Automatic Translation Error Analysis

or how to brute-force through exponential complexity algorithms by abusing beam search

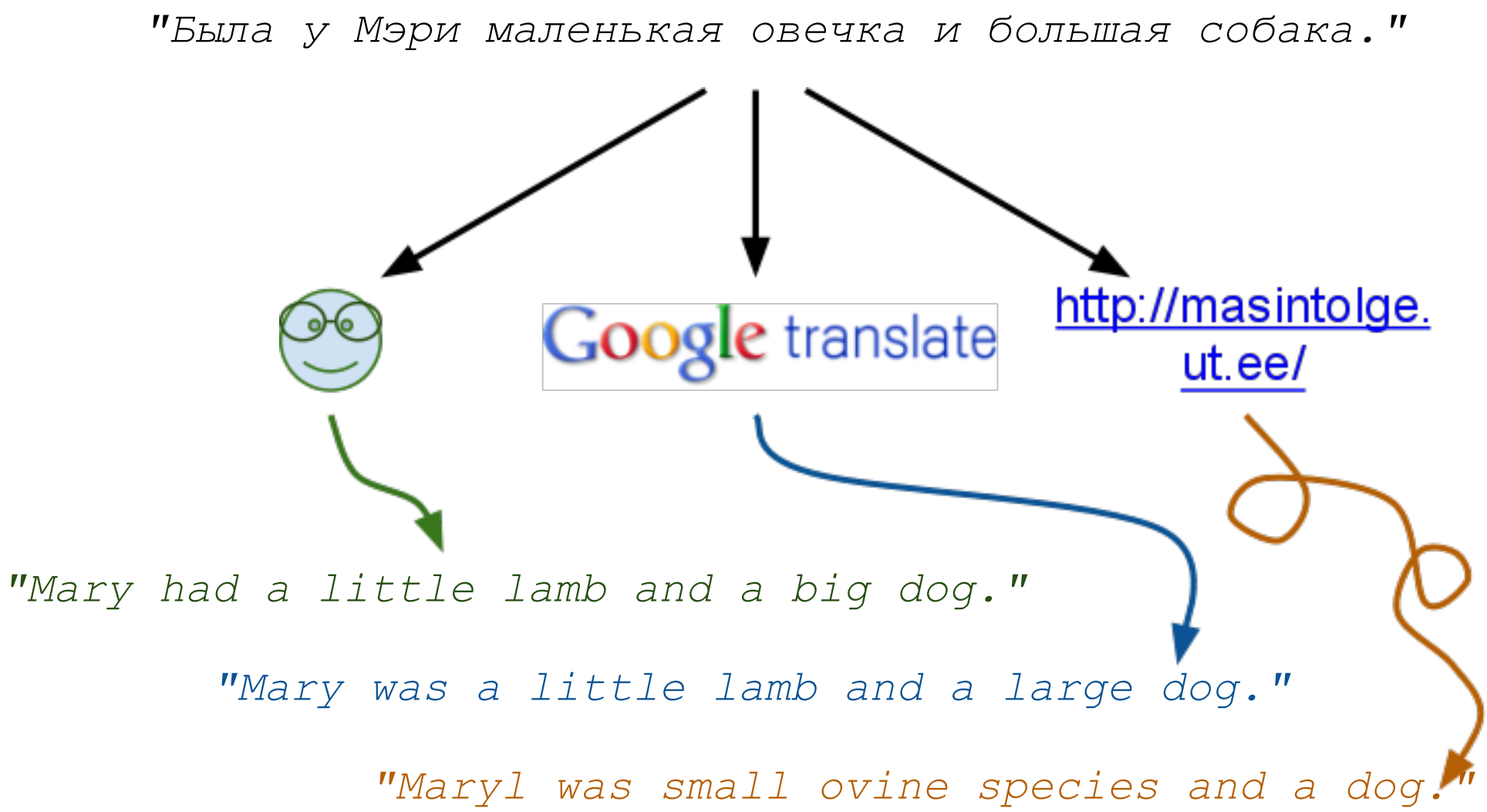
Mark Fishel, TÜ ATI

Feb. 5, 2011, Theory Days at Nelijärve

Outline

- Approaches to MT evaluation
- Automatic analysis of translation errors
  - alignment
  - error detection
  - error summarization
- Meta-evaluation
- First results
- Future work

Translation



Evaluation

Mostly done by comparison between the produced translation (hypothesis) and a correct one (reference)

	Manual	**Automatic**
Score	Adequacy/fluency, rank, HTER	WER, BLEU, NIST, METEOR, TER, SemPOS, LRscore, ... ad ∞
Analysis	(Vilar et al. 2006)	**Our work**

