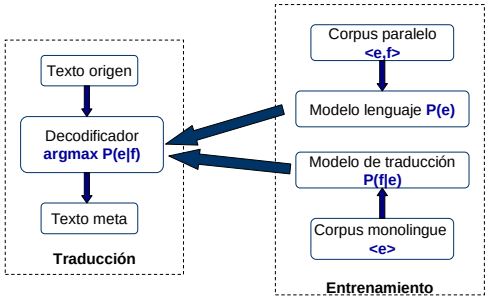
TA estadística



Modelos de la TAE

- El objetivo del modelo de lenguaje es asignar probabilidades altas a frases gramaticales en el LM
  - Básicamente prestado del área del reconocimiento del habla
  - Luego, veremos estos modelos en más detalle
- El modelo de traducción relaciona frases en LO a frases en LM
  - Describe cómo se originó f a partir de e
  - Para esto necesita "estudiar" relaciones previas, llamadas **alineamientos** de textos
- El proceso de descodificación implementa algún algoritmo de búsqueda que permita maximizar el producto de probabilidades
  - Algunos algoritmos usados son A*, Viterbi para HMMs, stack decoding, beam seach…

Alineamiento de textos (1/5)

- Dado un corpus paralelo bilingüe, *alinear* frases consiste en *conectar* cada frase en L1 a su equivalente semántico en L2
  - Frases de largo 1 = palabras o *unigramas*, de largo 2 = *bigramas*, largo n = *n-gramas*
- Además de su uso en TA, los textos alineados se usan para derivar diccionarios bilingües, BDs terminológicas, …
- Varias formas de alinear textos, por ej
  - Basadas en el largo de las frases medido en caracteres o palabras [Church&Gale '93; Dagan et al '93, Chang '93]
  - Basada en lexicones: usan la categoría semántica de palabras, POS tags, sinonimia [Ker et al '97; Wang '99]

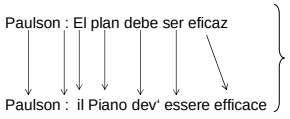
Alineamiento de textos (2/5)

LO = Español	LM = Italiano
<seg id="1"> La Bolsa de Praga terminó con menos puntos al final de la jornada </seg>	<seg id="1"> La Borsa di Praga alla chiusura del mercato ha avuto un crollo negativo.</seg>
<seg id="5"> Sin embargo , todavía no está ganado. </seg>	<seg id="5"> Ma non è ancora detta. </seg>
<seg id="9"> Según algunas fuentes, la votación sobre el plan se podría realizar en el Senado no antes del miércoles. </seg>	<seg id="9"> Al Senato, secondo alcune fonti, si potrebbe votare riguardo al piano probabilmente non prima di Lunedì. </seg>
<seg id="12"> Paulson: El plan debe ser eficaz </seg>	<seg id="12"> Paulson: il Piano dev'essere efficace </seg>

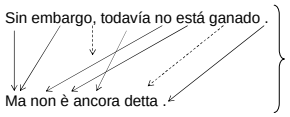
- Extracto corpus paralelo multilingüe Europarl [Koehn 005] [http://www.statmt.org/europarl/

Alineamiento de textos (3/5)

- Dado el corpus anterior, cuáles son los posibles alineamientos?



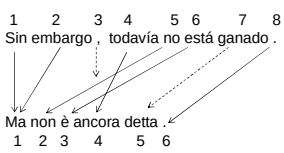
Ningún problema, conecciones 1-1



Algunos problemas: conecciones n-1, 1-Null y reordenamientos

Alineamiento de textos (4/5)

- Para formalizar la noción gráfica de alineamientos, se introduce una función a: i → j que mapea la palabra del LM en posición i a la palabra en LO en posición j
  - A(e,f) = conjunto de alineamientos de (f|e) = $U\{t_i \rightarrow s_j\}$
- Si cada palabra se conecta a una sola (ej 1 slide anterior), entonces A = {1→1, 2→2, 3→3, …}
- En cambio en el segundo ejemplo tenemos:



