Machine Translation
WiSe 2016/2017

# Example-Based Machine Translation

*Dr. Mariana Neves*

*January 23th, 2017*

# Overview

- Example-Based Machine Translation (EBMT)
- EBMT‘s Workflow
- EMBT‘s Working
- EBMT vs. Case-Based Reasoning (CBR)
- Text Similarity
- Recombination
- Conclusions

# Overview

- Example-Based Machine Translation (EBMT)
- EBMT‘s Workflow
- EMBT‘s Working
- EBMT vs. Case-Based Reasoning (CBR)
- Text Similarity
- Recombination
- Conclusions

# Fundamental ideas about MT (Nagao 1984)

„Man does not translate a simple sentence by doing deep linguistic analysis.“

```
(ROOT
  (S
    (NP (PRP$ My) (NN dog))
    (ADVP (RB also))
    (VP (VBZ likes)
      (S
        (VP (VBG eating)
          (NP (NN sausage)))))
    (. .)))
```

(http://nlp.stanford.edu:8080/parser/)

# Fundamental ideas about MT (Nagao 1984)

„Man does the translation …

- first, by properly decomposing an input sentence into certain fragmental phrases […]
- then by translation these fragmental phrases into other language phrases,
- and finally by properly composing these fragmental translations into one long sentence.“

also

My dog

sausage

likes eating

# Fundamental ideas about MT (Nagao 1984)

„Man does the translation …

- first, by properly decomposing an input sentence into certain fragmental phrases […]
- then by translating these fragmental phrases into other language phrases,
- and finally by properly composing these fragmental translations into one long sentence.“

auch

Mein Hund

Wurst

isst gern

# Fundamental ideas about MT (Nagao 1984)

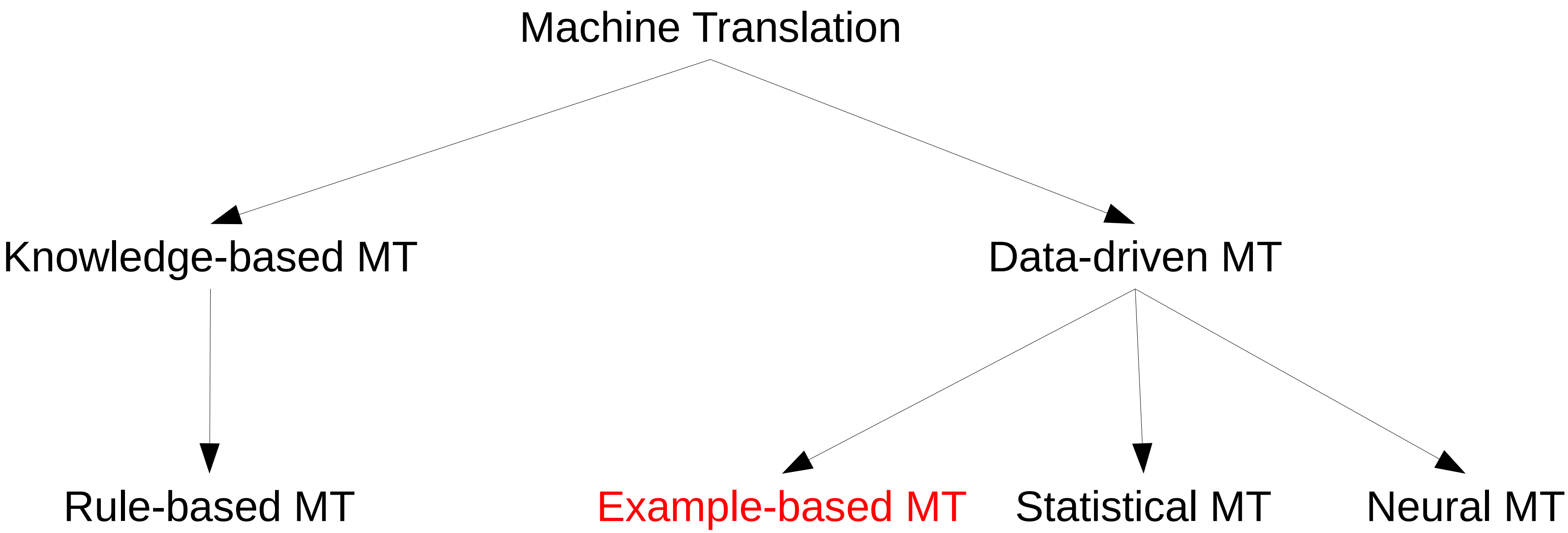
„Man does the translation …

- first, by properly decomposing an input sentence into certain fragmental phrases […]
- then by translation these fragmental phrases into other language phrases,
- and finally by properly composing these fragmental translations into one long sentence.“

