# Language Technologies in e-Commerce

## Lecture #8: Machine Translation (2)

http://www.cs.cmu.edu/~ralf/lectures.html

# Data-Driven MT

**Motivation:**

Instead of laboriously writing all the grammar rules, etc. for a translation system, why not have the computer learn them automatically? Incremental improvement should be possible by simply adding more data to the system.

Although there was some dabbling with statistical translation methods in the 1950s, it wasn't until the 1980s that computers were powerful enough to make corpus-based approaches practical.

**Statistical MT**:

build a mathematical model of the correspondences between source and target languages

**Example-Based MT**:

find best match in example base and use it as the basis for creating a translation

**Connectionist**:

train a neural network to perform translation

# Statistical MT

Various systems:

- Candide (Peter Brown, IBM)
- LINGSTAT (Jon Yamron, Dragon Systems)
- SBTG (Dekai Wu, HKUST)
- Bilingual Tiling (Alshawi, AT&T)
- Link Grammars (Lafferty/Sleator, CMU)
- EGYPT
  `www.clsp.jhu.edu/ws99/projects/mt/toolkit/`

IBM's Candide project -- came up with five translation models of increasing complexity:

- Model 1: probabilistic word-for-word dictionary
- Models 2-5: add fertilities and distortion probabilities

Training material was 100 million words of Canadian parliamentary proceedings (English/French); it took 10 person-years just to clean up the data.

# Probabilistic Dictionaries

Simple dictionaries are easy to create from a corpus: just count the number of times a source-language word co-occurs with each target-language word, which gives us the following model:

$$\{ P(s_i|t_j) \}$$

In practice, we don't actually use probabilities for all pairings because

- lowest-frequency items contain a lot of noise
- storing all probabilities takes up lots of space

Thus, the model is filtered, zeroing out the probabilities for pairings which are "not likely enough".

# Basic Dictionary Extraction

Count the number of co-occurrences between source-language and target-language words.

For a given word pair $W_s$ and $W_t$ (source- and target-language words), which co-occur $C$ times, the pair is considered a translation if

$$\min(\ P(W_t|W_s),\ P(W_s|W_t)\ ) \geq thr[C]$$

or

$$P(W_t|W_s) \geq thr_1[C] \text{ and } P(W_s|W_t) \geq thr_2[C]$$

or

$$P(W_t|W_s) \geq thr_2[C] \text{ and } P(W_s|W_t) \geq thr_1[C]$$

where $thr[C]$ is the symmetric threshold and $thr_1[C]$ and $thr_2[C]$ are the asymmetric threshold (the latter attempts to account for words which are polysemous in one language but not the other).

The thresholds are a function of the absolute number of co-occurrences of the word pair because frequent words may still co-occur by chance a small number of times, while rare words which co-occur the same number of times are likely to be translations of each other.

(ABI (ABI 4)(BEVERAGE 2)(AMALGAMATED 2))
(ALMAHDI (AL-SADIQ 1)(AL-MAHDI 1))
(ARABSAT (ARABSAT 6))
(BIOTOPES (BIOTOPOS 2))
(BLEACH (LEJÍA 1))
(COMPLEMENTARITY (COMPLEMENTARIEDAD 77))
(D-1 (D-1 91)(D-2 43))
(DEEPEN (PROFUNDIZAR 17))
(DYNAMICS (DINÁMICA 77))
(EBW (HAZ 6)(ELECTRONES 6)(SOLDADORA 6))
(ESCOBAR (ESCOBAR 30))
(EXTRACONTINENTAL (EXTRACONTINENTALES 1))
(GEOSYSTEMS (GEOEX-1986 1)(GEOSISTEMAS 1))
(HU (HU 2)(XIAODI 2))
(KG (KILOGRAMOS 16)(KG 10))
(MILITARY-IDEOLOGICAL
(MILITAR-IDEOLÓGICA 1))
(MONASTERY (MONASTERIO 2))
(NON-NUCLEAR-WEAPON (POSEEDORES 78))
(ORCI (OIRI 8))
(PASHTU (PASHTU 1)(BRISTISH 1))
(PYAT (PYAT 1)(CABARET 1)(PERODIN 1)
(VILLARD 1))
(RAVANDI (RAVANDI 1)(KATCHOUI 1))
(REDISCOVER (REENCONTRAR 1))
(SCENES (ESCENAS 5))
(SECRECY (SECRETO 53))
(SHANKANGA (SHANKANGA 1))
(TECNOLÓGICO (UCMM 1))
(XXVI (XXVI 86))
(|1506TH| (|1506A| 8))