A = {1→1, 1→2, 2 →5, 3→6, 4→4, 5→7, 6→8}

Alineamiento de textos (5/5)

- El ejemplo anterior muestra algunos fenómenos como reordenamientos, alineamientos múltiples (1-n) y alineamientos 1-0 (no conectados a nada)
- También se pueden dar
  - n-n: varias palabras origen a varias palabras meta; ej expresiones idiomáticas
  - 0-1: en LM debe aparecer una palabra no derivada explícitamente de la frase origen; ej men cook → *los* hombres cocinan
  - n-1: una palabra origen genera varias en LM; ej however → sin embargo
- Además de la dificultad intrínseca de alinear textos, pueden haber factores relacionados a la colección del corpus
  - Ej anterior "no antes del miércoles" → "non prima di Lunedi"
- O a la utilización del corpus (ej plan → Piano o piano ?)

Modelos IBM [Brown et al '93]

- Los modelos de traducción IBM 1al 5 indican cómo computar P(f|e) en términos de A(e,f)

$$P(f|e) = \sum_{a \in A} P(f, a|e)$$

- Donde $P(f, a|e) = P(m|e)P(a|m,e)P(f|a,m,e)$
- Cada modelo presenta una forma distinta de computar P(f|e) como la prob conjunta P(F=f, A=a, E=e)

Modelo 1 de IBM

- Es el mas simple de los 5
- No considera el orden de las palabras, toma cada frase como una bolsa de palabras
- No modela alineamientos 1-n, sólo 1-1
- Reserva la posición 0 de *e* por si hay que insertar una palabra no generada de *f* (1-Null)
- Con estas restricciones es un modelo fácil de implementar y eficiente de computar
- Útil para inicializar parámetros que serán utilizados por otros modelos

Modelo 1 de IBM

- Las probabilidades para este modelo son:

$$P(f, a|e) = P(m|e)P(a|m,e)P(f|a,m,e)$$

Cuando genero f y a a partir de e (de largo l) puedo elegir el largo de f (m) con lo que conozco de e

Modela dónde conectar una posición de f de acuerdo al conocimiento adquirido sobre e y el largo de f

Modela la elección de palabras en f de acuerdo al conocimiento adquirido sobre e, el largo de f y la posición en e a la que se conecta la posición de f

Modelo 1 de IBM

- Las probabilidades para este modelo son:

$$P(f, a|e) = P(m|e)P(a|m,e)P(f|a,m,e)$$

donde $P(m|e) = \varepsilon(m|l)$ → ditrib de prob de largo de las cadenas e de largo l se genera de f de largo m

Modelo 1 de IBM

- Las probabilidades para este modelo son:

$$P(f, a|e) = P(m|e)P(a|m,e)P(f|a,m,e)$$

donde $P(m|e) = \varepsilon(m|l)$

$P(a|m,e) = (l+1)^{-m}$ → Como este modelo toma todos los alineamientos igualmente válidos, cada una de las m posibles conecciones tiene peso 1/(l+1)

Modelo 1 de IBM

- Las probabilidades para este modelo son:

$$P(f, a \mid e) = P(m \mid e) P(a \mid m, e) P(f \mid a, m, e)$$

donde $P(m|e) = \varepsilon(m|l)$

$$P(a \mid m, e) = (l+1)^{-m}$$

$$P(f \mid a, m, e) = \prod_{j=1}^{m} t(f_j \mid e_{aj})$$

Probabilidad de traducir La palabra f_j como e_i, dado el alineamiento a_j

Modelo 1 de IBM

- Las probabilidades para este modelo son:

$$P(f, a \mid e) = P(m \mid e) P(a \mid m, e) P(f \mid a, m, e)$$

dondeε $P(m|e) = \varepsilon(m|l)$

$$P(a \mid m, e) = (l+1)^{-m}$$

$$P(f \mid a, m, e) = \prod_{j=1}^{m} t(f_j \mid e_{aj})$$

Ecuación final

$$P(f \mid e) = \sum_{a \in A} \frac{\varepsilon}{(l+1)^m} \prod_{j=1}^{m} P(f_j \mid e_{a_j})$$

