test output	Unknown	

Disadvantages of Factored models: Data sparseness

- Sparseness in generation:

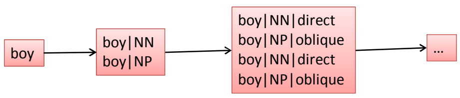
<table>
<tr><th></th><th colspan="2">Factored model
T: (Surface->lemma, Gender->suffix)
G: (lemma, suffix -> surface)</th></tr>
<tr><td>Training data</td><td>Ram|. eats|+musc food|.
Sita|. runs|-musc</td><td>राम|.|. खाना|.|.
खाताहै|खा|ताहै
सीता|.|. दौड़तीहै|दौड़|तीहै</td></tr>
<tr><td>Test input</td><td colspan="2">Sita|. eats|-musc</td></tr>
<tr><td>Test output</td><td colspan="2">सीता|.|. Unknown|खा|तीहै</td></tr>
</table>

Disadvantages of Factored models: Data sparseness

- Solutions:
  - Smoothing for the factor combinations absent in the training data
  - Augmenting training data with all the factor combinations possible

Disadvantages of Factored models: High decoding complexity

- Decoding of factored models may generate huge number of translation options
- The number of translation options increase exponentially with number of factors used



- Results in degraded translation output or it takes large time to translate

Disadvantages of Factored models: High decoding complexity

- Hence, it is not suggested to use many factors while designing a factored model
- Moses decoder allows four factors by default
- Solutions:
  - Heavy pruning of translation options
  - Less number of factors and simple mapping steps

Finding out optimal factor settings

- Huge space of factored model set-ups
- Automatic and Semi-automatic search through the space
- Estimating complexity of factored model

Huge space of factored model setups

- Possible factors on source and target side:
  lemma, POS tag, gender, number, person, tense, case, aspect, etc.
- We can't use all the factors at the same time, due to combinatorial explosion of options
- Even after choosing factors, we need to select appropriate factor mappings for them
- Thus, space of factored model setups is huge for a given language pair

Search through the space

- Finding the correct combination of steps and factors can not be done easily by brute force
- The number of possibilities explodes no matter which direction of exploration we take
- A clever automatic search in the space of configurations does not seem feasible due to
  - low reliability of automatic MT evaluation
  - frequent large variance in scores across different optimization runs

Estimating complexity of factored model

- Estimate the number of partial translation options generated in each step (without actual decoding)
- Use this estimate of complexity to prevent training of unrealistic setups
- Thus, automatic search through the space of factored setups can somewhat be made optimal

Outline

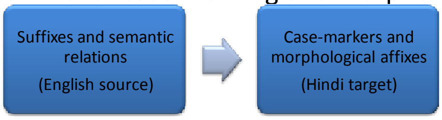
- Motivation
- What are factored models?
- Decomposition of Factored translation
- Statistical modeling of Factored models
- Disadvantages of Factored models
- **Case-studies**
  - English-Hindi (Ramanathan et. al., 2009)
  - English-Czech (Bojar ,2007)

Case-studies

- Ramanathan et. al., Case markers and Morphology: Addressing the crux of the fluency problem in English-Hindi SMT, *Proceedings of ACL/IJCNLP, ACL,* 2009.
- Ondrej Bojar, English-to-Czech Factored Machine Translation, *Proceedings of the Second Workshop on Statistical Machine Translation*, *ACL*, 2007.

Abstract

- English-to-Hindi translation
- English: Moderate case-marking and morphology
- Hindi: Richer case-marking and morphology



Factored model

- Log-linear model:

$$p(e|f) = 1/Z \exp \sum_i \lambda_i h_i(e, f)$$

Translation steps	Generation steps
(English lemma) -> (Hindi lemma) ex. Boy -> लड़क (ladak)	(Hindi lemma + suffix) -> (Hindi surface form) ex. लड़क (ladak) + ए (e) -> लडके (ladake)
(English suffix + semantic relation) -> (Hindi suffix/case-marker) ex. (-s + subj) -> ए (e)	

Motivation of factorization (1)

- Case-markers are decided by semantic relations and tense-aspect information in suffixes

Ex.

John ate an apple.

John|empty|subj eat|ed|empty an|empty|det apple|empty|obj

जॉन ने सेब खाया

John ne seb kahaya

(ed|empty + empty|obj -> ने(ne))

Motivation of factorization (2)

- Target language suffixes are largely determined by source language suffixes and case markers
- And source language case-markers are in turn largely determined by the semantic relations
- So, we need source suffix + semantic relations

Ex. The boys ate apples.

The|empt|det boy|s|subj eat|ed|empty apple|s|obj

लड़कों ने सेब खाये

ladakon ne seb khaye

Motivation of factorization (3)

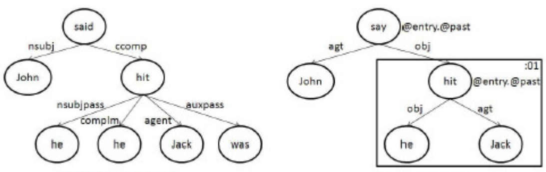
- The separation of the lemma and suffix helps in tiding over the data sparseness problem
- Allows suffix-case marker combination rather than the combination of the specific word and the case marker

Semantic relations

- Two different semantic relations used:
  - UNL (Universal Networking Language) relations
    - 44 binary relations
    - Ex. agent, object, co-agent, and partner, temporal relations, locative relations, conjunctive and disjunctive relations, comparative relations, etc.
  - Stanford parser grammatical relations
    - 55 binary relations
    - Ex. subject, object, objects of prepositions, and clausal complements, modifier relations like adjectival, adverbial, participial, and infinitival modifiers

UNL and Stanford relations differences

John said that he was hit by Jack.



Stanford Semantic graph

UNL Semantic graph

Experiments

- Corpus size:

	# Sentences	# Words
Training	12868	316508
Tuning	600	15279
Testing	400	8557

- Language model: SRILM
- Training, tuning and decoding: Moses toolkit
- Other tools: Stanford parser, morpha

Results

- BLEU and NIST evaluation:

MODEL	BLEU	NIST
Baseline (Surface)	24.32	5.85
lemma + suffix	25.16	5.87
lemma + suffix + unl	27.79	6.05
lemma + suffix + stanford	28.21	5.99

Note: All models had been preprocessed with source-side reordering

Discussions

- Better fluency and adequacy are achieved with the use of semantic relations
- The use of semantic relations, in combination with syntactic reordering, produces sentences that are reasonably fluent and convey most or all of the meaning

Outline

- Motivation
- What are factored models?
- Decomposition of Factored translation
- Statistical modeling of Factored models
- Disadvantages of Factored models
- **Case-studies**
  - English-Hindi (Ramanathan et. al., 2009)
  - English-Czech (Bojar ,2007)

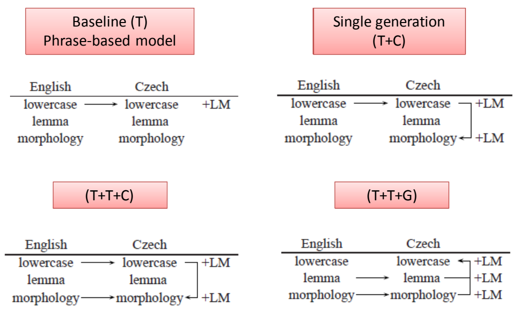
Abstract

- English-to-Czech translation
- Czech is a Slavic language with very rich morphology and relatively free word order
- Additional annotation of input and output tokens (multiple factors) is used to explicitly model morphology

