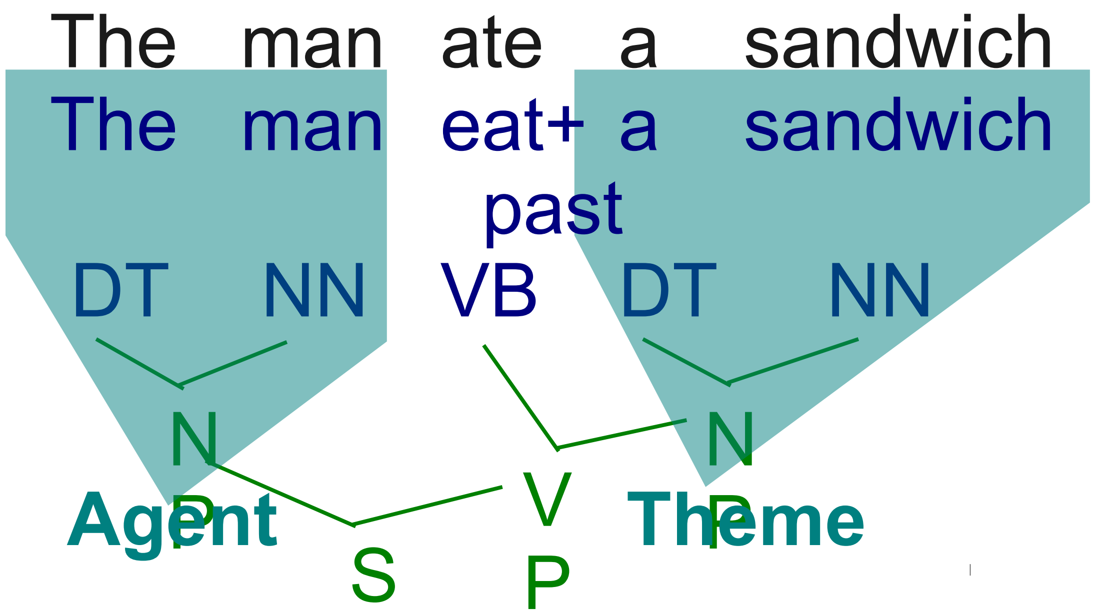
Part I

# Transfer Learning in Language

Part II

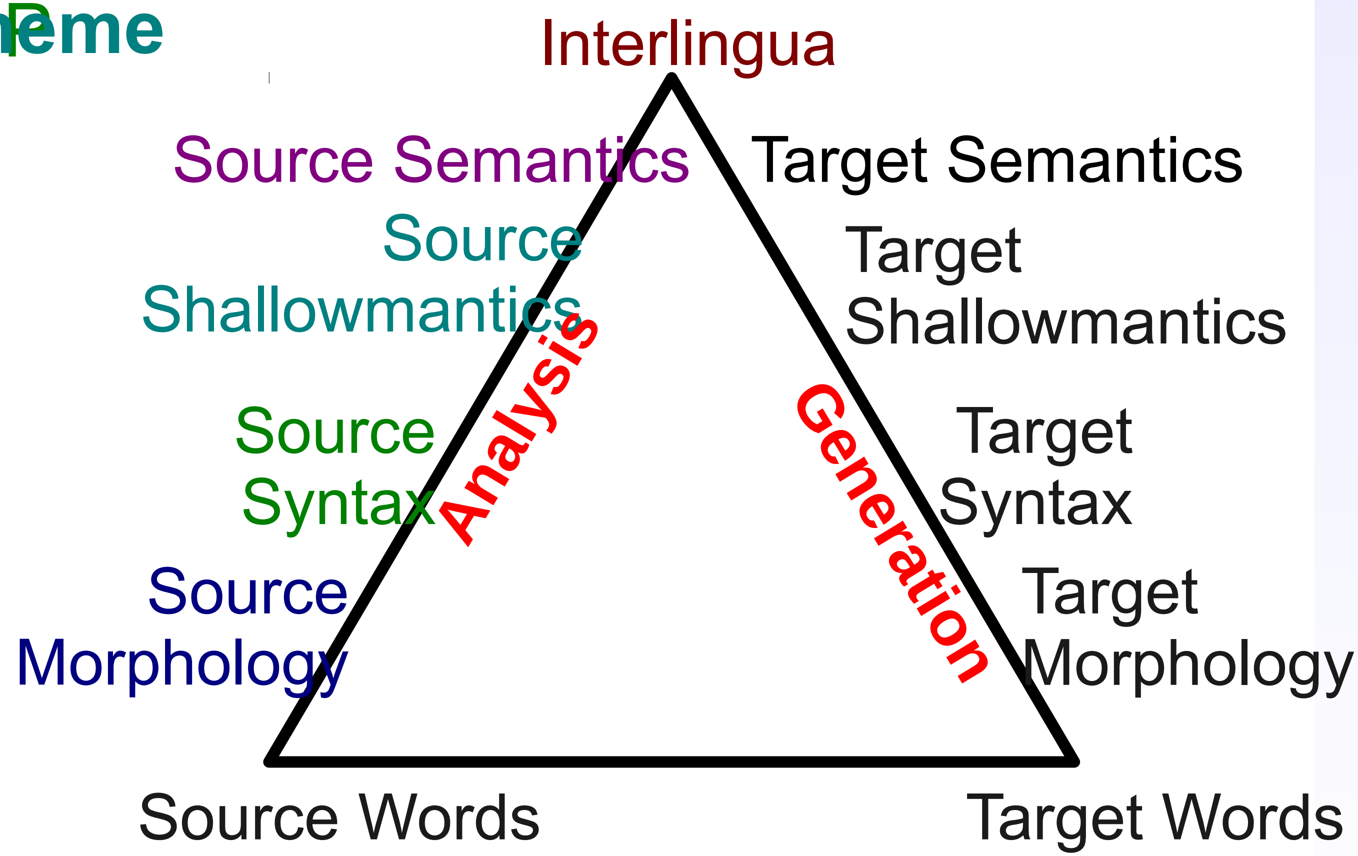
Hal Daumé III

# Typical NLP pipeline




Morphology
Tagging
Parsing
Role labeling
Interpretation

```
∃ a ∃ t ∃ e
  man(a) &
  sandwich(t) &
  eat(e,a,t) &
  past(e)
```



