A Weighted FSM implementation of alignment and translation models

Andrés Muñoz
David Álvarez-Melis

1Courant Institute, NYU

December 17, 2012
1 Background
2 Approach
3 Alignment Model
4 Translation Model
5 Experiments
6 Conclusion
Background
Source language sentences: \( f = f_1 \ldots f_m \), Target language sentences: \( e = e_1 \ldots e_l \).

**Task 1:** Given pairs \((e, f)\) of sentences, find the most likely alignment in the target language.

**Task 2:** Given sentences in source language, generate a translation in the target language.

**Noisy channel approach:**

\[
p(e|f) \propto p(f|e) \cdot p(e)
\]

*Translation Model* | *Language Model*
Source language sentences: $f = f_1 \ldots f_m$, Target language sentences: $e = e_1 \ldots e_l$.

Task 1: Given pairs $(e, f)$ of sentences, find the most likely alignment in the target language.

Task 2: Given sentences in source language, generate a translation in the target language.

Noisy channel approach:

$$p(e|f) \propto p(f|e) \cdot p(e)$$

Translation Model

Language Model
Source language sentences: \( f = f_1 \ldots f_m \), Target language sentences: \( e = e_1 \ldots e_l \).

Task 1: Given pairs \((e, f)\) of sentences, find the most likely alignment in the target language.

Task 2: Given sentences in source language, generate a translation in the target language.

Noisy channel approach:

\[
p(e|f) \propto p(f|e) \cdot p(e)
\]

Translation Model

Language Model
Approach
The IBM Models

- Very hard to model $p(f_1 \ldots f_m|e_1 \ldots e_l, m)$ directly. Instead define $p(f_1 \ldots f_m, a_1 \ldots a_m|e_1 \ldots e_l, m)$, where $(a_1, \ldots, a_l)$ are alignment variables.
- $a_j = k \Rightarrow f_j$ is aligned to $e_k$.
- IBM2 uses the following model:

$$p(f_1 \ldots f_m, a_1 \ldots a_m|e_1 \ldots e_l, m) = \prod_{i=1}^{m} q(a_i|i, l, m) t(f_i|e_{a_i})$$

Alignment Probability
The IBM Models

- Very hard to model \( p(f_1 \ldots f_m|e_1 \ldots e_l, m) \) directly. Instead define \( p(f_1 \ldots f_m, a_1 \ldots a_m|e_1 \ldots e_l, m) \), where \((a_1, \ldots, a_l)\) are alignment variables.
- \( a_j = k \Rightarrow f_j \) is aligned to \( e_k \).
- IBM2 uses the following model:

\[
p(f_1 \ldots f_m, a_1 \ldots a_m|e_1 \ldots e_l, m) = \prod_{i=1}^{m} q(a_i|i, l, m) \cdot t(f_i|e_{a_i})
\]

Alignment Probability \hspace{10cm} Transition Probabilities
The IBM Models

- Very hard to model \( p(f_1 \ldots f_m|e_1 \ldots e_l, m) \) directly. Instead define \( p(f_1 \ldots f_m, a_1 \ldots a_m|e_1 \ldots e_l, m) \), where \((a_1, \ldots, a_l)\) are alignment variables.
- \( a_j = k \Rightarrow f_j \) is aligned to \( e_k \).
- IBM2 uses the following model:

\[
p(f_1 \ldots f_m, a_1 \ldots a_m|e_1 \ldots e_l, m) = \prod_{i=1}^{m} q(a_i|i, l, m) t(f_i|e_{a_i})
\]
- $t(f|e)$ are trained by using an EM algorithm.
- How to model $q(a_i|i, l, m)$?
- For simplicity, assume alignment probabilities depend only on relative position (e.g. not in the particular words)

$$q(a_i|i, l, m) = e^{-\alpha|i-j|\frac{l}{m}}$$

- Note that we penalize alignments that move words too far from their position in the source language.
- Intuition: Words tend to remain in the same part of the sentence when translated.
It possible to represent the IBM Model as a FSM?

How to implement a translation machine (no target sentence provided) based on this model?