- Score -- good for comparison, but not informative
- Manual -- expensive

Translation errors by Vilar et al. (2006):

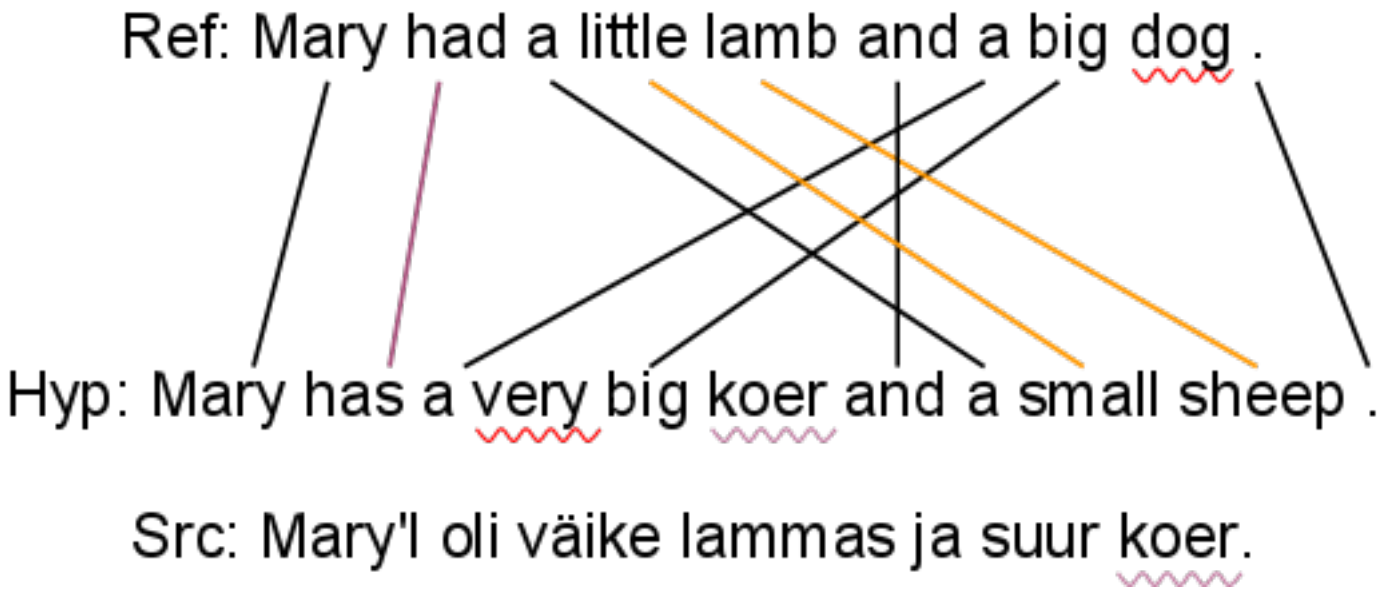
- Punctuation
- Missing words (in the reference)
  - Content word
  - Functional word
- Incorrect words (in the hypothesis)
  - Incorrect sense/form
  - Extra word
  - Style, idioms
- Unknown words (in the hypothesis)
  - Unknown stem/form
- Word order (in the hypothesis)
  - Short/long range
  - Word/phrase

Automatic error analysis

- Alignment between the hypothesis and the reference
- Error detection and classification
- Error summarization

- Result -- ~equivalent to Vilar et al.'s error classification

Alignment



Almost trivial, except for ambiguous alignment pairs

- repeating words (esp. punctuation, articles, etc.)
- surface forms of one lemma
- synonyms

Alignment solution

- Align using lemmas/synonym sets
- Alignment modelled as a HMM
  - observed variables -- hypothesis words
  - hidden variables -- reference words
  - emission probabilities allow matching words to align:

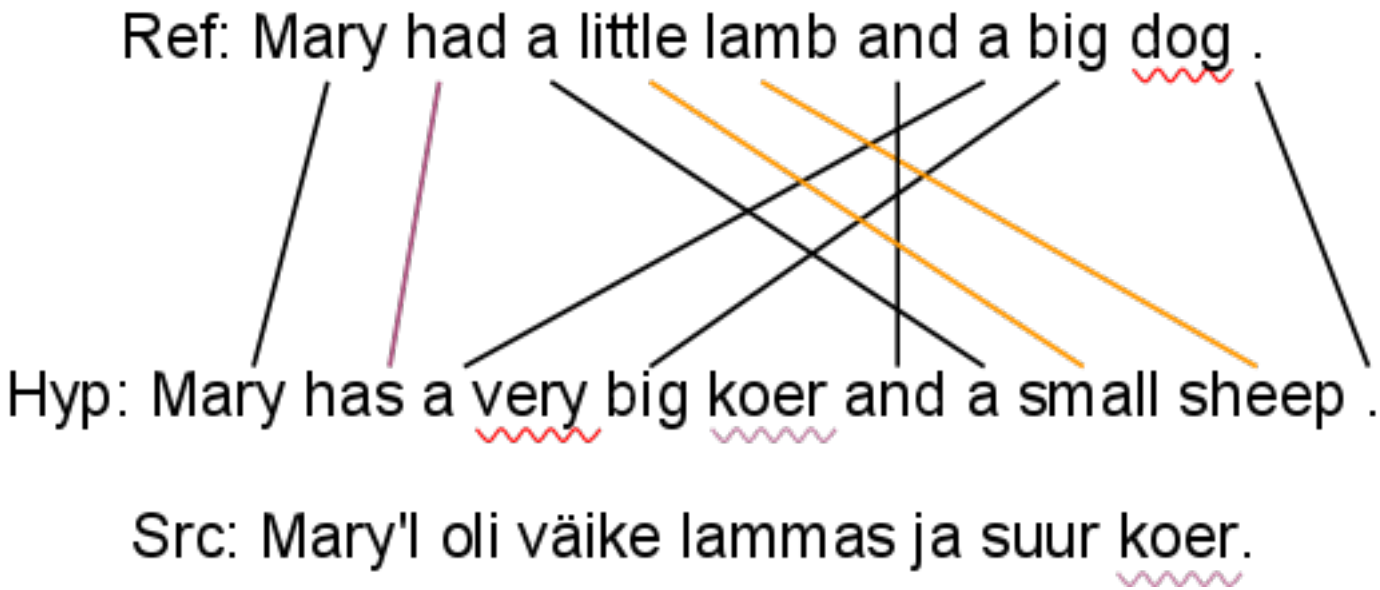
$$p(h_i|r_j) = (h_i == r_j)? \ \frac{1}{|\{h : h \in \text{hyp}, h = h_i\}|} : \ 0$$

  - transmission probabilities penalize long-distance reordering:

$$p(r_j|r_{j-1}) \sim (r_j - r_{j-1})^{-2}$$

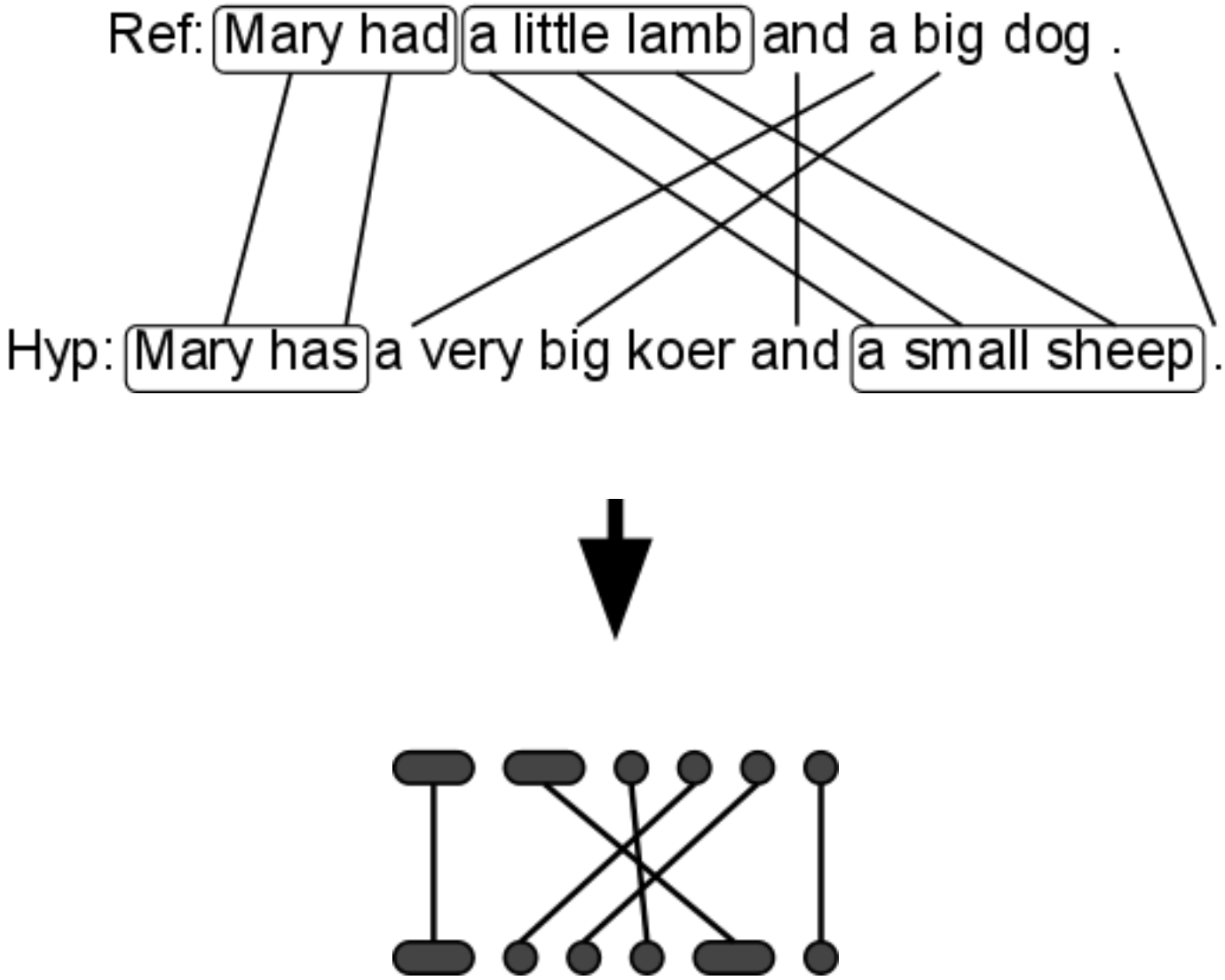
- We want only 1-to-1 alignments
  - makes search cost exponential
  - do a beam search

