
Statistical Language Processing

Ruvan Weerasinghe

Language Technology Research Centre
University of Colombo School of Computing
Colombo
Sri Lanka

What I'll try to do...

- Stress the need for Localization and Translation
 - Make a case for 'letting the data talk': The statistical approach to language processing
 - Provide a simple description of the *strange* world of statistical machine translation
 - Discuss some results of trying this for our languages: Sinhala, Tamil and English
-

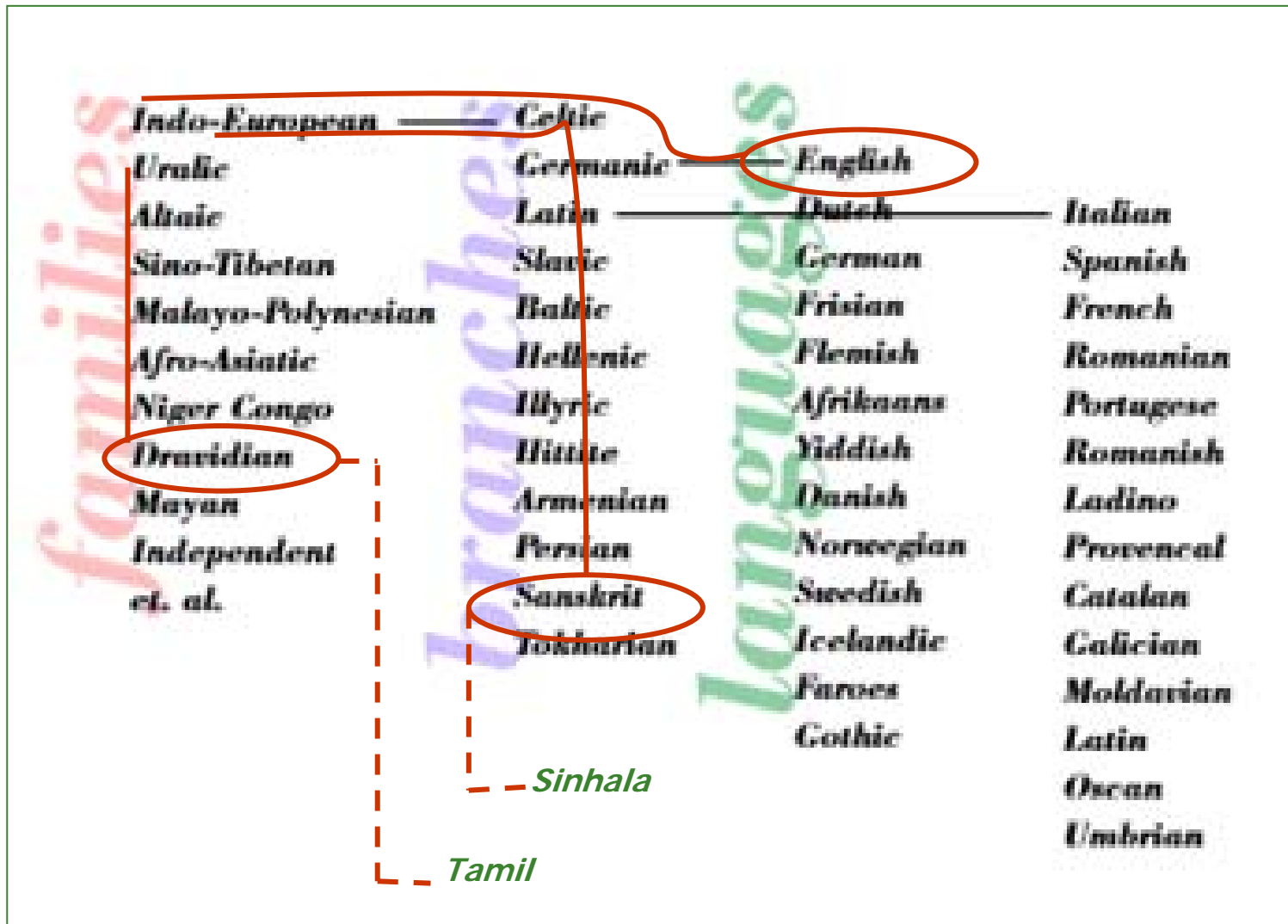
What I'll do now...