How many Transducers/Automata are needed?

Can these implementations be done time/space efficient?
Alignment Model
The FSM ingredients:

- Automata encoding $e$ and $f$ $\rightarrow \mathcal{E}, \mathcal{F}$.

\[
\begin{array}{c}
\text{0} & \text{The} & \text{1} & \text{red} & \text{2} & \text{house} & \text{3} \\
\end{array}
\]

\[
\begin{array}{c}
\text{0} & \text{La} & \text{1} & \text{maison} & \text{2} & \text{rouge} & \text{3} \\
\end{array}
\]

- A flower transducer $\mathcal{T}$, with word translation probabilities $t(f_i|e_j)$. Transitions: $f_i : e_j / t(f_i|e_j)$.

\[
\begin{array}{c}
\text{0} & \text{f:e/t(fle)} \\
\end{array}
\]
An automaton $E_P$, the same length as $f$ with permutations of the words of $e$, and transition weights $e^{-\alpha|i-j/l|m}$. 

![Diagram of automaton with states and transitions]
The full alignment model cascade: $\mathcal{E}_{P} \circ \mathcal{T} \circ \mathcal{F}$

The result (candidate alignments) are projected into the target space, and a best path (over the tropical semiring) is found.

FSM code:

```
fsmcompile -s log -i fwords.syms <AutoFr.txt | ...
fsmcompose - translator.fsm | fsmcompose - ...
Align.fsm | fsmconvert -s tropical | ...
fsmbestpath - | fsmproject -2 - >Predicted.fsm
```

Result:

```
La:The/3.873 1 maison:house/2.618 2 rouge:red/0.851 3/0
```
Computing Alignment Error with Transducers

- Given the gold alignment of each sentence $a^*$, with each alignment scored as “Possible” or “Sure”, how to measure the accuracy of the model?
- Compute AER: Alignment Error Rate (Och and Ney, 2003).

$$AER(A, P, S) = 1 - \frac{|P \cap A| + |S \cap A|}{|S| + |A|}$$
Flower alignment error $\mathcal{K}$ transducer with 0-1 loss:

$\mathtt{0/0} \quad \mathtt{i:j/1} \quad \mathtt{i:i/0}$

Predicted $\mathcal{P}$ and Gold $\mathcal{G}$ alignments encoded as automata from $X = \{1, \ldots, m\}$ to $Y = \{1, \ldots, l\}$. To, if $f_i$ is aligned to $e_j$, the $i$-th transition is labeled $j$.

Compose $\mathcal{P} \circ \mathcal{K} \circ \mathcal{G}$, find bestpath, project, push costs and read cost as number of wrong alignments.
Translation Model
• Problem: Given a french sentence try to give the best translation in English.
• Use the transducer $\mathcal{T}$ to solve this problem
• Search space is too big (French words can translate to several words)
• Unknown size of target sentence.
• Nondeterministic automaton of size $> 10^6$ before composition with language model.
- Prune the automaton with plausible translations
- Compose with bigram (faster than trigram) model to get the top n sentences
- Allow permutations on said sentences and rescore using trigram model.
- Find best path.
Source sentence

\[ 0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \]

- cette
- maison
- est
- rouge
Plausible translations after pruning
Best 5 sentences after composition with bigram model
• Final translation

0  this/8.798  1  house/15.47  2  is/8.558  3  red/25.80  4  </s>/10.77  5/0
Experiments
Dataset: Transcribed debates of the Canadian House of Commons, French ↔ English. IBM2 trained with ≈ 50,000 sentence pairs.

Test set: 447 sentences, along with their gold alignments.

AER: 42.1%, Precision 58.93%, Recall 68.72. 30 min computation. Not very accurate!

In translation the typical automaton of possible translations has more than $10^6$ transitions and
Conclusion
Conclusion

- IBM models are extremely simplistic - Outdated.
- AER obtained is poor.
- Yet, they provide a clear conceptual interpretation of MT
- More recent approaches:

  Phrase-Based → Syntax-Based → Hierarchical Phrase-based

- Possible improvement: add fertility model, allow for “null” alignments to appear in source language.
