# Introduction to Structured Prediction and Domain Adaptation

**Alexander Fraser**
CIS, LMU Munich

2017-10-24

WP1: Structured Prediction and Domain Adaptation

# Outline

- **Introduction to structured prediction and domain adaptation**
- Review of very basic structured prediction
- Domain adaptation for statistical machine translation

# Structured Prediction I

- Structured prediction is a branch of machine learning dealing with outputs that have structure
  - The output label is complex, such as an entire parse tree or a complete POS-tagging for a sentence
- Typically one can break down individual decisions into sequential steps, but each decision depends on all previous decisions
  - Often there is therefore a search problem involved in finding the best (structured) label

# Structured Prediction II

- Typical structured prediction problems in NLP include:
- Tagging tasks (such as POS-tagging or named entity recognition)
  - Here the structure is a sequence of labels (e.g., one per word, such as POS tags or IOB named entity labeling)
- Parsing tasks, such as syntactic parsing
  - Here the structure can be a parse tree (but as we will see later, parse trees can be viewed as sequences, this is popular at the moment)
- Word prediction tasks
  - Such as language modeling and machine translation (structure is the sequence of words chosen)

# Domain Adaptation I

- Domain adaptation is the problem in machine learning which occurs when one wishes to train on one distribution and test on another
  - For example, train a POS tagger on the German Tiger corpus, which is in the "news" domain
  - Test on German tweets (in the "tweet" domain?)
- However, the term is overloaded, meaning different things to different people
  - There are many different scenarios studied in the literature

# Domain Adaptation II

- Sometimes we are given an OLD domain training corpus (which is out of domain) and a NEW domain training corpus
- The baseline is training on NEW only
- The task is then to use the OLD domain corpus to improve performance
- One simple way to do this is to concatenate the two corpora and train on this new corpus
  - But this often results in OLD "overwriting" NEW, because OLD is often much larger

# Domain Adaptation III

- Domain adaptation of simple classifiers (like binary classifiers) is reasonably well-studied
- Two examples here include:
  - Frustratingly Easy by Daume (feature augmentation, more on this later)
  - Instance Weighting (downweight OLD training examples in training to try to get the best performance on NEW)
- There are many more approaches

# Combining Structured Prediction and Domain Adaptation

- Domain adaptation of structured prediction systems is particularly challenging
- Often it is easy to see domain effects on individual decisions, such as picking the part-of-speech of "monitor"
  - In the news domain, often a verb meaning "to watch"
  - In the information technology domain, often a noun, e.g., "computer monitor"
- But in domain adaptation one often wishes to use knowledge about the sequence that is coming from the wrong (OLD) domain
- It is difficult to do this!

# Outline

- Introduction to structured prediction and domain adaptation
- **Quick review of very basic structured prediction**
  - I will go through this very fast (many of you have seen some version of this before)
- Domain adaptation for statistical machine translation

# Example

**the seminar at <time> 4 pm will**

| Condition | Additional Knowledge | | | | Action |
|---|---|---|---|---|---|
| Word | Lemma | LexCat | case | SemCat | Tag |
| | | | | | |
| | at | | | | stime |
| | | Digit | | | |
| | | | | timeid | |
| | | | | | |



Fabio Ciravegna:
Adaptive Information Extraction from Text by Rule Induction and Generalisation
in Proceedings of 17th International Joint Conference on Artificial Intelligence (IJCAI 2001), Seattle, August 2001.

# Binary Classification

- I'm going to first discuss linear models for binary classification, using binary features
- Our classifier is trying to decide whether we have a <stime> tag or not at the current position (between two words in an email)
- The first thing we will do is encode the context at this position into a feature vector

# Feature Vector

- Each feature is true or false, and has a position in the feature vector
- The feature vector is typically sparse, meaning it is mostly zeros (i.e., false)
- The feature vector represents the full feature space. For instance, consider...

# Example

**the seminar at <time> 4 pm will**

| Condition | Additional Knowledge | | | | Action |
|---|---|---|---|---|---|
| Word | Lemma | LexCat | case | SemCat | Tag |
| the | the | Art | low | | |
| seminar | Seminar | Noun | low | | |
| at | at | Prep | low | | stime |
| 4 | 4 | Digit | low | | |
| pm | pm | Other | low | timeid | |
| will | will | Verb | low | | |



Fabio Ciravegna:
Adaptive Information Extraction from Text by Rule Induction and Generalisation
in Proceedings of 17th International Joint Conference on Artificial Intelligence (IJCAI 2001), Seattle, August 2001.

## Example

the seminar at <time> 4 pm will

| Condition | Additional Knowledge | | | | Action |
|---|---|---|---|---|---|
| Word | Lemma | LexCat | case | SemCat | Tag |
| the | the | Art | low | | |
| seminar | Seminar | Noun | low | | |
| at | at | Prep | low | | stime |
| 4 | 4 | Digit | low | | |
| pm | pm | Other | low | timeid | |
| will | will | Verb | low | | |

- Our features represent this table using binary variables
- For instance, consider the lemma column
- Most features will be false (false = off = 0)
- The lemma features that will be on (true = on = 1) are:

  -3_lemma_the
  -2_lemma_Seminar
  -1_lemma_at
  +1_lemma_4
  +2_lemma_pm
  +3_lemma_will

# Feature Vector

- We might use a feature vector like this:

(this example is simplified – really we'd have all features for all positions)

| | |
|---|---|
| 1 | Bias term |
| 0 | ... (say, -3_lemma_giraffe) |
| 1 | -3_lemma_the |
| 0 | ... |
| 1 | -2_lemma_Seminar |
| 0 | ... |
| 0 | ... |
| 1 | -1_lemma_at |
| 1 | +1_lemma_4 |
| 0 | ... |
| 1 | +1_Digit |
| 1 | +2_timeid |

# Weight Vector

- Now we'd like the dot product to be > 0 if we should insert a <stime> tag
- To encode the rule we looked at before we have three features that we want to have a positive weight
  - -1_lemma_at
  - +1_Digit
  - +2_timeid
- We can give them weights of 1
- Their sum will be three
- To make sure that we only classify if all three weights are on, let's set the weight on the bias term to -2