# Pipeline models break down (sorta)

- Tagging + Parsing + 0% / + 3%
- Parsing + Named Entities + 0.5% / + 4%
- Parsing + Role Identification + 0% / - 0.3%

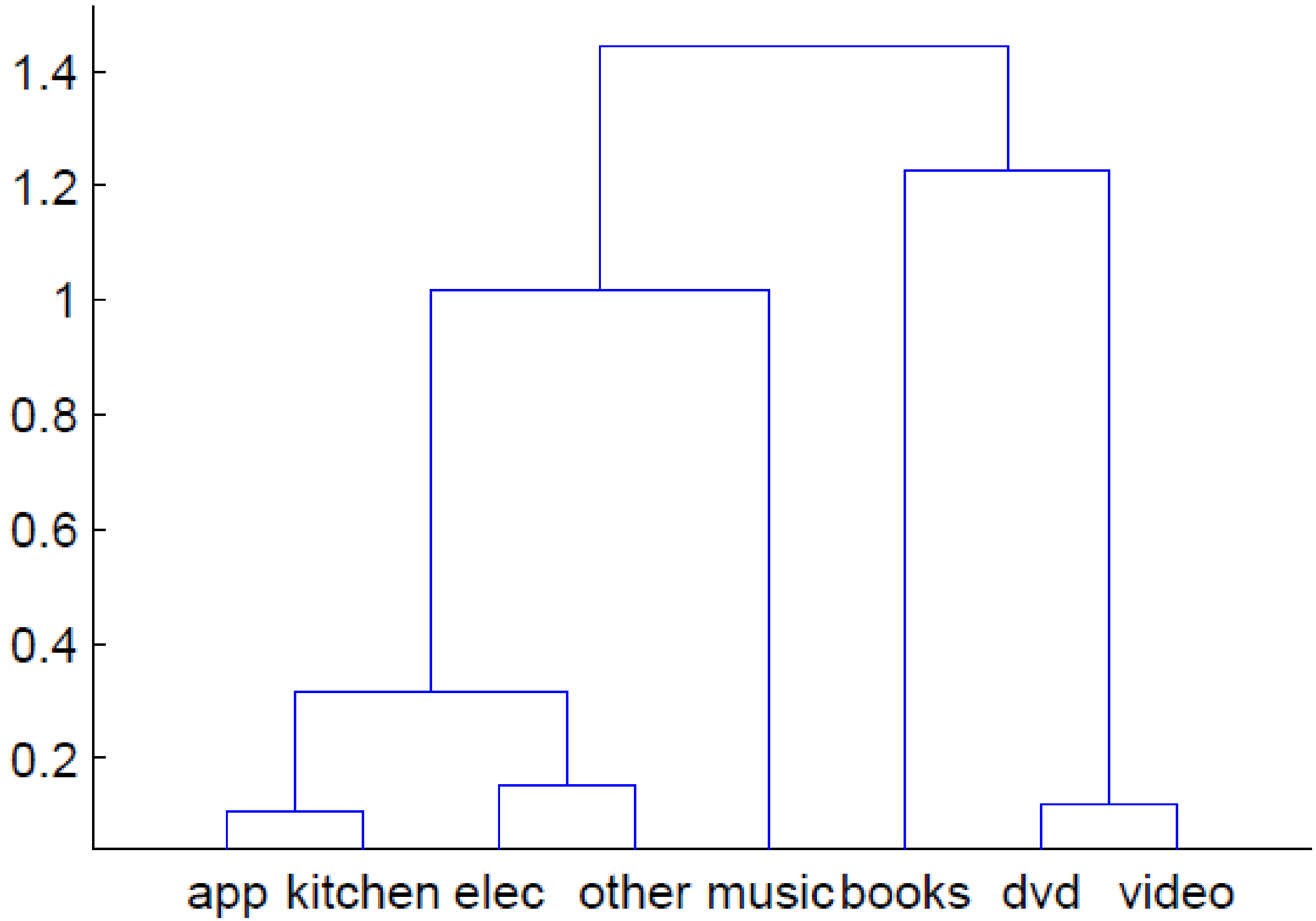
  *(upper bound: + 13% )*
- Named Entities + Coreference + 0.3% / + 1.3%

  *(upper bound: + 8% )*

Why? Maybe simpler model already has a lot of the fancier information? Maybe some of these tasks are more related than others?

# Tree-based model of task relatedness

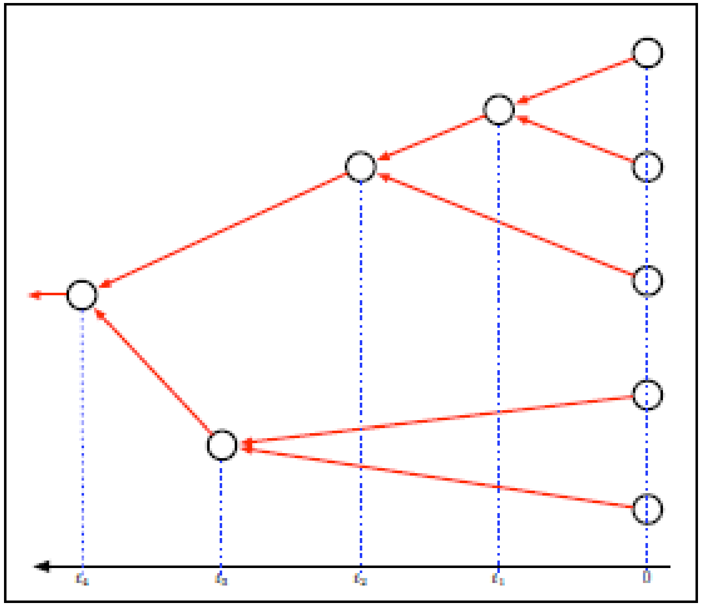
**Example 1:** Sentiment analysis for different product types




