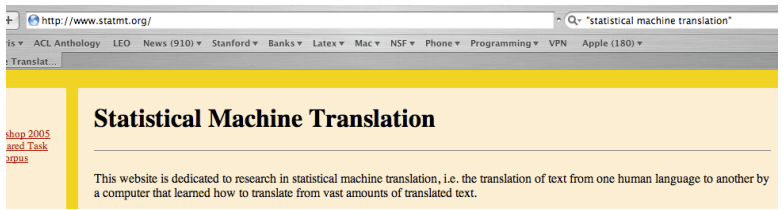
Translation

Statistical Machine Translation

Martin Kay

(with thanks to Kevin Knight and Philipp Koehn)

Stanford University
and the University of the Saarland



The Noisy Channel Model

A person wants to say e but, by the time it comes out, it has been corrupted by noise to become f. To make our best guess as to what was intended we reason about

1. The things English speakers are likely to say, and
2. The statistics of the corruption process

Statistical Machine Translation

- Find post probable English sentence given a foreign-language sentence
- Automatically align words and phrases within sentence pairs in a parallel corpus
- Probabilities are determined automatically by training a statistical model using the parallel corpus

Advantages of SMT

- Robustness
- Idioms
- Lexical ambiguity
- Minimal human effort

Disadvantages of (Current) SMT

- Finite-state equivalent
- No morphology
- Heavily domain-dependent

Probabilities

$$p(e|f)$$

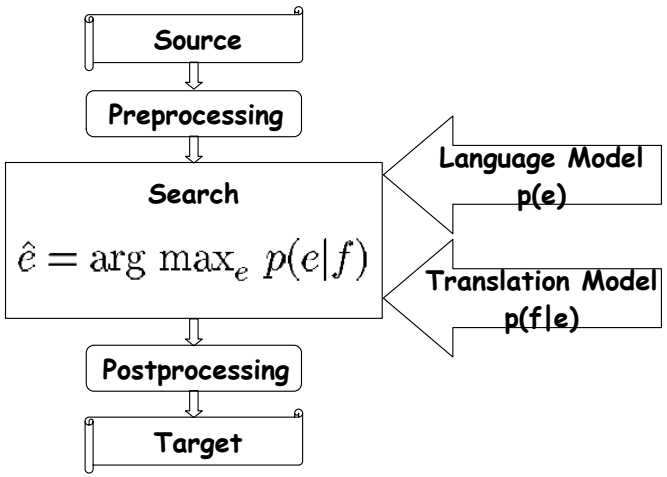
The translation Model

$$\hat{e} = \arg\max_e p(e|f)$$

$$p(e|f) = \frac{p(e)p(f|e)}{p(f)}$$

The Language Model

Bayes' rule



Find all the n English sentences on the web. If a sentence occurs k times, assign it a probability of k/n.

Problems

Big data base

Still does not contain many sentences

Language Model

- Component that ties to ensure that words come in the right order
- Some notion of grammaticality
- Standardly calculated with trigrams (Google uses 5-grams)
- Could use a statistical grammar (e.g. PCFG)

Trigram Language Model

p(the dog chased the cat)

≈

p(the | # #) *

p(dog | # the) *

p(chased | the dog) *

p(the | dog chased) *

p(cat | chased the)

Calculating the Probabilities

Unigrams

$$p(w_i) = \frac{\text{count}(w_i)}{\text{total words observed}}$$

Bigrams

$$p(w_i|w_{i-1}) = \frac{\text{count}(w_i w_{i-1})}{\text{count}(w_{i-1})}$$

Trigrams

$$p(w_i|w_{i-1}w_{i-2}) = \frac{\text{count}(w_i w_{i-1} w_{i-2})}{\text{count}(w_{i-1}w_{i-2})}$$

Backing Off

- Sparse counts are a big problem.
- If we haven't observed a sequence of words, then its count = 0.
- Because we are multiplying the n-gram probabilities to get the probability of a sentence, the whole probability = 0!

Smoothing (=Backing Off)

$$.8 \times p(w_i|w_{i-1}w_{i-2}) + \\ .15 \times p(w_i|w_{i-1}) + \\ .049 \times p(w_i) + \\ .001$$

Smoothing

let $n_k(s)$ be the probability estimate of s in the (k+1)-st order model.
Estimate probability of "abc" as $\lambda_3 n_3$("abc") + $\lambda_2 n_2$("bc") + $\lambda_1 n_1$("c") + λ_0 where $\lambda_3 + \lambda_2 + \lambda_1 + \lambda_0 = 1$.

Statistical Machine Translation

P(e | f) — The probability that e is an English translation of the given French sentence f.