- Stress the need for Localization and Translation
 - Make a case for 'letting the data talk': The statistical approach to language processing
 - Provide a simple description of the *strange* world of statistical machine translation
 - Discuss some results of trying this for our languages: Sinhala, Tamil and English
-

Why Localize, why translate?

- The top 5 stumbling blocks to crossing the digital divide
 - Computer literacy
 - Bandwidth
 - Network penetration
 - Cost
 - And the winner is...?
-

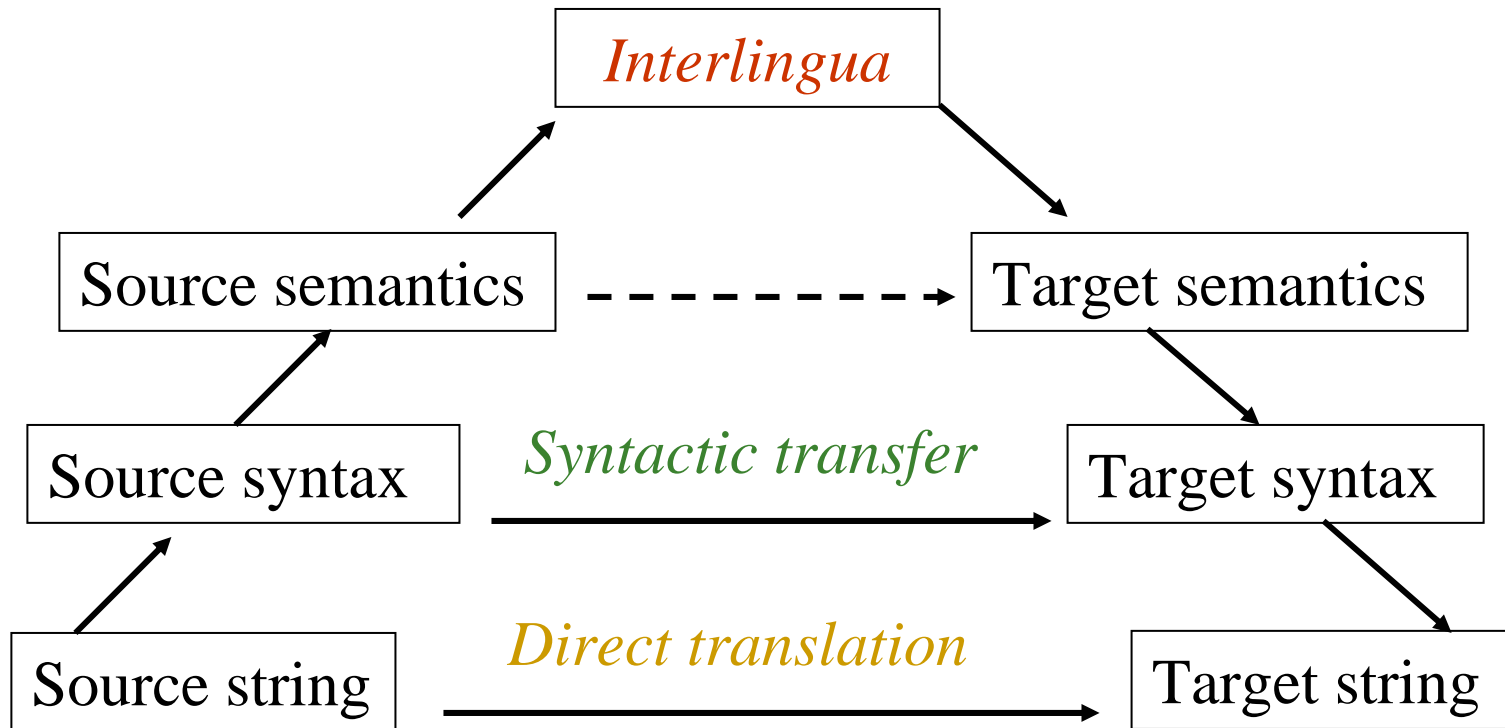
Why Localize, why translate?



What is MT anyway?

- It is not to do with computer languages – it's concerned with *real* ones: English, Spanish, Sinhala...
 - Input can be spoken or written/typed, but converting to electronic text (ASCII / Unicode) is not our problem!
 - *Understanding* by the machine is strictly optional!!
-

What is it anyway?



