## El menú de hoy

- Modelos por frases
  - Sintácticos [Yamada 01]
  - Probabilidad conjunta [Marcu 02]
  - Basados en noisy-channel [Koehn 03]
  - Alignment template [Och 03]
  - Factored models [Koehn 07]
- Estimación de parámetros (MERT)
- Alineamiento de frases
- Práctico 4

## Modelos de traducción

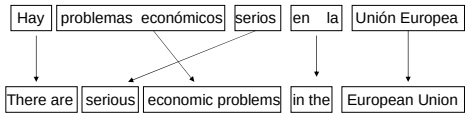
- Los modelos de traducción nacen en IBM y son la base de la TA pero tienen algunas limitaciones
  - Principalmente que solo modelan alineamientos 1-N de LM a LO
    - Es decir en la traducción EN-ES varias palabras en inglés pueden conectarse a una en español pero no al revés
  - Alineamientos mas generales N-M son difíciles de modelar dados los parámetros de estos modelos (n,d,p,t)
- Cuales son las consecuencias de estas restricciones? → textos mal o no traducidos

## Ejemplos

- DE-ES *Zahnarzttermin = cita/turno con el dentista*
- EN-FR *make with breadcrumbs = gratiner*
- DE-EN *Kindergarten = kindergarten* pero EN-ES *kindergarten = jardín de infantes*
- Dependiendo de los pares utilizados, con IBM estos pasajes podrían o no estar bien modelados

## Modelos por frases

- Modelos mas generales con mejor performance, capturan el contexto local de las palabras
  - Frases = unidades mayores a una palabra, no relacionado con estructura sintáctica de una oración
- Intuitivamente estos modelos funcionan así:




## El menú de hoy

- Modelos por frases
  - Sintácticos [Yamada 01]
  - Probabilidad conjunta [Marcu 02]
  - Basados en noisy-channel [Koehn 03]
  - Alignment template [Och 03]
  - Factored models [Koehn 07]
- Estimación de parámetros (MERT)
- Alineamiento de frases
- Práctico 4

## Diversidad de modelos

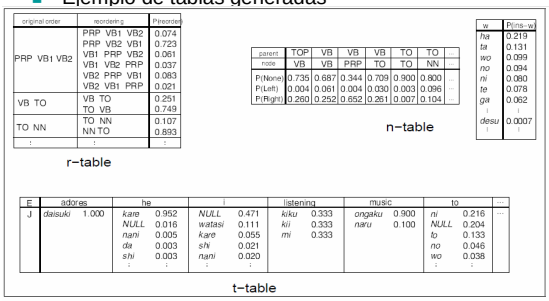
- Objetivo de un modelo de traducción es crear una tabla de traducción (t-table)
  - Usada durante la búsqueda (decoding)
- Existen muchos modelos pero pocos tienen alguna implementación disponible
- Veremos brevemente algunos modelos y con mas detalles los que se implementan en Moses
  - Decoder disponible libremente que usaremos en el próximo practico

## Sintácticos

- [Yamada&Knight 2001] proponen modelo sintáctico donde el árbol de parser se modifica para generar oración en LM sintácticos
  - Parecido a los modelos de lenguaje sintácticos de la clase pasada
- Entrenan su modelo sobre un corpus EN-JP y crean tablas de probabilidades para las operaciones sobre los árboles
  - Reordenamiento de nodos (sov eng/zh, svo jp/tr), inserción de nodos a izq o der, traducción

## Sintácticos

- Ejemplo de tablas generadas



| original order | reordering | Preorder |
|---|---|---|
| PRP VB1 VB2 | PRP VB1 VB2 | 0.074 |
| | PRP VB2 VB1 | 0.723 |
| | VB1 PRP VB2 | 0.061 |
| | VB1 VB2 PRP | 0.037 |
| | VB2 PRP VB1 | 0.083 |
| | VB2 VB1 PRP | 0.021 |
| VB TO | VB TO | 0.251 |
| | TO VB | 0.749 |
| TO NN | TO NN | 0.107 |
| | NN TO | 0.893 |
| : | : | : |