Experimental setup

- Data:
  - News commentary (NC) corpus
  - Train: 55, 676 sentence pairs
  - Tune: 1,023 sentence pairs
  - Test: 964 sentence pairs
- Factor generation:
  - English:
    - Tags: MXPOST (Ratnaparkhi, 1996)
    - Lemma: Morpha tool
  - Czech:
    - Tags and lemma: Tool by Hajic and Hladka (1998)

Scenarios



Results

- BLEU evaluation:

Model	BLEU
T+T+G	13.9±0.7
T+T+C	13.9±0.6
T+C	13.6±0.6
Baseline: T	12.9±0.6

References: Factored SMT

- Philipp Koehn and Hieu Hoang. Factored translation models. *Proceedings of the Joint Conference on Empirical Methods in Natural language Processing and Computational Natural Language Learning, ACL*, pages 868–876, 2007.
- Ananthakrishnan Ramanathan, Hansraj Choudhary, Avishek Ghosh, and Pushpak Bhattacharyya. Case markers and morphology: Addressing the crux of the fluency problem in English-Hindi SMT. *Proceedings of ACL/IJCNLP, ACL*, 2:800–808, 2009.
- Ale Tamchyna and Ondej Bojar. No free lunch in factored phrase-based machine translation. In *Computational Linguistics and Intelligent Text Processing*, volume 7817, pages 210–223. Springer Berlin Heidelberg, 2013.
- Ondrej Bojar, English-to-Czech Factored Machine Translation, *Proceedings of the Second Workshop on Statistical Machine Translation*, ACL, 2007.

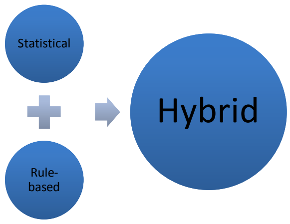
HYBRID MACHINE TRANSLATION

Outline

- **What is Hybrid machine translation?**
- Types of Hybrid machine translation
- Case studies

Hybrid Machine Translation: Get the best of both worlds

- Hybrid machine translation combines the strengths of both statistical and rule-based translation systems



Outline

- What is Hybrid Machine translation?
- **Types of Hybrid machine translation**
- Case studies

Rule-based vs. Statistical translation

- Rule-based machine translation:
  - Involves more information about the linguistics of the source and target languages
  - Uses the morphological and syntactic rules and semantic analysis of both languages
- Statistical machine translation:
  - Generates translations using statistical methods based on bilingual text corpora
  - No need of any linguistic information

Rule-based vs. Statistical translation

Rule-based translation system	Statistical translation system
Consistent and predictable quality	Unpredictable translation quality
Good out-of-domain translation	Poor out-of-domain translation
Knows grammatical rules	Does not know grammar
Lack of fluency	Good fluency
Hard to handle exceptions to rules	Good for catching exceptions to rules
Human efforts in developing rules	No human efforts needed

Types of Hybrid translation

- Rules post-processed by statistics:
  - Translations are performed using a rules based engine
  - Statistics are then used in an attempt to adjust/correct the output from the rules engine
- Statistics guided by rules:
  - Rules are used to pre-process data in an attempt to better guide the statistical engine
  - Rules are also used to post-process the statistical output to perform functions such as normalization
  - This approach has a lot more power, flexibility and control when translating

Outline

- What is Hybrid Machine translation?
- Types of Hybrid machine translation
- **Case studies**
  - Source-side reordering (Ramanathan et. al., 2008)
  - Clause-based reordering constraints
  - Rule-based translation with statistical post-editing

Reordering model

- Phrase-based models do not handle syntax in a natural way
- Reordering of phrases during translation is managed by distortion models
- Distortion models are not helpful enough to handle SVO-SOV reordering phenomenon
- Many preprocessing approaches have been suggested to overcome this problem
- One of them is: To reorder the English sentence so as to match the word order of the Indian language sentence

Source-side reordering

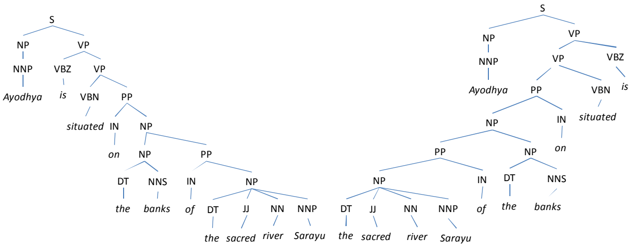
- Executes before SMT training or decoding
- Needs a constituency parse tree on the source side
- Approach is similar to the syntax-based model's reordering step

$$S\ S_m\ V\ V_m\ O\ O_m\ C_m \rightarrow C'_m\ S'_m\ S'\ O'_m\ O'\ V'_m\ V'$$

S: Subject   O: Object   V: Verb
C_m: Clause modifier
X': Corresponding constituent in Hindi (X=S, O, V)
X_m: Modifier of X

Example: Source-side reordering

Ayodhya is situated on the banks of the sacred river Sarayu .



Ayodhya the sacred river Sarayu of the banks on situated is .
अयोध्या पवित्र नदी सरयू के किनारे पर बसी है .

- Rules for reordering are found out manually

Experiments

- Data:

	# sentences	# words
Training	5000	120,153
Tuning	483	11,675
Test	400	8557
Monolingual (Hindi)	49,937	1,123,966

- Baseline system: Phrase-based model

Evaluation metric

- *BLEU(BiLingual Evaluation Understudy):*
  measures the precision of n-grams with respect to the reference translations, with a brevity penalty
- *mWER (multi-reference word error rate) :*
  measures the edit distance with the most similar reference translation
- *SSER(subjective sentence error rate):*

Score	Basis
0	Nonsense
1	Roughly understandable
2	Understandable
3	Good
4	Perfect

Results

Technique	Evaluation metric				
	BLEU	mWER	SSER	Roughly understandable+	Understandable+
Baseline	12.10	77.49	91.20	10%	0%
Baseline + Source reordering	16.90	69.18	74.40	42%	12%

* Ramanathan et. al.. Simple Syntactic and Morphological Processing Can Help English-Hindi Statistical Machine Translation, IJCNLP, 2008

Outline

- What is Hybrid Machine translation?
- Types of Hybrid machine translation
- **Case studies**
  - Source-side reordering
  - Clause-based reordering constraints (Ramanathan et. al., 2011)
  - Rule-based translation with statistical post-editing

Clause-based reordering constraints

- Problem statement:

Sentences translated: 225

Sentences having more than one clause: 120

Sentences having inter-clause reordering problem: 45

(Some words or phrases are wrongly placed where they do not belong)

Translation of finite and non-finite clauses

- Finite clauses:
  - Tensed clauses
  - Appear most commonly in conjunct or relative constructions
  - Each finite clause can be translated separately and glued together

Translation of finite and non-finite clauses

- Non-finite clauses:
  - Untensed clauses
  - Translation depends on the role in the sentence
  - Issues:
    - All or part of the non-finite clause could get reordered with the surrounding clause, or
    - The overall meaning is conveyed by a phrase or group of words from the non-finite clause and a surrounding or neighboring clause
  - Simply translating non-finite clauses separately with reordering constraints around them, will not lead to good translation