Translation model Language model

$$= \frac{P(f \mid e)\, P(e)}{P(f)}$$

Translation Model

- p(f|e) — the probability of some foreign-language string given a hypothesesed English translation

$$p(f|e) = \frac{count(f,e)}{count(e)}$$

- Decompose sentences into smaller chunks, as in language modeling

$$p(f|e) = \sum_a p(a,f|e)$$

Two models are better than

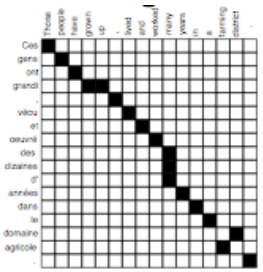
$$P(e \mid f)$$

Translation model — Language model

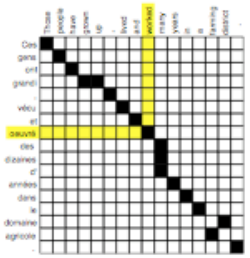
$$= \frac{P(f \mid e)\, P(e)}{P(f)}$$

... because they constrain one another, so neither has to take as much responsibility

Word Alignment



Alignment Probabilities

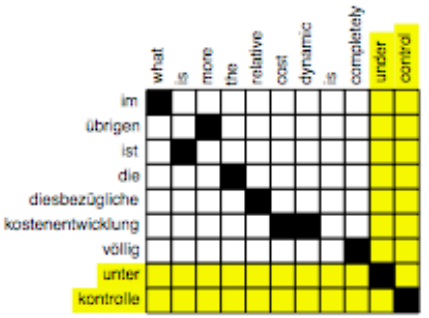


$$p(f|e) = \sum p(a, f|e)$$

$$p(a, f|e) = \prod_{j=1}^{m} t(f_j|e_i)$$

$$t(f_i|e_i) = \frac{\text{count}(f_j, e_i)}{\text{count}(e_i)}$$

Phrase Translation



Decoding

- Look up translations of every source phrase, using the phrase table
- Recombine the target language phrases that maximizes the translation model probability * the language model probability.
- This search over all possible combinations can get very large so we need to find ways of limiting the search space.

How does English become French?

- IBM Model 3
- Replace English word by French words that appear opposite them in a bilingual dictionary and then scramble their order

Translation can change length

- Each English word e_i has a fertility ϕ_i which gives the sequence number of the French word that will be generated for it.
- Each French word has a target position in its sentence which is a function of the position in the English sentence of the word it translates.

Translation as string rewriting

```
Hans ist nicht  in  dem Esszimmer          gegangen
```

Assign fertilities

```
1    0   1    1   1     2                    2
```

Apply fertilities

```
Hans     nicht in dem Esszimmer Esszimmer gegangen gegangen
```

Translate words

```
Hans     not   into the     dining    room      did      go
```

Permute

```
Hans did not go into the dining room
```

Parameters

- t(not, nicht): The probablity that German nicht will become English not.
- n(5 | 2) The probability that the English for a German word in position 2 of the sentence will be placed in position 5.
- p_1 The probability of a adding a spurious word, Add a word NULL at the beginning of the source sentence that can give rise to new (spurious) words in the target. These can be inserted anywhere after the other words have been arranged.

The Model-3 procedure

1. For each English word e_i choose a fertility ϕ_i with probability $n(\phi_i \mid e_i)$.
2. Choose the number ϕ_o of NULL words to insert with probability p1 + $\Sigma\phi_i$.
3. Let m = sum of all fertilities (including ϕ_o)
4. For i in (1..n) and k in in (1..ϕ_i), choose a French word τ_{ik} with probability $t(\tau_{ik} \mid e_i)$.
5. For i in (1..ϕ_o) choose a French position π_{ik} for a NULL translation with a total probability $1/\phi_o$.
6. Arrange and output the sentence

Parameters

n — fertilities
t — translations
d — position
p — NULL translations

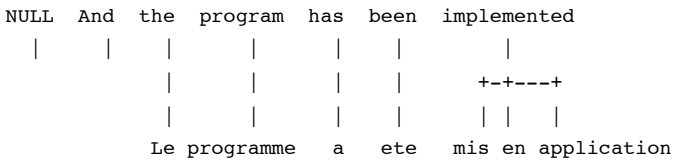
} 2 dimensons

1 dimension
Scaler

Parameter Values

Parameter values could be estimated easily on the basis of an English text and its translation into French where corresponding sentences have been aligned. An alignment of a pair of word strings is simply a mapping of the words of one string onto the words of the other. It can be represented by a vector A where A_i = j if the i-th English word is translated by the j-th French word. The frequncy of a translation, ordering, etc. is simply the number of alignments in which it is observed.

Allignment



[2, 3, 4, 5, 6, 6, 6]

P(e | f) = Allignments in which the pair appears / Allignments in which f appears

The distortion parameter

d(q | r, s, t) — probability that a word in original position r gives rise to a translation in position q when the respective lengths of the strings are s and t.

Alignments

Since alignments are not given, consider all k alignments for a given sentence pair, but add 1/k instead of 1 for the count for a given parameter.

Given values for the parameters, we can estimate probabilities of the aligments.

Given alignments, we can make better estimates of the parameters

⇒ **The EM-algorithm**

(Estimation Maximization)