# Dot Product - I

| | | |
|---|---|---|
| 1 | Bias term | -2 |
| 0 | | 0 |
| 1 | -3_lemma_the | 0 |
| 0 | | 0 |
| 1 | -2_lemma_Seminar | 0 |
| 0 | | 0 |
| 0 | | 0 |
| 1 | -1_lemma_at | 1 |
| 1 | +1_lemma_4 | 0 |
| 0 | | 0 |
| 1 | +1_Digit | 1 |
| 1 | +2_timeid | 1 |

To compute the dot product first take the product of each row, and then sum these

# Dot Product - II

| | | | | |
|---|---|---|---|---|
| 1 | Bias term | -2 | 1*-2 | 1*-2 |
| 0 | | 0 | 0*0 | |
| 1 | -3_lemma_the | 0 | 1*0 | |
| 0 | | 0 | 0*0 | |
| 1 | -2_lemma_Seminar | 0 | 1*0 | |
| 0 | | 0 | 0*0 | |
| 0 | | 0 | 0*0 | |
| 1 | -1_lemma_at | 1 | 1*1 | 1*1 |
| 1 | +1_lemma_4 | 0 | 1*0 | |
| 0 | | 0 | 0*0 | |
| 1 | +1_Digit | 1 | 1*1 | 1*1 |
| 1 | +2_timeid | 1 | 1*1 | 1*1 |
| | | | | ----- |
| | | | | 1 |

# Learning the Weight Vector

- The general learning task is simply to find a good weight vector!
  - This is sometimes also called "training"
- Basic intuition: you can check weight vector candidates to see how well they classify the training data
  - Better weights vectors get more of the training data right
- So we need some way to make (smart) changes to the weight vector
  - The goal is to make better decisions on the training data

# Feature Extraction

- We run **feature extraction** to get the feature vectors for each position in the text
- We typically use a text representation to represent true values (which are sparse)
- Often we define **feature templates** which describe the feature to be extracted and give the name of the feature (i.e., -1_lemma_ XXX)

| | |
|---|---|
| -3_lemma_the -2_lemma_Seminar -1_lemma_at +1_lemma_4 +1_Digit +2_timeid | STIME |
| -3_lemma_Seminar -2_lemma_at -1_lemma_4 -1_Digit +1_timeid +2_lemma_ will | NONE |

...

# How can we get more power in linear models?

- Change the features!
- For instance, we can create combinations of our old features as new features
- Sometimes these new compound features would be referred to as trigrams (they each combine three basic features)

# Feature Selection

- A task which includes automatically finding such new compound features is called **feature selection**
  - This is built into some machine learning toolkits
  - Or you can implement it yourself by trying out feature combinations and checking the training error
    - Use human intuition to check a small number of combinations
    - Or do it automatically, using a script
- Deep learning is conceptually doing something like this using **representation learning**

# Two classes

- So far we discussed how to deal with a single label
  - At each position between two words we are asking whether there is a <stime> tag
- However, we are interested in <stime> and </stime> tags
- How can we deal with this?
- We can simply train one classifier on the <stime> prediction task
  - Here we are treating </stime> positions like every other non <stime> position
- And train another classifier on the </stime> prediction task
  - Likewise, treating <stime> positions like every other non </stime> position
- If both classifiers predict "true" for a single position, take the one that has the highest dot product

# More than two labels

- What we have had up until now is called **binary classification**
- But we can generalize this idea to many possible labels
- This is called **multiclass classification**
  - We are picking one label (class) from a set of classes
- For instance, maybe we are also interested in the <etime> and </etime> labels
  - These labels indicate seminar end times, which are also often in the announcement emails (see next slide)

# CMU Seminars - Example

<0.24.4.93.20.59.10.jgc+@NL.CS.CMU.EDU (Jaime Carbonell).0>

Type: cmu.cs.proj.mt

Topic: <speaker>Nagao</speaker> Talk

Dates: 26-Apr-93

Time: <stime>10:00</stime> - <etime>11:00 AM</etime>

PostedBy: jgc+ on 24-Apr-93 at 20:59 from NL.CS.CMU.EDU (Jaime Carbonell)

Abstract:

<paragraph><sentence>This Monday, 4/26, <speaker>Prof. Makoto Nagao</speaker> will give a seminar in the <location>CMT red conference room</location> <stime>10</stime>-<etime>11am</etime> on recent MT research results</sentence>.</paragraph>

