Natural Language Processing: Phrase-Based Machine Translation

Christopher Manning

Borrows some slides from Kevin Knight and Dan Klein

Lecture Plan

1. **Searching for the best translation: Decoding** [3:00–3:05]
2. MT Evaluation [3:05–3:20]
3. Phrase-Based Statistical MT
   a) Introduction [3:20–3:25]
   b) Building a phrase table [3:25–3:35]
   c) Log-linear models for scoring hypotheses [3:35–3:50]
   d) Phrase-based decoders [3:50–4:10]
   e) Training machine learning models [4:10–4:15]
4. Extra time [4:15–4:20]

Searching for a translation

Of all conceivable English word strings, we want the one maximizing P(e) x P(f | e)

Language model, TBD

Exact search

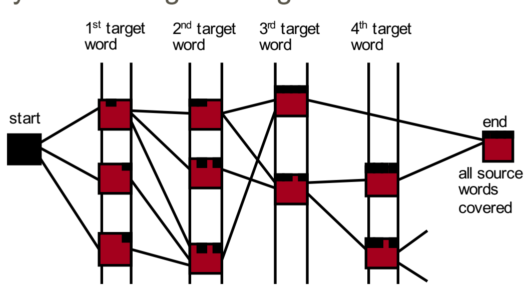
- Even if we have the right words for a translation, there are **n!** permutations
- We want the translation that gets the highest score under our model
- Finding the argmax with a n-gram language model is NP-complete [Germann et al. 2001]
- Equivalent to Traveling Salesman Problem

3

Searching for a translation

- Several search strategies are available
  – Usually a beam search where we keep multiple stacks for candidates covering the same number of source words
  – Or, we could try "greedy decoding", where we start by giving each word its most likely translation and then attempt a "repair" strategy of improving the translation by applying search operators (Germann et al. 2001)
- Each potential English output is called a *hypothesis*.

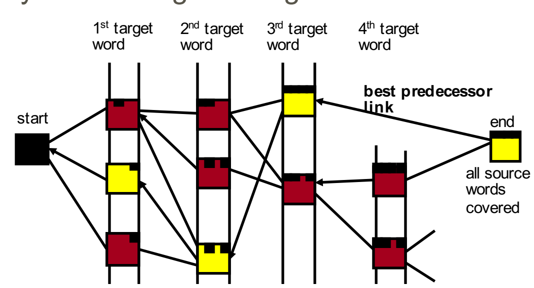
Dynamic Programming Beam Search



Each partial translation hypothesis contains:
- Last English word chosen + source words covered by it
- Next-to-last English word chosen
- Entire coverage vector (so far) of source sentence
- Language model and translation model scores (so far)

[Jelinek, 1969; Brown et al, 1996 US Patent; (Och, Ueffing, and Ney, 2001]

Dynamic Programming Beam Search



Each partial translation hypothesis contains:
- Last English word chosen + source words covered by it
- Next-to-last English word chosen
- Entire coverage vector (so far) of source sentence
- Language model and translation model scores (so far)

[Jelinek, 1969; Brown et al, 1996 US Patent; (Och, Ueffing, and Ney, 2001]

Lecture Plan

1. Searching for the best translation: Decoding [3:00–3:05]
2. **MT Evaluation** [3:05–3:20]
3. Phrase-Based Statistical MT
   a) Introduction [3:20–3:25]
   b) Building a phrase table [3:25–3:35]
   c) Log-linear models for scoring hypotheses [3:35–3:50]
   d) Phrase-based decoders [3:50–4:10]
   e) Training machine learning models [4:10–4:15]
4. Extra time [4:15–4:20]

Evaluating Alignments: Alignment Error Rate (Och & Ney 2000)

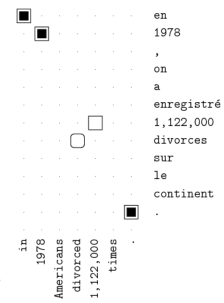
□ = Sure

○ = Possible

■ = Alignments (predicted)

$$AER(A,S,P) = \left(1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}\right)$$

$$= \left(1 - \frac{3+3}{3+4}\right) = \frac{1}{7}$$



Most work has used AER and we do, but it is problematic, and it's better to use an alignment F measure (Fraser and Marcu 2007)