Lexical error detection

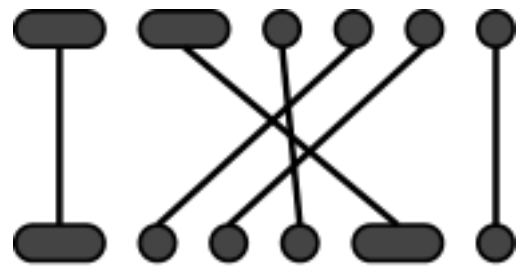


- unaligned ref words -- missing
- unaligned hyp words
  - present in src? untranslated
  - else, extra word
- aligned, different surface form
  - synonyms
  - or wrong surface form

Order error detection



Order error detection

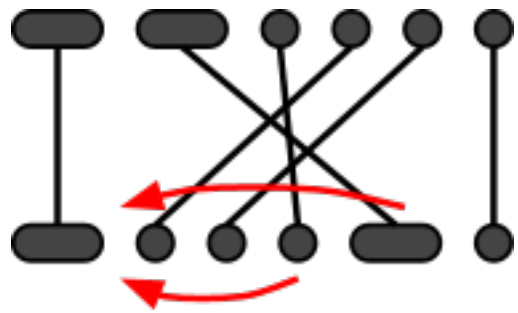


Can be used to

- calculate permutation distance
  - Hamming distance
  - Kendall's τ distance
  - Ulam's distance
  - Spearman's rank correlation coefficient

- Find misplaced words and phrases

Misplaced units



Breadth-first search for a minimum number of unit shifts

- vertices: permutations of the hypothesis ranks
- edge present if the two permutations differ by two adjacent symbols in the wrong order
- edge weight is 0 for block shift continuation, or 1 otherwise

- avoid exponential cost with beam search

Here: 1 word shift and 1 phrase shift

Error summarization

Can be performed on different levels

- keep list of errors for every translated sentence
  - usable for examining errors sentence-by-sentence

- summarize total number of errors, per category
  - apply part-of-speech tagging to classify content/functional words
  - present error numbers in percentage of total words in ref/hyp
  - usable for overall system weakness comparison

- linear combination of the ratio of different error types -- score!

Summary

- Fast
- Inexpensive
- Language-independent, but can benefit from linguistic analysis

Meta-evaluation

- For scores -- correlation with human judgements
- For analysis -- precision/recall of error detection
- Both require manual labor
- Manual analysis requires a lot of labor

First results

- 2656 sentences, from http://masintolge.ut.ee/ input, manually translated into English
- translated automatically with Google and 2 UT systems

	UT-Base	UT-Newer	Google
Missing	54.29%	51.79%	41.52%
Untranslated	10.08%	8.77%	2.40%
Extra	33.96%	38.77%	30.23%
Wrong form	2.40%	2.83%	3.05%
Misplaced	6.89%	7.09%	7.45%
Rho	0.905	0.904	0.921

Future work

- Improve alignment
- Structural order error detection, with syntactic analysis
- Perform meta-evaluation
- Scoring, tuning weights to fit dev set

Thank you!