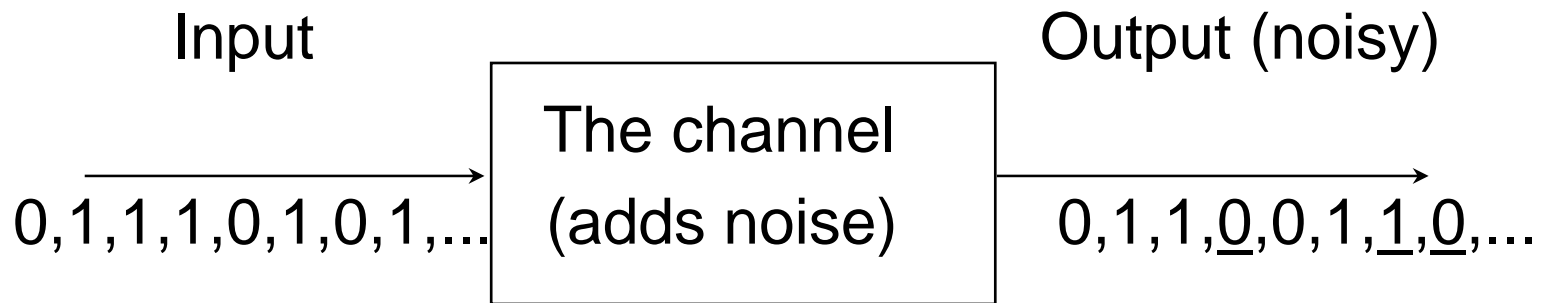
The 'translation cone'

What I'll do now...

- Stress the need for Localization and Translation
 - Make a case for 'letting the data talk': The statistical approach to language processing
 - Provide a simple description of the *strange* world of statistical machine translation
 - Discuss some results of trying this for our languages: Sinhala, Tamil and English
-

The Noisy Channel

- Prototypical case:



- Model: probability of error (noise):
- Example: $p(0|1) = .3$ $p(1|1) = .7$ $p(1|0) = .4$ $p(0|0) = .6$
- The Task:
known: the noisy output; want to know: the input (decoding)

Noisy Channel Applications

- OCR
 - straightforward: text → print (adds noise), scan → image
- Handwriting recognition
 - text → neurons, muscles (“noise”), scan/digitize → image
- Speech recognition (dictation, commands, etc.)
 - text → conversion to acoustic signal (“noise”) → acoustic waves
- Machine Translation
 - text in target language → translation (“noise”) → source language
- Also: Part of Speech Tagging
 - sequence of tags → selection of word forms → text

Noisy Channel: The Golden Rule of ...

OCR, ASR, HR, MT, ...

- Recall:

$$p(A|B) = p(B|A) p(A) / p(B) \quad (\text{Bayes formula})$$

$$A_{\text{best}} = \operatorname{argmax}_A p(B|A) p(A) \quad (\text{The Golden Rule})$$

- $p(B|A)$: the acoustic/image/translation/lexical model
 - application-specific name
 - will explore later
- $p(A)$: **the language model**

The Perfect Language Model

- Sequence of word forms [forget about tagging for the moment]
- Notation: $A \sim W = (w_1, w_2, w_3, \dots, w_d)$
- The big (modeling) question:

$$p(W) = ?$$

- Well, we know (Bayes/chain rule \rightarrow):

$$\begin{aligned} p(W) &= p(w_1, w_2, w_3, \dots, w_d) = \\ &= p(w_1) \cdot p(w_2|w_1) \cdot p(w_3|w_1, w_2) \cdot \dots \cdot p(w_d|w_1, w_2, \dots, w_{d-1}) \end{aligned}$$

- Not practical (even short $W \rightarrow$ too many parameters)

Markov Chain

- Unlimited memory (cf. previous foil):

- for w_i , we know all its predecessors

$w_1, w_2, w_3, \dots, w_{i-1}$

- Limited memory:

- we disregard “too old” predecessors

- remember only k previous words: $w_{i-k}, w_{i-k+1}, \dots, w_{i-1}$

- called “ k^{th} order Markov approximation”

- + stationary character (no change over time):

$$p(W) \cong \prod_{i=1..d} p(w_i | w_{i-k}, w_{i-k+1}, \dots, w_{i-1}), \quad d = |W|$$

n-gram Language Models

- $(n-1)^{\text{th}}$ order Markov approximation \rightarrow n-gram LM:

$$p(W) =_{\text{df}} \prod_{i=1..d} p(w_i | w_{i-n+1}, w_{i-n+2}, \dots, w_{i-1})$$

prediction \swarrow \searrow history !

