Gian Lecture 4.4: improved SMT

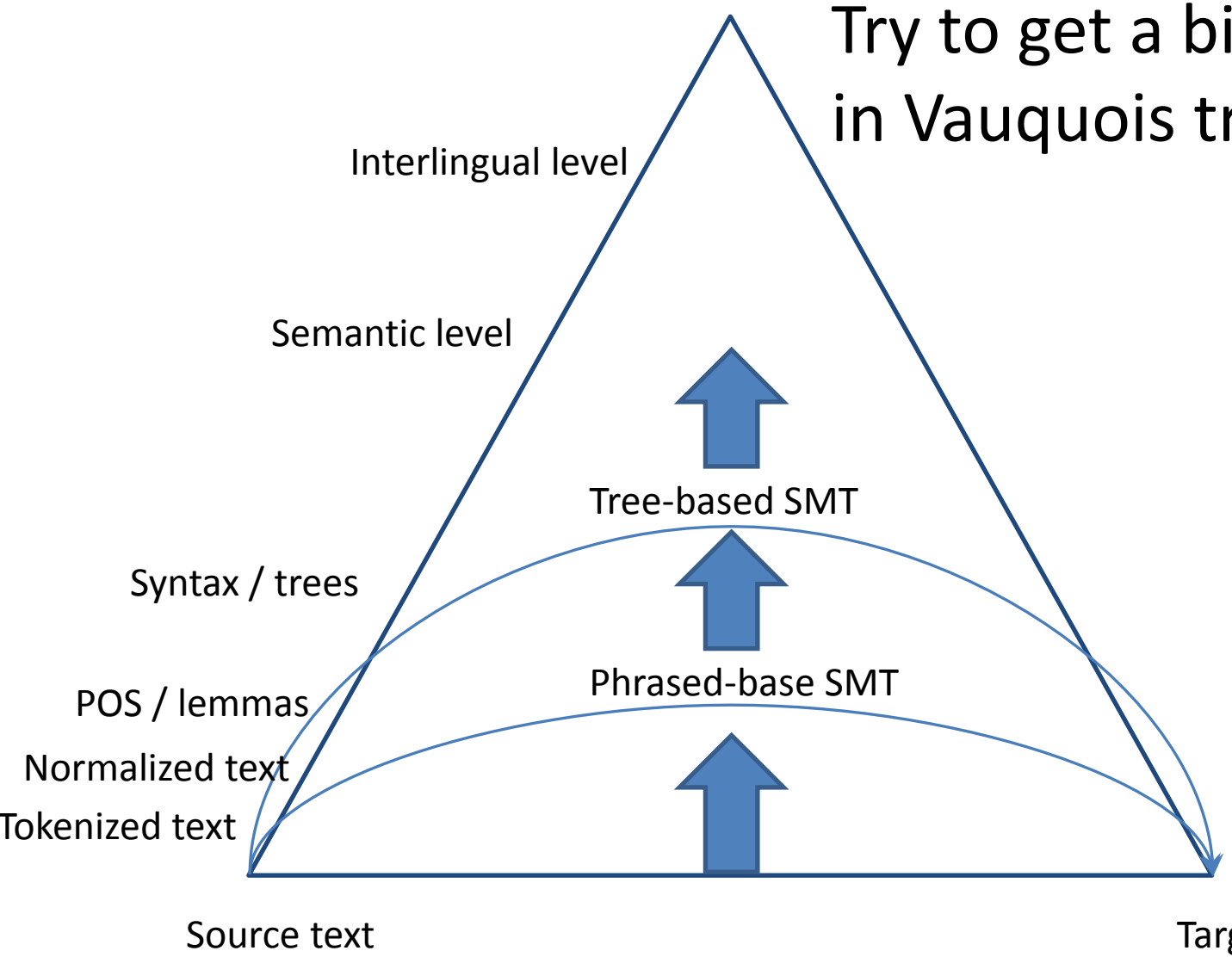
Factors (POS, lemmas)

Tree-based MT (Syntax-based / tree to string / syntax-enhanced…)

…

Improve PBMT?

Try to get a bit “higher” in Vauquois triangle



Tokenization

- Identify “tokens” in a sentence
- Even for “space-delimited” languages (identify punctuation)

  Tony ≠ Tony’s ≠ Tony. ≠ tony ≠ Tony, ≠ Tony's

– Compulsory for “unsegmented“ languages (Chinese, Thai etc.)

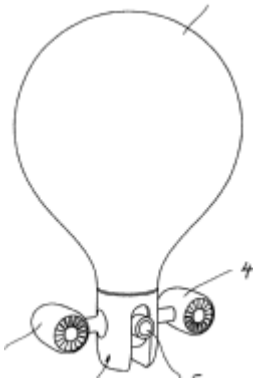
  “演奏音乐的娱乐装置” => “演奏 / 音乐 / 的 / 娱乐 / 装置”

Normalization

- Better generalization
- Casing, various options:
  - Keep original case
  - Lowercase all
  - True-case (keep “M. Larry Wall”)
- Decompounding:

  Gasballongetragener Flugroboter
  (lit. gas balloon carried flight robot)

  ➔Gas- ballon- getragener flug- roboter



Byte Pair Encoding

- Byte encoding: [Gage 1994]

Iteratively replace most frequent bytes pair by an unused byte

- In NLP: [Sennrich et al. 2016]

Iteratively replace most frequent bigram by a new token (the bigram)

Byte Pair Encoding minimal Python:

```
import re, collections

def get_stats(vocab):
  pairs = collections.defaultdict(int)
  for word, freq in vocab.items():
    symbols = word.split()
    for i in range(len(symbols)-1):
      pairs[symbols[i],symbols[i+1]] += freq
  return pairs

def merge_vocab(pair, v_in):
  v_out = {}
  bigram = re.escape(' '.join(pair))
  p = re.compile(r'(?<!\S)' + bigram + r'(?!\S)')
  for word in v_in:
    w_out = p.sub(''.join(pair), word)
    v_out[w_out] = v_in[word]
  return v_out

vocab = {'l o w </w>' : 5, 'l o w e r </w>' : 2,
         'n e w e s t </w>':6, 'w i d e s t </w>':3}
num_merges = 10
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
  print(best)
```

More information in the model

- Translate not only words, but also lemmas / part-of-speech / morphology / word class
- Better generalization
- Give more information to the model:
  - Reordering using more syntax
  - Language model using part-of-speech
  - Etc.

Factors in Moses

- Add information about each token : lemma / part-of-speech / morphology / word class
- Better generalization
- Give more information to the model:
  - Reordering using more syntax
  - Language model using part-of-speech
  - Etc.

  e.g. For a sentence “corruption flourishes” add lemma + part-of-speech:

  corruption|corruption|nn flourishes|flourish|nns

  See http://www.statmt.org/moses/?n=Moses.FactoredTutorial

Tree-based models

- http://*mt-class.org/jhu/slides/**lecture-syntax-based-models**.pdf*

- Syntax-based MT over the years…