# One against all

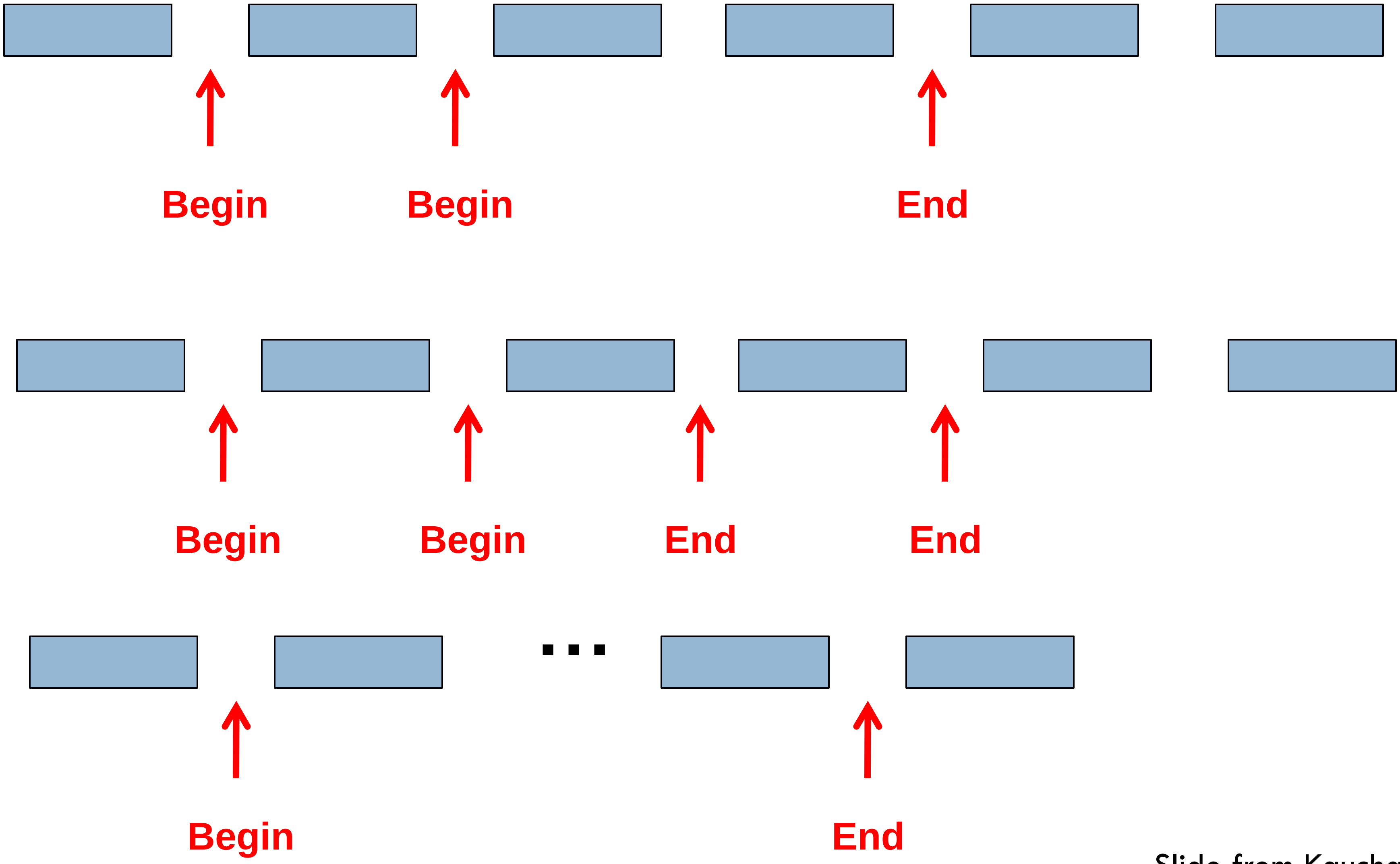
- We can generalize the way we handled two binary classification decisions to many labels
- Let's add the <etime> and </etime> labels
- We can train a classifier for each tag
  - Just as before, every position that is not an <etime> is a negative example for the <etime> classifier, and likewise for </etime>
- If multiple classifiers say "true", take the classifier with the highest dot product
- This is called **one-against-all**
- It is a quite reasonable way to use binary classification to predict one of multiple classes
  - It is not the only option, but it is easy to understand (and to implement too!)

# Binary classifiers and sequences

- We can detect seminar start times by using two binary classifiers:
  - One for <stime>
  - One for </stime>
- And recall that if they both say "true" to the same position, take the highest dot product

- Then we need to actually annotate the document
- But this is problematic...

# Some concerns




Slide from Kauchak

# A basic approach

- One way to deal with this is to use a greedy algorithm
- Loop:
  - Scan the document until the <stime> classifier says true
  - Then scan the document until the </stime> classifier says true
- If the last tag inserted was <stime> then insert a </stime> at the end of the document
- Naturally, there are smarter algorithms than this that will do a little better
- But relying on these two independent classifiers is not optimal

# How can we deal better with sequences?

- We can make our classification decisions dependent on previous classification decisions
- For instance, think of the Hidden Markov Model as used in POS-tagging
- The probability of a verb increases after a noun

# Basic Sequence Classification

- We will do the following
  - We will add a feature template into each classification decision representing the **previous classification decision**
  - And we will change the labels we are predicting, so that in the span between a start and end boundary we are predicting a different label than outside

# Basic idea

| | | | |
|---|---|---|---|
| Seminar | at | 4 | pm |
| | <stime> | in-stime | </stime> |

- The basic idea is that we want to use the previous classification decision
- We add a special feature template -1_label_XXX
- For instance, between 4 and pm, we have: -1_label_<stime>

- Suppose we have learned reasonable classifiers
- How often should we get a <stime> classification here? (Think about the training data in this sort of position)

# -1_label_<stime>

- This should be an extremely strong indicator not to annotate a <stime>

- What else should it indicate?
  - It should indicate that there must be either a in-stime or a </stime> here!

# Changing the problem slightly

- We'll now change the problem to a problem of annotating tokens (rather than annotating boundaries)
- This is traditional in IE, and you'll see that it is slightly more powerful than the boundary style of annotation
- We also make less decisions (see next slide)

# IOB markup

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Seminar | at | 4 | pm | will | be | on | ... |
| O | O | B-stime | I-stime | O | O | O | |

- This is called IOB markup (or BIO = begin-in-out)
- This is a standardly used markup when modeling IE problems as sequence classification problems

- We can use a variety of models to solve this problem
- One popular model is the Hidden Markov Model, which you have seen in Statistical Methods
  - There, the label is the state
- However, here we will (mostly) stay more general and talk about binary classifiers and one-against-all

# (Greedy) classification with IOB

| Seminar | at | 4 | pm | will | be | on | ... |
|---|---|---|---|---|---|---|---|
| O | O | B-stime | I-stime | O | O | O | |