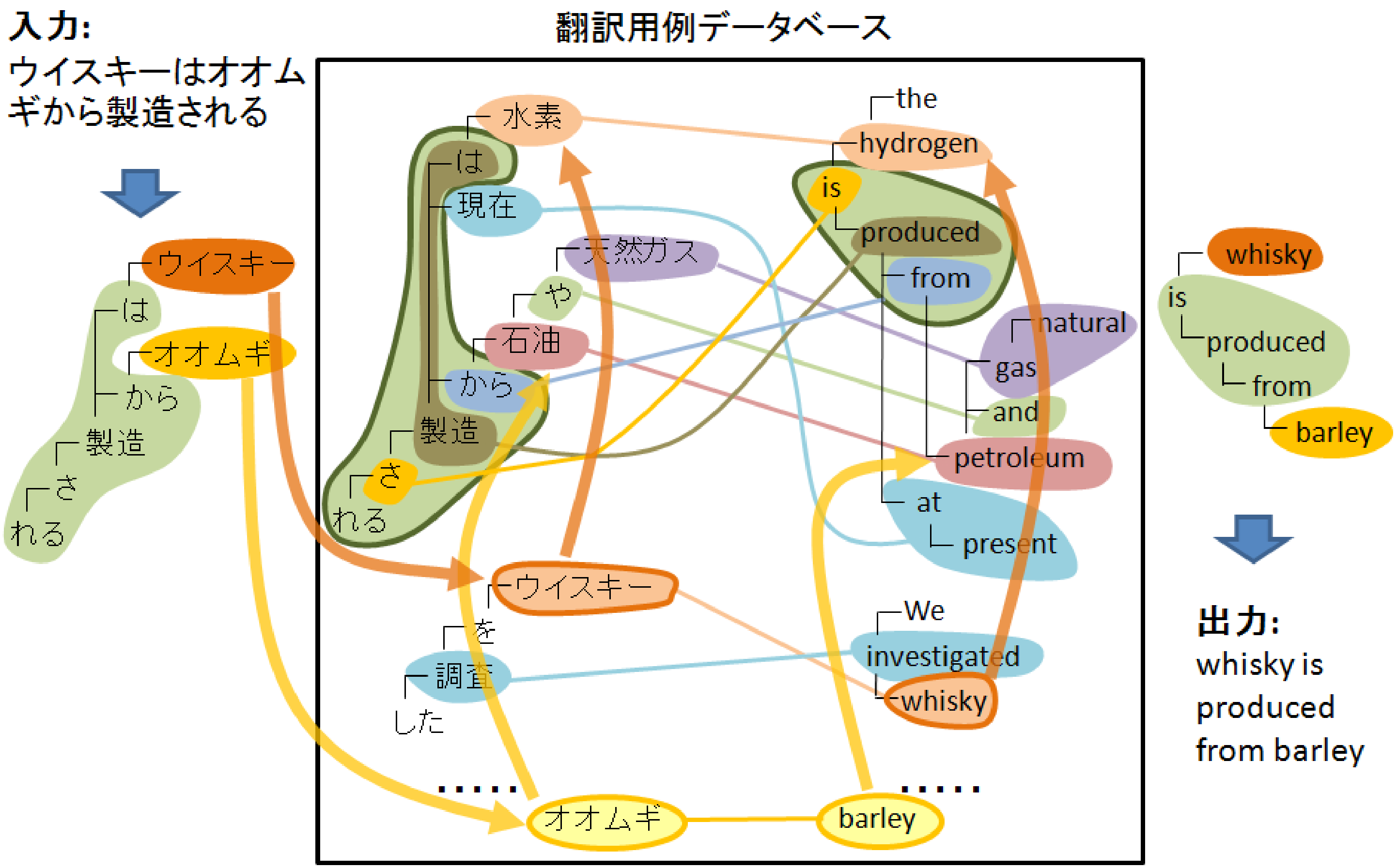
Mein Hund isst auch gern Wurst

# Example-Based MT

- Translation of fragmental phrases by <u>**analogy**</u>
- It is similar to SMT's decoding process



# Example-Based MT




(http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?plugin=attach&refer=KUROHASHI-KAWAHARA-LAB&openfile=EBMT.png)

# Example-Based MT (EBMT)

- Analogy (text similarity) is the key in EBMT

- Requirements for text similarity:
  - (1) measure of similarity: similar documents should be measured as similar, and vice-versa;
  - (2) large lexical knowledge networks to support similarity, e.g., WordNet, Wikipedia, etc.

# Processing of source sentences

Rule-Based MT
(meaning graphs)


Statistical MT
(phrases with
probabilities)

my dog ↔ mein Hund (p=0.75)
also ↔ auch (p=0.66)
..

Example-Based MT
(matched templates)

my dog ↔ mein Hund
also ↔ auch
sausage ↔ Wurst

(http://demo.ark.cs.cmu.edu/parse)

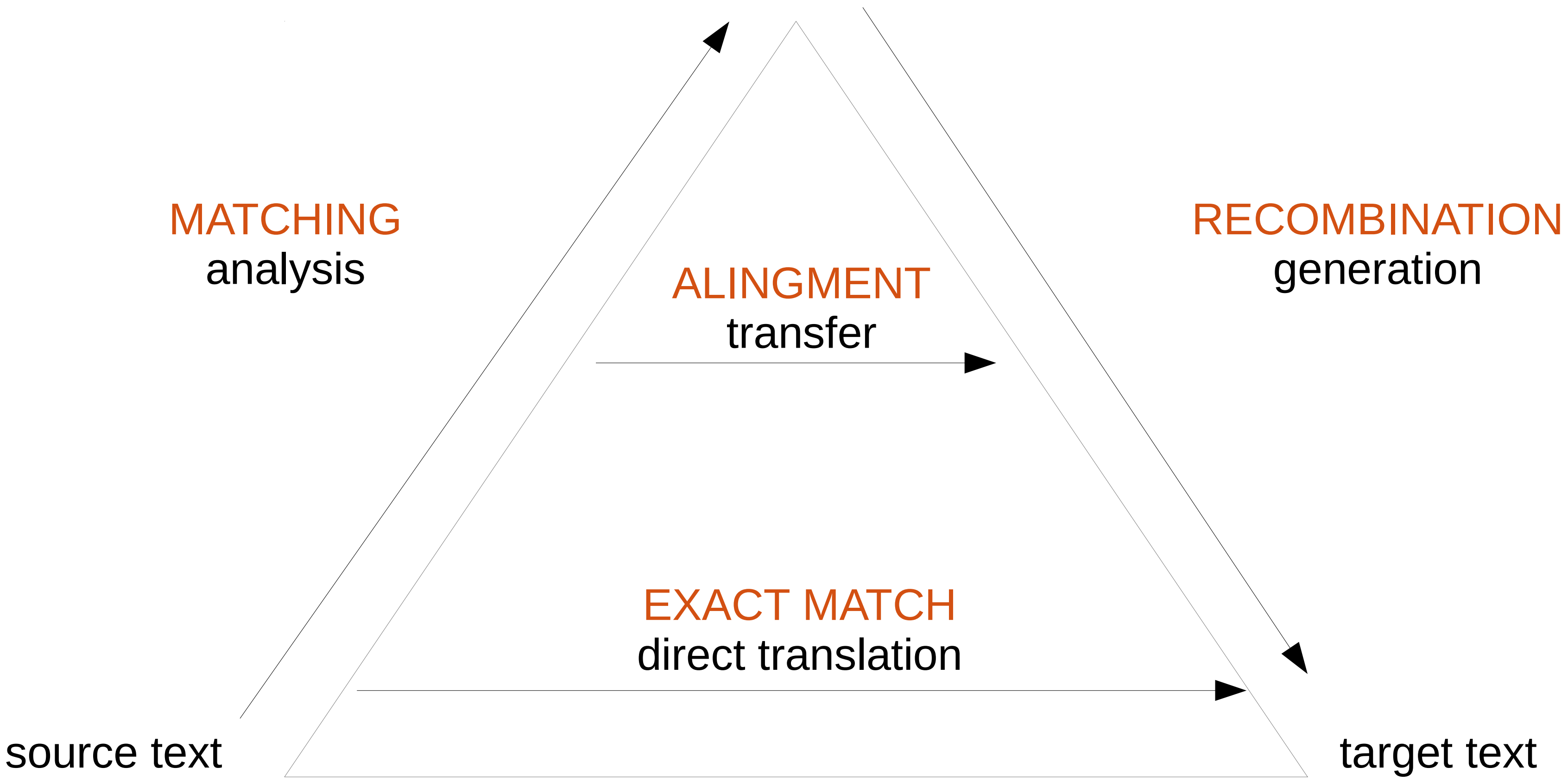
# Entry into the target language

- Rule-Based MT (interlingual): Late (top of the triangle)
- Statistical MT: immediate
- Example-Based MT (similar to RBMT transfer): intermediate level of NLP processing (middle of the triangle)

# Workflows

- Rule-Based MT
  - Lexical access, morphology generation, syntax planning
- Statistical MT
  - Mapping source sentence fragments, concatenation, scoring
- Example-Based MT
  - Template matching, recombination

# Vauquois triangle for EBMT



# Data-driven MT: EBMT vs. SMT

- Statistical MT
  - Probabilities to access the merit of candidates
  - Probabilities to rank candidates (decoding)
  - Stitching together translated fragments

- Example-based BMT
  - Similarity score between input fragments to fragments in database (most critical step!)
  - Syntactic and/or semantic similarity to rank candidates
  - NLP layers until (possibly) deep semantic analysis (closer to RBMT)
  - Stitching together translated fragments (closer to SMT)

# Overview

- Example-Based Machine Translation (EBMT)
- EBMT‘s Workflow
- EMBT‘s Working
- EBMT vs. Case-Based Reasoning (CBR)
- Text Similarity
- Recombination
- Conclusions

# Essential steps in EBMT

- Phrase fragment matching
- Translation of segments
- Recombination

# Base of examples

- He buys mangoes.
- Er kauft Mangos.