Experiments

- Baseline: DTM2 (a direct translation model)
- Word-alignments: HMM aligner
- The reordering restriction is applied by treating the relevant clause-boundaries as barriers
- Determining clause boundaries:
  1. Manually
  2. Using constituency parser
  3. Using a CRF-based clause-boundary classifier using parts-of-speech and parser features

Data and Evaluation

- Data:
  - Training: 289k sentences
  - Testing: 844 sentences
  - Language model: 1.5 million sentences
- Evaluation:
  - Automatic: BLEU score with single reference
  - Subjective: 5-point scale on 100 random sentences

Results

- Automatic evaluation:

	BLEU	Adequacy	Fluency
baseline	19.4	2.04	2.41
finite	20.4^{δ}	2.32^{δ}	2.67^{δ}
non-finite	19.6	2.17^{ψ}	2.5
finite + non-finite	19.8^{ψ}	2.17	2.51^{ψ}

Manually identified clauses. δ: 99% statistical significance; ψ: 95% statistical significance

Method	ACI accuracy	BLEU	Adequacy	Fluency
parser	0.42	19.3	-	-
CRF – word and pos	0.69	19.8^{ψ}	2.27^{δ}	2.59^{δ}

* Ramanathan et. al.. Clause-Based Reordering Constraints to Improve Statistical Machine Translation, IJCNLP, 2011

Results

- Subjective evaluation:

	improved	degraded
finite (manual)	36	8
finite (auto)	35	17
non-finite (manual)	17	10
finite + non-finite (manual)	19	11

Effect of clause-based reordering constraints

- Input:

  America claims that Iran wants to continue its nuclear program, and secretly builds atomic weapons.

- Baseline translation:

  अमेरिका का दावा है कि उसके परमाणु कार्यक्रम रहना चाहते हैं और ईरान परमाणु हथियार निर्माण करता है

  amerika kaa daavaa hai ki usake paramaanu kaaryakrama rahanaa caahate hain aur iraana paramaanu hathiyaara nirmaana karataa hai

- Clause-based translation:

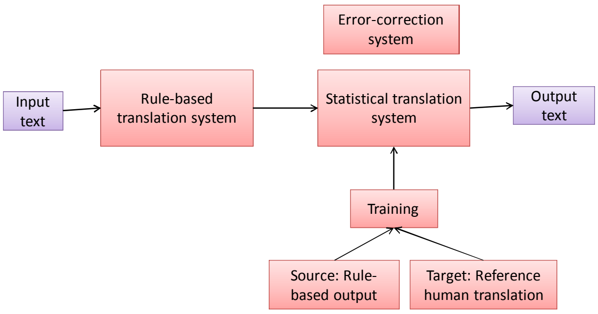
  अमेरिका का दावा है कि ईरान अपने परमाणु कार्यक्रम को जारी रखना चाहता है और परमाणु हथियार निर्माण करता है

  amerika kaa daavaa hai ki iran apane paramaanu kaaryakrama ko jaarii rakhanaa caahataa hai aura paramaanu hathiyaara nirmaana kartaa hai

Outline

- What is Hybrid Machine translation?
- Types of Hybrid machine translation
- **Case studies**
  - Source-side reordering
  - Clause-based reordering constraints
  - Rule-based translation with statistical post-editing (Simard et. al., 2007)

Overview of the system



System

- Rule-based system:
  - Initial source-to-target language translation done by SYSTRAN rule-based translation system (version 6)
- Statistical post-editing system:
  - Based on PORTAGE statistical phrase-based translation system (developed by NRC Canada)
  - Training data:
    - Source: Translation output of rule-based system on source text
    - Target: target text
    - English-French Europarl and News commenatry domain

Results

- BLEU score:

	en → fr	fr → en
Europarl (>32M words/language)		
SYSTRAN	23.06	20.11
PORTAGE	31.01	30.90
SYSTRAN+PORTAGE	31.11	30.61
News Commentary (1M words/language)		
SYSTRAN	24.41	18.09
PORTAGE	25.98	25.17
SYSTRAN+PORTAGE	28.80	26.79

Discussions

- Hybrid approach reduces post-editing efforts compared to simple rule-based system
- SYSTRAN+PORTAGE improves BLEU score significantly when compared to simple SYSTRAN rule-based system
- SYSTRAN+PORTAGE outperforms PORTAGE system in case of News commentary domain and performs at level in case of Europarl corpus

Summary

- Factored models are generic phrase-based models which make use of linguistic information
- Factored models can be used while translating from morphologically poor languages to morphologically richer languages
- Factored models face the problem of data sparseness, high decoding complexity and finding out optimal factored setup
- Case-studies over different language pairs show improvement after using factored model

- Source-side reordering improves translation fluency on large scale
- Clause-based reordering constraints on finite clauses improve translation quality
- Rule-based system can be augmented with a statistical error-correction system to improve the output quality and reduce post-editing efforts

Translation direction and Challenges

Challenges → Direction ↓	Reordering	Morphological inflections
English-to-Indian languages	Source-side reordering/ Clause-based constraints	Factored models
Indian languages-to-English	Source-side reordering/ Clause-based constraints	No explicit need

References: Hybrid SMT

- Ramanathan et. al.. Simple Syntactic and Morphological Processing Can Help English-Hindi Statistical Machine Translation. In *Proceedings of IJCNLP, 2008.*
- Ramanathan et. al.. Clause-Based Reordering Constraints to Improve Statistical Machine Translation. In *Proceedings of IJCNLP, 2011.*
- Michel Simard, Nicola Ueffing, Pierre Isabelle and Roland Kuhn. Rule-based Translation With Statistical Phrase-based Post-editing. *Proceedings of the Second Workshop on Statistical Machine Translation, ACL*, pages 203–206, 2007.
- http://en.wikipedia.org/wiki/Machine_translation

SYNTAX BASED SMT

Outline

- Motivation
- Different flavours of Tree based SMT models
- Synchronous Context Free Grammars
- Hierarchical Phrase Based Model

Problems with Phrase Based models

- Heavy reliance on lexicalization
  - Direct Translation method
  - No generalization
  - Lot of data is required

For similar sentences, sometimes reordering occurs, sometimes it does not

Correct reordering
Oracle bought Sun Microsystems in 2010
ओरेकल 2010 में सन माइक्रोसिस्टम्स को खरीदा

Incorrect Reordering
IBM approached Sun Microsystems in 2008
आईबीएम दरवाजा खटखटाया 2008 में सन माइक्रोसिस्टम्स का

Problems with Phrase Based models (2)

- Learning is very local in nature
  - Local reordering, sense disambiguation learnt
  - Phenomena like word order divergence, recursive structure are non-local

Word order divergence (SVO-SOV) is not learnt

[The USA] [is not engaging] [in war] [with Iran]
[अमरीका] [संलग्न नहीं है] [युद्ध में] [ईरान के साथ]

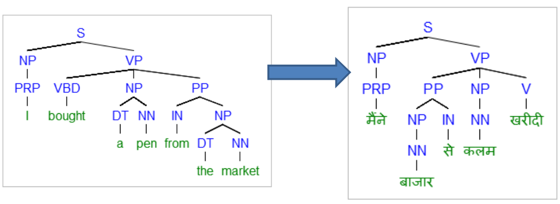
Recursive structure: phrase boundaries are not maintained

[[It is necessary [that the person [who is travelling for the conference]] should get approval prior to his departure]]
यह सम्मेलन के लिए यात्रा कर रहा है, जो व्यक्ति पहले अपने प्रस्थान से अनुमोदन प्राप्त करना चाहिए कि आवश्यक है

Tree based models

- Source and/or Target sentences are represented as trees
- Translation as Tree-to-Tree Transduction
  - As opposed to string-to-string transduction in PB-SMT
- Parsing as Decoding
  - Parsing of the source language sentence produces the target language sentences

Example



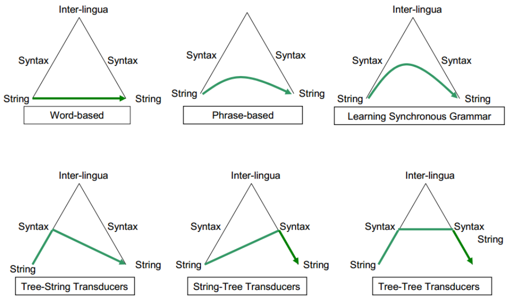
Source Tree

Target Tree

Why tree based model?