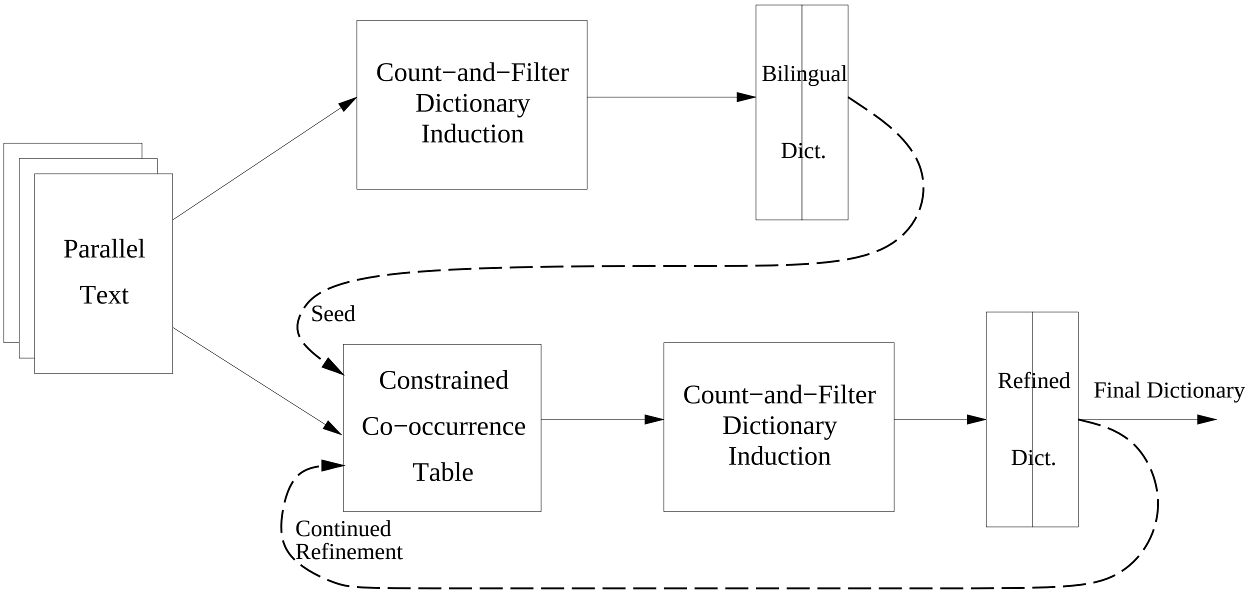
# Constrained Correspondence Table

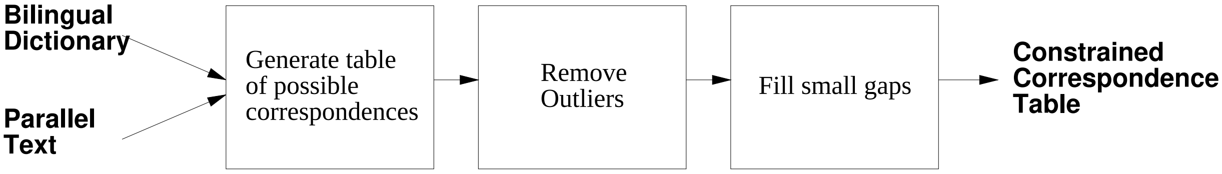
Using the initial dictionary, we can generate a correspondence table for each sentence pair.  This is constrained by assuming the coherence of local constituents between the sentences.  A new, more accurate dictionary is generated by only counting the remaining correspondences and filtering the counts as before.

Local coherence is enforced by computing an expected range for the translation of each word, and removing candidates outside of the range if there are also candidates inside the range.  The expected range is the range of possible correspondences for the two adjacent words, expanded by $N$ words in each direction.

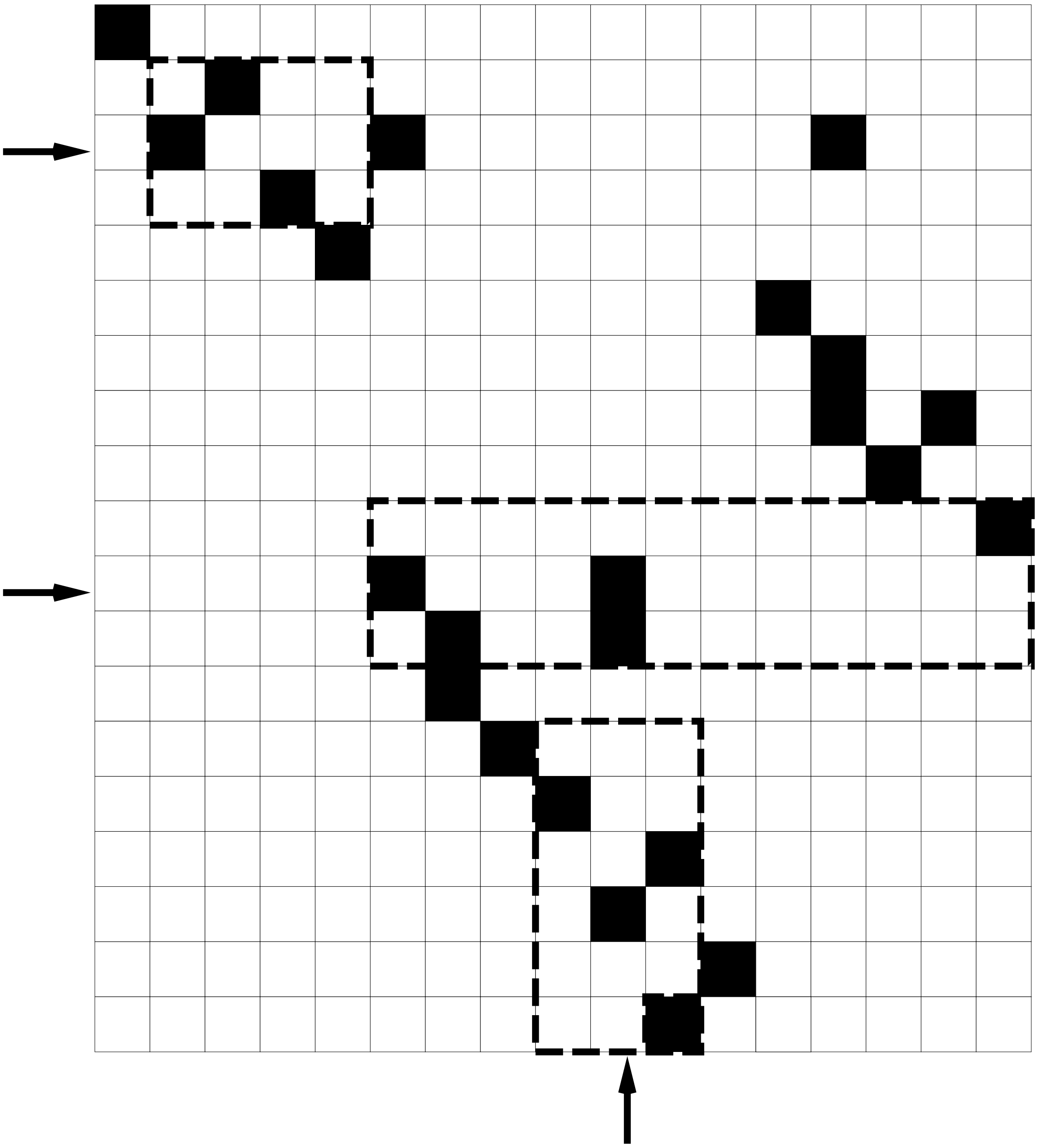
Small gaps are then filled on the assumption that any remaining words with no known translations probably correspond to words on the target-language side with no known correspondences.