- This is a matter of international politics.
- Das ist ein Thema der internationalen Politik.

- They read a book.
- Sie lesen ein Buch.

# New sentence:

„He buys a book on international politics.“

| | |
|---|---|
| He buys | **Er kauft** (Mangos) |
| a book | (Sie lesen) **ein Buch** |
| international politics | (Das ist ein Thema der) **internationalen Politik** |

“Er kauf ein Buch (on) internationalen Politik”

# Essential questions

- Which sentences from the example base are useful?
- Which parts of the input sentence to match? (words, phrases, etc)
- How is the matching done efficiently?
- Should the matching be on instances or classes?
  (dog,cat vs. animal)

# Essential questions

- How are function words (articles, prepositions) inserted and where?

- Which function words are picked? (functional or non-functional)

# Overview

- Example-Based Machine Translation (EBMT)
- EBMT‘s Workflow
- EMBT‘s Working
- EBMT vs. Case-Based Reasoning (CBR)
- Text Similarity
- Recombination
- Conclusions

# Translation of a sentence

- (1) <u>Parts</u> of the new sentence match parts of existing sentences (similar to SMT)

| | |
|---|---|
| He buys | **Er kauft** (Mangos) |
| a book | (Sie lesen) **ein Buch** |
| international politics | (Das ist ein Thema der) **internationalen Politik** |

# Translation of a sentence

- (2) Properties of the new sentence match properties of existing sentences (linguistically-enriched SMT)

    - Morphosyntactic structures (POS tags, chunks)
    - Parse trees (constituent and dependency)
    - Semantic graphs (disambiguated words, semantic roles, speech acts)

# Parse tree similarity

```
(ROOT
  (S
    (NP (PRP He))
    (VP (VBZ buys)
      (NP
        (NP (DT a) (NN book))
        (PP (IN on)
          (NP (JJ international) (NNS politics)))))
    (. .)))
```

```
(ROOT
  (S
    (NP (PRP They))
    (VP (VBP read)
      (NP (DT a) (NN book)))
    (. .)))
```

```
(ROOT
  (S
    (NP (DT This))
    (VP (VBZ is)
      (NP
        (NP (DT a) (NN matter))
        (PP (IN of)
          (NP (JJ international) (NNS politics)))))
    (. .)))
```

```
(ROOT
  (S
    (NP (PRP He))
    (VP (VBZ buys)
      (NP (NNS mangoes)))
    (. .)))
```

# Word matching

- Explotation of semantic similarity

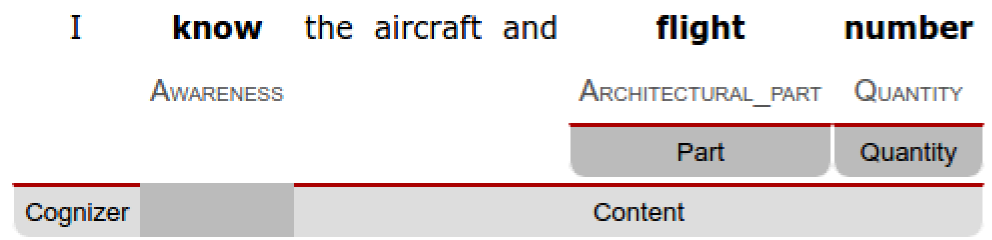
- Use of knowledge networks, such as WordNet
  - hypernyms: „mango“ and „fruit“
  - sisters: „water“ and „tea“

- Measures for semantic similarity: Resnick; Lin; Jiang and Conrath

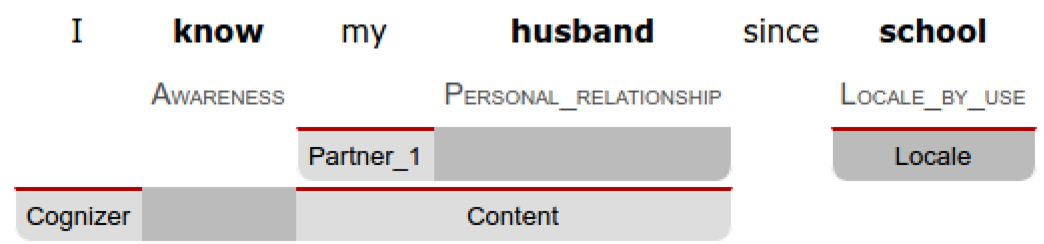
# Ambiguity (semantic role labeling)

- „know“ (English) to „kennen“ or „wissen“ (German)

I know the aircraft and flight number.



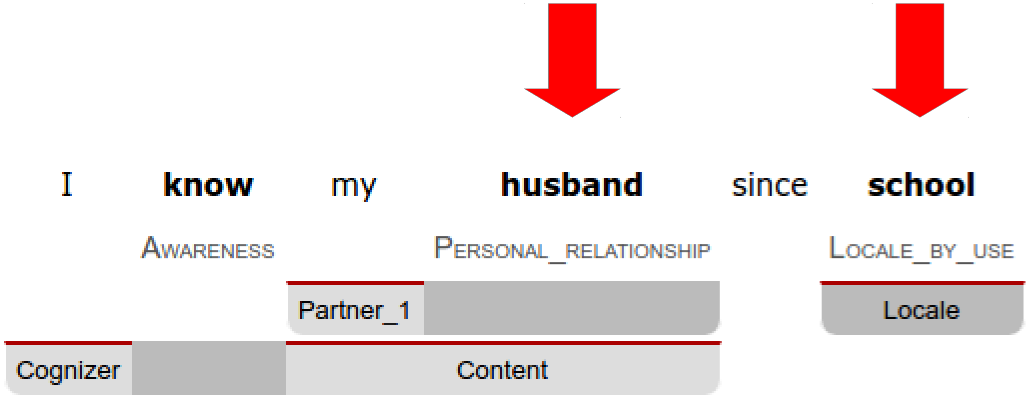

I know my husband since school




(http://demo.ark.cs.cmu.edu/parse)

# How does one know which words are important?

- Semantic role labeling to get the roles of the words



# EBMT is good for sublanguage phenomena

- e.g., learning various templates for possesives

| | | | |
|---|---|---|---|
| 子供の犬 | Children's dog | N1 の N2 | N1's N2 |
| 木の根 | Root of a tree | N1 の N2 | N2 of N1 |
| 本のページ | Pages in book | N1 の N2 | N2 in N1 |