- Natural language sentences have a tree-like structure
- Syntax based Reordering
- **Source side tree**: guides decoding by constraining the possible rules that can be applied
- **Target side tree** ensures grammatically correct output

Different flavours of tree-based models



Slide from Amr Ahmed

Synchronous Context Free Grammar

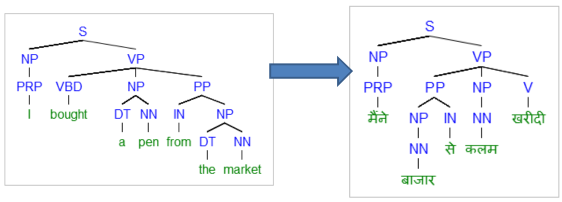
- Fundamental formal tool for Tree-based translation models
- An enhanced Context Free Grammar for generating two related strings instead of one
- Alternatively, SCFG defines a tree transducer

Definition

CFG	SCFG
S → NP VP	S → < NP_1 VP_2 , NP_1 VP_2 >
VP → V	VP → < V_1 , V_1 >
VP → V NP	VP → < V_1 NP_2 , NP_2 V_1 >
VP → VP NP PP	VP → < V_1 NP_2 PP_3 , PP_3 NP_2 V_1 >
NP → NN	NP → < NN_1 , NN_1 >
NN → market	NN → < market , बाजार >

- **Differences of SCFG from CFG:**
  - 2 components on the RHS of production rule
  - Same number of non-terminals
  - Non-terminals have one-one correspondence (index-linked)

Example



Source Tree

Target Tree

Example SCFG for English-Hindi

1. S → < NP_1 VP_2 , NP_1 VP_2 >
2. VP → < V_1 , V_1 >
3. VP → < V_1 NP_2, NP_2 V_1 >
4. VP → < V_1 NP_2 PP_3 , PP_3 NP_2 V_1 >
5. NP → < NN_1 , NN_1 >
6. NP → < PRP_1 , PRP_1 >
7. PP → < IN_1 NP_2 , NP_2 IN_1 >
8. NN → < market , बाजार >
9. NN → < pen , कलम >
10. PRP → < I , मैंने >
11. V → < bought, खरीदी >
12. IN → < from , से >
13. DT → < the , ε >
14. DT → < a ,ε >

Derivation

Parsing as Decoding!

- S
- < NP_1 VP_2 , NP_1 VP_2 >
- < NP_1 VP_2 , NP_1 VP_2 >
- < PRP_3 VP_2 , PRP_3 VP_2 >
- < I VP_2 , मैंने VP_2 >
- < I V_3 NP_4 PP_5 , मैंने PP_5 NP_4 V_3 >
- < I bought NP_4 PP_5 , मैंने PP_5 NP_4 खरीदी>
- < I bought DT_6 NN_7 PP_5 , मैंने PP_5 DT_6 NN_7 खरीदी>
- < I bought a NN_7 PP_5 , मैंने PP_5 NN_7 खरीदी>
- < I bought a pen PP_5 , मैंने PP_5 कलम खरीदी>
- < I bought a pen IN_8 NP_9 , मैंने NP_9 IN_8 कलम खरीदी>
- < I bought a pen from NP_9 , मैंने NP_9 से कलम खरीदी>
- < I bought a pen from DT_{10} NN_{11} , मैंने DT_{10} NN_{11} से कलम खरीदी>
- < I bought a pen from the NN_{11} , मैंने NN_{11} से कलम खरीदी>
- < I bought a pen from the market, मैंने बाजार से कलम खरीदी>

Reordering and Relabeling among Child Nodes

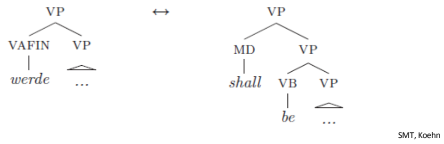
- The only operations a SCFG allows is:
  - reordering among child nodes

    $VP \rightarrow \langle V_1\ NP_2\ PP_3\ ,\ PP_3\ NP_2\ V_1 \rangle$
  - Re-labelling of nodes

    $VP \rightarrow \langle V_1\ NP_2\ PP_3\ ,\ PREPP_3\ NP_2\ V_1 \rangle$

    $PP/PREPP \rightarrow \langle IN_1\ NP_2\ ,\ NP_2\ IN_2 \rangle$
- The condition is overly restrictive, hardly any pair of languages would follow such a grammar
- Useful for representing non-linguistic formalisms like hierarchical model, Inverse Transduction Grammar

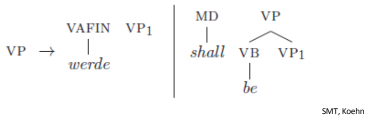
No raising or lowering of nodes



- *'werde'* in German maps to *'shall be'* in complex ways
- Cannot be captured by SCFG
- Child node reodering restriction

Synchronous Tree Substitution Grammar

- Restriction can be overcome by S-TSG
- Synchronous extension of Tree Substitution Grammar
- RHS components can be tree fragments instead of string on non-terminals



Chomsky Normal Form

- Rank of CFG/SCFG: maximum number of non-terminals on RHS
- Any CFG can be converted to weakly equivalent rank-2 CFG (Chomsky Normal Form)
- SCFG of rank-3 can be converted to CNF
- However, in general, CNG may not exist for SCFG
- Has implications for efficient parsing

Hierarchical Phrase Based Models

- Learns a SCFG purely from data
  - no source, target side parsers used
- Learns an undifferentiated grammar
  - Grammar does not have notion of different types of non-terminals (eg. NP, VP, etc.)
  - Only one type of non-terminal, called X
- Production rules are of the form

  $X \rightarrow \langle \alpha\ X_1\ \beta\ X_2\ \gamma\ X_3\ ,\ X_2\ \alpha'\ \beta'\ X_3\ X_1 \rangle$
- Useful in generalizing learning of reordering among phrases

Formal, Not Linguistic

- "Formal", but not linguistic
  - The SCFG grammar learnt would not correspond to the notion of a language
  - only one non-terminal
  - "non-linguistic" phrases (not words) as basic units
- Built on top of phrase based model
  - Leverages the strengths of PBSMT
  - PBSMT performs best when not restricted to just linguistic phrases
- The HPBSMT model defines a formal SCFG model for reordering of these "phrases"
- A custom designed engineering solution for a purpose