- In = review
- Out = rating
- Bag of words
- Crawled from Amazon

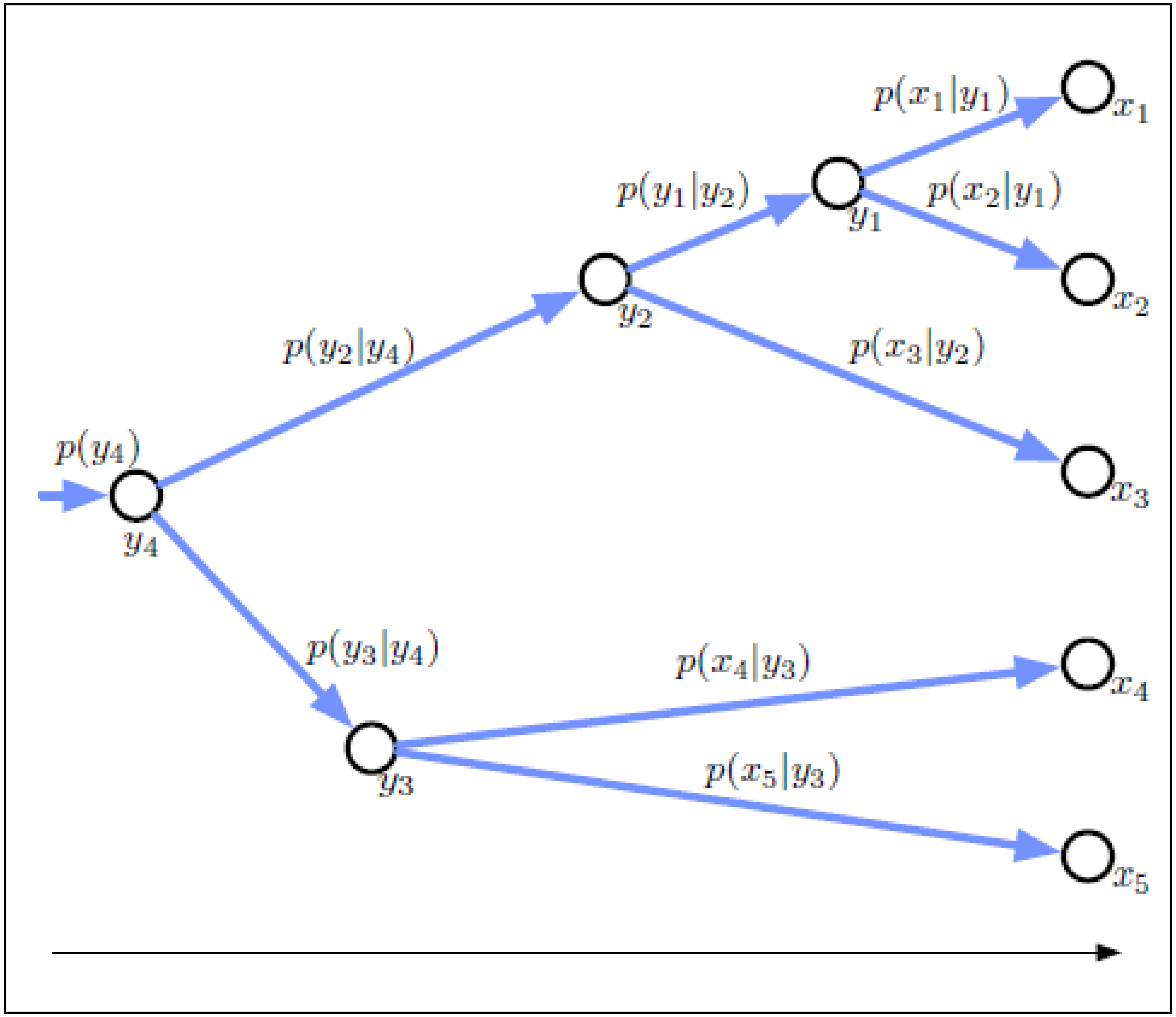
# A probabilistic model for trees

- Kingman's coalescent is the standard model for the genealogical history of populations.
- It is assumed that each organism has exactly one parent (haploid).
- Thus the genealogy of a population of organisms is a tree.
- Kingman's coalescent is a particularly elegant and simple distribution over genealogical trees of the population.

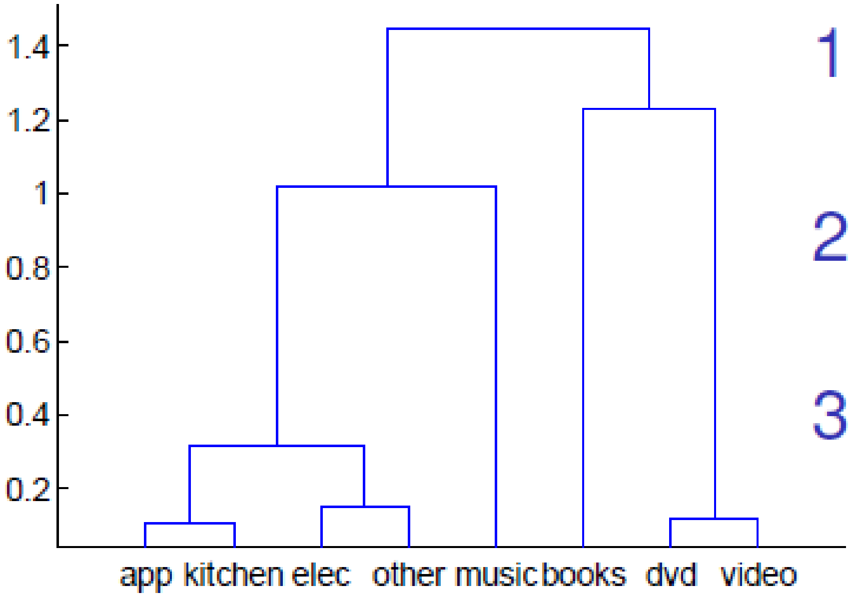


# From trees to priors...

Place a simple Markov process defined on the tree for $p(X|T)$ which evolves forward in time



# Inference




Inference by **EM**:

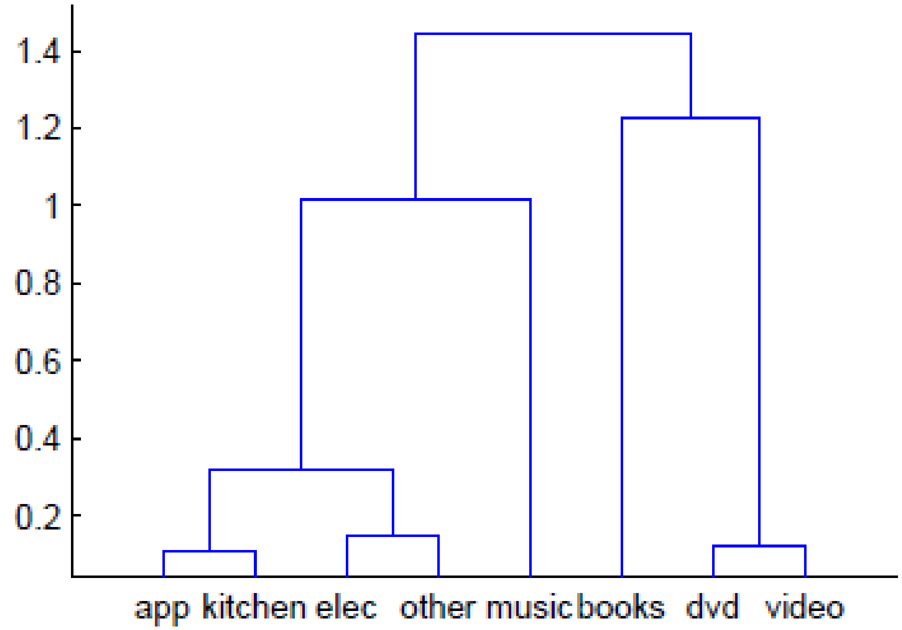
E: Compute expectations over $\boldsymbol{w}$s

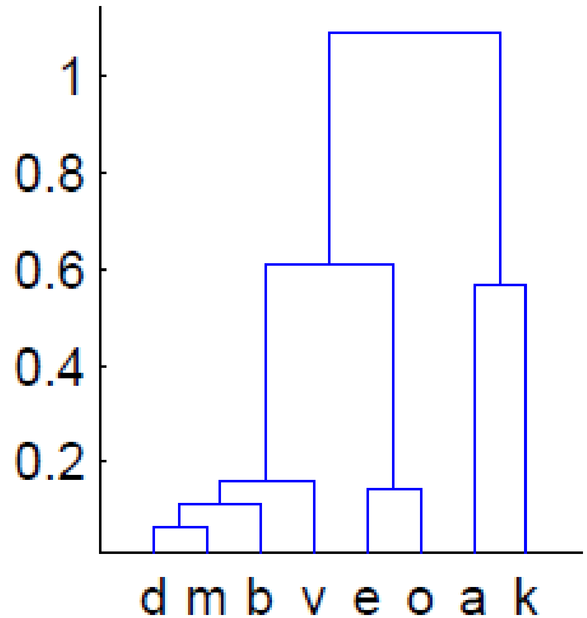
M: Maximize $(\pi, \delta, \Lambda)$, integrating out internal nodes

1. Choose global params: $(\boldsymbol{\mu}^{(0)}, \boldsymbol{\Lambda}) \sim \mathcal{N}or\mathcal{IW}(0, \sigma^2\mathbf{I}, D+1)$
2. Choose a tree structure: $(\pi, \boldsymbol{\delta}) \sim$ *Coalescent*
3. For each non-root $i \in \pi$:
   3.1 Choose $\boldsymbol{\mu}^{(i)} \sim \mathcal{N}or(\boldsymbol{\mu}^{(p_\pi(i))}, \delta_i\boldsymbol{\Lambda})$, where $p_\pi(i)$ is the parent of $i$
4. For each domain $k \in [K]$:
   4.1 Denote by $\boldsymbol{w}^{(k)} = \boldsymbol{\mu}^{(i)}$ where $i$ is the leaf corresponding $k$.
   4.2 For each example $n \in [N_k]$:
      4.2.1 Choose $\boldsymbol{x}_n^{(k)} \sim \mathcal{D}^{(k)}$.
      4.2.2 Choose $y_n^{(k)}$ by $F(\boldsymbol{w}^{(k)\top}\boldsymbol{x}_n^{(k)})$

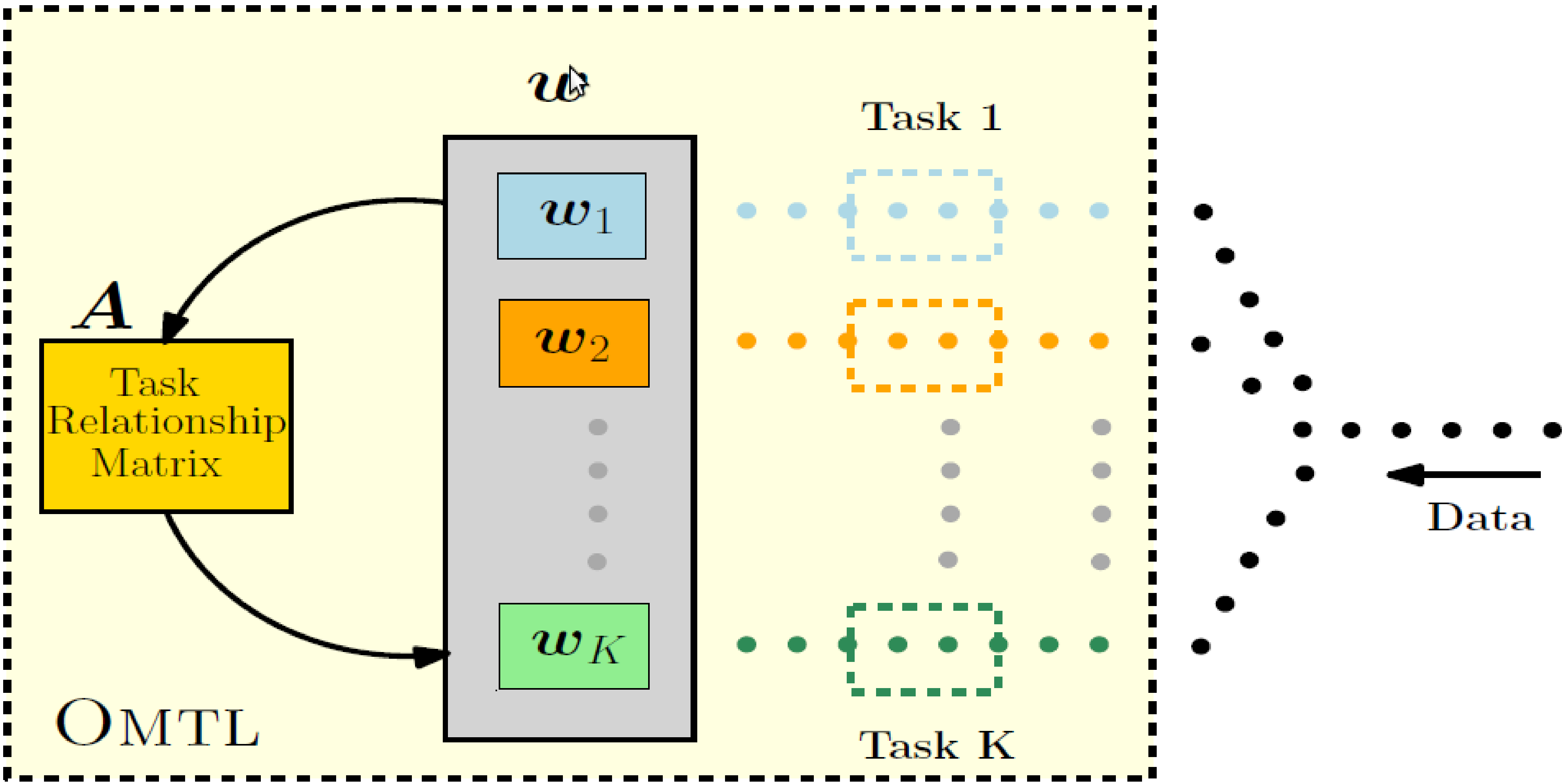
# Experiments (selected)