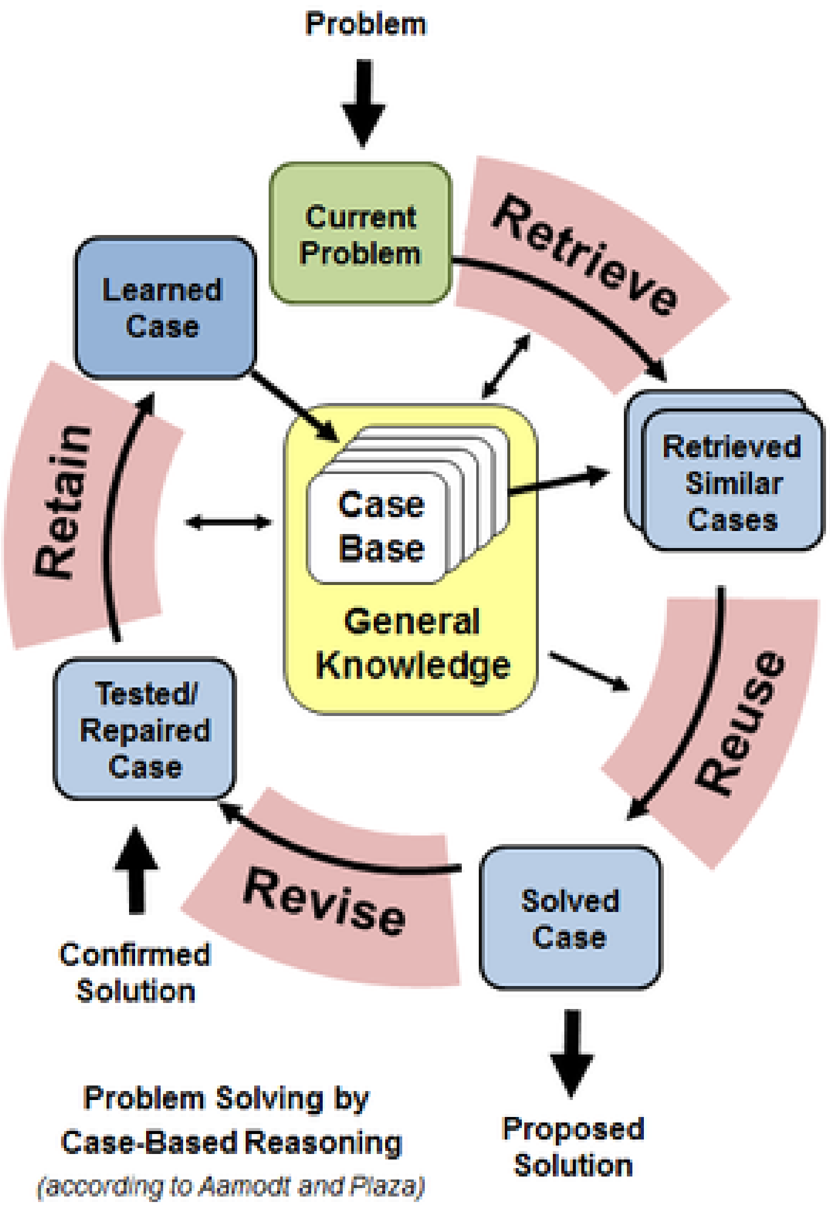
# Overview

- Example-Based Machine Translation (EBMT)
- EBMT‘s Workflow
- EMBT‘s Working
- EBMT vs. Case-Based Reasoning (CBR)
- Text Similarity
- Recombination
- Conclusions

# Case-based reasoning (CBR)

- CBR: learning by analogy
- EBMT: translation by analogy

# CBR




(https://www.uni-trier.de/index.php?id=44502&L=2)

# CBR and EBMT

(new sentence to be translated)

(matching rules exploiting features)

(new sentence-translation pair)

Case base: parallel corpora (example base)

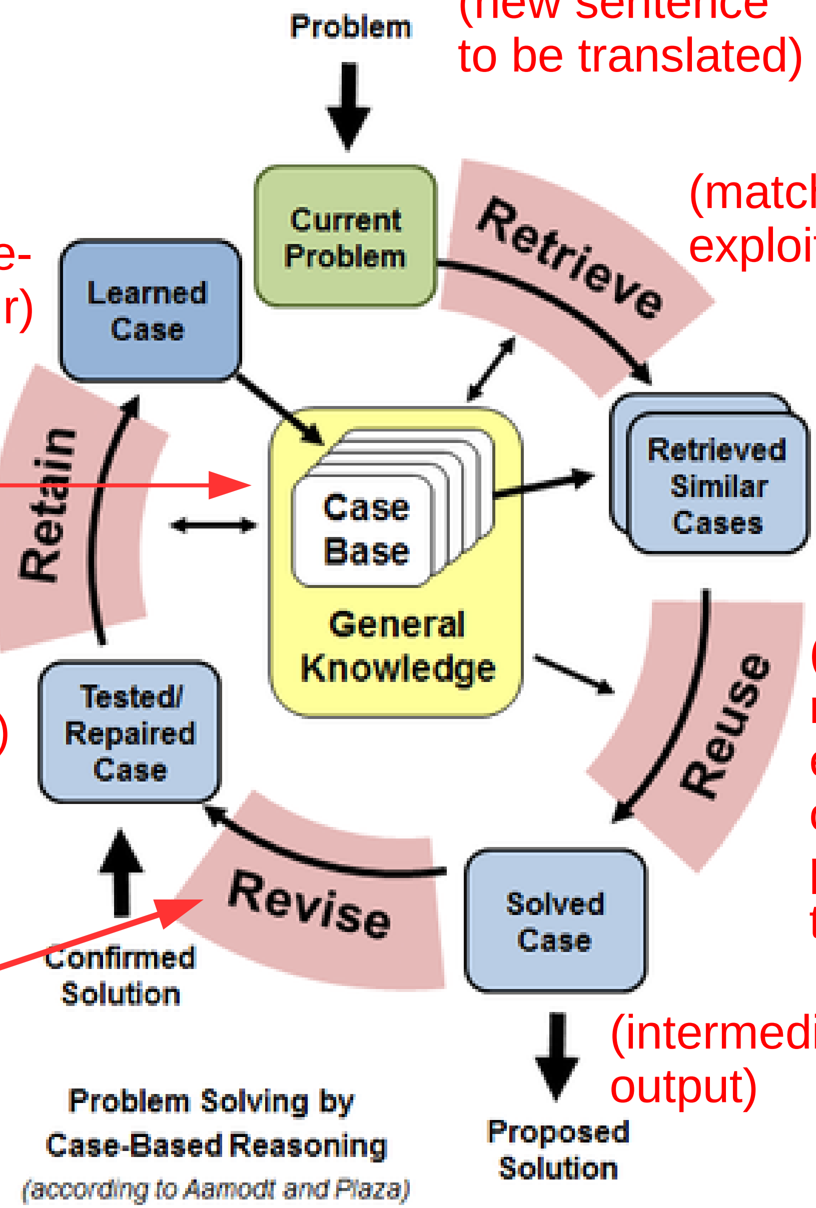
(matched sentences found by similarity)