The SCFG for the Hierarchical Model

- A rule is of the form:

$$X \rightarrow \langle \gamma, \alpha, \sim \rangle$$

where, ~ is one-one correspondence between non-terminals

X → < with X_1 , X_1 के साथ >

- In addition, there are "glue" rules for the initial state

$$S \rightarrow \langle S_{\boxed{1}} X_{\boxed{2}}, S_{\boxed{1}} X_{\boxed{2}} \rangle$$
$$S \rightarrow \langle X_{\boxed{1}}, X_{\boxed{1}} \rangle$$

The Probabilistic Model

- The translation model is a log-linear model
- The weight of each rule is given by:

$$w(X \rightarrow \langle \gamma, \alpha \rangle) = \prod_i \phi_i(X \rightarrow \langle \gamma, \alpha \rangle)^{\lambda_i}$$

where ϕ_i – feature function
λ_i – feature weight

- Features used: analogous to PB-SMT
  - Rule probability, inverse rule probability, lexical weights, phrase penalty
- For the glue rules:

$$S \rightarrow \langle S_{\boxed{1}} X_{\boxed{2}}, S_{\boxed{1}} X_{\boxed{2}} \rangle \qquad 1$$
$$S \rightarrow \langle X_{\boxed{1}}, X_{\boxed{1}} \rangle \qquad \exp(-\lambda_g)$$

The Probabilistic Model (2)

- The weight of a derivation is

$$w(D) = \prod_{\langle r,i,j \rangle \in D} w(r) \times p_{lm}(e)^{\lambda_{lm}} \times \exp(-\lambda_{wp}|e|)$$

  - Derivation weight is a combination of product of rule weights, language model score and word penalty
- Decision Rule: Choose the target sentence for which derivation score is maximum

$$e^* = \arg\max w(D(f, e))$$

Learning Grammar Rules

- Rules are learnt from phrase alignments provided by phrase based model
- The phrases in the phrase table are called "initial phrase pairs"

1. If $\langle f_i^j, e_{i'}^{j'} \rangle$ is an initial phrase pair, then

$$\mathrm{X} \rightarrow \langle f_i^j, e_{i'}^{j'} \rangle$$

is a rule.

2. If $r = \mathrm{X} \rightarrow \langle \gamma, \alpha \rangle$ is a rule and $\langle f_i^j, e_{i'}^{j'} \rangle$ is an initial phrase pair such that $\gamma = \gamma_1 f_i^j \gamma_2$ and $\alpha = \alpha_1 e_{i'}^{j'} \alpha_2$, then

$$\mathrm{X} \rightarrow \langle \gamma_1 \mathrm{X}_{\boxed{k}} \gamma_2, \alpha_1 \mathrm{X}_{\boxed{k}} \alpha_2 \rangle$$

is a rule, where k is an index not used in r.

Example of rule generation

	Prof	C.N.R.	Rao	was	honoured	with	the	Bharat	Ratna
प्रोफेसर	■								
सी.एन.आर		■	■						
राव		■	■						
को							■	■	■
भारतरत्न							■	■	■
से						■			
सम्मानित					■				
किया				■					
गया				■					

Extracted Phrase alignments

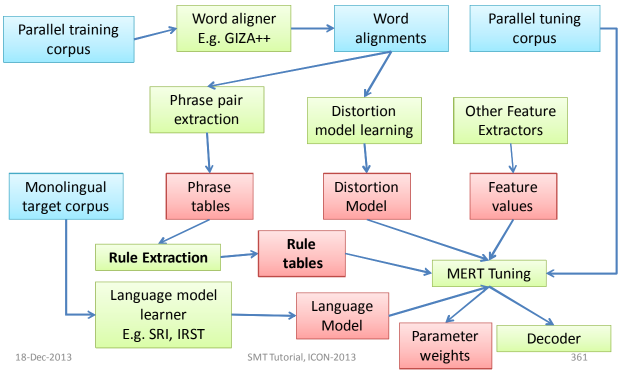
Extracted Rules

Phrase Pair	Extracted Rule
(was honoured, सम्मानित किया गया)	X → < was X_1 , X_1 किया गया >
(with the Bharat Ratna, भारतरत्न से)	X → < with X_1 , X_1 से >
(was honoured with the Bharat Ratna, भारतरत्न से सम्मानित किया गया)	X → < was X_1 with X_2, X_2 से X_1 किया गया >

Restrictions on rule generation

- Problems with rule generation
  - Given a sentence pair with *n* phrase pairs, and allowing *k* non-terminals in a rule, a single sentence can generate C(n,k) rules
  - Spurious Ambiguity: Multiple rules leading to same derivation
- Hence, necessary to constrain the rules that can be created
- Maximum of two non-terminals per rule
- Length of initial phrases limited to 10
- Length of RHS of rules limited to 5
- Length of RHS of rules should be greater than 2 (remove unit productions)

Overall Training Process for Hierarchical-PB-SMT



Some Results

- Chinese to English Translation
- 24 M rules generated, filtered to 2.2 M from the development set

System	BLEU
Phrase based	0.2676
Hierarchical	0.2877

Summary

- Tree based models can better handle syntactic phenomena like reordering, recursion
- Basic formalism: Synchronous Context Free Grammar
- Decoding: Parsing on the source side
  - CYK Parsing
  - Integration of the language model presents challenge
- Parsers required for learning syntax transfer
- Without parsers, some weak learning is possible with hierarchical PBSMT
- Lot of active research: dependency based models, TAG/TSG based models, faster decoding, etc.

References: Syntax Based SMT

- David Chiang. *An Introduction to Synchronous Grammars* <http://www.isi.edu/~chiang/papers/synchtut.pdf>. 2006.
- Philip Koehn. *Tree-based models.* In Statistical Machine Translation. Cambridge University Press. 2010.
- Chiang, David. *A hierarchical phrase-based model for statistical machine translation*. ACL. 2005.
- Chiang, David. *Hierarchical phrase-based Translation*. *Computational Linguistics*. 2007.
- Yamada, Kenji, and Kevin Knight. *A syntax-based statistical translation model*. ACL. 2001.
- Christopher Manning and Hierich Schutze. *Probabilistic Parsing*. In Foundations of Statistical Natural Language Processing. 1999.

MACHINE TRANSLATION EVALUATION

Acknowledgments: Aditya Joshi (Ph. D student), Kashyap Popat (M.Tech student), CSE, IITB

Introduction and formulation of BLEU

Motivation

How do we judge a good translation?
Can a machine do this?
Why should a machine do this?
Because humans take time!

Outline

- Evaluation
- Formulating BLEU Metric
- Understanding BLEU formula
- Shortcomings of BLEU
- Comparison with other metrics

R. Ananthakrishnan, Pushpak Bhattacharyya, M. Sasikumar and Ritesh M. Shah, *Some Issues in Automatic Evaluation of English-Hindi MT: More Blues for BLEU*, **ICON 2007**, Hyderabad, India, Jan, 2007.

Evaluation

- Assign scores to specific qualities of output
  - Intelligibility: How good the output is as a well-formed target language entity
  - Accuracy: How good the output is in terms of preserving content of the source text

For example, I am attending a lecture

मैं एक व्याख्यान बैठा हूँ
Main ek vyaakhyan baitha hoon
I a lecture sit (Present-first person)
I sit a lecture : Accurate but not intelligible
मैं व्याख्यान हूँ
Main vyakhyan hoon
I lecture am
I am lecture: Intelligible but not accurate.