Modelo 1 de IBM

- Para este modelos los parámetros libres que se estiman durante el proceso de *entrenamiento son:*

$$\prod_{j=1}^{m} t(f_j \mid e_{aj}) \qquad \varepsilon(m|l)$$

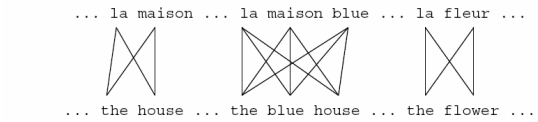
- Entrenamiento = inicialización parámetros + EM
  - EM = expectation-maximization

Aprendizaje no supervisado (EM)

- Queremos buscar el EMV de las probabilidades sobre el conjunto de datos de entrenamiento
  - En este caso las variables latentes u ocultas son alineamientos
- Algoritmo Expectation-Maximization (EM)
  - E-Step: predecir parámetros de acuerdo a los valores actuales
  - M-Step: re-estimar a partir de las predicciones
  - Iterar: con suerte converge a los EMV de los parámetros o a un máximo local
- Este proceso se considera la decodificación

16

EM algorithm



- Initial step: all alignments equally likely
- Model learns that, e.g., *la* is often aligned with *the*

Slide from Koehn 2008

EM algorithm

- After one iteration
- Alignments, e.g., between *la* and *the* are more likely

Slide from Koehn 2008

EM algorithm

- After another iteration
- It becomes apparent that alignments, e.g., between *fleur* and *flower* are more likely (**pigeon hole principle**)

Slide from Koehn 2008

EM algorithm

- Convergence
- Inherent hidden structure revealed by EM

Slide from Koehn 2008

EM algorithm


```
p(la|the) = 0.453
p(le|the) = 0.334
p(maison|house) = 0.876
p(bleu|blue) = 0.563
...
```

- Parameter estimation from the aligned corpus

Slide from Koehn 2008

IBM Model 1 and EM

- EM Algorithm consists of two steps
- **Expectation-Step**: Apply model to the data
  - parts of the model are hidden (here: alignments)
  - using the model, assign probabilities to possible values
- **Maximization-Step**: Estimate model from data
  - take assign values as fact
  - collect counts (weighted by probabilities)
  - estimate model from counts
- Iterate these steps until **convergence**

Probabilidad de alineamientos

Count de (f|e) = esperanza #de veces que e se conecte a f

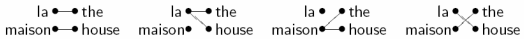
Slide from Koehn 2008

IBM Model 1 and EM

- **Probabilities**

$p(\text{the}|\text{la}) = 0.7$ $p(\text{house}|\text{la}) = 0.05$
$p(\text{the}|\text{maison}) = 0.1$ $p(\text{house}|\text{maison}) = 0.8$

- **Alignments**



$p(\mathbf{e}, a|\mathbf{f}) = 0.56$ $p(\mathbf{e}, a|\mathbf{f}) = 0.035$ $p(\mathbf{e}, a|\mathbf{f}) = 0.08$ $p(\mathbf{e}, a|\mathbf{f}) = 0.005$

$p(a|\mathbf{e}, \mathbf{f}) = 0.824$ $p(a|\mathbf{e}, \mathbf{f}) = 0.052$ $p(a|\mathbf{e}, \mathbf{f}) = 0.118$ $p(a|\mathbf{e}, \mathbf{f}) = 0.007$

- **Counts**

$c(\text{the}|\text{la}) = 0.824 + 0.052$ $c(\text{house}|\text{la}) = 0.052 + 0.007$
$c(\text{the}|\text{maison}) = 0.118 + 0.007$ $c(\text{house}|\text{maison}) = 0.824 + 0.118$