r-table

| parent | TOP | VB | VB | VB | TO | TO | |
|---|---|---|---|---|---|---|---|
| node | VB | VB | PRP | TO | TO | NN | |
| P(None) | 0.735 | 0.687 | 0.344 | 0.709 | 0.900 | 0.800 | ... |
| P(Left) | 0.004 | 0.061 | 0.004 | 0.030 | 0.003 | 0.096 | ... |
| P(Right) | 0.260 | 0.252 | 0.652 | 0.261 | 0.007 | 0.104 | ... |

n-table

| w | P(ins-w) |
|---|---|
| *ha* | 0.219 |
| *ta* | 0.131 |
| *wo* | 0.099 |
| *no* | 0.094 |
| *ni* | 0.080 |
| *te* | 0.078 |
| *ga* | 0.062 |
| : | : |
| *desu* | 0.0007 |
| : | : |

| E | adores | | he | | I | | listening | | music | | to | | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| J | *daisuki* | 1.000 | *kare* | 0.952 | *NULL* | 0.471 | *kiku* | 0.333 | *ongaku* | 0.900 | *ni* | 0.216 | ... |
| | | | *NULL* | 0.016 | *watasi* | 0.111 | *kii* | 0.333 | *naru* | 0.100 | *NULL* | 0.204 | |
| | | | *nani* | 0.005 | *kare* | 0.055 | *mi* | 0.333 | | | *to* | 0.133 | |
| | | | *da* | 0.003 | *shi* | 0.021 | | | | | *no* | 0.046 | |
| | | | *shi* | 0.003 | *nani* | 0.020 | | | | | *wo* | 0.038 | |
| | | | : | : | : | : | | | | | : | : | |

t-table

Table 1: Model Parameter Tables

## Modelos de probabilidad conjunta

- [Marcu&Wong 2002] proponen aprender pares de oraciones (e,f) en vez de aprender cómo e se genera desde f
  - Aprenden directamente de un corpus paralelo
- Idealmente si se aprende la prob conjunta P(e,f) se aprende también P(e|f) y P(f|e)
  - Consistente con modelos IBM pero usando frases
- Algunos problemas son la cantidad exponencial de posibles frases alineadas, el costo computacional
  - Solución: aprender frases de largo <6, probar con DP, …

## Modelo por frases [Koehn 03]

- Usa el canal ruidoso como los modelos IBM
  - Luego en FR-EN, modelamos P(f|e)*P(e) por Bayes
- Cada oración se segmenta en J frases $f_1^J = f_1, \ldots, f_J$ con distribución uniforme sobre todos los posibles segmentos
- Cada frase $f_i$ se traduce por una frase $e_i$
  - ahora la probabilidad de traducción dada por la distribución $\varphi(f_i|\, e_i)$

## Modelo por frases [Koehn 03]

- El reordenamiento de frases se modela con la prob de distorsión relativa $d(start_i, end_{i-1})$
  - *start* = posición en origen donde comienza la frase que se traduce por la i-ésima frase destino, análogo para *end*
  - Formalmente se estima con $\alpha^{|start_i, end_{i-1}|}$
- Se introduce un parámetro que calibra la longitud de las traducciones ω (costo)
- Durante el proceso de búsqueda (decoding) se intenta maximizar la ecuación dada por

## Modelo por frases [Koehn 03]

- $\underset{e}{\operatorname{argmax}}\, P(e|f)$

  $= \underset{e}{\operatorname{argmax}}\, P(f|e)*P(e)$

  $= \prod_{i=1}^{J} \varphi(f_i|\, e_i)\, \alpha^{|start_i, end_{i-1}|}\, P(e)\, \omega^{|e|}$

## Modelos log-linear [Och 2003]

- Una alternativa al canal ruidoso es modelar directamente P(e|f) y encontrar $\text{argmax}_e P(e_1^J, f_1^K)$
- Combinan funciones características $h_i(e_1^J, f_1^K)$ i=1..M que describen distintos aspectos de la traducción (*features*)
  - $\hat{e} = \text{argmax}_e \Pi_{i=1}^M \rho_i^{hi(e|f)}$
  - $\rho_i$ es el peso asignado a cada $h_i$

## Modelos log-linear

- Si elegimos features relativas a los modelos de lenguaje y traducción como $h_1(e,f) = \log_{\rho 1} P(e)$ , $h_2(e,f) = \log_{\rho 2} P(f|e)$
- Podemos reescribir la ecuación anterior:
  $\hat{e} = \text{argmax}_e \Pi_{i=1}^2 \rho_i^{hi(e|f)}$
  $= \text{argmax}_e \rho_1^{h1(e|f)} * \rho_2^{h2(e|f)}$
  $= \text{argmax}_e P(e) * P(f|e)$