Comparative results (AER)

[Och & Ney 2003]

Model	Training scheme	Size of training corpus: 0.5K	8K	128K	1.47M
Dice		50.9	43.4	39.6	38.9
Dice+C		46.3	37.6	35.0	34.0
Model 1	1^5	40.6	33.6	28.6	25.9
Model 2	$1^5 2^5$	46.7	29.3	22.0	19.5
HMM	$1^5 H^5$	26.3	23.3	15.0	10.8
Model 3	$1^5 2^5 3^3$	43.6	27.5	20.5	18.0
	$1^5 H^5 3^3$	27.5	22.5	16.6	13.2
Model 4	$1^5 2^5 3^3 4^3$	41.7	25.1	17.3	14.1
	$1^5 H^5 3^3 4^3$	26.1	20.2	13.1	9.4
	$1^5 H^5 4^3$	26.3	21.8	13.3	9.3
Model 5	$1^5 H^5 4^3 5^3$	26.5	21.5	13.7	9.6
	$1^5 H^5 3^3 4^3 5^3$	26.5	20.4	13.4	9.4
Model 6	$1^5 H^5 4^3 6^3$	26.0	21.6	12.8	8.8
	$1^5 H^5 3^3 4^3 6^3$	25.9	20.3	12.5	8.7

Common software: GIZA++/Berkeley Aligner

Illustrative translation results

- *nous avons signé le protocole .* — *(Foreign Original)*
- we did sign the memorandum of agreement . — (Reference Translation)
- we have signed the protocol . — (IBM4+N-grams+Stack)

- *où était le plan solide ?* — *(Foreign Original)*
- but where was the solid plan ? — (Reference Translation)
- where was the economic base ? — (IBM4+N-grams+Stack)

对外经济贸易合作部今天提供的数据表明，今年至十一月中国实际利用外资四百六十九点五九亿美元，其中包括外商直接投资四百点零七亿美元。

the Ministry of Foreign Trade and Economic Cooperation, including foreign direct investment 40.007 billion US dollars today provide data include that year to November china actually using foreign 46.959 billion US dollars and

- See more – including the output of Stanford systems – at:
  - http://matrix.statmt.org/

MT Evaluation

- Manual (the best!?):
  - SSER (subjective sentence error rate)
  - Correct/Incorrect
  - **Adequacy and Fluency** (5 or 7 point scales)
  - Error categorization
  - **Comparative ranking of translations**
- Testing in an application that uses MT as one sub-component
  - E.g., question answering from foreign language documents
    - May not test many aspects of the translation (e.g., cross-lingual IR)
- Automatic metric:
  - WER (word error rate) – why problematic?
  - **BLEU (Bilingual Evaluation Understudy)**

BLEU Evaluation Metric

(Papineni et al, ACL-2002)

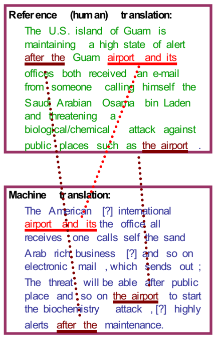
Reference (human) translation:
The U.S. island of Guam is maintaining a high state of alert after the Guam airport and its offices both received an e-mail from someone calling himself the Saudi Arabian Osama bin Laden and threatening a biological/chemical attack against public places such as the airport .

Machine translation:
The American [?] international airport and its the office all receives one calls self the sand Arab rich business [?] and so on electronic mail , which sends out ; The threat will be able after public place and so on the airport to start the biochemistry attack , [?] highly alerts after the maintenance.