- In particular (assume vocabulary $|V| = 60\text{k}$):
 - 0-gram LM: uniform model, $p(w) = 1/|V|$, 1 parameter
 - 1-gram LM: unigram model, $p(w)$, $6 \cdot 10^4$ parameters
 - 2-gram LM: bigram model, $p(w_i | w_{i-1})$, $3.6 \cdot 10^9$ parameters
 - 3-gram LM: trigram model, $p(w_i | w_{i-2}, w_{i-1})$, $2.16 \cdot 10^{14}$ parameters

LM: Observations

- How large n ?
 - nothing is enough (theoretically)
 - but anyway: as much as possible (\rightarrow close to “perfect” model)
 - empirically: **3**
 - parameter estimation? (reliability, data availability, storage space, ...)
 - 4 is too much: $|V|=60k \rightarrow 1.296 \cdot 10^{19}$ parameters
 - but: 6-7 would be (almost) ideal (having enough data): in fact, one can recover original from 7-grams!
- Reliability $\sim (1 / \text{Detail})$ (\rightarrow need compromise)
- For now, keep word forms (no “linguistic” processing)

Statistical Machine Translation

- Why SMT?
 - Letting the ‘data talk’ – a machine learning approach (aka data mining/pattern recognition)
 - Advantages of the ‘direct approach’
 - Non-reliance on expert knowledge
 - Learnability/Trainability
 - Theoretical basis
 - Other direct translation methods
 - EBMT: reliance on more examples...
-

What I'll do now...

- Stress the need for Localization and Translation
 - Make a case for 'letting the data talk': The statistical approach to language processing
 - Provide a simple description of the *strange* world of statistical machine translation
 - Discuss some results of trying this for our languages: Sinhala, Tamil and English
-

Statistical Machine Translation

The screenshot shows a text editor window titled "NoteTab Light" with two panes. The left pane contains the source text in English, and the right pane contains the target text in Sinhala. Red circles highlight the words "2001" and "17th" in both the source and target text, illustrating the alignment process in SMT.

Source Text (English):

```
<div>
<seg> An exchange on socialism and human nature.
<seg> By Nick Beams . </seg>
<seg> 1 May 2001. </seg>
<seg> The following is a reply by Nick Beams , a
<seg> BM , who describes himself as a Reagan Con
<seg> Centralised planning does not work and hist
<seg> The freer the society the more prosperous i
<seg> The full text of BM's email is posted at R
<seg> Thank you for your e-mail for it provides u
<seg> Your defence of the capitalist market boils

<seg> I propose to begin my reply with an examina
<seg> Then I shall turn to Ronald Reagan . </se
<seg> The political ideas of Jefferson , and the

<seg> The social context in which these ideas wer
<seg> In the latter part of the 17th century John
<seg> According to Locke , every man was the sole
<seg> The Canadian political theorist C.B. Macphe
<seg> Past societies had , of course , developed
<seg> What was new in the 17th century was the de
<seg> This involved a sharp break with the previo
<seg> The significance of Locke , as Macpherson d
<seg> [ I ]f the new kind of property required by
<seg> The universal basis was found in 'labour' .
<seg> Every man had a property in his own labour
<seg> And from the postulate that a man's labour
<seg> The postulate reinforced the concept of pro
<seg> As his labour was his own , so was the land
<seg> This was the principle that Locke made cent
<seg> By the time of Jefferson , a century later
<seg> In his Discourse on the Origin and Foundat
<seg> In man's natural state the earth and its fr
```

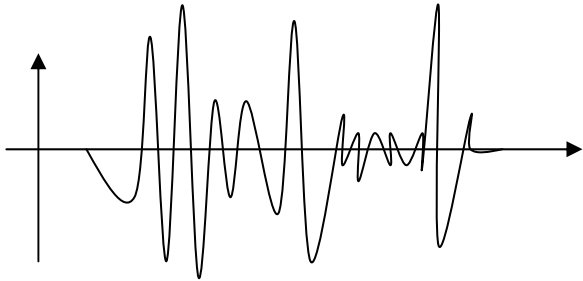
Target Text (Sinhala):