(rules for recombination, e.g., extraction of sentence parts and templates)

(translation)

(recombination rules for solving boundaries friction)

(intermediate output)



# Overview

- Example-Based Machine Translation (EBMT)
- EBMT‘s Workflow
- EMBT‘s Working
- EBMT vs. Case-Based Reasoning (CBR)
- Text Similarity
- Recombination
- Conclusions

# Text similarity

- Computing similarity for a pair of sentences containing a set of words:

$$S_1 : u_1 , u_2 , u_3 ... u_n$$

$$S_2 : v_1 , v_2 , v_3 ... v_n$$

- Word-based similarity
- Tree and graph-based similarity
- CBR‘s similarity adapted to EBMT

# Word-based similarity

- Edit distance: number of insertions, deletions and substitutions to transform one sentence into another:
  - Levenshtein distance

| | | k | i | t | t | e | n |
|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| **s** | 1 | 1 | 2 | 3 | 4 | 5 | 6 |
| **i** | 2 | 2 | 1 | 2 | 3 | 4 | 5 |
| **t** | 3 | 3 | 2 | 1 | 2 | 3 | 4 |
| **t** | 4 | 4 | 3 | 2 | 1 | 2 | 3 |
| **i** | 5 | 5 | 4 | 3 | 2 | 2 | 3 |
| **n** | 6 | 6 | 5 | 4 | 3 | 3 | 2 |
| **g** | 7 | 7 | 6 | 5 | 4 | 4 | 3 |

| | | S | a | t | u | r | d | a | y |
|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| **S** | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| **u** | 2 | 1 | 1 | 2 | 2 | 3 | 4 | 5 | 6 |
| **n** | 3 | 2 | 2 | 2 | 3 | 3 | 4 | 5 | 6 |
| **d** | 4 | 3 | 3 | 3 | 3 | 4 | 3 | 4 | 5 |
| **a** | 5 | 4 | 3 | 4 | 4 | 4 | 4 | 3 | 4 |
| **y** | 6 | 5 | 4 | 4 | 5 | 5 | 5 | 4 | 3 |

(https://en.wikipedia.org/wiki/Levenshtein_distance)

# Word-based similarity

- Bag of words: unordered set of words of each sentence

$$B(S_1)=\{u_1, u_2, u_3 \ldots u_n\}$$

$$B(S_2)=\{v_1, v_2, v_3 \ldots v_n\}$$

$$Dice(B(S_1), B(S_2))=\frac{|B(S_1) \cap B(S_2)|}{|B(S_1)|+|B(S_2)|}$$

$$Jackard(B(S_1), B(S_2))=\frac{|B(S_1) \cap B(S_2)|}{|B(S_1) \cup B(S_2)|}$$

# Word-based similarity

- Bag of words: unordered set of words of each sentence

$S_1$ = “Peter hired a car for the trip.”
$S_2$ = “For the trip, a car was hired by Peter.”

$$Dice(B(S_1), B(S_2)) = \frac{7}{7+9}$$

$$Jackard(B(S_1), B(S_2)) = \frac{7}{9}$$

# Word-based similarity

- Vector-based similarity: used for information retrieval and based on the vocabulary of the corpus

$$V(S_1)=\{0,0,1,0,\ldots 1\}$$

$$V(S_2)=\{1,0,0,0,\ldots 1\}$$

$$cosine(V(S_1),V(S_2))=\frac{V(S_1)\cdot V(S_2)}{|V(S_1)|\cdot|V(S_2)|}$$

(dot product)

(scalar product)

# Word-based similarity

- Vectors with term frequencies: vectors of integers instead of binaries

$$V(S_1) = \{0,0,2,0,\ldots 1\}$$

$$V(S_2) = \{4,0,0,0,\ldots 2\}$$

$$cosine(V(S_1), V(S_2)) = \frac{V(S_1) \cdot V(S_2)}{|V(S_1)| \cdot |V(S_2)|}$$

# Word-based similarity

- Vectors with TF and IDF: term frequencies (TF) and inverse document (sentence) frequencies (IDF)

$$idf(w) = \log\left(\frac{N}{|S; w \in S|}\right)$$

Each component of $V(S_1)$ and $V(S_2)$ is then given by:

$$tf(w_p \in S_1) \cdot idf(w_p)$$

# Tree and graph-based similarity

- Based on constituency and dependency parse trees of $S_1$ and $S_2$ .

```
(ROOT
  (S
    (NP (PRP$ My) (NN dog))
    (ADVP (RB also))
    (VP (VBZ likes)
      (S
        (VP (VBG eating)
          (NP (NN sausage)))))
    (. .)))
```

```
nmod:poss(dog-2, My-1)
nsubj(likes-4, dog-2)
advmod(likes-4, also-3)
root(ROOT-0, likes-4)
xcomp(likes-4, eating-5)
dobj(eating-5, sausage-6)
```

(http://nlp.stanford.edu:8080/parser/)

# Tree and graph-based similarity

- For constituency matching, the non-terminals and terminals of the two trees should match when the trees are transversed in identical order.

```
(ROOT
  (S
    (NP (PRP He))
    (VP (VBZ buys)
      (NP
        (NP (DT a) (NN book))
        (PP (IN on)
          (NP (JJ international) (NNS politics)))))
    (. .)))
```

```
(ROOT
  (S
    (NP (PRP He))
    (VP (VBZ buys)
      (NP (NNS mangoes)))
    (. .)))
```

# Tree and graph-based similarity

- Constituency tree similarity:

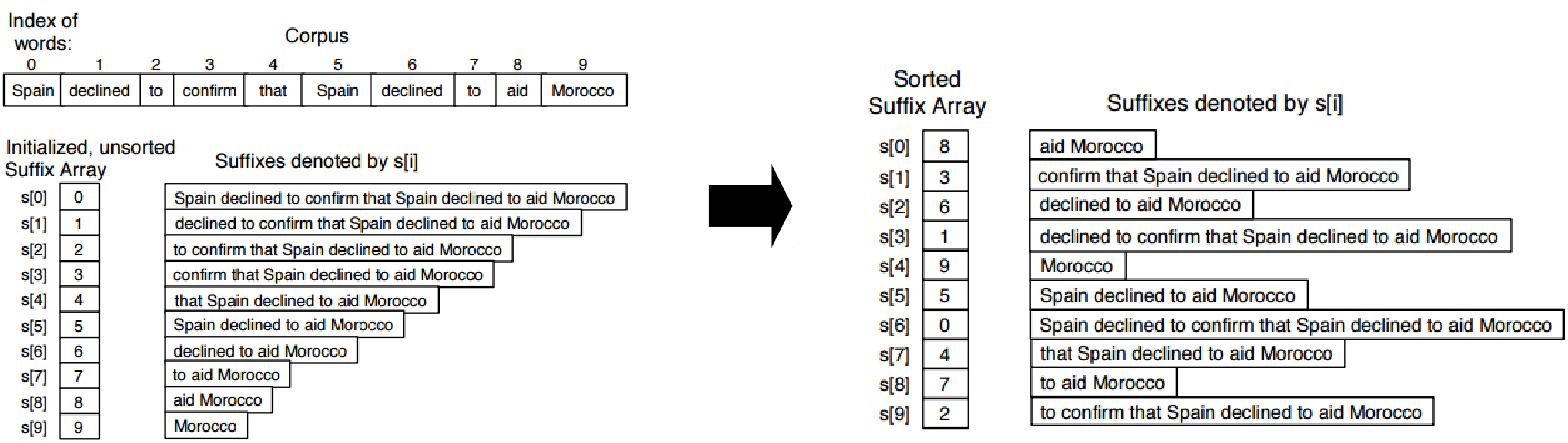
$$S(S_1, S_2) = \frac{M}{max(N_1, N_2)}$$

$N_1$ is the number of nodes in $S_1$ constituency tree;
$N_2$ is the number of nodes in $S_2$ constituency tree;
M is the number of nodes matched in a particular traversal order.

If $S(S_1,S_2)$ is above a certain threshold, the sentences are considered similar.

# Tree and graph-based similarity

- Time complexity is linear to the length of the longer S.
- For performance issues, suffix trees is usually employed:




(https://www.cs.jhu.edu/~ccb/publications/compact-data-structure-for-searchable-translation-memories.pdf)

# Tree and graph-based similarity

- For dependency tree matching, we compare the relations and their arguments

nsubj(buys-2, he-1)
root(ROOT-0, buys-2)
det(book-4, a-3)
dobj(buys-2, book-4)
case(politics-7, on-5)
amod(politics-7, international-6)
nmod(book-4, politics-7)

⟺

nsubj(buys-2, he-1)
root(ROOT-0, buys-2)
dobj(buys-2, mangoes-3)

# Tree and graph-based similarity

- Dependency tree matching is given by a weighted matching
  - Highest weight for predicate/relation
  - Less weight for the two arguments

$$w_r + w_{arg\,1} + w_{arg\,2} = 1$$

$$S(S_1, S_2) = \frac{\sum_{i=1}^{|D_1|} \sum_{j=1}^{|D_2|} [w_r \delta(R_i^1, R_j^2) + w_{arg\,1} \delta(A_1^i, A_2^i) + w_{arg\,2} \delta(B_1^j, B_2^j)]}{max(D_1, D_2)}$$

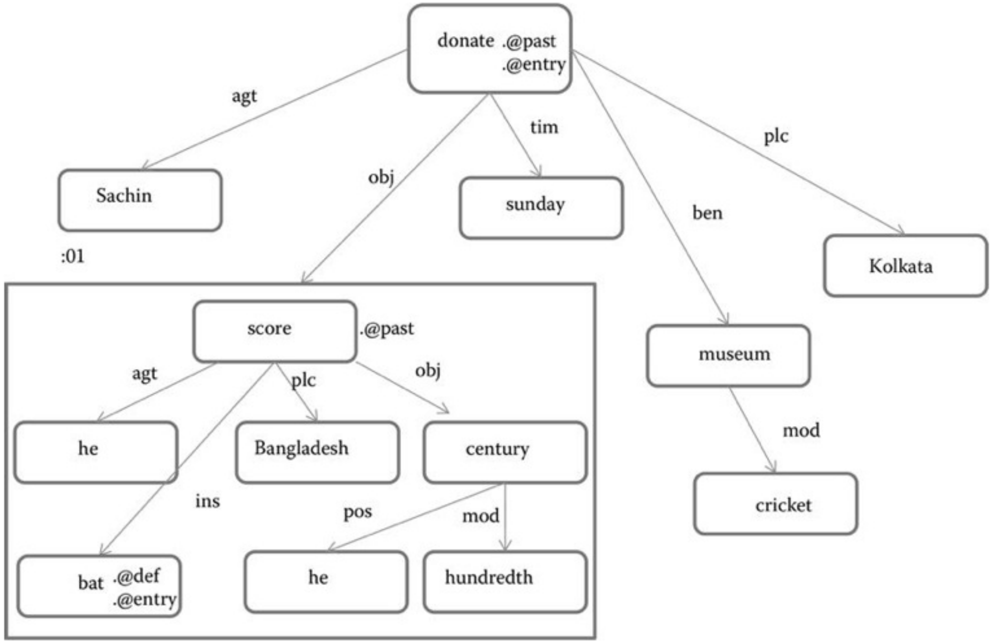
$$\delta(x, y) = 1, if\ (x = y)$$

$D_1$: no. of relations in $S_1$
$D_2$: no. of relations in $S_2$
$R_i^1$: ith relation in $S_1$
$R_j^2$: jth relation in $S_2$

$A_1^i$: first argument of the ith relation in $S_1$
$A_2^j$: second argument of the ith relation in $S_1$
$B_1^i$: first argument of the jth relation in $S_2$
$B_2^j$: second argument of the jth relation in $S_2$

# Tree and graph-based similarity

- Deep semantic graph-based similarity: deep semantic include sense disambiguation, relations, speech acts, co-references, etc.




- An advanced similarity can be calculated based on such graphs

(Figure from Bhattacharyya 2015)

# CBR‘s similarity adapted to EBMT

- The similarity below is a common measure in CBR:

$$S(I,R)=\frac{\sum_{i=1}^{n} w_i \times s\left(f_i^I, f_i^R\right)}{\sum_{i=1}^{n} w_i}$$

I: input case (sentence)
R: retrieved case (sentence)

*f* are features on which *s* operates as similarity measure with respective weighting function *w*

# Sentence features and similarities

| | | | |
|---|---|---|---|
| 1 | Length | Integer | Equality |
| 2 | Active/passive | 1 (active)/0 (passive) | Equality |
| 3 | Parse tree | – | Tree similarity between two parse trees |
| 4 | Concatenation of vectors of words forming the sentence | Vector of Boolean/real values | Cosine similarity |
| 5 | Bag of words forming the sentence | Set | Dice/Jackard and such other similarity measures |
| 6 | Position of nouns of the sentence in the wordnet hypernymy hierarchy | A function combining the *information content* of the individual nouns | Equality |
| 7 | Position of the two main verbs of the sentence in Verb Ocean | Distance between the two main verbs in Verb Ocean[a] | A rule that says similar or dissimilar, depending on the distance being within a threshold or not |
| 8 | Main verb, its type and argument frame as given by the VerbNet,[b] types of nouns semantically related to it | A slot-filler structure for each sentence | Equality or subset check on the slots and their fillers |
| 9 | Frame semantic representation of the sentence as per Framenet[c] | Slot-filler structure | Equality or subset check on the slots and their fillers |