| Model | N=100 | N=6400 |
|---|---|---|
| Indp | 62.1% | 75.8% |
| Pool | 67.3% | 74.5% |
| FEDA | 63.6% | 75.7% |
| YaXue | 67.8% | 72.3% |
| Bickel | 68.0% | 72.5% |
| Coal: | | |
| Full | 72.2% | 80.5% |
| Diag | 71.9% | 80.4% |
| Data | 70.1% | 75.8% |



# Learning task relationships




[Saha, Rai, D., Venkatasubramanian, DuVall AIStats11]

# Task Relationship Learning

- *multitask instance* $\phi_t(x) \in \mathbb{R}^{Kd} = (\underbrace{0,\ldots,0}_{d(i_t-1) times} \quad x_t \quad \underbrace{0,\ldots,0}_{d(K-i_t) times})$
- compound weight vector $\boldsymbol{w}_s^T = (\boldsymbol{w}_1^T, \boldsymbol{w}_2^T, \ldots, \boldsymbol{w}_K^T) \in \mathbb{R}^{Kd}$
- update rules: $\boldsymbol{w}_s = \boldsymbol{w}_{s-1} + y_t(A \otimes I_d)^{-1}\phi_t$ and $s$ denotes the update number $(s < t)$

  where $\boldsymbol{A} = \begin{bmatrix} K & -1 & \ldots & -1 \\ -1 & K & \ldots & -1 \\ \ldots & \ldots & \ldots & \ldots \\ -1 & -1 & \ldots & K \end{bmatrix}$ and $\boldsymbol{A}^{-1} = \dfrac{1}{K+1}\begin{bmatrix} 2 & 1 & \ldots & 1 \\ 1 & 2 & \ldots & 1 \\ \ldots & \ldots & \ldots & \ldots \\ 1 & 1 & \ldots & 2 \end{bmatrix}$
- $K \times K$ **interaction matrix** $\boldsymbol{A}$ controls the updates
- update scheme:
  - **fixed** *full update* for the current task $i_t$
  - **fixed** *half update* for the remaining $(K-1)$ tasks

[Saha, Rai, D., Venkatasubramanian, DuVall AIStats11]

# Joint learning of relationships

- **key idea:** joint minimization of $\boldsymbol{A}$ and $\boldsymbol{w}$

  $\arg\min_{\boldsymbol{w}\in\mathbb{R}^{Kd}, \boldsymbol{A}\succ 0} \left[ D_w(\boldsymbol{w}||\boldsymbol{w}_s) + D_A(\boldsymbol{A}||\boldsymbol{A}_s) + \sum_1^t l_t(\boldsymbol{w}) \right]$

- in this work: hinge loss for $l_t(\boldsymbol{w})$, mahalanobis distance for $D_w(\cdot||\cdot)$, log-det divergence and von-neumann divergence for $D_A(\cdot||\cdot)$
- update rules after **alternating minimization**:
  - $\boldsymbol{w}_s = \boldsymbol{w}_{s-1} + y_t(\boldsymbol{A}_{s-1} \otimes \boldsymbol{I}_d)^{-1}\phi_t$
  - $\boldsymbol{A}_s = \boldsymbol{f}^{-1}\left( \boldsymbol{f}(\boldsymbol{A}_{s-1}) - \eta \ \mathbf{sym}\left(\nabla_{\boldsymbol{A}} \frac{1}{2} tr(\boldsymbol{W}_{s-1}\boldsymbol{A}\boldsymbol{W}_{s-1}^T)\right)\right)$

  where, $tr(\boldsymbol{W}_{s-1}\boldsymbol{A}\boldsymbol{W}_{s-1}^T) = \boldsymbol{w}_{s-1}^T(\boldsymbol{A} \otimes I_d)\boldsymbol{w}_{s-1}$, and $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K] \in \mathbb{R}^{d\times K}$
- **issue:** when to start updating $\boldsymbol{A}$ ? our approach:
  - initially learn $K$ independent classifiers
  - start updating $\boldsymbol{A}$ after reaching *priming duration* (EPOCH)

[Saha, Rai, D., Venkatasubramanian, DuVall AIStats11]

# Experimental Results (sample)

| Method | Accuracy (Standard Deviation) | | |
|---|---|---|---|
| | 20newsgroups | Sentiment | Spam |
| STL | 56.94(±3.32) | 66.31(±2.14) | 76.45(±1.56) |
| IPL | 75.20(±2.35) | 67.24(±1.40) | 91.02(±0.77) |
| CMTL | 73.14(±2.35) | 67.38(±1.82) | 90.17(±0.66) |
| OMTLLOG | **81.83(±0.46)** | **73.49(±0.53)** | **91.35(±1.12)** |
| OMTLVON | 76.51(±1.54) | 67.60(±0.83) | 91.05(±1.05) |

Accuracy for *full training data* (EPOCH $= 0.5$).

[Saha, Rai, D., Venkatasubramanian, DuVall AIStats11]

# Transfer Learning in Language

aka: why everything I've told you so far isn't useful for some problems...