- N-gram precision (score is between 0 & 1)
  - What percentage of machine n-grams can be found in the reference translation?
    - An n-gram is an sequence of n words
  - Not allowed to match same portion of reference translation twice at a certain n-gram level (two MT words *airport* are only correct if two reference words *airport*; can't cheat by typing out "the the the the the")
  - Do count unigrams also in a bigram for unigram precision, etc.
- Brevity Penalty
  - Can't just type out single word "the" (precision 1.0!)
- It was thought quite hard to "game" the system (i.e., to find a way to change machine output so that BLEU goes up, but quality doesn't)

BLEU Evaluation Metric

(Papineni et al, ACL-2002)



- BLEU is a weighted geometric mean, with a brevity penalty factor added.
  - Note that it's precision-oriented
- BLEU4 formula
  (counts n-grams up to length 4)

exp (1.0 * log p1 +
0.5 * log p2 +
0.25 * log p3 +
0.125 * log p4 –
max(words-in-reference/words-in-machine – 1, 0)

p1 = 1-gram precision
P2 = 2-gram precision
P3 = 3-gram precision
P4 = 4-gram precision

Note: only works at corpus level (zeroes kill it);
there's a smoothed variant for sentence-level

BLEU in Action

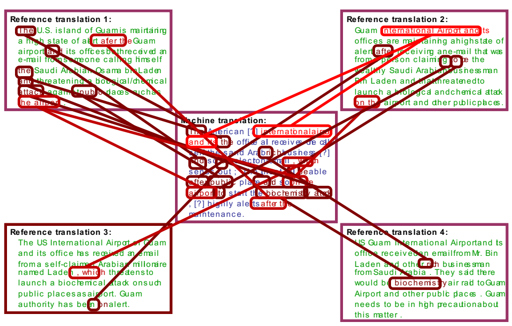
枪手被警方击毙。 **(Foreign Original)**

the gunman was shot to death by the police . **(Reference Translation)**

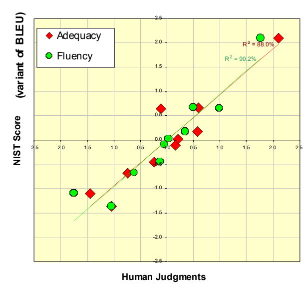
the gunman was police kill . **#1**
wounded police jaya of **#2**
the gunman was shot dead by the police . **#3**
the gunman arrested by police kill . **#4**
the gunmen were killed . **#5**
the gunman was shot to death by the police . **#6**
gunmen were killed by police ?SUB>0 ?SUB>0 **#7**
al by the police . **#8**
the ringer is killed by the police . **#9**
police killed the gunman . **#10**

green = 4-gram match (good!)
red = word not matched (bad!)

Multiple Reference Translations



Initial results showed that BLEU predicts human judgments well



slide from G. Doddington (NIST)

Automatic evaluation of MT

- People started optimizing their systems to maximize BLEU score
  - BLEU scores improved rapidly
  - The correlation between BLEU and human judgments of quality went way, way down
  - StatMT BLEU scores now approach those of human translations but their true quality remains far below human translations
- Coming up with automatic MT evaluations has become its own research field
  - There are many proposals: TER, METEOR, MaxSim, SEPIA, our own RTE-MT
  - TERpA is a representative good one that handles some word choice variation.
- MT research **requires** *some* automatic metric to allow a rapid development and evaluation cycle.

Lecture Plan

1. Searching for the best translation: Decoding [3:00–3:05]
2. MT Evaluation [3:05–3:20]
3. **Phrase-Based Statistical MT**
   a) **Introduction** [3:20–3:25]
   b) Building a phrase table [3:25–3:35]
   c) Log-linear models for scoring hypotheses [3:35–3:50]
   d) Phrase-based decoders [3:50–4:10]
   e) Training machine learning models [4:10–4:15]
4. Extra time [4:15–4:20]

MT Problems to Address: Flaws of Word-based MT

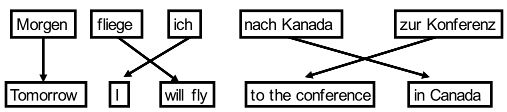
- The funny asymmetry of IBM models
  - You can't have multiple English words for one French word
  - IBM models can do one-to-many (fertility) but not many-to-one
- Adding more features for better translation quality
- Working with larger chunks than just words
  - Phrase-based systems
    - "real estate", "note that", "interested in"
    - There's a lot of multiword idiomatic language use