→ modelos log-linear son una generalización del modelo anterior (noisy-channel)

## Log-linear phrase-based

- La ventaja de los modelos log-linear es que pueden asignarle pesos a sus componentes
  - Ej modelo de lenguaje o al de traducción
- Otra ventaja es la simplificación que permiten al tomar log a ambos lados
  $\log P(e|f) = \log \Pi_{i=1} \rho_i^{hi(e|f)}$
  $= \Sigma_{i=1} h_i(e|f) * \log \rho_i$
  - $\log \rho_i$ son constantes ($\lambda_i$) que reprsentan el peso de cada feature y suman 1

## Log-linear phrase-based

- Permite tomar cada componente del modelo por frases y darle un peso distinto
  - Se explicitan en vez de que cada probabilidad salga de los alineamientos como en IBM
  - → $\lambda_1 \varphi(f_i|\, e_i) + \lambda_2\, \alpha^{|starti,\, endi-1|} + \lambda_3 P(e) + \lambda_4\, \omega^{|e|}$
- Se puede usar EM para estimar el valor de los $\lambda_i$ → tuning del modelo

## Ejemplo tabla de traducción

a bilateral ||| des deux cotés de ||| 0.03125
a bilateral ||| des deux cotés ||| 0.031746
a bilateral ||| deux cotés de la tête ||| 1
anxious for more than six days ||| anxieux pendant plus de six jours ||| 0.5
anxious for more than six ||| anxieux pendant plus de six ||| 0.25
anxious for more than ten minutes ||| anxieux pendant plus de dix minutes ||| 0.5
anxious for more than ten months ||| anxieux pendant plus de dix mois ||| 1

## Modelos factorizados [Koehn 07]

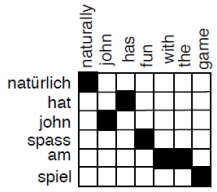
- Análogo a los modelos de lenguaje factorizados
  - Una palabra w es una colección de factores, se representa como un vector
    - Ej la = (“la”, artículo), gato = (“gato”, sustantivo), …
- Morfología es un buen ejemplo de la utilidad de los modelos factorizados
  - Si entrenamos el sistema solo con sustantivos en plural (ej casas, autos, perros, …) no podremos traducir ningún sustantivo en singular
- Por otro lado, idiomas ricos morfológicamente producen muchas formas por palabra
  - Necesitaría tener cada una de esas formas en el corpus

## Modelos factorizados

- Para traducir de factores a factores es necesario partir el proceso en dos
  - Mapeo de factores por separado (nivel de la frase)
  - Generación de palabra en LM a partir de factores
- Son una extensión de modelos por frases
  - Como cada paso es modelado con una feature encaja bien en modelos log-linear
  - Los pesos se obtienen con proceso de tuning

## Phrase-based training

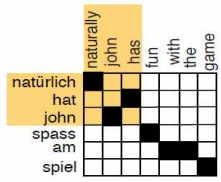
- Establish word alignment (GIZA++ and symmetrization)




Slide de Koehn 2009

## Phrase-based training

- Extract phrase

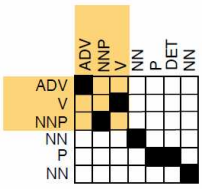



⇒ *natürlich hat john — naturally john has*

Slide de Koehn 2009

## Factored training

- Annotate training with factors, extract phrase




⇒ *ADV V NNP — ADV NNP V*

Slide de Koehn 2009

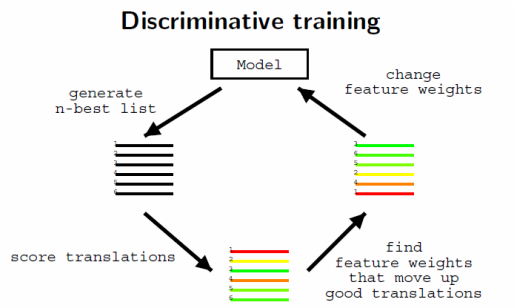
## El menú de hoy

- Modelos por frases
  - Sintácticos [Yamada 01]
  - Probabilidad conjunta [Marcu 02]
  - Basados en noisy-channel [Koehn 03]
  - Alignment template [Och 03]
  - Factored models [Koehn 07]
- Estimación de parámetros (MERT)
- Alineamiento de frases
- Práctico 4