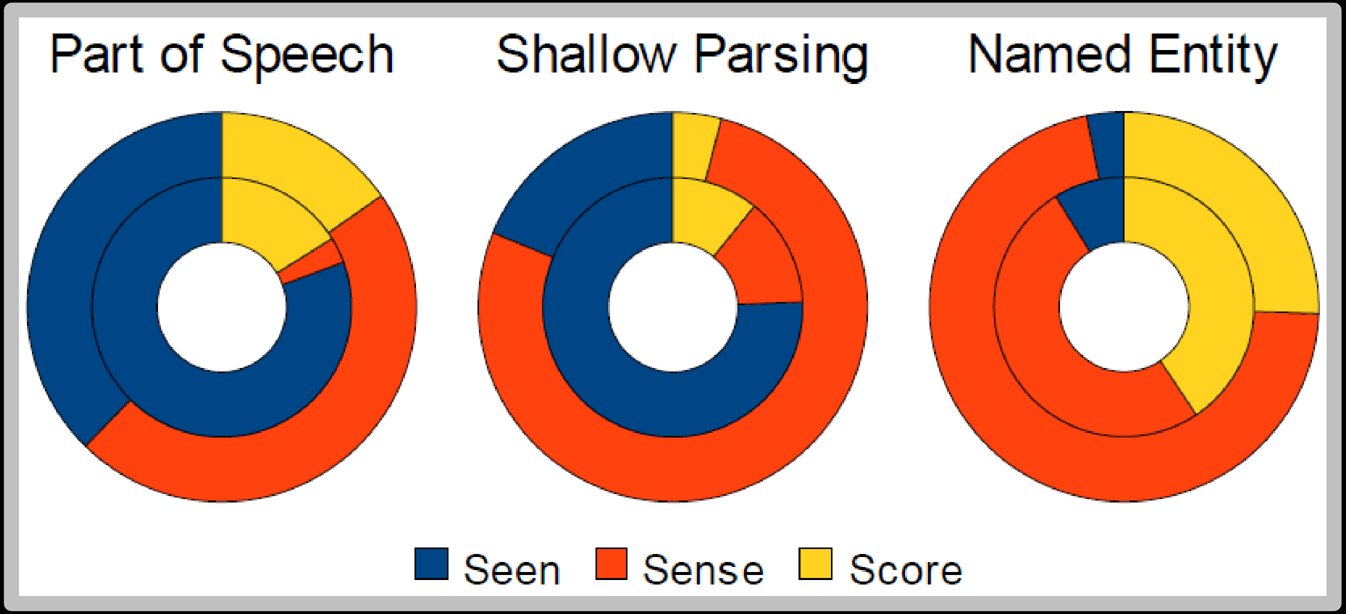
# Domains really are different

- Can you guess what domain each of these sentences is drawn from?

| | |
|---|---|
| **News** | Many factors contributed to the French and Dutch objections to the proposed EU constitution |
| **Parliament** | Please rise, then, for this minute's silence |
| **Medical** | Latent diabetes mellitus may become manifest during thiazide therapy |
| **Science** | Statistical machine translation is based on sets of text to build a translation model |
| **Step-mother** | I forgot to mention in yesterdays post that I also trimmed an overgrown HUGE hedge that spams the entire length of the front of my house and is about 3' accrossed. |

# $S^4$ ontology of adaptation effects

- **Seen:** Never seen this word before
  - News to medical: "diabetes mellitus"
- **Sense:** Never seen this word used in this way
  - News to technical: "monitor"
- **Score:** The wrong output is scored higher
  - News to medical: "manifest"
- **Search:** Decoding/search erred *(ignored)*




(inside=old domain
outside=new domain)

# Translating across domains is hard

| Old Domain (Parliament) | |
|---|---|
| **Original** | monsieur le président, les pêcheurs de homard de la région de l'atlantique sont dans une situation catastrophique. |
| **Reference** | mr. speaker, lobster fishers in atlantic canada are facing a disaster. |
| **System** | mr. speaker, the lobster fishers in atlantic canada are in a mess. |

| New Domain | |
|---|---|
| **Original** | comprimés pelliculés blancs pour voie orale. |
| **Reference** | white film-coated tablets for oral use. |
| **System** | white **pelliculés** tablets to oral. |

| New Domain | |
|---|---|
| **Original** | mode et voie(s) d'administration |
| **Reference** | method and route(s) of administration |
| **System** | **fashion** and **voie(s)** of **directors** |

**Key Question: What went wrong?**

# Adaptation effects in MT

- Quick observations:
  - New D language model helps (10%-63% impro
  - Tuning on new D data helps (10%-90% improve
  - Weighting new D data helps (4%-150% improve

Consistent in:
* movie subtitles
* scientific pubs
* PHP tech docs

- Identifying errors in MT (w/o parallel ne data):
  - **Seen:** old-only model + unseen input word pairs
  - **Sense:** old-only model + seen input/unseen out t pairs
  - **Score:** intersect old and mixed model, score fr m old

| | News | Medical |
|---|---|---|
| Seen | Little effect | ~ 40% of error |
| Sense | Little effect | ~ 40% of error |
| Score | ~ 90% of error | ~ 20% of error |

(as measured by Bleu score)