MT Problems to Address: Linguistic structure

- Syntactic Transformations
  - Verb at the beginning in Arabic
    - Translation model penalizes any proposed re-ordering
    - Language model may not strong enough to force the verb to move to the right place
- Hey, what about some linguistic structure to help translation?
- These issues point to hierarchical, syntactic or grammar-based systems
  - See, e.g., Chiang (2005) Hiero reading
  - Unfortunately, we won't have time to discuss these today

[Koehn et al, 2003]

Phrase-Based Statistical MT: The Pharaoh/Moses Model



- Foreign input segmented into phrases
  - "phrase" is any subsequence of words – not a linguistic phrase
- Each phrase is probabilistically translated into English
  - P(to the conference | zur Konferenz)
  - P(into the meeting | zur Konferenz)
- Phrases are probabilistically re-ordered

See J&M or Lopez 2008 for an intro.

This is still pretty much the state-of-the-art!

Advantages of Phrase-Based

- Many-to-many mappings can handle non-compositional phrases
- Local context is very useful for disambiguating
  - "interest rate" → …
  - "interest in" → …
- The more data, the longer the learned phrases
  - Sometimes whole sentences

Lecture Plan

1. Searching for the best translation: Decoding [3:00–3:05]
2. MT Evaluation [3:05–3:20]
3. Phrase-Based Statistical MT
   a) Introduction [3:20–3:25]
   b) Building a phrase table [3:25–3:35]
   c) Log-linear models for scoring hypotheses [3:35–3:50]
   d) Phrase-based decoders [3:50–4:10]
   e) Training machine learning models [4:10–4:15]
4. Extra time [4:15–4:20]

How to Learn the Phrase Translation Table?

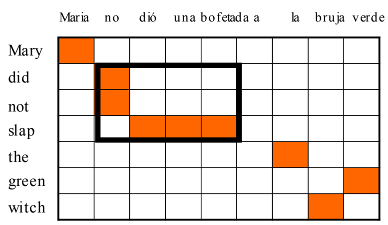
- Main method: "alignment templates" (Och et al, 1999)
- Start with "symmetrized" word alignment, build phrases from that.

This word-to-word alignment is a by-product of training a translation model like IBM-Model-3.

This is the best (or "Viterbi") alignment.

How to Learn the Phrase Translation Table?

- One method: “alignment templates” (Och et al, 1999)
- Start with word alignment, build phrases from that.

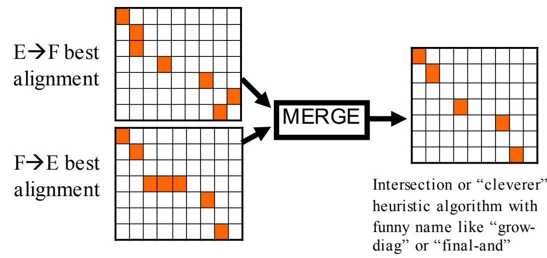


This word-to-word alignment is a by-product of training a translation model like IBM-Model-3.

This is the best (or “Viterbi”) alignment.

IBM Models are 1-to-Many

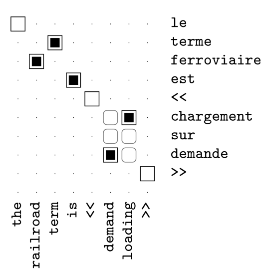
- Run IBM-style aligner both directions, then merge:



Intersection or “cleverer” heuristic algorithm with funny name like “grow-diag” or “final-and”

Symmetrization

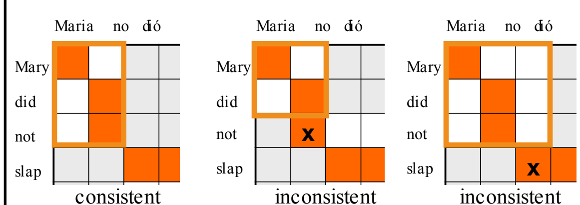
- Standard practice is to train models in each direction then to intersect their predictions
- Second model is basically a filter on the first
  - Precision jumps, recall drops
  - End up not guessing hard alignments



Model	P/R	AER
Model 1 E→F	82/58	30.6
Model 1 F→E	85/58	28.7
Model 1 AND	96/46	34.8

How to Learn the Phrase Translation Table?