Evaluation in MT [1]

- Operational evaluation
  - "Is MT system A operationally better than MT system B? Does MT system A cost less?"
- Typological evaluation
  - "Have you ensured which linguistic phenomena the MT system covers?"
- Declarative evaluation
  - "How does quality of output of system A fare with respect to that of B?"

Evaluation bottleneck

- Typological evaluation is time-consuming
- Operational evaluation needs accurate modeling of cost-benefit

- Automatic MT evaluation: Declarative

  BLEU: Bilingual **E**valuation **U**nderstudy

Deriving BLEU [2]

Incorporating Precision
Incorporating Recall

How is translation performance measured?

The closer a machine translation is to a professional human translation, the better it is.

- A corpus of good quality human reference translations
- A numerical “translation closeness” metric

Preliminaries

- **Candidate Translation(s):** Translation returned by an MT system
- **Reference Translation(s):** ‘Perfect’ translation by humans

Goal of BLEU: To correlate with human judgment …. To evaluate translation quality

Formulating BLEU (Step 1): Precision

I had lunch now.

Reference 1: मैने अभी खाना खाया
maine abhi khana khaya
I now food ate
I ate food now.

Reference 2 : मैने अभी भोजन किया
maine abhi bhojan kiyaa
I now meal did
I did meal now

Candidate 1: **मैने** अब **खाना खाया**
maine ab khana khaya
I now food ate
I ate food now

matching unigrams: 3,
matching bigrams: 1

Candidate 2: **मैने अभी** लंच एट
maine abhi lunch ate.
I now lunch ate
I ate lunch(OOV) now(OOV)

matching unigrams: 2,
matching bigrams: 1

Unigram precision: Candidate 1: 3/4 = 0.75, Candidate 2: 2/4 = 0.5
Similarly, bigram precision: Candidate 1: 0.33, Candidate 2 = 0.33

Precision: Not good enough

Reference: मुझ पर तेरा सुरूर छाया
mujh-par tera suroor chhaaya
me-on your spell cast
Your spell was cast on me

Candidate 1: मेरे **तेरा सुरूर छाया** matching unigram: 3
mere tera suroor chhaaya
my your spell cast
Your spell cast my

Candidate 2: **तेरा तेरा तेरा सुरूर** matching unigrams: 4
tera tera tera suroor
your your your spell

Unigram precision: Candidate 1: 3/4 = 0.75, Candidate 2: 4/4 = 1

Formulating BLEU (Step 2): Modified Precision

- Clip the total count of each candidate word with its maximum reference count
- $Count_{clip}$(n-gram) = min (count, max_ref_count)

Reference: मुझ पर तेरा सुरूर छाया
mujh-par tera suroor chhaaya
me-on your spell cast
Your spell was cast on me

Candidate 2: **तेरा तेरा तेरा सुरूर**
tera tera tera suroor
your your your spell

- matching unigrams:
  (**तेरा** : min(3, 1) = 1) (सुरूर : min (1, 1) = 1)

Modified unigram precision: 2/4 = 0.5

Modified n-gram precision

For entire test corpus, for a given n,

n-gram: Matching n-grams in C

$$\boxed{p_n} = \frac{\boxed{\sum_{C \in \{Candidates\}}} \boxed{\sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}}{\boxed{\sum_{C' \in \{Candidates\}}} \boxed{\sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')}}$$

Modified precision for n-grams

Overall candidates of test corpus

n-gram’: All n-grams in C

Formula from [2]

Recall for MT (1/2)

- Candidates shorter than references
- Reference: क्या ब्लू लंबे वाक्य की गुणवत्ता को समझ पाएगा?

  kya blue lambe vaakya ki guNvatta ko samajh paaega?

  Will blue long sentence-of quality (case-marker) understandable(III-person-male-singular)?

  Will blue be able to understand quality of long sentence?

Candidate: लंबे वाक्य

lambe vaakya

long sentence

long sentence

modified unigram precision: 2/2 = 1

modified bigram precision: 1/1 = 1

Recall for MT (2/2)

- Candidates longer than references

Reference 1: मैने भोजन किया

maine bhojan kiyaa

I meal did

I had meal

Reference 2: मैने खाना खाया

maine khaana khaaya

I food ate

I ate food

Candidate 1: मैने खाना भोजन किया

maine khaana bhojan kiya

I food meal did

I had food meal

Modified unigram precision: 1

Candidate 2: मैने खाना खाया

maine khaana khaaya

I food ate

I ate food

Modified unigram precision: 1

Formulating BLEU (Step 3): Incorporating recall

- Sentence length indicates 'best match'
- Brevity penalty (BP):
  - Multiplicative factor
  - Candidate translations that match reference translations in length must be ranked higher

Candidate 1: लंबे वाक्य

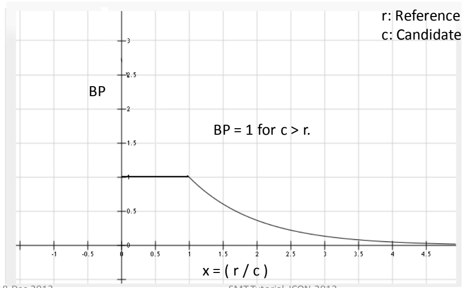
Candidate 2: क्या ब्लू लंबे वाक्य की गुणवत्ता समझ पाएगा?

Formulating BLEU (Step 3): Brevity Penalty

$$\mathrm{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

r: Reference sentence length

c: Candidate sentence length



Formula from [2]

Graph drawn using www.fooplot.com

BP leaves out longer translations

Why?

Translations longer than reference are already penalized by modified precision

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')}$$

Formula from [2]

BLEU score

Recall -> Brevity Penalty

$$\mathrm{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

Precision -> Modified n-gram precision

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')}$$

$$\mathrm{BLEU} = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

Formula from [2]

Understanding BLEU

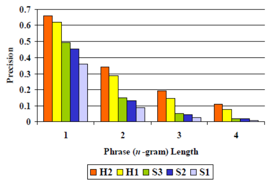
Dissecting the formula

Decay in precision

Why log p_n?