# Refining the Dictionaries

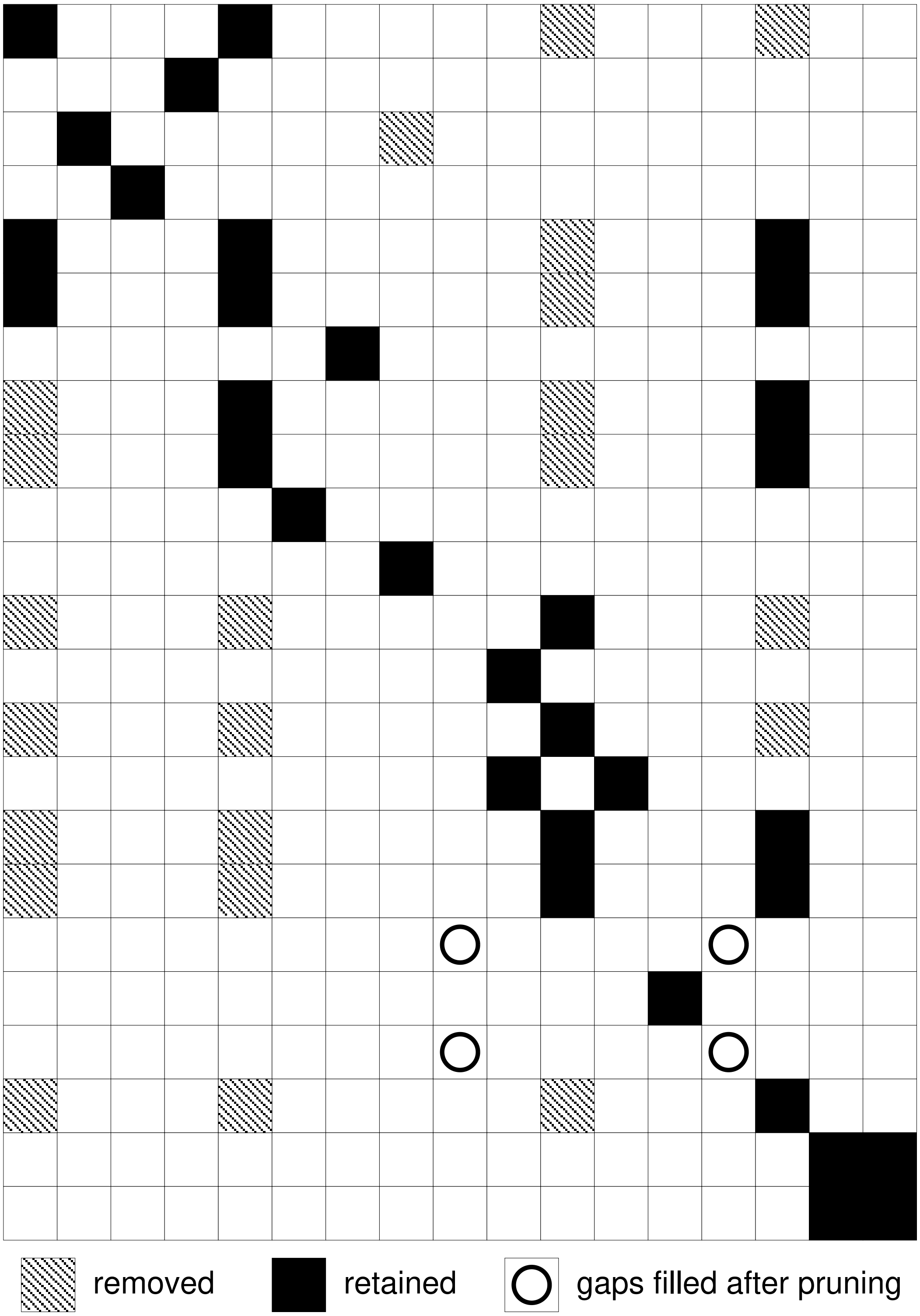


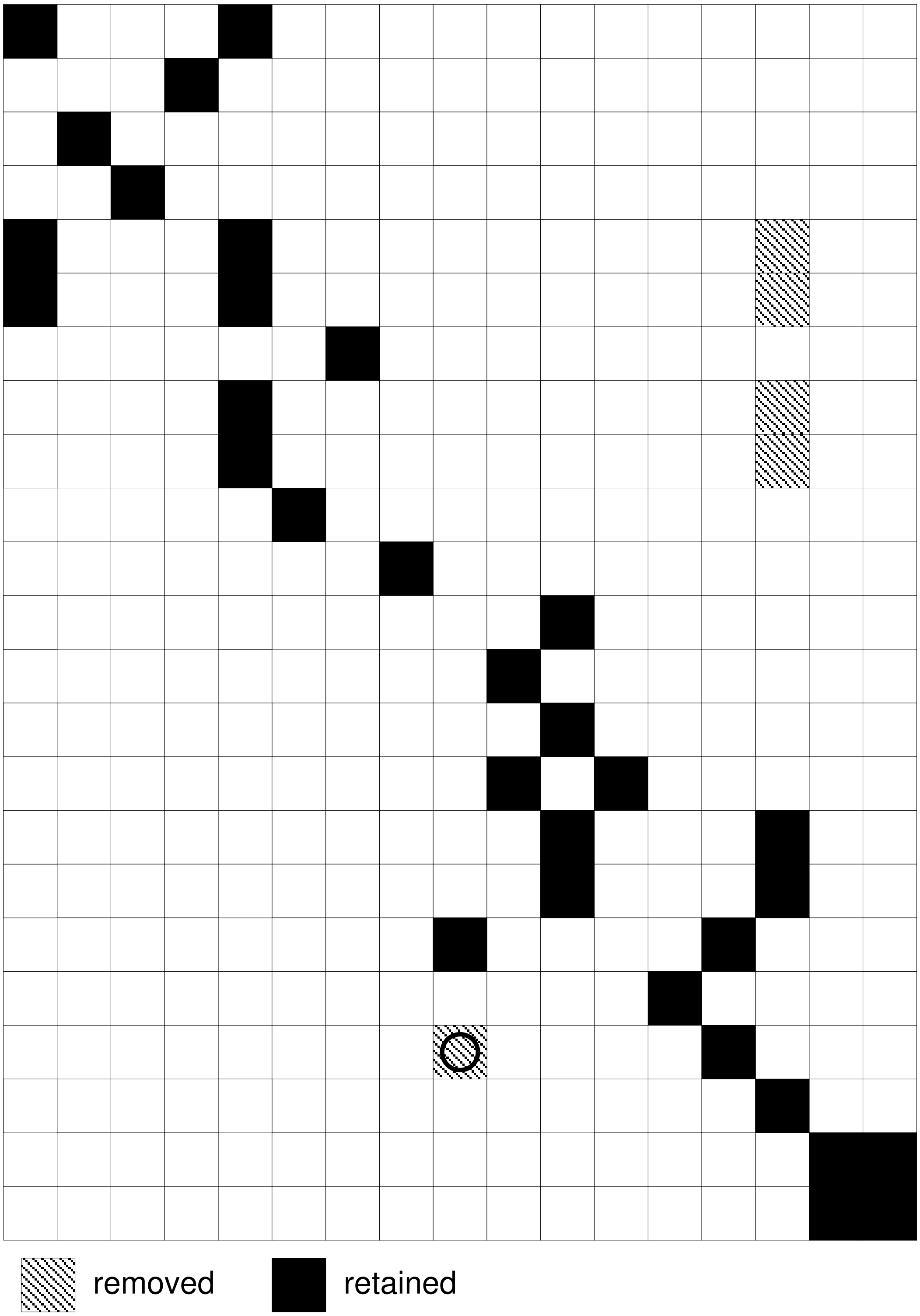



# Expected Ranges



# Pruning Correspondences




removed
retained

# Example-Based MT

Different levels of analysis possible:

- Sato (1984) parsed both source and target halves of corpus and did matching based on parse trees
- ReVerb
- Gaijin: full-sentence match by sequences of part-of-speech tags
- CMU: partial string matching and subsentential alignment
- Translation Memory

Translation units may also vary in size from words to entire paragraphs.

As with statistical systems, corpus will probably require cleaning prior to use.