- Collect all phrase pairs *that are consistent with the word alignment*



- Phrase alignment must contain all alignment points for all the words in both phrases!
- These phrase alignments are sometimes called *beads*

The phrase table becomes our translation model. How do we put goodness values on phrases?

开发 ||| the development ||| (1) ||| () (0) ||| -3.43 -2.72 -3.43 -2.76
开发 ||| the development of ||| (1) ||| () (0) () ||| -4.03 -2.72 -4.26 -5.31
开发 ||| development ||| (0) ||| (0) ||| -2.97 -2.72 -0.86 -0.95
开发 ||| development of ||| (0) ||| (0) () ||| -3.41 -2.72 -3.22 -3.50
进行 监督 ||| that carries out a supervisory ||| (1,2,3) (4) ||| () (0) (0) (0) (1) ||| 0.0 -3.68 -7.27 -21.24
进行 监督 ||| carries out a supervisory ||| (0,1,2) (3) ||| (0) (0) (0) (1) ||| 0.0 -3.68 -7.27 -17.17
监督 ||| supervisory ||| (0) ||| (0) ||| -1.03 -0.80 -3.68 -3.24
监督 检查 ||| supervisory inspection ||| (0) (1) ||| (0) (1) ||| 0.0 -2.33 -6.07 -4.85
检查 ||| inspection ||| (0) ||| (0) ||| -1.54 -1.53 -2.05 -1.60
尽管 ||| in spite ||| (1) ||| () (0) ||| -0.90 -0.50 -3.56 -6.14
尽管 ||| in spite of ||| (1) ||| () (0) () ||| -1.11 -0.50 -3.93 -8.68
尽管 ||| in spite of the ||| (1) ||| () (0) () () ||| -1.06 -0.50 -4.77 -10.50
尽管 ||| in spite of the fact ||| (1) ||| () (0) () () () ||| -1.18 -0.50 -6.54 -18.19
尽管 ||| spite ||| (0) ||| (0) ||| -0.78 -0.50 -3.34 -2.88
尽管 ||| spite of ||| (0) ||| (0) () ||| -0.96 -0.50 -3.71 -5.43
尽管 ||| spite of the ||| (0) ||| (0) () () ||| -0.90 -0.50 -4.54 -7.25
尽管 ||| spite of the fact ||| (0) ||| (0) () () () ||| -0.99 -0.50 -6.25 -14.93
尽管 ||| spite of the fact that ||| (0) ||| (0) () () () () ||| -1.03 -0.50 -6.35 -19.00

Lecture Plan

1. Searching for the best translation: Decoding [3:00–3:05]
2. MT Evaluation [3:05–3:20]
3. Phrase-Based Statistical MT
   a) Introduction [3:20–3:25]
   b) Building a phrase table [3:25–3:35]
   c) **Log-linear models for scoring hypotheses** [3:35–3:50]
   d) Phrase-based decoders [3:50–4:10]
   e) Training machine learning models [4:10–4:15]
4. Extra time [4:15–4:20]

The "Fundamental Equation of Machine Translation" (Brown et al. 1993)

$$\hat{e} = \underset{e}{\text{argmax}}\ P(e \mid f)$$

$$= \underset{e}{\text{argmax}}\ P(e) \times P(f \mid e) / P(f)$$

$$= \underset{e}{\text{argmax}}\ P(e) \times P(f \mid e)$$

What StatMT people do in the privacy of their own homes

$$\underset{e}{\text{argmax}}\ P(e \mid f) =$$

$$\underset{e}{\text{argmax}}\ P(e) \times P(f \mid e) / P(f) =$$

$$\underset{e}{\text{argmax}}\ P(e)^{1.9} \times P(f \mid e)$$ ... works better!

Which model are you now paying more attention to?

