Word Alignment

Dr. Mariana Neves
(adapted from the original slides of Prof. Philipp Koehn)

November 14th, 2016
Overview

- Further discussion on word alignment, such as problems and quality measurement
- Present a method on word alignment based on the IBM models
Given a sentence pair, which words correspond to each other?

michael  geht  davon  aus  dass  er  in  haus  bleibt

michael
assumes
that
he
will
stay
in
the
house
Word alignment

- It does not need to be one-by-one.
- Words can have multiple or no alignment points.
Is the English word does aligned to the German wohnt (verb) or nicht (negation) or neither?
How do the idioms *kicked the bucket* and *biss ins grass* match up? Outside this exceptional context, *bucket* is never a good translation for *grass*.
The better solution here is a phrasal alignment!
Sure alignments:
  - John to John

Possible alignments:
  - kicked to biss
  - the to im
  - bucket to Grass
Manually align corpus with *sure* \((S)\) and *possible* \((P)\) alignment points \((S \subseteq P)\)

Alignment Error Rate (AER): common metric for evaluation word alignments

\[
AER(S, P; A) = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}
\]

AER = 0: alignment \(A\) matches all sure, any number of possible alignment points
IBM Models create a **many-to-one** mapping
- words are aligned using an alignment function
- a function may return the same value for different input (one-to-many mapping)
- a function cannot return multiple values for one input (no many-to-one mapping)

Real word alignments have **many-to-many** mappings
Symmetrizing Word Alignments

Intersection of GIZA++ bidirectional alignments
The **intersection** usually contains good alignment points (high precision), but not all of them.

The **union** usually contains most of the desired align points (high recall), but also faulty points.

We want to explore the space between the two extremes:

- Take the all alignment points in the intersection (reliable).
- Add some of the points from the union (neighboring candidates), incrementally.
Growing heuristic

grow-diag-final(e2f,f2e)
  1: neighboring = {(-1,0),(0,-1),(1,0),(0,1),(-1,-1),(-1,1),(1,-1),(1,1)}
  2: alignment A = intersect(e2f,f2e); grow-diag(); final(e2f); final(f2e);
grow-diag()
  1: while new points added do
  2:    for all English word e ∈ [1...e_n], foreign word f ∈ [1...f_n], (e, f) ∈ A do
  3:      for all neighboring alignment points (e_new, f_new) do
  4:        if (e_new unaligned OR f_new unaligned) AND (e_new, f_new) ∈ union(e2f,f2e) then
  5:          add (e_new, f_new) to A
  6:      end if
  7:    end for
  8:  end for
  9: end while
final()
  1: for all English word e_new ∈ [1...e_n], foreign word f_new ∈ [1...f_n] do
  2:    if (e_new unaligned OR f_new unaligned) AND (e_new, f_new) ∈ union(e2f,f2e) then
  3:      add (e_new, f_new) to A
  4:  end if
  5: end for
Suggested reading

- Statistical Machine Translation, Philipp Koehn (section 4.5).