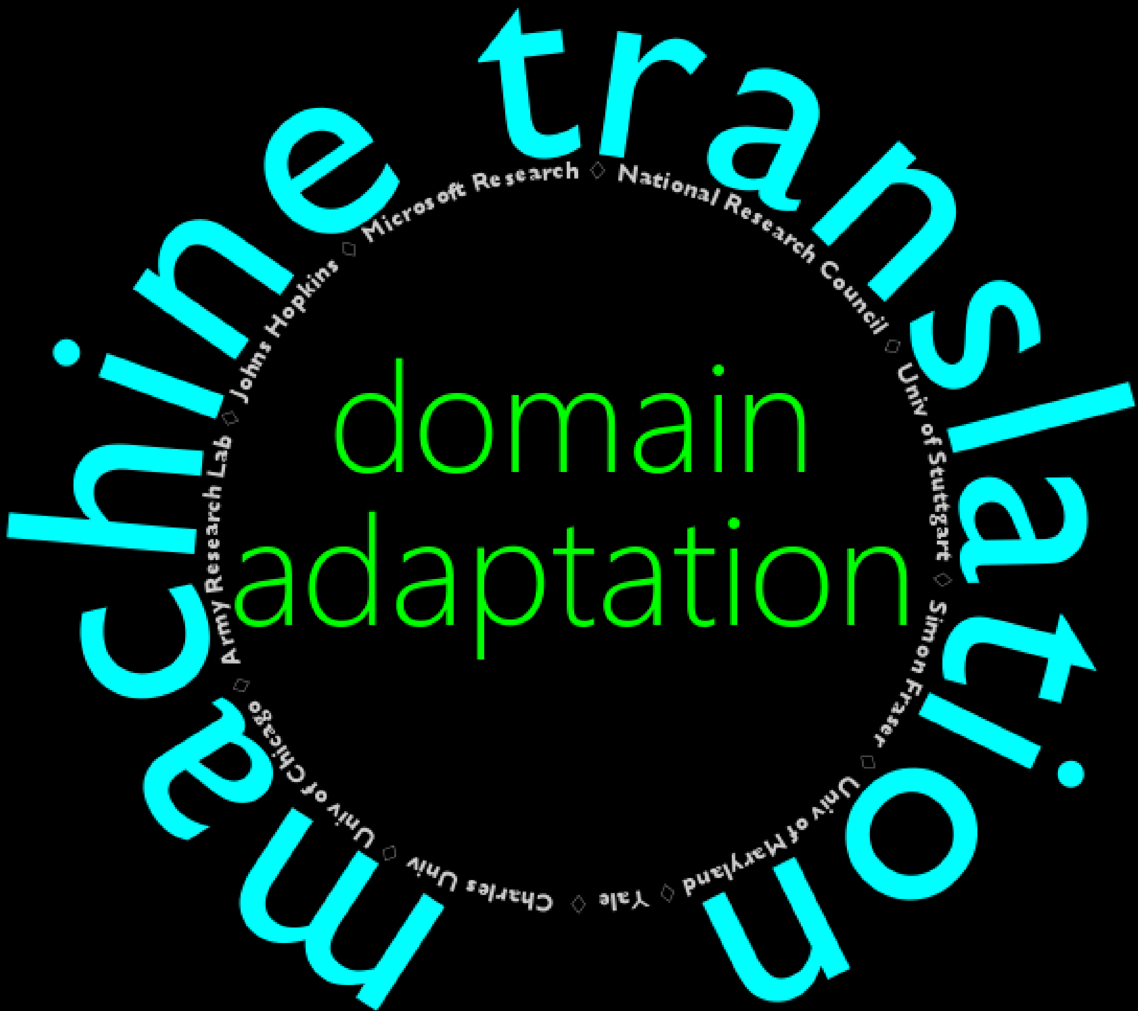
- To perform greedy classification, first run your classifier on "Seminar"
- You can use a label feature here like
  -1_Label_StartOfSentence
- Suppose you correctly choose "O"
- Then when classifying "at", use the feature:
  -1_Label_O
- Suppose you correctly choose "O"
- Then when classifying "4", use the feature:
  -1_Label_O
- Suppose you correctly choose "B-stime"
- Then when classifying "pm", use the feature:
  -1_Label_B-stime
- Etc...

# Summary: very simple structured prediction

- I've taught you the basics of:
  - Binary classification using features
  - Feature selection (vs. representation learning)
  - Multiclass classification (using one-against-all)
  - Sequence classification (using a feature that uses the previous decision)
    - And IOB labels
- I've skipped a lot of details!
  - I haven't told you how to actually learn the weight vector in the binary classifier in detail (beyond the perceptron rule)
  - I also haven't talked about non-greedy ways to do sequence classification
  - And I didn't talk about probabilities, which are used directly, or at least approximated, in many kinds of commonly used linear models
- Hopefully what I did tell you is fairly intuitive and helps you understand classification, that is the goal

# Outline

- Introduction to structured prediction and domain adaptation
- Review of very basic structured prediction
- **Domain adaptation for statistical machine translation**
  - I probably can't make it through all of these slides, but hopefully this gives you an idea




Fabienne Braune
Marine Carpuat
Ann Clifton
Hal Daumé III
Alex Fraser
Katie Henry
Anni Irvine
Jagadeesh Jagarlamudi
John Morgan
Chris Quirk
Majid Razmara
Rachel Rudinger
Ales Tamchyna

<u>Special thanks:</u>
George Foster
Dragos Munteanu
Everyone at CLSP

Based on the Report of the 2012 JHU Workshop On Domain Adaptation for Machine Translation

# 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 |
| **(Science?) Joel Tetreault sings Greg Crowther** | Jenny, what is this number?<br>Tell me how it's defined.<br>Jenny, plug in this number:<br>Three point one four one five nine.<br>(Three point one four one five nine). |

# 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** |

# Outline

- Quick introduction to domain adaptation for SMT
- What is the problem really?
  - a new taxonomy for domain-related SMT errors
- Towards solving the errors
  - with comparable corpora
  - with parallel corpora

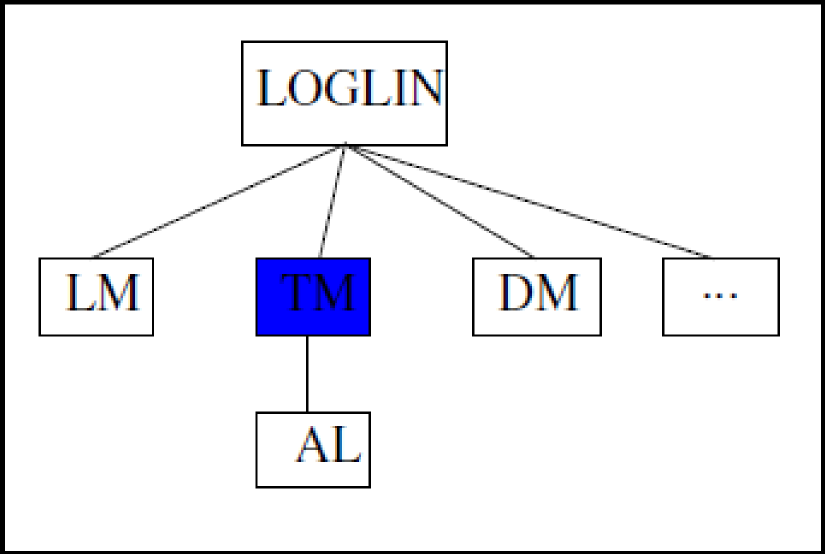
# Domain Adaptation for SMT

- Problem: **domain mismatch** between test and training data can cause severe degradation in translation quality
- General solution: adjust SMT parameters to optimize performance for a test set, based on some knowledge of its domain
- Various settings:
  - amount of in-domain training data: small, dev-sized, none (just source text)
  - nature of out-of-domain data: size, diversity, labeling
  - monolingual resources: source and target, in-domain or not, comparable or not
  - latency: offline, tuning, dynamic, online, (interactive)