What StatMT people do in the privacy of their own homes

$$\underset{e}{\text{argmax}}\ P(e \mid f) =$$

$$\underset{e}{\text{argmax}}\ P(e) \times P(f \mid e) / P(f)$$

$$\underset{e}{\text{argmax}}\ P(e)^{1.9} \times P(f \mid e) \times 1.1^{\text{length}(e)}$$

Rewards longer hypotheses, since these are 'unfairly' punished by P(e)

What StatMT people do in the privacy of their own homes

$$\underset{e}{\text{argmax}}\ \underbrace{P(e)^{1.9} \times P(f \mid e) \times 1.1^{\text{length}(e)} \times KS^{3.7} \ldots}$$

Lots of knowledge sources vote on any given hypothesis. Each has a weight

"Knowledge source" = "feature function" = "score component".

Log-linear feature-based MT

$\text{argmax}_e\ 1.9 \times \log P(e) + 1.0 \times \log P(f \mid e) +$
$1.1 \times \log \text{length}(e) + 3.7 \times KS + \ldots$

$= \text{argmax}_e \sum_i w_i f_i$

So, we have two things:

- "Features" f, such as log language model score
- A weight w for each feature that indicates how good a job it does at indicating good translations

Numeric Features for Phrases: Log Phrase Pair Probabilities

- A certain phrase pair (f-f-f, e-e-e) may appear many times across the bilingual corpus.
- No EM training
- Simplest features are just relative frequency!
- $$P(\text{f-f-f} \mid \text{e-e-e}) = \frac{\text{count(f-f-f, e-e-e)}}{\text{count(e-e-e)}}$$
- P(e-e-e | f-f-f)
- Model 1 score P(f|e)
- Model 1 score P(e|f)

Other Numeric Features

- log language model score
- amount of "distortion" [reordering] in the translation hypothesis

- Other good ideas….
  – Average word frequency relative to source??

Categorical Features

- Categorical features are often represented by a symbol (a String)
- Mathematically, they're a feature whose value is 0 or 1
- Final feature value is number of time it fires in a hypothesis
  – Source phrase contains verb but target phrase doesn't: TRANS_NO_VERB
  – Source phrase contains period but target phrase doesn't: TRANS_NO_PERIOD
  – Target phrase contains the word "the": THE
  – Word part-of-speech trigam is X Y Z [feature for each X Y Z]

Feature weights

- **How to set the weights for features?**
  – Done for you, by optimization procedure
  – One way (which we look at later doing NER): maxent (softmax/logistic) models
  – The standard way in MT is "MERT" (minimum error rate training)
  – There are more recent proposals like "PRO" (pairwise ranking maxent optimization)
- **Basically, a positive weight if feature indicates good translation, negative if indicates a bad translation, magnitude is how good or bad (how positive/negative correlated)**

Feature gains … for PA1

- The core numeric features should get you a decent system
- Expect and be pleased by getting small incremental gains from features you devise
- 0.25 BLEU from a feature is good
- 0.5 BLEU from a feature is fantastic

Lecture Plan

1. Searching for the best translation: Decoding [3:00–3:05]
2. MT Evaluation [3:05–3:20]
3. Phrase-Based Statistical MT
   a) Introduction [3:20–3:25]
   b) Building a phrase table [3:25–3:35]
   c) Log-linear models for scoring hypotheses [3:35–3:50]
   d) **Phrase-based decoders** [3:50–4:10]
   e) Training machine learning models [4:10–4:15]
4. Extra time [4:15–4:20]

Phrase-based decoder

Input: lo haré | rápidamente | .

Translations: I'll do it | quickly | .

quickly | I'll do it | .

The decoder…

tries different segmentations,

translates phrase by phrase,

and considers reorderings.

Phrase-Based Translation

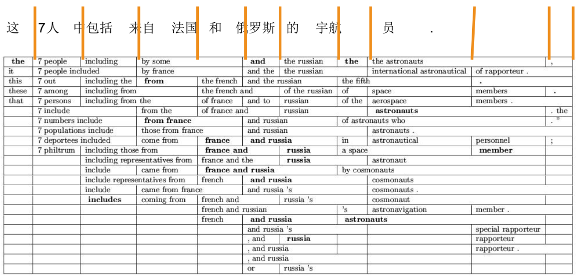


Table 1: #11# the seven - member crew includes astronauts from france and russia .

