A Weighted FSM implementation of alignment and translation models

Andrés Muñoz [1]
David Álvarez-Melis [1]

[1]Courant Institute, NYU

December 17, 2012

Background

- Source language sentences: $f = f_1 \dots f_m$, Target language sentences: $e = e_1 \dots e_l$.
- 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:

$$p(e|f) \propto \boxed{p(f|e)} \cdot \boxed{p(e)}$$

Translation Model Language Model

- Source language sentences: $f = f_1 \dots f_m$, Target language sentences: $e = e_1 \dots e_l$.
- 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:

$$p(e|f) \propto \boxed{p(f|e)} \cdot \boxed{p(e)}$$

Translation Model

Language Model

- Source language sentences: $f = f_1 \dots f_m$, Target language sentences: $e = e_1 \dots e_l$.
- 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:

$$p(e|f) \propto \underbrace{\boxed{p(f|e)}}_{\textit{Translation Model}} \cdot \underbrace{\boxed{p(e)}}_{\text{Language Model}}$$

Approach

The IBM Models

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

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

Alignment Probability

The IBM Models

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

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

Alignment Probability

Transition Probabilities

The IBM Models

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

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

Alignment Probability

Transition Probabilities

- $t(f|e)$ 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)

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

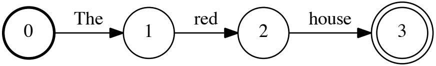
- Note that we penalize alignments that move words too far from their position in the source language.
- Intuition: Words tend to remain in the same part of the sentence when translated.

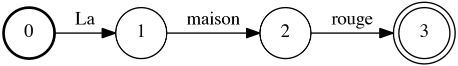
- It possible to represent the IBM Model as a FSM?
- How to implement a translation machine (no target sentence provided) based on this model?
- How many Transducers/Automata are needed?
- Can these implementations be done time/space efficient?

Alignment Model

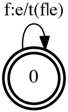
The FSM ingredients:

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

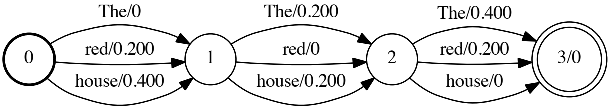




- A flower transducer $\mathcal{T}$, with word translation probabilities $t(f_i|e_j)$. Transitions: $f_i : e_j/t(f_i|e_j)$.



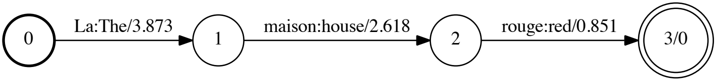
- An automaton $\mathcal{E}_P$, the same length as f with permuations of the words of e, and transition weights $e^{-\alpha\left|i-j\frac{l}{m}\right|}$.



- 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:

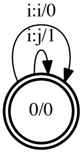


Computing Alignment Error with Transducers

- 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:



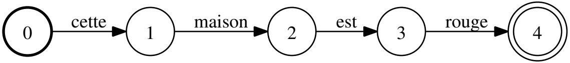
- Predicted $\mathcal{P}$ and Gold $\mathcal{G}$ alignments encoded as automata from $X = \{1, \ldots, m\}$ to $Y = \{1, \ldots, l\}$. To, if f_i is aligned to e_j, the i-th transition is labeled j.
- Compose $\mathcal{P} \circ \mathcal{K} \circ \mathcal{G}$, find bestpath, project, push costs and read cost as number of wrong alignments.

Translation Model

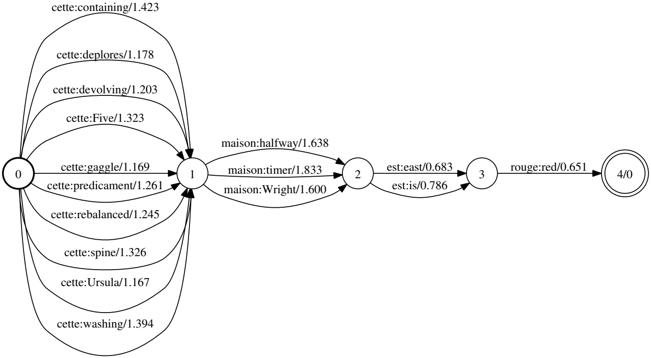
- Problem: Given a french sentence try to give the best translation in English.
- Use the transducer $\mathcal{T}$ to solve this problem
- Search space is too big (French words can translate to several words)
- Unknown size of target sentence.
- Nondeterministic automaton of size $> 10^6$ before composition 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.

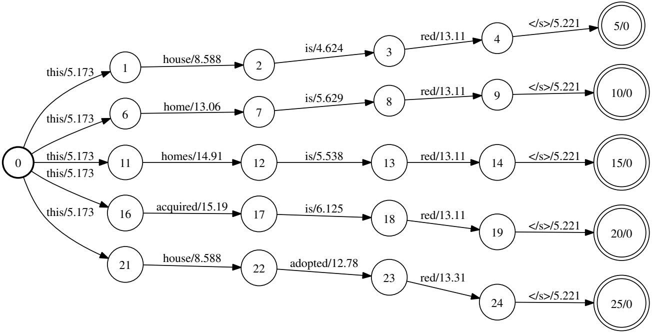
- Source sentence



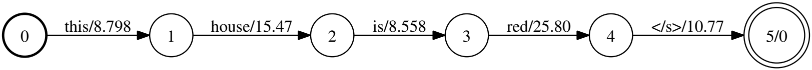
- Plausible translations after pruning



- Best 5 sentences after composition with bigram model



- Final translation



Experiments

- Dataset: Transcribed debates of the Canadian House of Commons, French $\leftrightarrow$ English. IBM2 trained with $\approx 50,000$ 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 $\rightarrow$ Syntax-Based $\rightarrow$ Hierarchical Phrase-based

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

- D. Jurafsky and J. H. Martin, *Speech and Language Processing - An introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Pearson Prentice Hall, second ed., 2008.
- K. Knight and Y. Al-Onaizan, *Translation with finite-state devices*, in Proceedings of the Third Conference of the AMTA, Springer-Verlag, 1998, pp. 421–437.
- S. Kumar and W. Byrne, *A weighted finite state transducer implementation of the alignment template model for statistical machine translation*, in Proceedings of HLT-NAACL, Association for Computational Linguistics, 2003, pp. 63–70.