# Lexical Example-Based Machine Translation

**Approach**:

Perform string matching against indexed corpus of bilingual examples, finding all possible phrasal matches (let MEMT sort things out later). For each match, perform a subsentential alignment to find the translation for that fragment of the input.

Advantages:

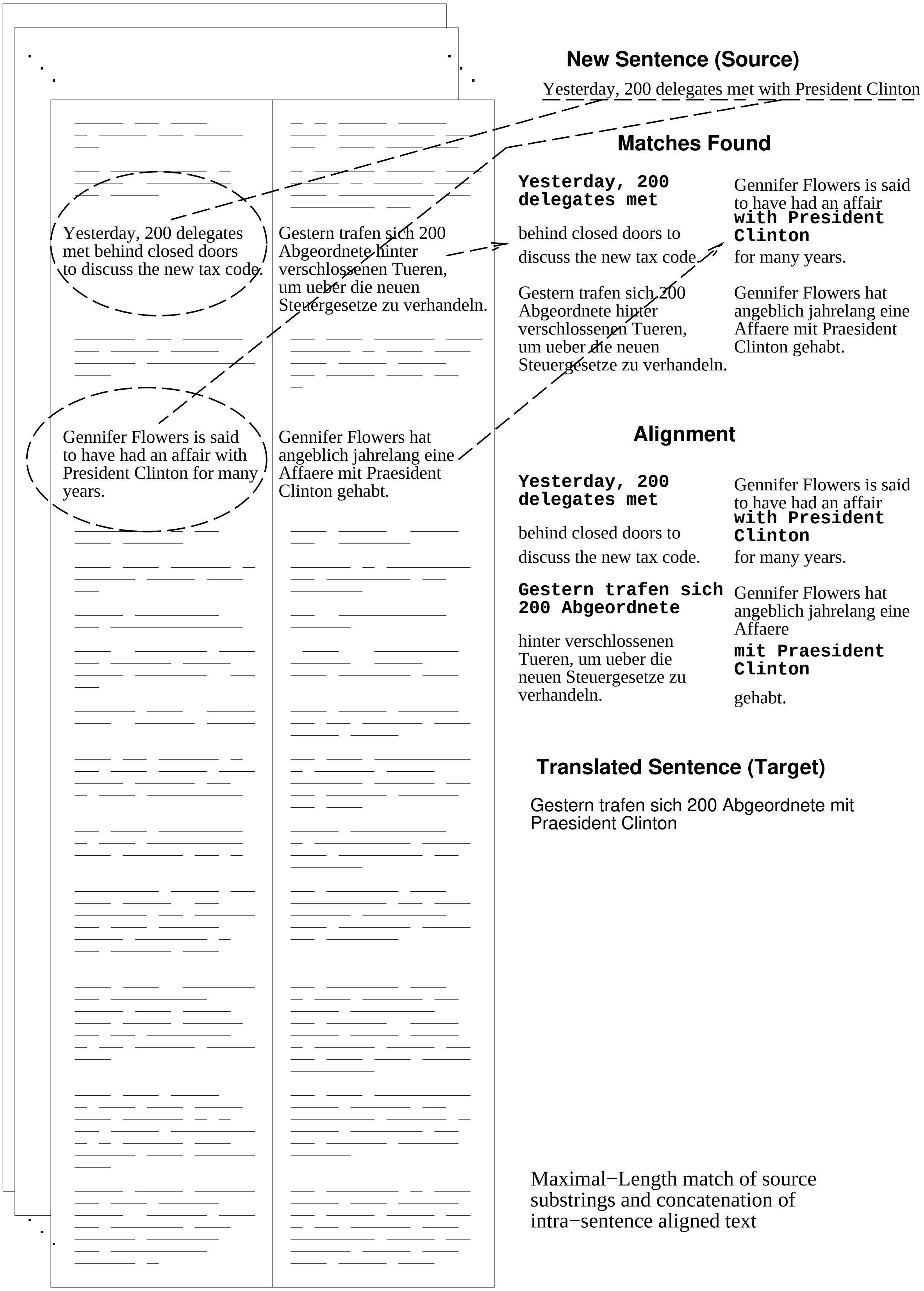
- Very quick to develop if sufficient parallel text available
- Can hire non-linguist translators to create more parallel text

Disadvantages:

- Needs **lots** of parallel text

To reduce data requirements, convert examples (and text to be translated) into templates, and perform string matches on templates.

## EBMT Paradigm



# Simple Matching

The simplest form of matching for EBMT is an exact match of surface forms between the sentence to be translated and the example base.

Thus,

200 delegates to the convention in London.

would completely match itself, but only the boldfaced portion of

`205` **delegates to the convention in** `Paris.`

After finding all exactly-matching phrases (single words are not matched), subsentential alignment is attempted to find the proper portion of the translated sentence for the source phrase. If this fails to find a sufficiently good alignment, the match is discarded and the next one examined.

# Subsentential Alignment

Alignment is driven by a bilingual dictionary, with a set of heuristic scoring functions to determine the best-matching fragment in the target language.

First, use the dictionary to build a table of possible correspondences, and try to find a least one word in the matched chunk that has a unique, unambiguous translation listed. Each such word forms the anchor for the alignment.

Next, scan left and right from the anchor in the target half until words are found which can’t possibly be translations. This yields the maximal substring to check, while the sequence between first and last anchors is the minimal substring to check.

For each possible substring between minimal and maximal segments, apply the heuristic scoring functions:

1. number of SL words without TL correspondence
2. number of TL words without SL correspondence
3. minimum of 1 and 2
4. number of possible correspondences outside segment
5. shared sentence boundaries
6. difference in length
7. elidable SL words
8. insertable TL words
9. number of equivalence-class markers (makes system prefer exact matches)
10. number of words excised

# Generalizing with Equivalence Classes

Since there are certain sets of words which can be freely substituted for each other yet still yield valid translations, we can improve the coverage of the system by converting the example base into templates using those equivalence classes.

Equivalence classes may be semantic:

- numbers
- days of the week
- names of cities
- colors
- shapes

or syntactic:

- masculine nouns
- plural adjectives
- etc.

More complex equivalences are also possible:

- **multiple words**: West Germany
- **patterns**:
  <month> <number>, <number> == <date>
  <honorific> <firstname> <lastname> == <person>

Occurrences of members of an equivalence class are replaced by the name of the equivalence class, so

```
Three people took the elevator to the
fourth floor, two to the fifth.
```

is treated as

```
<number>1 people took the elevator to the
<ordinal>1 floor, <number>2 to the
<ordinal>2.
```

# Equivalence Classes and Grammars

Training examples may be further generalized by replacing entire patterns which are, in some sense, equivalent. Thus, if all forms of a date are placed into a single equivalence class, the templates created from the example base will be more general than if the templates only contained the pattern for the particular form of a date expression that actually occurred in each example.