Scoring: Try to use phrase pairs that have been frequently observed.
Try to output a sentence with frequent English word sequences.

Phrase-Based Translation

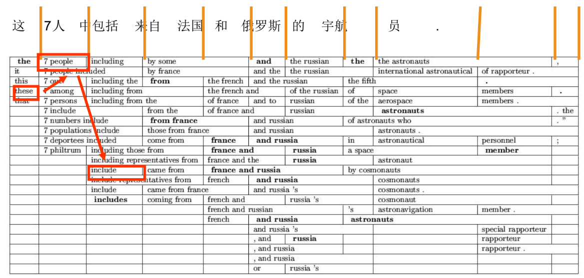


Table 1: #11# the seven - member crew includes astronauts from france and russia .

Scoring: Try to use phrase pairs that have been frequently observed.
Try to output a sentence with frequent English word sequences.

Phrase-Based Translation

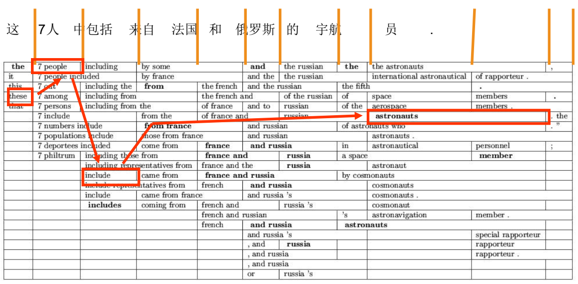


Table 1: #11# the seven - member crew includes astronauts from france and russia .

Scoring: Try to use phrase pairs that have been frequently observed.
Try to output a sentence with frequent English word sequences.

Phrase-Based Translation

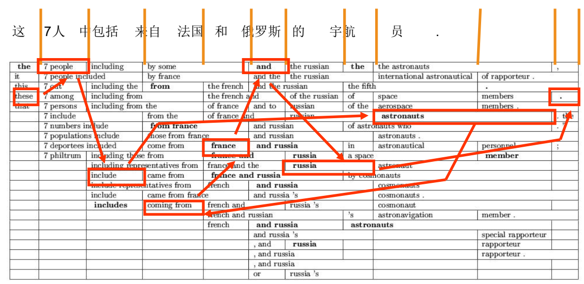
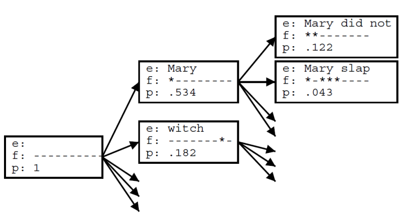


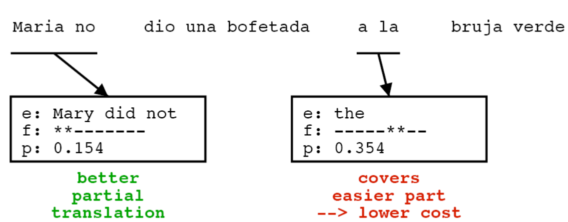
Table 1: #11# the seven - member crew includes astronauts from france and russia .

Scoring: Try to use phrase pairs that have been frequently observed.
Try to output a sentence with frequent English word sequences.

Non-Monotonic Phrasal MT



Pruning: Beams + Forward Costs



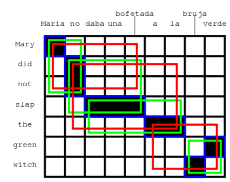
- Problem: easy partial analyses are cheaper
  - Solution 1: use beams per foreign subset
  - Solution 2: estimate forward costs (A*-like)

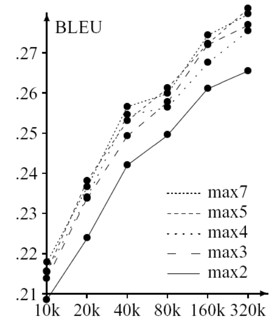
“Distortion”