(Table from Bhattacharyya 2015)

# Sentence features and similarities

| SEMANTIC RELATION | EXAMPLE | *Transitive* | *Symmetric* | *Num in* VERBOCEAN |
|---|---|---|---|---|
| *similarity* | produce :: create | Y | Y | 11,515 |
| *strength* | wound :: kill | Y | N | 4,220 |
| *antonymy* | open :: close | N | Y | 1,973 |
| *enablement* | fight :: win | N | N | 393 |
| *happens-before* | buy :: own; marry :: divorce | Y | N | 4,205 |

| | | | |
|---|---|---|---|
| 7 | Position of the two main verbs of the sentence in Verb Ocean | Distance between the two main verbs in Verb Ocean[a] | A rule that says similar or dissimilar, depending on the distance being within a threshold or not |

(http://demo.patrickpantel.com/demos/verbocean/)

# Sentence features and similarities

**Roleset id: abandon.01 , *leave behind*, Source: , vncls: , framnet:**

**abandon.01**: ABANDON-V NOTES: Verbnet class leave-51.2, other framed members include leave. Comparison with 'leave'. (from abandon.01-v)

**Aliases:**

| Alias | FrameNet | VerbNet |
|---|---|---|
| **abandon** (v.) | Quitting_a_place Departing | |

**Roles:**

**Arg0-PPT**: *abandoner* (vnrole: 51.2-theme)
**Arg1-DIR**: *thing abandoned, left behind*
**Arg2-PRD**: *attribute of arg1*

| | | | |
|---|---|---|---|
| 8 | Main verb, its type and argument frame as given by the VerbNet,[b] types of nouns semantically related to it | A slot-filler structure for each sentence | Equality or subset check on the slots and their fillers |

(http://verbs.colorado.edu/propbank/framesets-english-aliases/abandon.html)

(https://framenet.icsi.berkeley.edu/fndrupal/about)

# Sentence features and similarities

## Abandonment

**Definition:**

An Agent leaves behind a Theme effectively rendering it no longer within their control or of the normal security as one's property.

Carolyn ABANDONED her car and jumped on a red double decker bus.

Perhaps he LEFT the key in the ignition

ABANDONMENT of a child is considered to be a serious crime in many jurisdictions.

There are also metaphorically used examples:

She LEFT her old ways behind .

**FEs:**

**Core:**

| | |
|---|---|
| Agent [Age] | The Agent is the person who acts to leave behind the Theme. |
| Theme [The] | The Theme is the entity that is relinguished to no one from the Agent's possession. |

**Non-Core:**

| | |
|---|---|
| Degree [] | The extent to which the Agent leaves the Theme behind. |
| Depictive [] | The FE Depictive describes the Agent during the abandoning event. |

| | | | |
|---|---|---|---|
| 9 | Frame semantic representation of the sentence as per Framenet[c] | Slot-filler structure | Equality or subset check on the slots and their fillers |

# Overview

- Example-Based Machine Translation (EBMT)
- EBMT‘s Workflow
- EMBT‘s Working
- EBMT vs. Case-Based Reasoning (CBR)
- Text Similarity
- Recombination
- Conclusions

# Recombination

- Once examples are retrieved from the database, their translation need to be adapted to produce the target output

- Based on sentence parts
- Based on properties of sentence parts
- Based on parts of semantic graphs

# Recombination based on Sentence Parts

- Null adaptation: exact match of sentences.

- Reinstantiation: Input and example are structurally similar but differ in values of elements.

Input: “Tomorrow, today will be yesterday.”

Example: “Yesterday, today was tomorrow.”

(Tomorrow, today and yesterday are hyponyms of day.)

(will be and was both derived from the verb to be)

# Recombination based on Sentence Parts

- Adjustment (boundary friction) in predicates and matching arguments are necessary

Input: “Tomorrow, today will be yesterday.”

Example: “Yesterday, today was tomorrow.”

Example translation: “Gerstern, heute war morgen.”

Output translation: “Morgen, heute ist gerstern.”

# Recombination based on Properties of Sentence Pairs

- Abstraction and respecialization:
  - When input and example differ in small parts

  - Look for abstraction of small pieces
  - Try specialization of the abstraction
  - Need a hierarchical organization of concepts

# Recombination based on Properties of Sentence Pairs

- Abstraction and respecialization (example):

Input: “Wir malen die Wand.”

Example: “Wir malen die Mauer.”

Example translation: “We paint the wall.”

Output: “We paint the wall.”

# Recombination based on Properties of Sentence Pairs

- Case-based substitution: there are matches in the attributes of the words of the input sentence and the example sentence.
  - Properties can be features such as word, lemma, gender, number (singular/plural), person (3rd, 2nd), tense (past,future), voice (passive,active), POS tag, etc.

# Recombination based on Properties of Sentence Pairs

- Case-based substitution (example):

Input: “The new museum was inaugurated.”

Example 1: “The new stadium was inaugurated.”

Example 2: “The stadium is new.”

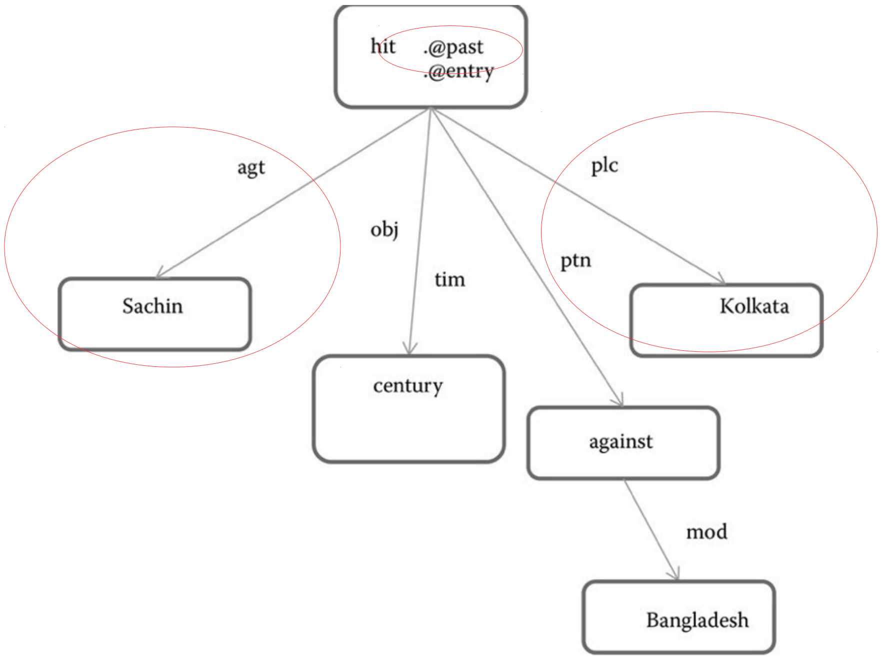
# Recombination based on Parts of Semantic Graph

- Correspondences that do not appear in linear sequences show up clearly in syntax trees and semantic graphs.