Since the token representing a pattern may itself be included in another pattern, we can form a paired production-rule grammar.

```
;;;(TOKEN <N-M>)
book
livre

;;;(TOKEN <NP-M>)
the <N-m>
le <N-m>

;;;(TOKEN <NP-M>)
<poss-m> <N-m>
<poss-m> <N-m>

;;;(TOKEN <NP-M>)
the <number> <N-m>
le <number> <N-m>

;;;(TOKEN <NP-M>)
the <adj-m> <N-m>
le <N-m> <adj-m>

;;;(TOKEN <NP-M>)
the <adj-m>1 <adj-m>2 <N-m>
le <N-m> <adj-m>2 <adj-m>1

;;;(TOKEN <NP-M>)
the <ordinal> <N-m>
le <N-m> <ordinal>

;;;(TOKEN <NP-M>)
the <national-m> <N-m>
le <N-m> <national-m>
```

To avoid over-generalization, the production rules can incorporate semantic or linguistic constraints through the appropriate use of tags:

```
    the affordable painters
       ==> the (<adj-s> <adj-p>) <noun-m-p>
       ==> the <adj-p> <noun-m-p>
       ==> <np-m>

       ==> <np-m>
       ==> los <noun-m-p> <adj-p>
       ==> los pintores accesibles
```

The grammar used by the EBMT system *need not be complete* -- even a few simple rules may suffice.  Any parts of the examples (and input to be translated) which is not covered by a grammar rule will be matched normally.

# Applying Grammar Rules

Grammar rules are applied in much the same way as tokenization patterns. The main differences are that matches are not unconditionally applied, and tokenization patterns take precedence over grammar rules.

When indexing, search the corpus for matching tagged entries (grammar rules and morphological data), and store the result obtained by substituting the tags from the matched entries. Matching/substituting is repeated until no more matches are found.

When translating, again search the corpus for matching tagged entries and repeat the match/substitute cycle until there are no more matches. However, keep all the intermediate results, and do a match of the entire lattice against the full corpus to perform the actual translation.

Example:

**The red ball bounced on the pavement.**

would permit any appropriate combination of the parts of the following patterns (and the original input) to be matched:

**<det>1 <color> <noun>1 bounced <prep> <det>2 <noun>2 .**
**<np> bounced <pp> .**

# Sample Output (1)

Input:

La motion de M. Lewis est adoptée par 147 voix contre 77.

(*Mr. Lewis' motion is adopted by 147 votes to 77*.)

707,000-word corpus, no generalization:

| | |
|---|---|
| "la motion de m . lewis" (1) | "motion of Mr . Lewis" |
| "adoptée par" (0) | "adopted by" |
| "147 voix" (0) | "147 votes" |
| "77 ." (0) | "77 ." |

307,000-word corpus, *with* generalization:

| | |
|---|---|
| "la motion de m ." (2.21) | "the motion for Mister" |
| "motion de m . lewis" (0.4) | "Mister Lewis's motion" |
| "est adoptée par 147" (1.825) | "is adopted by 147" |
| "par 147 voix contre 77 ." (1) | "by 147 voice against 77 ." |

# Sample Output (2)

Input:

Je crois que cela serait opportun.

(*I believe that that would be appropriate.*)

907,000-word corpus, no generalization:

| | |
|---|---|
| "je crois que cela" (0) | "I believe this" |
| "je crois que" (0) | "I believe that no" |
| "je crois" (0) | "I think" |
| "que cela" (0.5) | "That" |
| "que cela" (0.5) | "this" |
| "cela serait" (0) | "That would" |

407,000-word corpus, *with* generalization:

| | |
|---|---|
| "je crois que cela serait" (1) | "i believe that that would" |
| "je crois que cela" (0) | "i believe that that" |
| "je crois que" (0) | "i think this" |
| "crois que cela serait" (1.825) | "believe that that would" |
| "crois que cela" (0.75) | "believe that that" |
| "que cela serait opportun" (19.8) | "that should be opportune to" |
| "que cela serait" (0) | "that this should be" |
| "cela serait" (5.95) | "This should be" |
| "cela serait" (0) | "that would" |
| "serait opportun ." (5.4) | "should be opportune !" |
| "serait opportun" (4) | "opportune should be" |
| "serait opportun" (4) | "should be opportune" |
| "serait opportun" (0) | "should be" |
| "opportun ." (0) | "opportune !" |

Surprisingly, even with 10 million words of ungeneralized text, performance on the previous example is (overall) worse than with generalization!

| | |
|---|---|
| "je crois que cela" (0) | "I think this" |
| "je crois que" (0) | "I believe we" |
| "je crois" (0) | "I think I" |
| "que cela serait" (1) | "than That would" |
| "que cela serait" (0.5) | "that would" |
| "que cela" (0) | "than that" |
| "cela serait" (0) | "That would" |
| "opportun ." (0) | "appropriate ." |

# Automatic Equivalence Classes

Adding all of the information needed for converting examples into templates requires a considerable amount of manual labor. Can we determine that information automatically?

The answer is a qualified yes -- the manually-generated generalization data is better, but the EBMT system requires substantially less example text with automatically-determined generalization data than it would without any generalization.

Since equivalence classes are meant to contain words and phrases that can be used interchangeably, it is not unreasonable to expect the members of the equivalence classes to appear in similar contexts. Thus, clustering is an obvious approach.

# Word Clustering with Document-Clustering Techniques

**Observation**: if the context in which a word appears is defined as the sum of the words in the immediate neighborhoods of its occurrences, we can use standard document-clustering techniques to perform word clustering.

**Approach**: create a pseudo-document for each word, containing all the words surrounding its occurrences; make the word the document identifier.

**Problem**: this yields only a monolingual clustering, but we need a set of bilingual pairs.

**Solution**: use the approach of Barrachina and Vilar to inject bilingual information into the clustering.

# Injecting Bilingual Information into Monolingual Clustering

To generate a bilingual word pair:

1. use a bilingual dictionary to create a rough bi-text mapping between the source-language and target-language halves of a sentence pair.
2. whenever there is a unique correspondence indicated by the bi-text mapping, generate a bilingual word pair consisting of the word and its translation.
3. treat those word pairs as indivisible tokens in further processing

These bilingual word pairs also serve to provide a rough separation of a word into its senses, which aids the clustering process. The word pairs do not give us strict word senses since a particular sense may have multiple translations, and multiple senses may have the same translation.

# Creating Pseudo-Documents

For every unique bilingual word pair found:

1. create an empty pseudo-document with the word pair as its identifier
2. scan the corpus for the occurrences of the word pair; for each occurrence, add the $N$ words preceding the occurrence and the $N$ words following the occurrence to the pseudo-document
3. weight the total counts for each neighboring word inversely to its distance from the occurrence of the word-pair

Why weight the terms according to distance?

- The further away the neighbor is, the less likely it is to actually be linked.