- If our model were great, we’d let it rearrange phrases as much as it wants to
- In practice, that make translations **slow** and **bad**
- Commonly people put a hard limit on the size of reorderings
  - We do this in Phrasal in PA1

Phrase Size

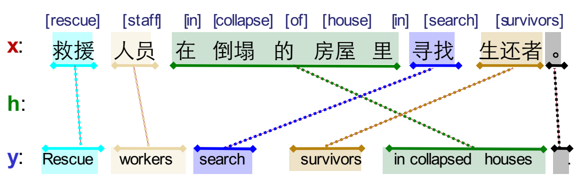
- Phrases help
  - But long ones often don’t help much
  - Why should this be?





Local syntax in phrase-based systems

[Och et al., 1999; Och and Ney; 2004]

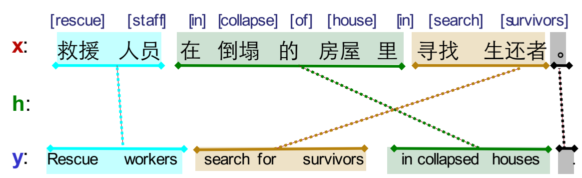


Phrases capture multi-word expressions,
help select correct function words,
and enable local reorderings.

51

Local syntax in phrase-based systems

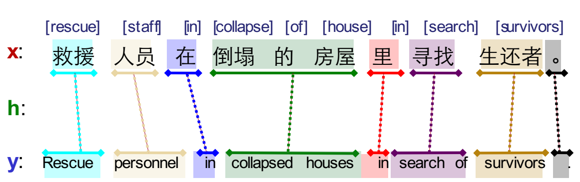
[Och et al., 1999; Och and Ney; 2004]



Phrases capture multi-word expressions,
help select correct function words (e.g., now also “for”),
and enable local reorderings.

52

Phrase-based models at test time



Google translate ’s actual output, 2010

Oct 2015 output: Rescue workers in collapsed buildings in search of survivors.

Long test phrases are often unseen in training.
Short phrases yield poor translations.
Need a more effective model to account for non-local dependencies!

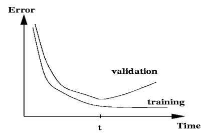
53

Lecture Plan

1. Searching for the best translation: Decoding [3:00–3:05]
2. MT Evaluation [3:05–3:20]
3. Phrase-Based Statistical MT
   a) Introduction [3:20–3:25]
   b) Building a phrase table [3:25–3:35]
   c) Log-linear models for scoring hypotheses [3:35–3:50]
   d) Phrase-based decoders [3:50–4:10]
   e) **Training machine learning models** [4:10–4:15]
4. Extra time [4:15–4:20]

Training models and pots of data

- The big danger when training models is that you **overfit** to what you are training on
  - The model correctly describes what happened to occur in particular data you trained on, but the patterns are not general enough patterns to be likely to apply to new data
- The way to monitor and avoid overfitting is using **independent** validation and test sets …



Training models and pots of data

- You build (estimate/train) a model on a **training set**.
- Commonly, you then set further hyperparameters on another, independent set of data, the **tuning set**
  - The tuning set is the training set for the hyperparameters!
- You measure progress as you go on a **dev set** (development test set or validation set)
  - If you do that a lot you overfit to the dev set so it's good to have a second dev set, the **dev2** set
- **Only at the end**, you evaluate and present final numbers on a **test set**
  - Use final test set **extremely** few times … ideally only once

Training models and pots of data

- The **train**, **tune**, **dev**, and **test** sets need to be completely distinct
- It is invalid to test on material you have trained on
  - You will get a falsely good performance. We usually overfit on train
- You need an independent tuning set
  - The hyperparameters won't be set right if tune is same as train
- If you keep running on the same evaluation set, you begin to overfit to that evaluation set
  - Effectively you are "training" on the evaluation set … you are learning things that do and don't work on that particular training set and using that
- To get a valid measure of system performance you need another untrained on, **independent** test set … hence dev2 and final test
  - Ideally, you only test on it once … definitely extremely few times