Slide adapted from Foster 2012

# What to adapt?

- Log-linear model
  - limited scope if in-domain tuning set (dev) is available
- Language model (LM)
  - effective and simple
  - previous work from ASR
  - perplexity-based interpolation popular
- Translation model (TM):
  - most popular target, gains can be elusive
- Other features: little work so far
- Alignment: little work, possibly limited scope due to “one sense per discourse"




Slide adapted from Foster 2012

# How to adapt to a new domain?

- Filtering training data
  - select from out-of-domain data based on similarity to our domain
- Corpus weighting (generalization of filtering)
  - Done at sub-corpora, sentence, or phrase-pair levels
- Model combination
  - train sub-models on different sub-corpora
- Self-training
  - generate new parallel data with SMT
- Latent semantics
  - exploit latent topic structure
- Mining comparable corpora
  - extend existing parallel resources

Slide adapted from Foster 2012

# 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?**

# $S^4$ taxonomy 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

Working with *no* new domain parallel data!

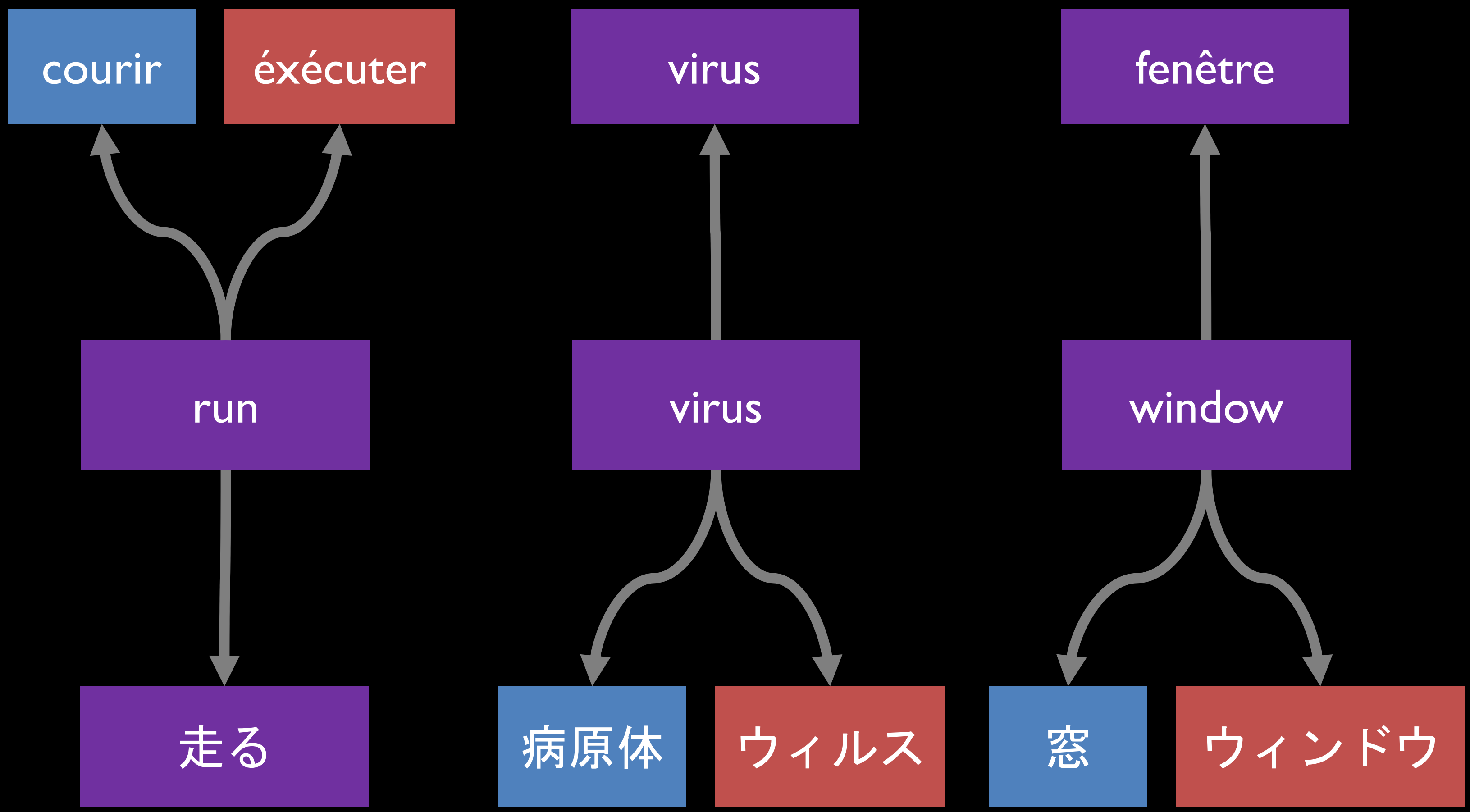
# Macro-analysis of $S^4$ effects

- Evaluation using BLEU

| | News | Medical | Science | Subtitles |
|---|---|---|---|---|
| **Seen** | +0.3% | +8.1% | +6.1% | +5.7% |
| **Sense** | +0.6% | +6.6% | +4.4% | +8.7% |
| **Score** | +0.6% | +4.5% | +9.9% | +8.4% |

- Hansard: 8m sents 161m fr-tokens
- News: 135k sents 3.9m fr-tokens
- Medical: 472k sents 6.5m fr-tokens
- Science: 139k sents 4.3m fr-tokens
- Subtitles: 19m sents 155m fr-tokens