# Pseudo-Document

Consider the word pair `cinq` + `five`:

```
<NUL> <NUL> Le cinq jours depuis la
<NUL> <NUL> The five days since the

elles commenceront en cinq jours . <NUL>
they will begin in five days.
```

| Word | Occur | Weight |
|---|---:|---|
| <NUL>(-3) | 1 | 0.333 |
| elles(-3) | 1 | 0.333 |
| <NUL>(-2) | 1 | 0.667 |
| commenceront(-2) | 1 | 0.667 |
| Le(-1) | 1 | 1.000 |
| en(-1) | 1 | 1.000 |
| jours(1) | 2 | 2.000 |
| depuis(2) | 1 | 0.667 |
| .(2) | 1 | 0.667 |
| la(3) | 1 | 0.333 |
| <NUL>(3) | 1 | 0.333 |

# Clustering

Several variations tried, subjectively best one used

- agglomerative (bottom-up) clustering:
    1. start with one cluster per pseudo-document
    2. repeatedly merge the two most similar clusters until similarity too low
- nearest-neighbor cosine similarity metric as cluster similarity measure

Clusters whose similarity is below a pre-set threshold will never be merged. The threshold varies according to the number of occurrences of the word pair -- the fewer occurrences contribute to the term vector, the more a coincidental appearance of the same word in the context raises the similarity score. Two singleton words achieve a similarity score of 0.321 (of 1.000) if just one of the immediately adjacent words is the same.

# Sample Clusters

| | |
|---|---|
| 238 | HISTOIRE HISTORY<br>ÉCONOMIE ECONOMY |
| 260 | CERTAINEMENT CERTAINLY<br>CERTAINEMENT SURELY<br>CERTES SURELY<br>JAMAIS NEVER<br>PAS NOT<br>PEUT-ÊTRE MAY<br>PROBABLEMENT PROBABLY<br>QUE ONLY<br>RIEN NOTHING<br>SÛREMENT CERTAINLY<br>SÛREMENT SURELY<br>VRAIMENT REALLY |
| 348 | CONSERVATEUR CONSERVATIVE<br>CONSERVATEUR TORY<br>DÉMOCRATIQUE DEMOCRATIC<br>DÉMOCRATIQUE NDP<br>LIBÉRAL LIBERAL |
| 522 | DERNIÈRES LAST<br>DERNIÈRES PAST<br>DERNIÈRES RECENT<br>PROCHAINES NEXT<br>QUELQUES FEW<br>QUELQUES SOME |
| 535 | AVONS HAVE<br>SOMMES ARE |
| 1375 | ÉLECTORALE CAMPAIGN<br>ÉLECTORALE ELECTION |

# Sample Clusters (2)

| | |
|---|---|
| 1386 | FÉDÉRALES-PROVINCIALES FEDERAL-PROVINCIAL<br>INDUSTRIELLES INDUSTRIAL<br>OUVRIÈRES LABOUR |
| 1528 | FAÇON EVENT<br>ÉVIDENCE CLEARLY<br>ÉVIDENCE OBVIOUSLY |
| 1563 | HOMMES POLITICIANS<br>PRISONNIERS PRISONERS |
| 1652 | RETOUR BACK<br>REVENIR BACK |
| 2008 | CONVENU AGREED<br>SIGNÉ SIGNED<br>VU SEEN |
| 2182 | AGRICOLE AGRICULTURE<br>ENTIER AROUND<br>ENTIER THROUGHOUT<br>OCCIDENTAL WESTERN |
| 2472 | AVEUGLES BLIND<br>CHAUSSURES SHOES<br>CONSTRUCTEURS BUILDERS<br>PENSIONNÉS PENSIONERS<br>RETRAITÉS PENSIONERS<br>VÊTEMENTS CLOTHING |
| 3539 | POISSON FISH<br>PORC PORK |

# Seeding Clusters

Word pairs may be assigned to initial equivalance classes to seed the clustering process.

Seeding has the advantage of

- preventing words which should not be treated equivalently from clustering together
- merging otherwise-separate clusters

The primary disadvantage of seeding is that it requires manual effort.

To determine effects of seeding, the top 500 words by frequency were matched against the dictionary used in manual generalization and filtered. The resulting seed sets had initial cluster assignments for

- French: 426 word pairs
- Spanish: 911 word pairs (multiple definitions per word)

# Ongoing Work: Grammar Induction

Since we have lots of translated examples, we can try to look for patterns bigger than a single word, and use those to form grammar rules.

**Observation**

If we have two nearly identical sentences, their difference is probably a valid constituent of the language:

`John hit the wall with` **his fist**`.`

`John hit the wall with` **a hammer**`.`

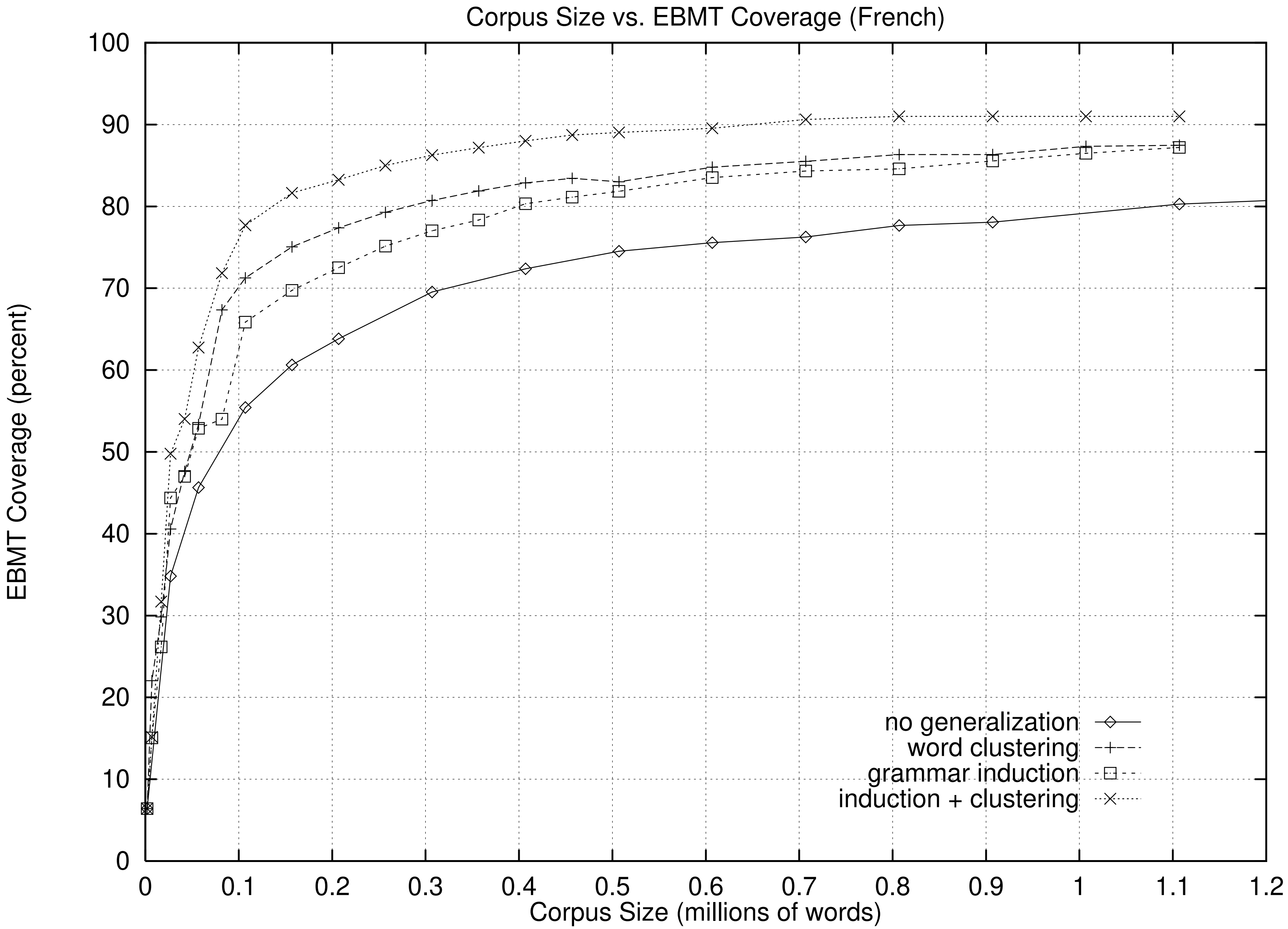
If the translations of those sentences also differ in only a small segment,