To accommodate decay in precision values

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$



Formula from [2]
Graph from [2]

Dissecting the Formula

Claim: BLEU should lie between 0 and 1

Reason: To intuitively satisfy “1 implies perfect translation”

Understanding constituents of the formula to validate the claim

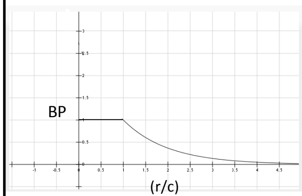
$$\text{BLEU} = \underbrace{\text{BP}}_{\text{Brevity Penalty}} \cdot \exp\left(\sum_{n=1}^{N} \underbrace{w_n}_{\text{Set to 1/N}} \log \underbrace{p_n}_{\text{Modified precision}}\right)$$

Formula from [2]

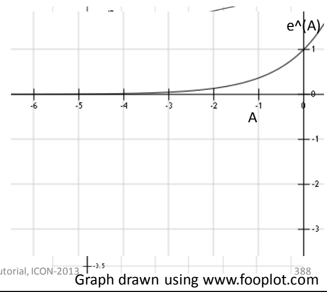
Validation of range of BLEU

$$\text{BLEU} = \text{BP} \cdot \exp\overbrace{\boxed{\left(\sum_{n=1}^{N} w_n \log p_n\right)}}^{A}$$

p_n : Between 0 and 1
log p_n : Between –(infinity) and 0
A: Between –(infinity) and 0
e ^ (A): Between 0 and 1



BP: Between 0 and 1



Graph drawn using www.fooplot.com

BLEU v/s human judgement [2]

Target language: English

Source language: Chinese

Setup

Five systems perform translation:

- 3 automatic MT systems
- 2 human translators

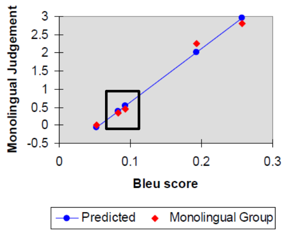
BLEU scores obtained for each system

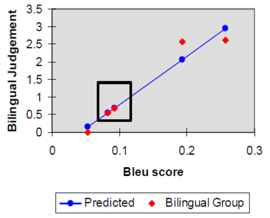
Human judgment (on scale of 5) obtained for each system:

- Group 1: Ten Monolingual speakers of target language (English)
- Group 2: Ten Bilingual speakers of Chinese and English

BLEU v/s human judgment

- Monolingual speakers:
Correlation co-efficient: 0.99
- Bilingual speakers:
Correlation co-efficient: 0.96



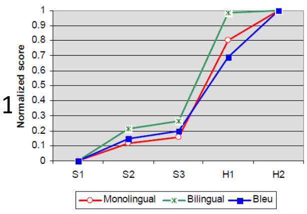


BLEU and human evaluation for S2 and S3

Graph from [2]

Comparison of normalized values

- High correlation between monolingual group and BLEU score
- Bilingual group were lenient on ‘fluency’ for H1
- Demarcation between {S1-S3} and {H1-H2} is captured by BLEU



Graph from [2]

Shortcomings of BLEU

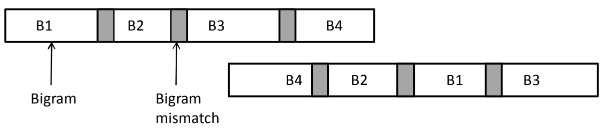
Admits too much variation

- BLEU relies on n-gram matching only
- Puts very few constraints on how n-gram matches can be drawn from multiple reference translations

$$\text{BLEU} = \underbrace{\text{BP}}_{\text{Brevity Penalty (Incorporating recall)}} \cdot \exp\left(\sum_{n=1}^{N} \underbrace{w_n}_{\text{Set to } 1/N} \log \underbrace{p_n}_{\text{Modified precision}}\right)$$

Permuting phrases [3]

- Reordering of unmatched phrases does not affect precision
- Bigram mismatch sites can be freely permuted



Possible to randomly produce other hypothesis translations that have the same BLEU score

Issues with precision (1/2)

The king and the queen went to the jungle to hunt.

Reference 1:
राजा और रानी जंगल को शिकार के लिए गये
raaja aur raani jangal ko shikaar ke liye gaye
King and queen to-jungle for-hunting went

Reference 2:
राजा और उनकी बीवी शिकार करने जंगल गये
raaja aur unki biwi shikaar karne jangal gaye
king and his wife to-do-hunting jungle went

Candidate: राजा और रानी शिकार करने जंगल में गये
raaja aur raani shikaar karne jungal mein chale gaye
King and queen to-do-hunting to-jungle went

Matching bi-grams = 4 / 8

Candidate: राजा और रानी शिकार करने जंगल गये में
raaja aur raani shikaar karne gaye jungle mein
King and queen to-do-hunting went jungle to (grammatically incorrect)

Matching bi-grams = 4 / 8

Issues with precision (2/2)

The king and the queen went to the jungle to hunt.

Reference 1:
राजा और रानी जंगल को शिकार के लिए गये
raaja aur raani jangal ko shikaar ke liye gaye
King and queen to-jungle for-hunting went

Reference 2:
राजा और उनकी बीवी शिकार करने जंगल गये
raaja aur unki biwi shikaar karne jangal gaye
king and his wife to-do-hunting jungle went

Candidate: राजा और रानी शिकार करने जंगल में गये
raaja aur raani shikaar karne jungal mein chale gaye
King and queen to-do-hunting to-jungle went

Matching bi-grams = 4 / 8

Candidate: शिकार करने जंगल राजा और रानी में गये
shikaar karne jungle raaja aur raani mein gaye
to-do hunting jungle raja and rani in went (grammatically incorrect)

Matching bi-grams = 4 / 8

Permuting phrases, in general

- For *'b'* bi-gram matches in a candidate translation of length *'k'*,
  (k – b)! possible ways to generate similarly score items using only the words in this translation

In sentence of length k,
total bigrams = k – 1
matched bigrams = b

no. of mismatched bigrams = k – 1 – b
no. of matched chunks = k – 1 – b + 1
= k – b

These (k-b) chunks can be reordered in (k – b)! ways

In our example, (8-4)! = 24 candidate translations

Overview of MT Evaluation Metrics

Outline

- **Manual Evaluation**
- **Automatic Evaluation**
  - **BLEU**
  - TER
  - METEOR
  - GTM

Manual evaluation [11]

Common techniques:

1. Assigning fluency and adequacy scores on five *(Absolute)*
2. Ranking translated sentences relative to each other *(Relative)*
3. Ranking translations of syntactic constituents drawn from the source sentence *(Relative)*

Manual evaluation: Assigning Adequacy and fluency

Adequacy:
is the meaning translated correctly?
5 = All
4 = Most
3 = Much
2 = Little
1 = None

Fluency:
Is the sentence grammatically valid?
5 = Flawless English
4 = Good English
3 = Non-native English
2 = Disfluent English
1 = Incomprehensible

मैं एक व्याख्यान बैठा हूँ
Main ek vyaakhyan baitha hoon
I a lecture sit (Present-first person)
I sit a lecture
Adequate but not fluent

मैं व्याख्यान हूँ
Main vyakhyan hoon
I lecture am
I am lecture
Fluent but not adequate

- Evaluators use their own perception to rate
- Often adequacy/fluency scores correlate: undesirable

Outline

Manual Evaluation

Automatic Evaluation

- **BLEU**
- TER
- METEOR
- GTM

Translation edit rate[5] (TER)

- Introduced in GALE MT task

Central idea: Edits required to change a hypothesis translation into a reference translation

Shift Deletion Substitution Insertion

Prime Minister of India will address the nation today

Reference translation:भारत के प्रधान-मंत्री आज राष्ट्र को संबोधित करेंगे
Bhaarat ke pradhaan-mantri aaj raashtra ko sambodhit karenge

Candidate translation: प्रधान-मंत्री भारत के आज को अड्रेस करेंगे आज
Pradhaan-mantri bhaarat ke aaj ko address karenge aaj

Formula for TER

TER = # Edits / # Avg number of reference words

- Cost of shift 'distance' not incorporated
- Mis-capitalization also considered an error

Shift Deletion Substitution Insertion

Prime Minister of India will address the nation today

TER = 4 / 8

Reference translation:भारत के प्रधान-मंत्री आज राष्ट्र को संबोधित करेंगे
Bhaarat ke pradhaan-mantri aaj raashtra ko sambodhit karenge

Candidate translation: प्रधान-मंत्री भारत के आज को अड्रेस करेंगे आज
Pradhaan-mantri bhaarat ke aaj ko address karenge aaj

TER v/s BLEU

	TER	BLEU
Handling incorrect words	Substitution	N-gram mismatch
Handling incorrect word order	Shift or delete + insert incorporates this error	N-gram mismatch
Handling recall	Missed words become deleted words	Precision cannot detect 'missing' words. Hence, brevity penalty!