# Translating across domains is hard

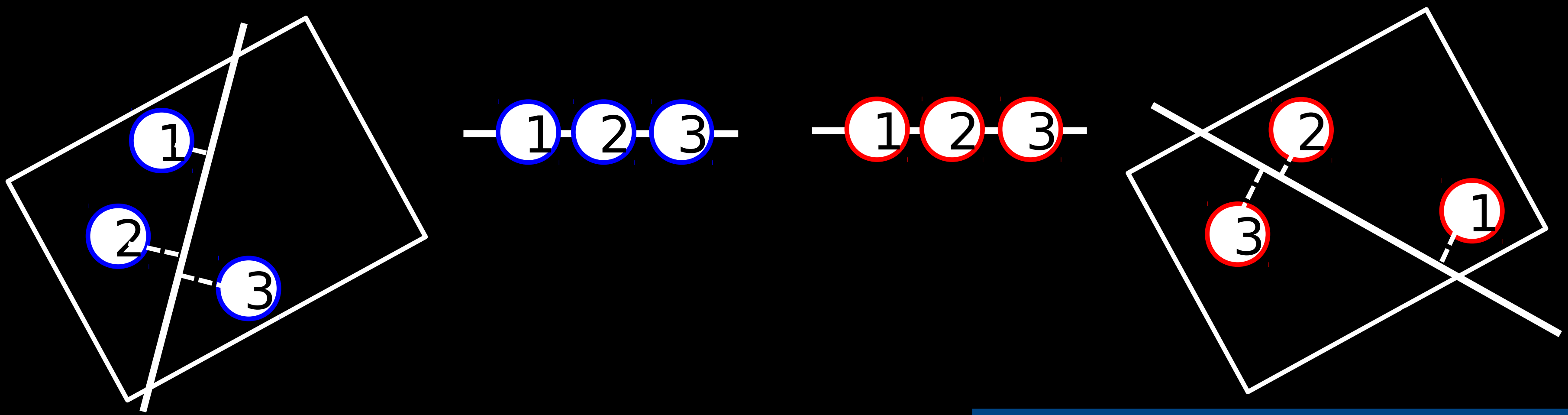
| Dom | Most frequent OOV Words | | | |
|---|---|---|---|---|
| News (17%) | behavior | favor | neighbors | fueled |
| | neighboring | abe | wwii | favored |
| | favorable | zhao | ahmedinejad | bernanke |
| | favorite | phelps | ccp | skeptical |
| Medical (49%) | renal | hepatic | subcutaneous | irbesartan |
| | ribavirin | olanzapine | serum | patienten |
| | dl | eine | sie | pharmacokinetics |
| | ritonavir | hydrochlorothiazide | erythropoietin | efavirenz |
| Movies (44%) | gonna | yeah | mom | hi |
| | b**** | daddy | s*** | later |
| | f*****g | f*** | gotta | wanna |
| | uh | namely | bye | dude |

[Daumé III & Jagarlamudi, 2011]

# Dictionary mining for "seen" errors

[Haghighi, Liang & Klein, 2009; Daumé III & Jagarlamudi, 2011]

- Find frequent terms in new domain
- Use those that exist in old domain as "training data"
- Extract context and orthographic features
- Find low-dimensional subspace on training data (CCA)

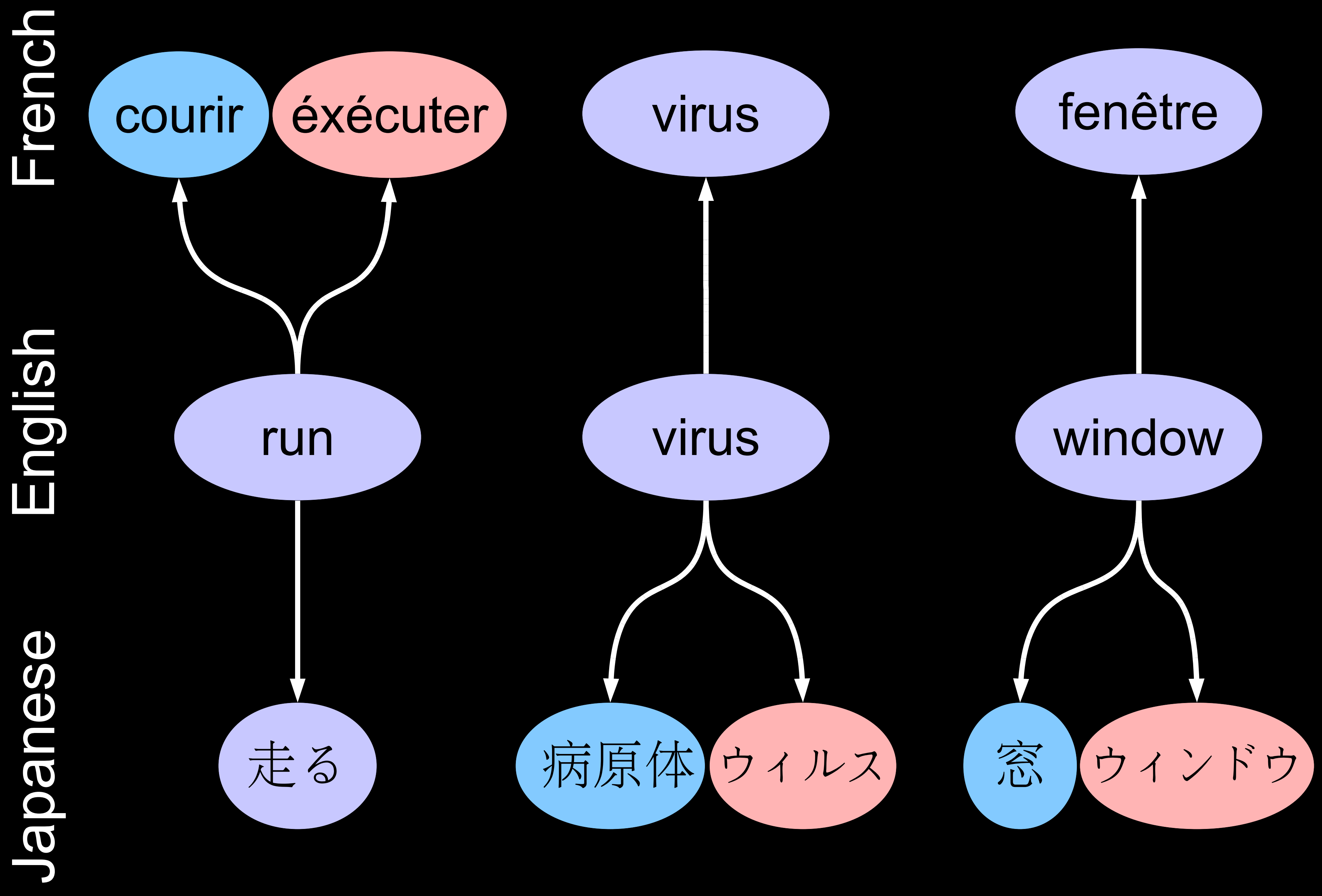



- Pair input words with <=5 output words
- Add four features to SMT model
- Rerun parameter tuning

| | DE | FR |
|---|---|---|
| News | +0.80 | +0.36 |
| Emea | +1.44 | +1.51 |
| Subs | +0.13 | +0.61 |
| PHP | +0.28 | +0.68 |

(Bleu score improvements)