`John schlug mit` **seiner Faust** `die Wand.`

`John schlug mit` **einem Hammer** `die Wand.`

we can declare a bilingual equivalence class containing those differences:

```
begin <CL_1>
"his fist"  "seiner Faust"
"a hammer"  "einem Hammer"
end <CL_1>
```


Corpus Size vs. EBMT Coverage (French)
EBMT Coverage (percent)
Corpus Size (millions of words)
0
10
20
30
40
50
60
70
80
90
100
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.1
1.2
no generalization
word clustering
grammar induction
induction + clustering

# iKBMT

Instructible Knowledge-Based MT:
A new project at the LTI for taking EBMT's grammar induction to the extreme and form a set of transfer rules from an annotated corpus.

# Multi-Engine MT

Each translation technology has its strengths and weaknesses. What if we could combine them so that only the strengths of each are used, not the weaknesses?

We can also exploit the differences to achieve:

- General-Purpose MT
- Rapid-deployment MT: bring up initial system in days, gradually improve it over time
- "Anytime" MT for real-time applications

# MEMT Development

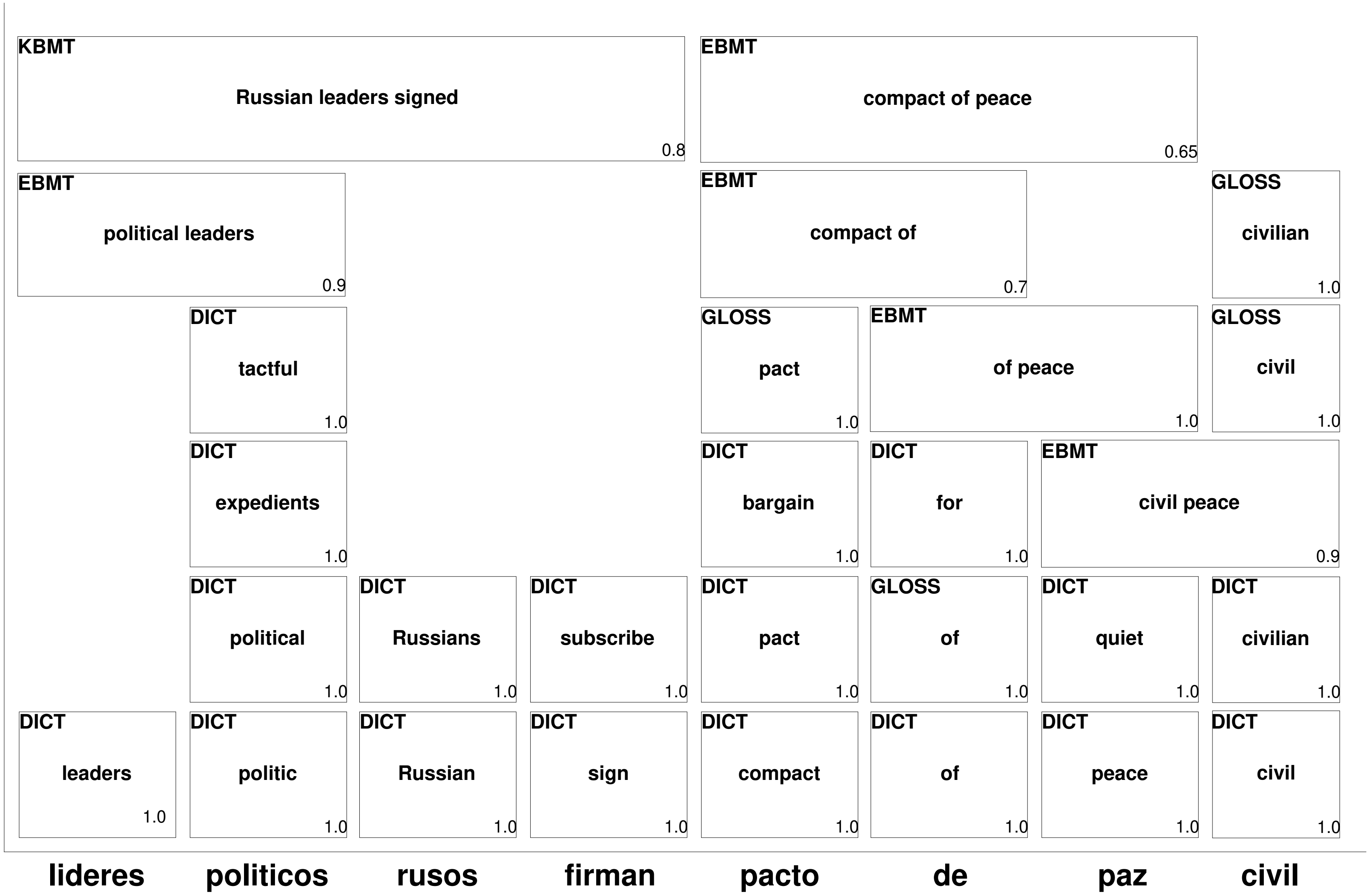
Started as a simple fall-back scheme (which is still used by many systems)

Initial chart-based version used dynamic programming

- assumes initial scores are correct and context-free
- exhaustive $N^3$ algorithm

Current implementation uses language modeling.