TER = # Edits / # Avg number of ref. words

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

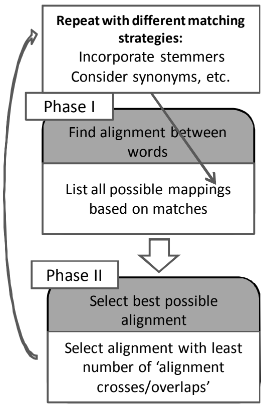
METEOR [6]

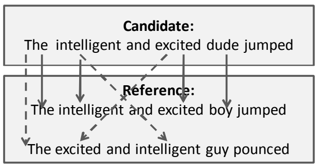
Aims to do better than BLEU

Central idea: Have a good unigram matching strategy

METEOR: Criticisms of BLEU

- Brevity penalty is punitive
- Higher order n-grams may not indicate grammatical correctness of a sentence
- BLEU is often zero. Should a score be zero?





METEOR: Process

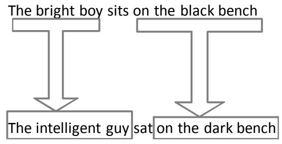
METEOR: The score

- Using unigram mappings, precision and recall are calculated. Then,

harmonic mean:

$$Fmean = \frac{10PR}{R+9P}$$

Penalty: Find 'as many chunks' that match



$$Score = Fmean * (1 - Penalty)$$

$$Penalty = 0.5 * \left(\frac{\#chunks}{\#unigrams\_matched} \right)$$

More accurate -> Less #chunks, Less penalty
Less accurate -> More #chunks, more penalty

METEOR v/s BLEU

	METEOR	BLEU
Handling incorrect words	Alignment chunks. Matching can be done using different techniques: Adaptable	N-gram mismatch
Handling incorrect word order	Chunks may be ordered in any manner. METEOR does not capture this.	N-gram mismatch
Handling recall	Idea of alignment incorporates missing word handling	Precision cannot detect 'missing' words. Hence, brevity penalty!

$$Score = Fmean * (1 - Penalty)$$

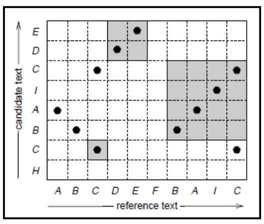
$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n \right)$$

GTM [9]

- General Text Matcher
- F-score: uses precision and recall
- Does not rely on 'human judgment' correlation
  - What does BLEU score of 0.006 mean?
- Comparison is easier

GTM Scores: Precision and Recall

- MMS: Maximum Match Size



$$\text{precision}(C|R) = \frac{\text{MMS}(C,R)}{|C|}$$

$$\text{recall}(C|R) = \frac{\text{MMS}(C,R)}{|R|}$$

$$\text{size}(M) = \sqrt{\sum_{r \in M} \text{length}(r)^2}$$

GTM v/s BLEU

	GTM	BLEU
Handling incorrect words	Precision based on maximum Match Size	N-gram mismatch
Handling incorrect word order	By considering maximum runs	N-gram mismatch
Handling recall	Recall based on maximum match size	Precision cannot detect 'missing' words. Hence, brevity penalty!

$$\text{precision}(C|R) = \frac{\text{MMS}(C,R)}{|C|}$$

$$\text{recall}(C|R) = \frac{\text{MMS}(C,R)}{|R|}$$

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n \right)$$

Conclusion

- Introduced different evaluation methods
- Need of Automatic Evaluation
- Formulated BLEU score
- Analyzed the BLEU score
- Compared BLEU with human judgment
- Shortcoming: Permutation of phrases possible
- Different evaluation metrics
- Compared the evaluation metrics

References: MT Evaluation

[1] Doug Arnold, Louisa Sadler, and R. Lee Humphreys, Evaluation: an assessment. *Machine Translation, Volume 8, pages 1–27.* 1993.

[2] K. Papineni, S. Roukos, T. Ward, and W. Zhu. Bleu: a method for automatic evaluation of machine translation, *IBM research report rc22176 (w0109-022). Technical report, IBM Research Division, Thomas, J. Watson Research Center.* 2001.

[3] Chris Callison-Burch, Miles Osborne, Phillipp Koehn, *Re-evaluating the role of Bleu in Machine Translation Research, European ACL (EACL) 2006, 2006.*

[4] R. Ananthakrishnan, Pushpak Bhattacharyya, M. Sasikumar and Ritesh M. Shah, *Some Issues in Automatic Evaluation of English-Hindi MT: More Blues for BLEU*, ICON 2007, Hyderabad, India, Jan, 2007.

[5] Matthew Snover and Bonnie Dorr and Richard Schwartz and Linnea Micciulla and John Makhoul, "A study of translation edit rate with targeted human annotation",In Proceedings of Association for Machine Translation in the Americas,2006

[6] Satanjeev Banerjee and Alon Lavie, "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments", Proceedings of the ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization, 2005

References: MT Evaluation

[7] Doddington, George, "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics", Proceedings of the second international conference on Human Language Technology Research, HLT 2002

[8] Maja and Ney, Hermann, "Word error rates: decomposition over Pos classes and applications for error analysis", Proceedings of the Second Workshop on Statistical Machine Translation, StatMT 2007

[9] Joseph Turian and Luke Shen and I. Dan Melamed, "Evaluation of Machine Translation and its Evaluation", In Proceedings of MT Summit IX, pages 386-393, 2003

[10] Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz and Josh Schroeder, "(Meta-) Evaluation of Machine Translation", ACL Workshop on Statistical Machine Translation 2007

CONCLUSIONS

Summary

- Introduction: perspective, MT paradigms, language divergence, alignment
- Word based models
- SMT of Indian languages
- Comparable and Parallel corpora
- Phrase based MT
- Decoding
- Factored SMT
- Tree based SMT
- Evaluation

Conclusions: No surprises!

- “Hybrid” is the way to go
- Interesting unverified-fully observations for translations involving Marathi
  - RBMT good for nominals, SMT for verbals
- Factored SMT beset by complexity barriers
- Decoding is immensely helped if a tracing of decoding is done
- Evaluation- BLEU not suitable for free word order languages
- Need to leverage multilinguality: parameter projection from pair of languages to another

SMT Resources at CFILT, IIT Bombay

- **Publications:**
  - http://www.cse.iitb.ac.in/~pb/pubs-yearwise.html
- **Śata-Anuvādak:** Phrase based SMT systems and extensions for 11 Indian languages
  - http://www.cfilt.iitb.ac.in/indic-translator
- Comparative Analysis of Phrase based and Factor Models
  - http://www.cfilt.iitb.ac.in/SMT
- Simple Experiment Management for Moses
  - https://bitbucket.org/anoopk/moses_job_scripts
- Indic NLP library: Unicode normalization and transliteration for Indian languages
  - https://bitbucket.org/anoopk/indic_nlp_library

ITS FINALLY OVER!