# Senses are domain/language specific

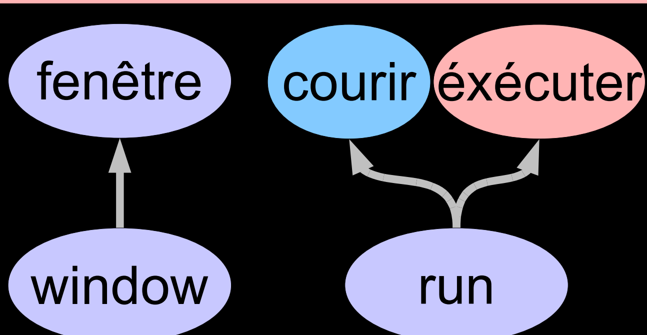


# Automatically identifying new senses

- **Context + existence of translations in comparable data**

| | |
|---|---|
| is a **window** of opportunity | via une **fenêtre** insérée . |
| have a **window** of opportunity | vers ma **fenêtre** ou vers |
| in the **run** up to | voulons pas **courir** le risque |
| , we **run** the risk | , sans **courir** le risque |

| | |
|---|---|
| the browser **window** ' s | dans la **fenêtre** . cet |
| in the **window** to give | dans la **fenêtre** . </s> |
| time to **run** when applied | *courir not found* |
| or have **run** vcvars.bat , | |




ne pouvez **éxécuter** que les
pour l' **éxécuter** elle va

# Spotting New Sen

- Binary classification
  - +ve: French token
  - -ve: French token i
- Lots of features con
  - Frequency of words
  - Language model pe
  - Topic model "misr
  - Marginal matching
  - Translation "flow" in

Given:

- A joint p(x,y) in the old domain
- Marginals q(x) and q(y) in the new domain

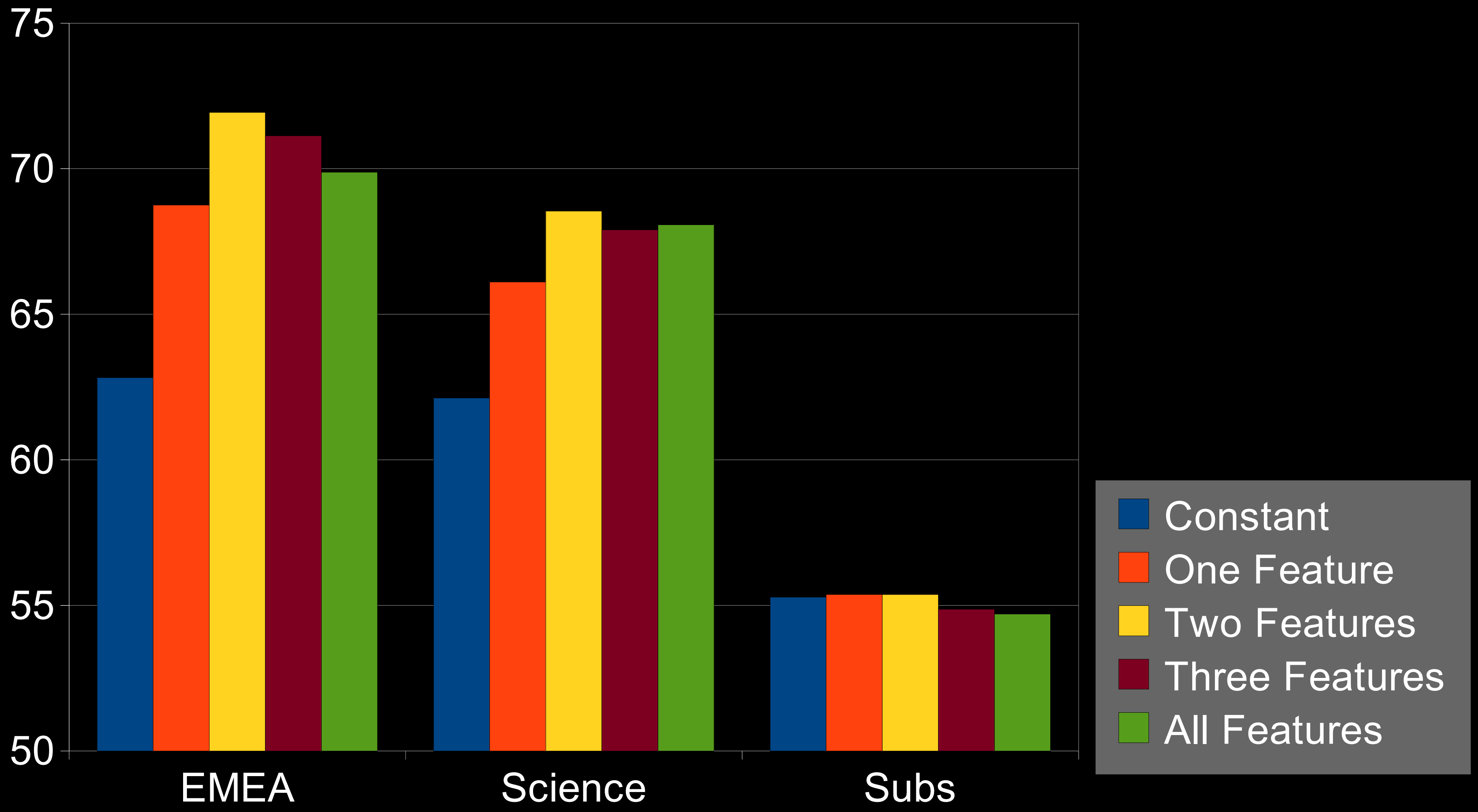
Recover:

- Joint q(x,y) in the new domain

We formulate as a L1-regularized linear program

Easier alternative: we have *many* such q(x) and q(y)s

# Experimental Results




Selected features:

```
EMEA:    ppl   || matchm flow   || matchm topics flow
Science: ppl   || matchm ppl    || matchm topics ppl
Subs:    topcs || matchm topics || matchm topics flow
```

# Conclusions

- Transfer Learning...
  - Assuming fixed task/domain relatedness is a bad idea
  - Key question: what type of representation is "right"?
  - Can do subspaces, trees, clusters, etc. etc. etc.

- In Language...
  - ML addresses only part of the adaptation picture
  - So far, specialized approaches for addressing other parts
    - Mining translations from comparable data
    - Automatically spotting new word senses

Thanks! Questions?