## MERT [Och03]

- Minimum Error-Rate Training se usa para estimar automáticamente pesos de parámetros del MT
  - También podríamos setear los $\lambda_i$ a mano
- Usa el ML y alineamientos generados para entrenar MT y trata de mejorar (optimizar) los $\lambda_i$ usando EM
  - La idea mejorar la calidad total del sistema (overall)
- La optimización se hace respecto a una métrica automática de la calidad de la traducción
  - Mas adelante veremos evaluación en mas detalle

## MERT [Och03]




Slide de Koehn 2009

## MERT [Och03]

- En la practica
  - Se entrena el MT con los parámetros por default
  - Se ejecuta MERT sobre un corpus de desarrollo
    - Corpus de "dev" es una porción separada del corpus original
  - MERT genera una lista de pesos para cada feature que luego se pasa al decoder para la traducción del corpus de test
- Este proceso de tuning automático no siempre mejora la calidad de las traducciones
  - En realidad solo mejora un score automático (BLEU) que depende de los n-grams del corpus
  - → idea de proyecto: probar otras métricas en mert

## El menú de hoy

- Modelos por frases
  - Sintácticos [Yamada 01]
  - Probabilidad conjunta [Marcu 02]
  - Basados en noisy-channel [Koehn 03]
  - Alignment template [Och 03]
  - Factored models [Koehn 07]
- Estimación de parámetros (MERT)
- Alineamiento de frases
- Práctico 4

## Alineamiento de frases

- Es necesario aprender como se relacionan pares de frases para crear una tabla de traducción (modelo)
- Dependiendo del MT usado el alineamiento se puede hacer
  - A partir de palabras
  - Directamente del corpus
  - Por frases sintácticas …
- Una vez alineados los pares de frases se crea una distribución de probs

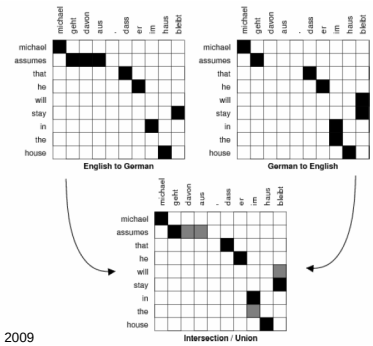
## Alineamiento de frases

- [Och 2003] genera alineamientos en ambas direcciones, $A_e$ y $A_f$
- Para aumentar la calidad de los alineamientos se pueden combinar
  - $A=A_e \cap A_f$ contiene solo alineamientos viterbi → son muy confiables = alta precisión
  - $A=A_e \cup A_f$ contiene todos los viterbi → alto cubrimiento
  - Incremental: tomar ∩ primero, agregar $(i,j) \in A_e$ ó $A_f$ si ni $f_j$ ni $e_i$ tiene alguna conexión en A o si se dan estas dos condiciones
    - (i,j) tiene vecino horizontal (i+/-1,j) o vertical (i,j+/-1)ϵ A
    - AU{(i,j)} no contiene alineamientos con vecinos ver/hor

## Alineamiento de frases

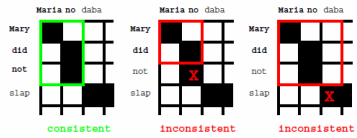
- La idea es extraer frases "consistentes" con los alineamientos por palabras dados por modelos IBM
  - Consistencia: palabras de un par de frases alineadas solo entre ellas (no fuera del par)
- Formalmente consistencia definida por
  **BP**$(f_1^J, e_1^J, A) = \{ ( f_j^{j+m}, e_i^{i+n} ) \}$:
  forall (i',j') in A : j<=j' <= j+m <-> i <= i' <= i+n
- Modificaciones de este algoritmo usan un paso adicional para agregar alineamientos "no consistentes"
  - Con algun criterio para que no introduzacn ruido