# Senses are domain/language specific



# Case 1: No NEW domain parallel data

- Common situation
  - Lots of data in some OLD domain (e.g., government documents)
  - Need to translate many NEW domain documents
- Acquiring additional NEW domain translations is critical!
- Lots of past work in term mining
  - Distributional similarity [Rapp 1996]
  - Orthographic similarity
  - Temporal similarity

# Marginal matching for "sense" errors

**Given:**

- Joint $p(x, y)$ in old domain
- Marginals $q(x)$ and $q(y)$ in the new domain

**Recover:**

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

We formulate as a L1-regularized linear program

Easier: *many* $q(x)$ and $q(y)$s

| | grant | tune | … | Σ |
|---|---|---|---|---|
| accorder | 9 | 1 | … | 10+… |
| … | … | … | … | … |
| Σ | 9+… | 1+… | … | |

| | grant | tune | … | Σ |
|---|---|---|---|---|
| accorder | ??? | ??? | ??? | 5 |
| … | ??? | ??? | ??? | … |
| Σ | 1 | 5 | … | |

# Additional features

- Sparsity: # of non-zero entries should be small
- Distributional: document co-occurrence ⇔ translation pair
- Spelling: Low edit dist ⇔ translation pair
- Frequency: Rare words align to rare words; common words align to common words

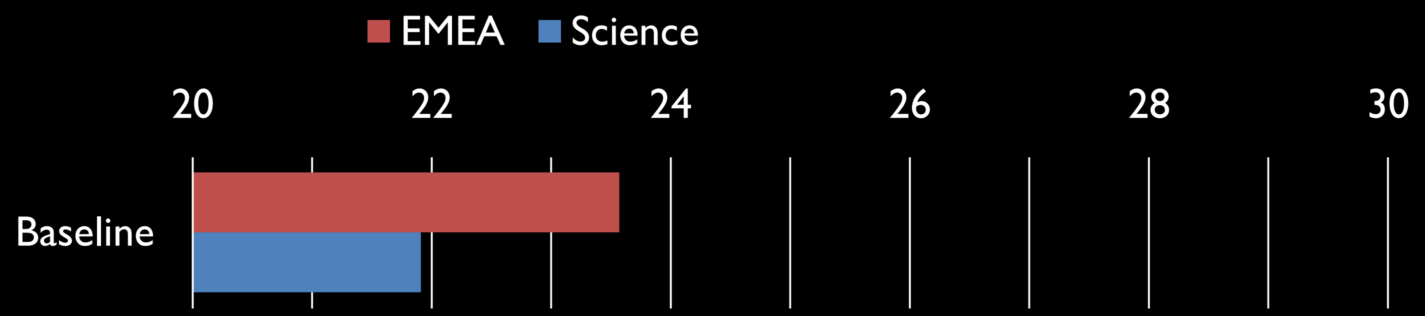
```
c-aractérisation
characterization
```

| E | F |
|---|---|
| the | le |
| … | … |
| spiders | araignées |
| … | … |

# Example learned translations (Science)

| French | Correct English | Learned Translations |
| --- | --- | --- |
| cisaillement | shear | viscous<br>crack<br>shear |
| chromosomes | chromosomes | chromosomes<br>chromosome<br>chromosomal |
| caractérisation | characterization | characterization<br>characteristic |
| araignées | spiders | spiders<br>ant<br>spider |
| tiges | stems | usda<br>centimeters<br>flowering |

# BLEU Scores

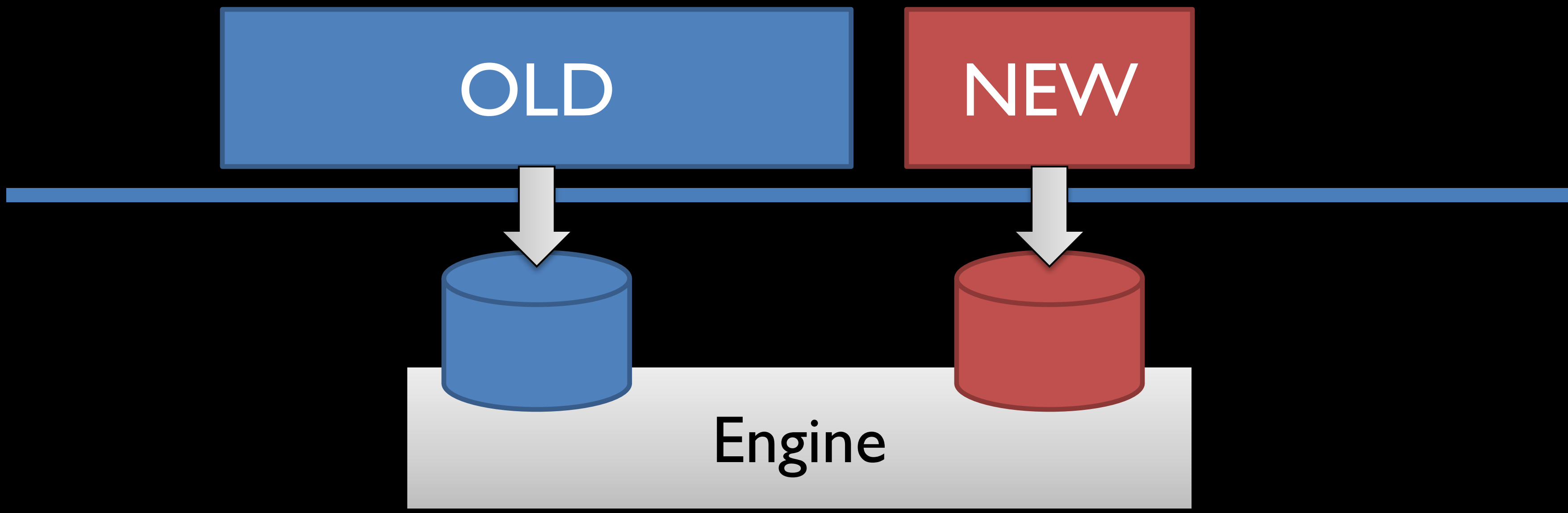


# Case 2: Add NEW domain parallel data

- Say we have a NEW domain translation memory
- How can we leverage our OLD domain to achieve the greatest benefit?