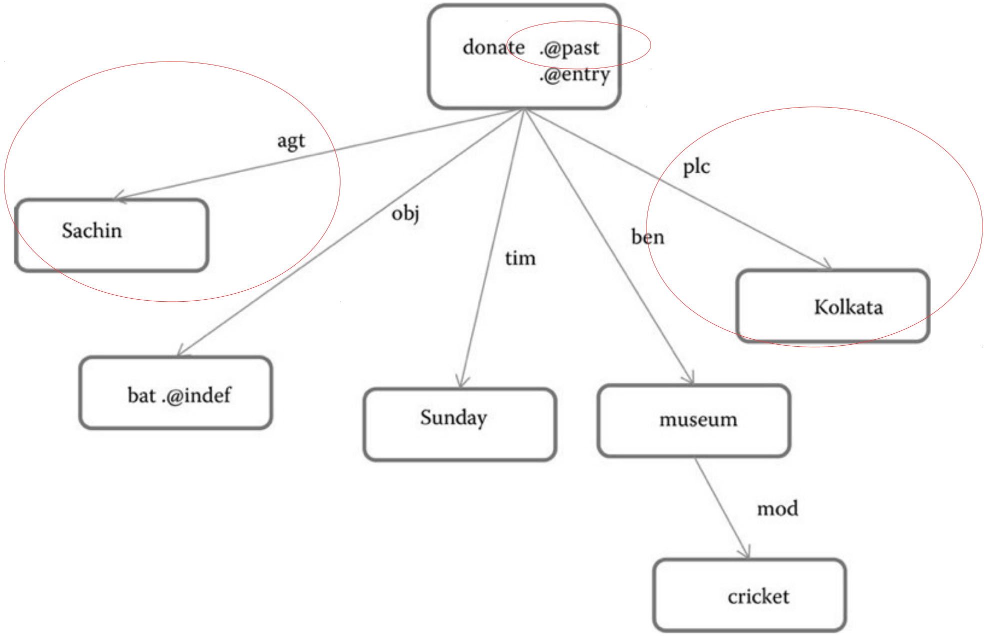
Input: “Sachin hit a century in Kolkata against Bangladesh.”

Example: “In Kolkata, Sachin donated a bat to the cricket museum on Sunday.”

# Recombination based on Parts of Semantic Graph




(input sentence)




(example sentence)

(Figures from Bhattacharyya 2015)

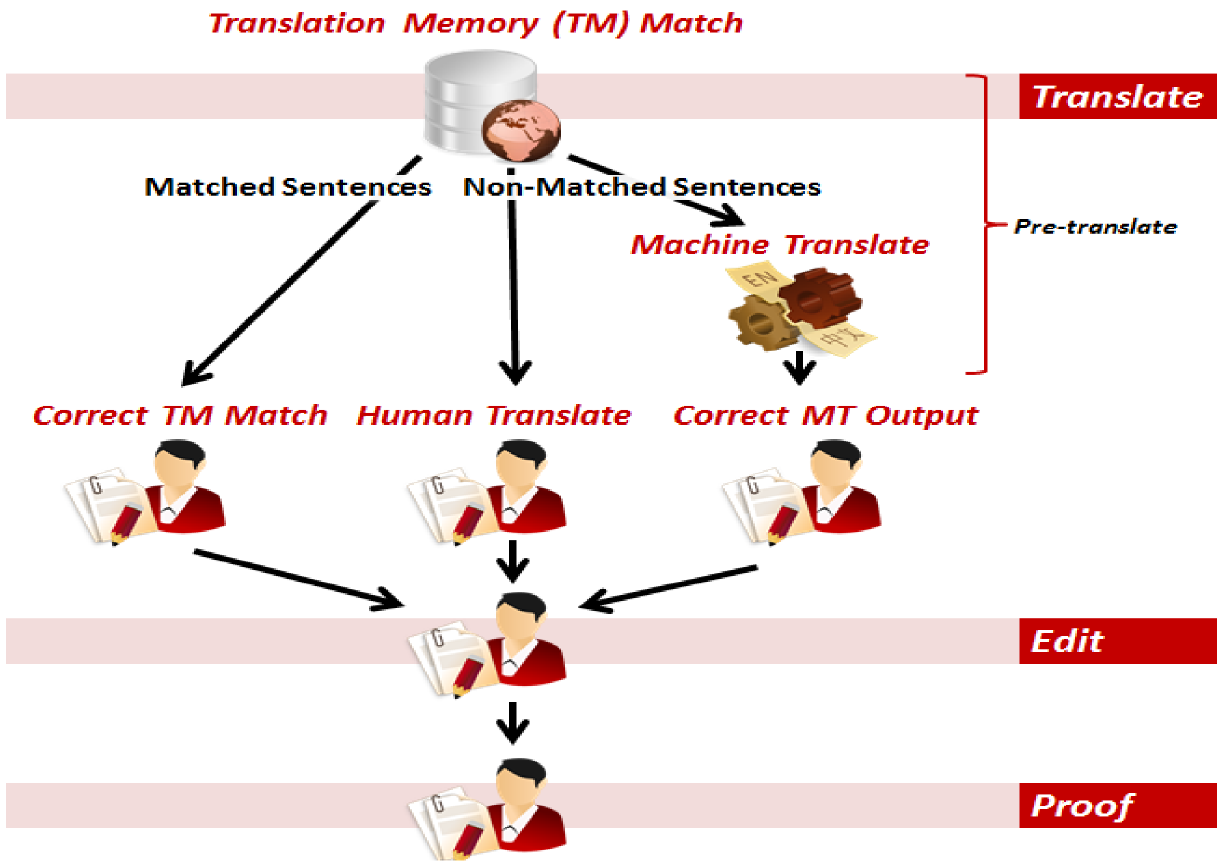
Hasso Plattner Institut

# Overview

- Example-Based Machine Translation (EBMT)
- EBMT‘s Workflow
- EMBT‘s Working
- EBMT vs. Case-Based Reasoning (CBR)
- Text Similarity
- Recombination
- Conclusions

# EBMT vs. Translation Memory (TM)

- TM includes input from users (human translators)




(http://www.languagestudio.com/LanguageStudioDesktop.aspx)

# Hybrid systems: EBMT + SMT

- Kyoto-EBMT and CMU-EBMT:
  - statistical alignment during analysis phase
  - Alignment also for parse trees and semantic graphs

# Summary

- Data-driven paradigm of MT
- Translation by analogy
- Makes use of rules to find matches and to recombine aligned parts and build translation
- Similar to CBR: makes use of text similarity, language resources and abundant parallel corpora
- Similarity based on words and/or structure
- Recombination: adapt matched translation parts to build a new translation (boundary friction)

# Suggested reading

- Machine Translation, Pushpak Bhattacharyya, (Chapter 6)

- Kyoto-EBMT (http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?KyotoEBMT)
- CMU-EBMT (http://www.cs.cmu.edu/~ralf/ebmt/ebmt.html)