## Ejemplo de ∩/∪




Slide de Koehn 2009

## Consistent with word alignment




- **Consistent with the word alignment** :=

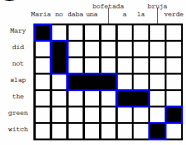
  phrase alignment has to *contain all alignment points* for all covered words

$$(\overline{e}, \overline{f}) \in BP \Leftrightarrow \quad \forall e_i \in \overline{e} : (e_i, f_j) \in A \rightarrow f_j \in \overline{f}$$
$$\text{AND} \quad \forall f_j \in \overline{f} : (e_i, f_j) \in A \rightarrow e_i \in \overline{e}$$

Slide de Koehn 2009

## Ejemplo

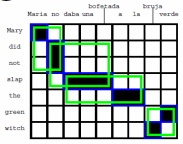
### Word alignment induced phrases



(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green)

Slide de Koehn 2009

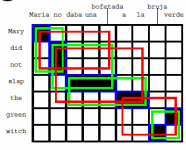
### Word alignment induced phrases



(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green),
(Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the),
(bruja verde, green witch)

Slide de Koehn 2009

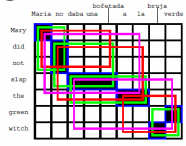
### Word alignment induced phrases



(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green),
(Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the),
(bruja verde, green witch), (Maria no daba una bofetada, Mary did not slap),
(no daba una bofetada a la, did not slap the), (a la bruja verde, the green witch)

Slide de Koehn 2009

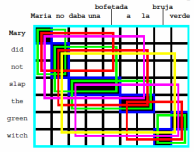
### Word alignment induced phrases



(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green),
(Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the),
(bruja verde, green witch), (Maria no daba una bofetada, Mary did not slap),
(no daba una bofetada a la, did not slap the), (a la bruja verde, the green witch),
(Maria no daba una bofetada a la, Mary did not slap the),
(daba una bofetada a la bruja verde, slap the green witch)

Slide de Koehn 2009

## Word alignment induced phrases (5)




(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green), (Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the), (bruja verde, green witch), (Maria no daba una bofetada, Mary did not slap), (no daba una bofetada a la, did not slap the), (a la bruja verde, the green witch), (Maria no daba una bofetada a la, Mary did not slap the), (daba una bofetada a la bruja verde, slap the green witch), (no daba una bofetada a la bruja verde, did not slap the green witch), (Maria no daba una bofetada a la bruja verde, Mary did not slap the green witch)

Slide de Koehn 2009

## Resumen

- Hoy vimos como modelar por frases en vez de palabras
  - Distintos modelos el que nos interesa es log-linear phrase-based [Koehn03]
- Vimos como se estiman los parámetros de cada función característica (tuning)
  - MERT [Och03]
- Vimos como se obtienen las frases alineadas para estos modelos

→ estamos en condiciones de probar la traducción por frases en un nuevo práctico!

## El menú de hoy

- Modelos por frases
  - Sintácticos [Yamada 01]
  - Probabilidad conjunta [Marcu 02]
  - Basados en noisy-channel [Koehn 03]
  - Alignment template [Och 03]
  - Factored models [Koehn 07]
- Estimación de parámetros (MERT)
- Alineamiento de frases
- Práctico 4

## Practico 4

- Usar el mismo corpus que en el practico 3 para entrenar modelo por frases
- Traducir el mismo corpus de test del practico 3
- Comparar los resultados
- Herramientas GIZA++, Moses, SRI-LM
  - http://www-speech.sri.com/projects/srilm/download.html
  - http://fjoch.com/GIZA++.html
  - http://www.statmt.org/moses/

- anxious for more than ||| anxieux pendant plus de ||| (0) (1) (2) (2,3) ||| (0) (1) (2,3) (3) ||| 0.32 0.0275453 0.888889 0.0570425 2.718

- Currently, five different phrase translation scores are computed:

- * phrase translation probability f(f|e)
- * lexical weighting lex(f|e)
- * phrase translation probability f(e|f)
- * lexical weighting lex(e|f)
- * phrase penalty (always exp(1) = 2.718)

## Bibliografía


- A Syntax-based Statistical Translation Model http://wing.comp.nus.edu.sg/acl/P/P01/P01-1067.pdf
- A Phrase-Based, Joint Probability Model for Statistical Machine Translation http://www.isi.edu/~marcu/papers/jointmt2002.pdf
- Effective Phrase Translation Extraction from Alignment Models http://acl.ldc.upenn.edu/acl2003/main/pdfs/Venugopal.pdf