# Initial adaptation baselines

# Use both models (log-linear mixture)



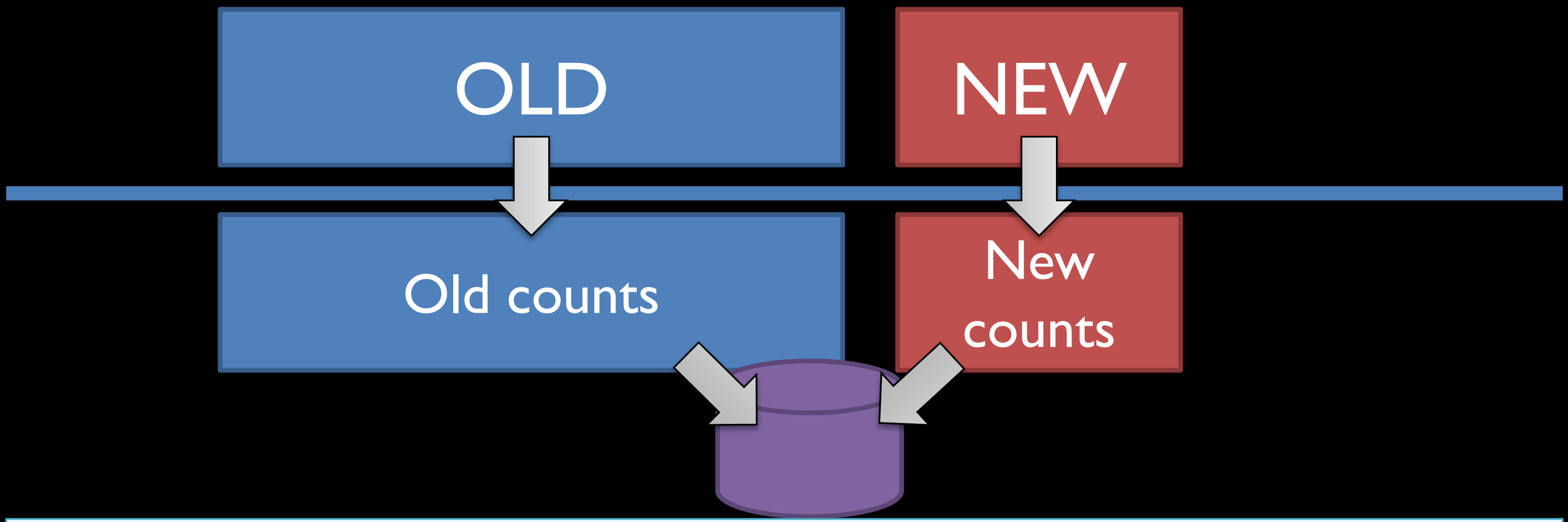

**Baseline:**

$$\alpha_1 \log p(f|e) + \alpha_2 \log p(e) + \ \ldots$$

**New:**

$$\alpha_{1OLD} \log p_{OLD}(f|e) + \alpha_{1NEW} \log p_{NEW}(f|e) + \alpha_2 \log p(e) + \cdots$$

# Combine models (linear mixture)




**Baseline:**

$$p(f|e) = \frac{c(f,e)}{c(e)}$$

**New** – mix with $\lambda$ picked on dev set:

$$p(f|e) = \lambda \frac{c_{old}(f,e)}{c_{old}(e)} + (1-\lambda) \frac{c_{new}(f,e)}{c_{new}(e)}$$

# BLEU results

| | OLD | NEW | OLD+ NEW | Use both models | Combine models |
|---|---|---|---|---|---|
| News | 23.8 | 21.7 | 22.0 | 16.4 | 21.4 |
| EMEA | 28.7 | 34.8 | 34.8 | 32.9 | 36.6 |
| Science | 26.1 | 32.3 | 27.5 | 30.9 | 32.2 |
| Subtitles | 15.1 | 20.6 | 20.5 | 18.4 | 18.5 |

# Next steps

- These mixtures are simple but coarse
- More fine-grained approaches:
  - Data selection: pick OLD data most like NEW
  - Data reweighting: use fractional counts on OLD data; greater weight to sentence pairs more like NEW
  - Can reweight at the word or phrase level rather than sentence pair [Foster et al., 2010]
- Similar in spirit to statistical domain adaptation
  - but existing machine learning algorithms can't be applied
  - because SMT is not a classification task

# Phrase Sense Disambiguation (PSD)

Proposed solution: Phrase Sense Disambiguation

[Carpuat & Wu 2007]

- Incorporate context in lexical choice
  - Yields P(e|f, context) features for phrase pairs
  - Unlike usual P(e|f) relative frequencies
- Turns phrase translation into discriminative classification
  - Just like standard machine learning tasks

[Chan et al. 2007, Stroppa et al. 2007, Gimenez & Màrquez 2008, Jeong et al. 2010, Patry & Langlais 2011, ...]

# Why PSD for domain adaptation?

Disambiguating English senses of **rapport**

P(e|f) in Hansard

| | |
|---|---|
| report | Il a rédigé un **rapport** . |
| relationship | Quel est le **rapport** ? |
| ratio | le **rapport** longueur / largeur |
| balance | le **rapport** bénéfique / risque |
| … | |

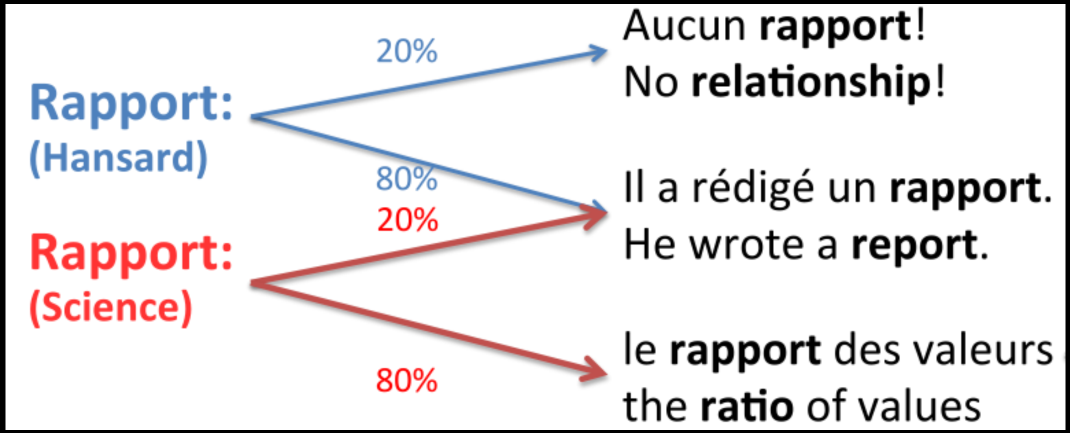
Highest P(e|f) in Science!