හඩඉවද හසළවද සමාජවාදය සහ මිනිස් ස්වභාවය පිලිබදව
හසළවද හික් ක්ෂිප් විසිනි . හඹුසළවද
හසළවද 2001 මැයි 01 . හඹුසළවද
හසළවද පහත පලවන්නේ සමාජවාදය සහ ආර්ථික
හසළවද හමන් රිකනිකානු කොන්සර්වේටිව් රිපබ්ලිකානු
හසළවද මධ්‍යතන සැලසුම්කරනය නො කෙරෙන්නක් ඩි
හසළවද සමාජය නිදහස් වන තරමට එය වඩා සොය
හසළවද ක්ෂිප්ගේ සම්පූර්ණ ර්-මේල් පහ නික් ක්ෂිප්ට
හසළවද ඔබගේ ර්මේල් පහට ස්තූතියක් වෙමි . මෑ
හසළවද ධනවාද වෙලද පොල ආරක්ෂා කිරීමේ කමි
හසළවද සහන ප්‍රධානතය නිදහස් වෙලද පද්ධතිය ආ
හසළවද ඔබගේ හර්තයෙහි ඇරට්ව හෝමස් පේලර්ස
හසළවද ඇනතුරුව මම රොනල්ඩ් රේකන් වෙත හැරෙ
හසළවද පේලර්සන්ගේ සහ ඇමරිකානු විප්ලවයේ ඇ
හසළවද ගුන් බොහොමයක් , පසුව ඉං ආ සි කිවිල්
හසළවද පවිල ඇදහක් ඉදිරිපත් කල දේපලාලන සන්ද
හසළවද 17 වන සියවසේ පසුභාගයේ දී පෝන් ලොක
හසළවද ලොක්ව් ඇනුව , සෑම මිනිසෙකුම ඔහුගේ පෙ
හසළවද කැපවීමකින් දේපලාලන න්‍යායාචාරය සි.කී. ම
හසළවද හත්තනින් ම ඇති සමාජයන් දේපල පිලිබද
හසළවද 17 වන සියවසෙහි නවමු දෙය වූයේ , ධන
හසළවද පෙර පැවති , භූමිය සහ පෑතිවියෙහි මල්
හසළවද මැකර්සන් පෙන්වා දුන් ඇසුරු ලොක්ගේ ව
හසළවද ධනවාද වෙලද පොල සමාජයට ඇවසන කර
හසළවද විවිධය පදනම ඉමය ආල සොයා ගැනුනි
හසළවද සෑම මිනිසෙකුටම ඔහුගේ ම ඉමය ආල දේ
හසළවද එසේ ම , මිනිසකුගේ ඉමය ඇනනනය ය පුළු
හසළවද මෙම ඉල සහනය දේපල පිලිබද සංකල්පය
හසළවද ඔහුගේ ඉමය ඔහුගේ ම වන හෙයින් , ඔහු
හසළවද ලොක් , දේපල පිලිබද ලිබරල් සංකල්පයට
හසළවද සියවසකින් ඇනතුරුව , පේලර්සන්ගේ යුගය
හසළවද 1755 දී රූසෝ ප්‍රසිද්ධ කල මිනිසන් ඇතර
හසළවද මිනිසාගේ ස්වභාවික හත්වයේ දී පෑතිවිය ස
හසළවද පොද්කලික දේපල ස්ථාපිත විම ඇසමානතාව
හසළවද ඇමරිකානු නිදහස් ප්‍රධානතය හඹුසළවද

Statistical Machine Translation

- The *usual* (KB/Interlingua) story:
 - Input text tokenized (somehow)
 - e.g. “dZOn_IVvz_meIrl” => John loves Mary => [john, love+s, mary]
 - Parsed into trees and logical forms (say FOPL)
 - e.g. S(Np(john), Vp(V(love+s), Np(mary))) => loves(john,mary)
 - Meaning is common in any language (is it really?)
 - Generation step creates new sentences in target language from these logical forms (non-trivial for realistic output)
 - e.g. => John aime Mary (Voilà!)

Statistical Machine Translation

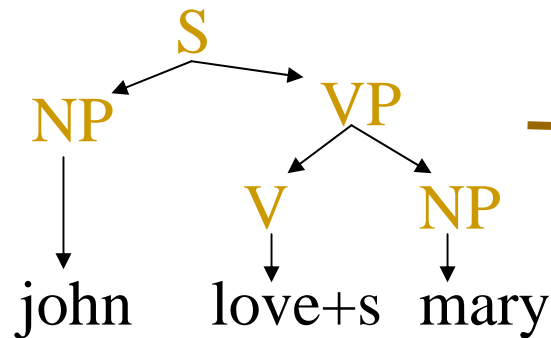


“dZOn_lVvz_meIrI”

John loves mary.