- technique borrowed from speech recognition
- can work with initial scores which are only approximately correct
- trigram model of target language


KBMT
Russian leaders signed
0.8
EBMT
compact of peace
0.65
EBMT
political leaders
0.9
EBMT
compact of
0.7
GLOSS
civilian
1.0
DICT
tactful
1.0
GLOSS
pact
1.0
EBMT
of peace
1.0
GLOSS
civil
1.0
DICT
expedients
1.0
DICT
bargain
1.0
DICT
for
1.0
EBMT
civil peace
0.9
DICT
political
1.0
DICT
Russians
1.0
DICT
subscribe
1.0
DICT
pact
1.0
GLOSS
of
1.0
DICT
quiet
1.0
DICT
civilian
1.0
DICT
leaders
1.0
DICT
politic
1.0
DICT
Russian
1.0
DICT
sign
1.0
DICT
compact
1.0
DICT
of
1.0
DICT
peace
1.0
DICT
civil
1.0
lideres
politicos
rusos
firman
pacto
de
paz
civil

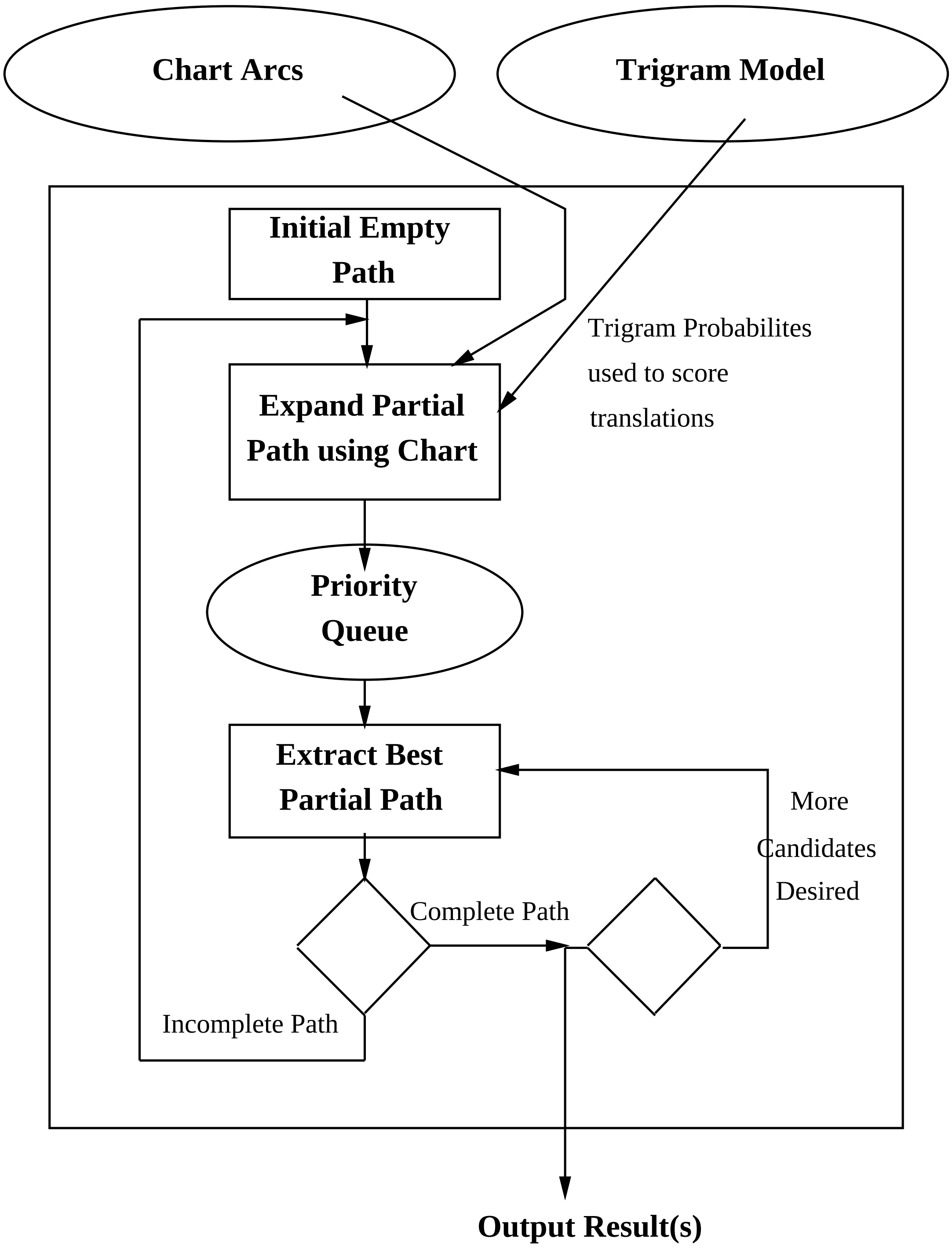


Chart Arcs
Trigram Model
Initial Empty Path
Trigram Probabilites used to score translations
Expand Partial Path using Chart
Priority Queue
Extract Best Partial Path
More Candidates Desired
Complete Path
Incomplete Path
Output Result(s)

# Weighting Factors

In addition to the trigram probabilities, the modeler uses

- translation engine weight
- quality-of-translation score assigned by translation engine
- bonuses for
  - longer translation units
  - agreement between engines
  - first alternative proposed by engine
  - use of certain preferred glossaries by Transfer engine
- penalties for
  - untranslated strings
  - excessive verbosity in translation
  - out-of-vocabulary words
  - null translations (eliding words from output)

# Arc Reordering

Since the Transfer engine operates on a word-for-word basis, differences in word order between source and target languages will greatly affect the trigram probabilities assign to partial paths. To compensate, the modeler performs some simple reordering as it constructs partial paths.

Sample Reordering Rules

| | |
|---|---|
| Noun Adj Adj | Adj Adj Noun |
| Noun Adj Conj Adj | Adj Conj Adj Noun |
| Clitic Verb | Verb Clitic |
| "not" AuxVerb | AuxVerb "not" |
| ". | ." |

These rearrangements ensure the applicability of the trigram model to the word-for-word translations despite differences in word order.

## Advantages + Disadvantages

MEMT's advantages:

- can exploit differences between MT technologies
  - general-purpose MT with high-quality subdomain
  - rapid-deployment MT
- engines can share data
- language model trained on monolingual data, which is much easier to acquire
- allows individual engines to be simpler

Disadvantages:

- not as accurate as KBMT
- hard to correctly select best edges
- selection based only on target language trigrams

# The SLT Project

(at SRI Cambridge)

- robust multi-engine anytime translation
- combines semantic translation and surface translation
- back-up surface translation rules from semi-automatic corpus-based method
- cheap translation methods run first -- gives "anytime" behavior of always having an answer available, with the quality improving given more processing time.