New sense in medical domain!

Occurs in new domains but not as often as in Hansard!

Source context can prevent translation errors when shifting domain

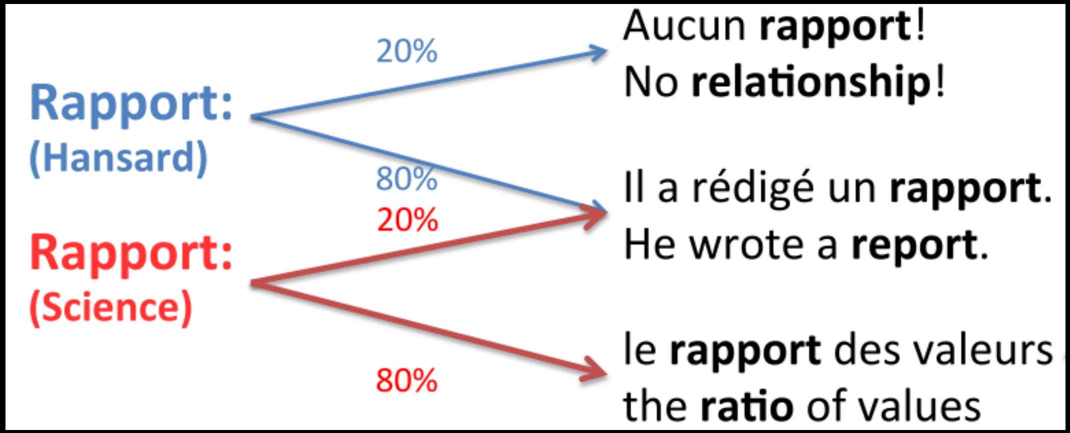
# Phrase Sense Disambiguation




- PSD = phrase translation as classification
- PSD at test time
  - use context to predict correct English translation of French phrase
  - local lexical and POS context , global sentence and document context
- PSD at train time
  - extract French phrases with English translations from word alignment
  - throw into off-the-shelf classifier + adaptation techniques

[Blitzer & Daumé 2010]

# Domain adaptation in PSD




- Train a classifier over OLD and NEW data
- Allow classifier to:
  - share some features
    `{rédigé …} rapport → report`
  - keep others domain specific
    `rapport {… valeurs} → ratio`

# Feature augmentation I

| | OLD | NEW |
|---|---|---|
| | $\varphi_{e,f} \mapsto \langle \varphi_{e,f}, \varphi_{e,f}, 0 \rangle$ | $\varphi_{e,f} \mapsto \langle \varphi_{e,f}, 0, \varphi_{e,f} \rangle$ |
| Original features | {rédigé …} rapport → report<br>{aucun …} rapport → relationship | {rédigé …} rapport → report<br>rapport {… valeurs} → ratio |

# Feature augmentation II

Feature augmentation is a very simple way to carry out domain adaptation

For more details on the basic approach (applicable to any feature-based classifier), see the paper:

**Frustratingly Easy Domain Adaptation**

Hal Daumé III

ACL 2007

# PSD in Moses: VW-Moses integration

- First general purpose classifier in Moses
- Tight integration
  - Can be built and run out-of-the-box, extended with new features, etc
  - Fast!
    - 180% run time of standard Moses, fully parallelized in training (multiple processes) and decoding (multithreading)

# Other areas of investigation

PSD for Hierarchical phrase-based translation

Discovering latent topics from parallel data

Spotting new senses: determining when a source word gains a new sense (needs a new translation)

# Discussion

- Introduced taxonomy and measurement tools for adaptation effects in MT
- "Score" errors – target of prior work – only a part of what goes wrong in translation
- Marginal matching introduced as a model for addressing *all* $S^4$ issues simultaneously: +2.4 BLEU
- Data and outputs released for you to use (both in MT and as a stand-alone lexical selection task)
- Feature-rich approaches integrated into Moses via VW library, applied to adaptation
- Range of other problems to work on: identifying new senses, cross-domain topic models, etc.

Marine Carpuat
NRC-CNRC

Hal Daume
U Maryland

Chris Quirk
MSR

# Summary

- Defined structured prediction
  - And presented a very simple approach
- Presented the abstract problem of domain adaptation
- Talked about domain adaptation in statistical machine translation
  - Raised lots of questions about how to define the problem, data and modeling
  - Parallel questions will come up throughout the semester

# Reminder: Getting a Grade

- You will make a presentation in English for 25 minutes on the paper
  - Using latex, powerpoint, etc
  - Include slide numbers (useful for discussion)
  - Send me the slides after class
  - Important technical note: this room only has *VGA*
- This will be followed by 20 minutes of discussion by everyone
- Three weeks after your presentation, a 6-page Hausarbeit is due
  - Written prose version of your slides
  - With inline citations, looking just like a standard scientific article!
    - References in a standard format!!!
  - If you need a review of how to do this, please check my slides on this in a previous seminar I have taught
    - (Or the new slides in the Informationsextraktion seminar, to be presented on Wednesday and Thursday this week)

# Outlook

- In the seminar, we will start by reading a number of recent but classic papers on structured prediction
  - Using neural networks
  - These are all deep networks, in the sense that they are deep over time
  - Nearly all of the papers we look at will model sequences (even the parsing paper)
- Then we will begin to look at domain adaptation papers applied to structured prediction
  - We'll see that very basic approaches work well, advanced work in this area is in its infancy
  - So now is a good time to acquire a basic understanding!
- Please read the two papers that will be discussed next time
  - They are important papers to understand, setting much of the groundwork on structured prediction using neural networks

- But don't forget that Tuesday next week is a holiday!

- Thank you for your attention!