John loves Mary

[john, love+s, mary]



loves(john, mary)

Statistical Machine Translation

- The *strange* story:
 - Input sentence expanded by repeating *high fertility* words in source language (removing zero fertility ones)
 - e.g. John does not love Mary => john not not love mary
 - Some NULL words inserted to account for 'extra words' of target needed e.g. => NULL john not not love mary
 - Resulting 'sentence' translated largely word-for-word
 - e.g. NULL john ne pas aime mary
 - 'Translated' stream re-ordered using frequent patterns (n-gram) in target language
 - e.g. john ne aime pas mary => John n'aime pas Mary

Statistical Machine Translation

John **doesn't** love Mary

John does **not** love Mary

john **not not** love mary

NULL john not not love mary

john **ne pas** aime mary

john ne aime pas mary

John n'aime pas Mary

Statistical Machine Translation

- Why this weird way of doing things?
 - Divide and conquer (our 'old friend')
 - Difficult to know/model $P(s|t)$ – need very accurate model
 - Thanks to Bayes rule we decompose as:
 $P(t|s) \times P(s) / P(t)$
 - i.e. we need a 'translation model' of Tamil to Sinhala and a 'language model' just for Sinhala
-

Statistical Machine Translation

- What do we gain/achieve by this?
 - we distribute the burden of accuracy between *two* models instead of *one*
 - the ‘translation model’, $P(t|s)$, may predict many hypothesis (some really crazy ones!)
 - the ‘language model’, $P(s)$ will ‘select’ the more ‘natural sounding’ ones of these
-

Building SMT Models

- Where do we get these figures from?
 - Not available in dictionaries ☹
 - Calculate from real sentences (texts/utterances)
 - Estimating the *language model*, $P(s)$
 - Each sentence has a VERY low probability
 - Many will have zero (unseen before!)
 - Approximate sentence probability by the number of previously seen *n-grams* that sentence has
-

Building SMT Models – P(e)

- Estimating P(s)
- E.g. if we consider *bigram* probabilities:
 $P(\text{මම බන් කමි}) \sim$
 $P(\text{මම} | \langle \text{start} \rangle) \times P(\text{බන්} | \text{මම}) \times P(\text{කමි} | \text{බන්}) \times$
 $P(\langle \text{end} \rangle | \text{කමි})$
- E.g. if we consider *trigram* probabilities:
 $P(\text{මම බන් කමි}) \sim$
 $P(\text{මම} | \langle \text{start} \rangle \langle \text{start} \rangle) \times P(\text{බන්} | \langle \text{start} \rangle \text{මම}) \times$
 $P(\text{කමි} | \text{මම බන්}) \times P(\langle \text{end} \rangle | \text{බන් කමි}) \times \dots$

Building SMT Models – P(e)

- What if there's ONE such *n-gram* unseen?
 - The probability of the sentence becomes ZERO!
- Smoothing – graceful degradation
 - E.g. if we use trigrams, we can 'fall back' to bigrams or even unigrams:

$$P(x|y z) = 0.95 \times (\# \text{ of "xyz"}) / (\# \text{ of "xy"}) + \\ 0.04 \times (\# \text{ of "yz"}) / (\# \text{ of "z"}) + \\ 0.008 \times (\# \text{ of "z"}) / (\text{total words}) + \\ 0.002$$

Building SMT Models – $P(f|e)$

- Estimating the *translation model*
- Each source language word is said to have a *fertility* – the number of words it translated to in the target language on average
 - E.g. we assumed that ‘not’ has fertility 2 in the example and generated 2 French words for it
 - How do we get these?
 - Estimate from texts of course!

Building SMT Models – $P(f|e)$

- Each French word is set to a particular position (slot) according to the English word that generated it (referred to as *distortion*)
 - E.g. $P(5|2,4,6)$ will estimate prob. of an English word in position 2 of a sentence of length 4 ending up in position 5 of a French sentence of length 6!
 - So “... **ne pas** aime ...” becomes “... **ne** aime **pas**...”
- Insert NULL English words for each position of the French sentence with some probability to account for ‘spurious’ French words!
 - We slot their French translations into the slots remaining after generating the ‘real’ French words (above) based on a single probability for the entire sentence

Building SMT Models – $P(f|e)$

- A Spanish example from Knight(1999):

Mary did not **slap** the green witch (input)