Slide from Koehn 2008

IBM Model 1 and EM: Pseudocode

```
initialize t(e|f) uniformly
do until convergence
  set count(e|f) to 0 for all e,f
  set total(f) to 0 for all f
  for all sentence pairs (e_s,f_s)
    for all words e in e_s
      total_s(e) = 0
      for all words f in f_s
        total_s(e) += t(e|f)
    for all words e in e_s
      for all words f in f_s
        count(e|f) += t(e|f) / total_s(e)
        total(f)   += t(e|f) / total_s(e)
  for all f
    for all e
      t(e|f) = count(e|f) / total(f)
```

Slide from Koehn 2008

Modelo 2

- Modelo 2 incorpora concepto de reordenamiento absoluto de palabras
  - La probabilidad de una conexión ***a*** depende de la posición en e,f y el largo de ambas
    - Es decir, depende del orden de las palabras en e y f
    - Son alineamientos más generales que el M 1

$$P(a \mid m,e) = (l+1)^{-m} \rightarrow P(a \mid m,e) = a(a_i \mid j,m,l)$$

- EM es calculado exactamente excepto que hay varios máximos locales
- Como Modelo 1 es caso particular de Modelo 2 → se usa para inicializar EM

Modelo 3

- Incorpora fertilidad y distorsión
  - Fertilidad: probabilidad de que una palabra en e se conecte a n palabras de f
  - Distorsión: prob que una palabra de e en posición i se conecte a una palabra en posición j (dado el largo de e y f)
    - Posiciones absolutas
- Trata conexiones 1-NULL con una distrib de proba uniforme
  - Potencialmente todas las palabras de f podrían no estar conectadas a alguna palabra en e
- Iterar sobre todos los alineamientos existentes es imposible
  - Técnica para calcular los ***a*** más probables → pegging

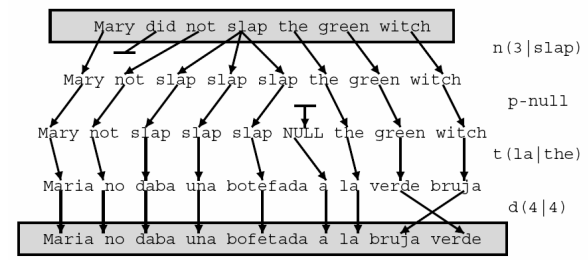
Pegging – Modelo 3

- Primero se obtiene un conjunto de mejores ***a*** entrenando Modelo 1 y luego se ajustan con Modelo 3 haciendo pequeños cambios
  - Movimiento: se cambia una conexión (j,i) a (j,i')
  - Swap: se intercambia un par (j,i)-(j',i') por (j,i')-(j',i)
  - El conj de ***a*** resultante se llama vecindad (neighborhood)
- Si uno de estos cambios aumenta la prob de traducción se agrega al conjunto de alineamientos

Deficiencia del modelo

- El principal problema es que asigna probabilidades a frases imposibles de generar
  - Por ejemplo Is possible
- Esto ocurre por las probabilidades de distorsión que generan distintas palabras en la misma posición
- Sin embargo, podría ser posible filtrarlas mas adelante

IBM Model 4



Slide from Koehn 2008

Modelos 1–5

- Modelo 1: bolsa de palabras
  - Algoritmo EM calculado exactamente (no aproximado)
  - Máximos locales únicos
- Modelo 2: alineamientos generales
  - Como en modelo 1 EM calculado exactamente
  - Agrega reordenamientos absolutos
- Modelo 3: agrega fertilidad: n(k | e)
  - De este en adelante solo aproximaciones de EM
  - Cuenta solo vecinos (Model 3–5)
  - Es deficiente (Model 3–4)
- Modelo 4: distorsion relativa d(k|e), clases de palabras
- Modelo 5: arregla el problema de la deficiencia

Limitaciones de modelos IBM

- Solo consideran mapeos de palabras 1-N
- Toda la informacion sintactica se limita a clases de palabras y distorsion
- No es posible hacer reodenamientos de larag distancia
- La gramaticalidad (fluency) de las traducciones depende exclusivamente del modelo de lenguaje usado