Mary not **slap slap slap** the green witch (fertilities)

Mary not slap slap slap **NULL the** green witch (spurious words)

Mary no daba una botefada **a la verde bruja** (choose translations)

Mary no daba una botefada a la **bruja verde** (choose positions)

Building SMT Models – $P(f|e)$

- Big question: how do we know which English words to map to which French one(s)?
- If only...
 - We had a probabilistic ‘translation lexicon’ (dictionary)
 - We had word-aligned sentences
- As usual we estimate this from data too!
 - Start by assuming any word maps to any
 - Use *EM algorithm* to improve alignment

Building SMT Models

- Is that all?
 - Not quite. There is subtlety in the last bit!
 - The ‘IBM model’ for this consists of 5 sub-models (model 1 through 5)
 - Each model gets more sophisticated and “feeds” partially trained parameters to the next
 - The EM algorithm is an essential tool in ‘chicken & egg’ situations...
-

Building SMT Models

- It is thus a boot-strapping scheme
 - The good news:
 - There's an excellent tutorial (thanks to Kevin Knight)
 - There's a well documented toolkit – GIZA
 - A less well documented extension – GIZA++
 - There are other support tools collected under the 1999 JHU summer workshop on SMT
-

Building SMT Models

- We now have the language and translation models, $P(s)$ and $P(t|s)$
- We still need to search through all the candidate translations of a sentence for the ‘best’ (one which maximizes product) – this is a ‘hard’ problem 😞
- We need a ‘decoder’ (borrowing from ASR)
 - Using Viterbi style search
 - ISI Rewrite is a public domain decoder

What I'll do now...

- Stress the need for Localization and Translation
 - Make a case for 'letting the data talk': The statistical approach to language processing
 - Provide a simple description of the *strange* world of statistical machine translation
 - Discuss some results of trying this for our languages: Sinhala, Tamil and English
-

Does SMT *really* work?

- Not if you think the present translate button doesn't work at all!
 - Most current commercial systems use hand-crafted rules, machine readable dictionaries and yet are not FAHQMT!
 - Current laboratory SMT systems approach and even surpass these 'knowledge based' systems
 - They are easy to experiment with repeatedly
-

Does SMT *really* work?

- Surely we are done now... not quite: how do we recognize a 'good translation'?
 - What is a better translation?
 - Problem of evaluation
 - What is good for me, is it good for you?
 - Can the experts agree?
 - Importance of automatic metrics
 - For comparing 'reference' (professional) translations with SMT
 - IBM BLEU: one way to proceed
-

Does SMT *really* work?

- Intuitively humans still far better!
 - Only approximately *twice as good* on current metrics!
 - on a 0 to 1 (IBM BLEU) score
 - SMT systems score 0.05 - 0.25
 - Mere mortals such as you and me would be hard pressed to score more than about 0.4!
 - Point: How good need a *good* translation be
-

Does SMT *really* work?

- For the English-Sinhala translations we got BLEU scores of only 0.02 – 0.06 ☹️
- For the Sinhala-Tamil translations we get BLEU scores of 0.12 – 0.14
- How does this measure up?
 - State of the art French-English ~ 0.25 with 2-4 ‘reference translations’
 - Number of reference translations affects score
 - ‘Adjusted’ Sinhala-Tamil score ~ 0.185

Does SMT *really* work?

- Sinhala-Tamil translation still unintelligible ☹️
 - Statistical Machine Translation results in general
 - Many applications would be happy even with current levels of accuracy (e.g. cross-language IR?)
 - Some humans would also be not unduly upset at current levels (e.g. when it's the ONLY way to survive/proceed!)
 - Further work is needed for when the gist is just not good enough...
-

Conclusions

- The 'translate button' is work-in-progress!
 - Translation is non-trivial – an utterance can be translated in an infinite number of ways at different levels of acceptability!!
 - SMT is doable – at least the input and expected output are definable (not in KB approach)
 - There is no serious knowledge acquisition bottleneck for 'new' language pairs
 - Our experiments found SMT to work better for two languages from distinct families (S & T) than for two from the same (S & E) – case for re-drawing family tree!
-

=> Bohoma Isthuthi, Ayubowan!

working...

[=> thanks hello - TM Hyp 1

=> much thanks goodbye - TM Hyp2

=> ...

=> many thank you hello - TM Hypn]

=> Thank you and Goodbye - LM Best 1

=> Many thanks, Good day - LM Best2
