



Universität Hamburg



Example Based Machine Translation for Crosslingual Retrieval

Cristina Vertan

University of Hamburg, Natural Language Systems Division

vertan@informatik.uni-hamburg.de

Why this tutorial instead of visiting Copenhagen?

- In order to visit Copenhagen one may know what is interesting to visit so:
 - either you buy a book (if you live in a Hamburg it will take some time to go to a library...)
 - Or you go to the tourism office (but for this you may know where it is)
 - Or you search in Google. This seems to be the faster and easier solution. And I can read maybe some informations in other languages.

But...

Is it of any use for the web if I speak more languages?

- „Informationen über Kopenhagen“
 - 31 pages only in German
- Informationen über Kopenhagen
 - 2020 pages only in German, including the page of „AutoEurope“..
- Information about Copenhagen
 - 13700 pages in German and English !!

But the settings in my browser were
„Retrieval in German, English, Romanian,
Portuguese, Spanish, French“

What is going wrong?

- Apparently the search engine works multilingual but not always
- If I restrict too much the query I will not get what I want but
- If not I will get too many documents.
- Why no documents in the other languages I specified were found

Is it really worth to invest time in finding some solutions ?

- For finding information about Copenhagen maybe not
- But if you are an Exchange student coming to Hamburg you will like to find out which courses you want to follow, and which prerequisites are stipulated
- The web page of the Department CS in Hamburg is 90% only in GERMAN, and contains terms like
- *„Anmeldebescheinigung“*, *„Zulasungskriterien“*
„Arbeitsbereichleiter“ which you will not find immediately in a common paper dictionary

On-line Translation and Retrieval

- Is it not easier to use existent on-line translation systems and translate my query ?
- **Assumption**: the translation will not be perfect but for retrieval should be enough !

Some examples of using On-Line Translation (Babelfish)-1-

- *Query:* When can I enroll for the **summer term**?
- *Translation:* Wenn ich für **Sommerbezeichnung** einschreiben kann

Some examples of using On-Line Translation(Babelfish)-2-

- *Query:* How long lasts Bachelor in Computer Science ?
- *Translation:* Wie langer Letzte Junggeselle in der Informatik

Some examples of using On-Line Translation(Babelfish)-3-

- *Query:* Which **lectures** are obligatory in the first semester
- *Translation:* Welche **Vorträge** sind im ersten Semester obligatorisch?
- An incorrect translation can damage a lot the retrieval results. Here the search engine will find the die Kolloquium -Vorträge which are invited talks of the department!

Why is on-line Translation problematic for retrieval

- More than 70% of the incorrect translations of on-line systems are due to false lexical choice
- This will influence dramatically the retrieval results.
- On-line systems rely on general lexicons. For specialised domains such systems are unreliable

What is needed for good retrieval results?

- A good lexical translation (no need in investing a lot in obtaining a 100% correct syntactic translation)
- A translation of the query beyond word-by word (multi-terms are considered)
- Easy to adapt for different language pairs, also when no linguistic processing tools are available (statistical translation will work only if you have big parallel aligned corpora)

What is this tutorial about

- Basic principles and methods of Information Retrieval
 - How to make Information retrieval to operate multilingual
-
- Basic principles and methods of EBMT
 - EBMT and Crosslingual Retrieval in use: the system LT4eL, and other pilot studies

- How it works
- Data structures used
- Pros and cons of different approaches

Basic principles and methods in Information Retrieval

IR

CLIR

EBMT

EBMT+CLIR

Information Retrieval problem

- Information Retrieval (IR) deals with finding documents that:
 - Are usually unstructured (text and/or images)
 - satisfy an information need
 - Belong to large collections (local computer servers or Internet)
- IR is becoming the dominant form of information access
- Usually together with IR a certain clustering is also performed
- Documents are :
 - Either fully unstructured
 - Or semi-structured (contain some XML mark-up or at least meta-data)

IR

CLIR

EBMT

EBMT+CLIR

Why do we need IR methods? -1-

Document collection : „Shakespeare’s Collected Works

IR Task: Find out which plays of Shakespeare contain the words **Brutus AND Caesar AND NOT Calpurnia**

Easiest solution:

-grep through text , i.e.

-Write a regular expression corresponding to the query to be matched against each text in the collection

`(Brutus) (Caesar) (! Calpurnia)`

Why do we need IR methods? -2-

- For modest collections of text grep is enough
- But:
 - Large document collections (bilions, trillions of words) cannot be processed quickly with grep
 - Flexible matching operations like
 - „Brutus NEAR Caesar“ where
NEAR = maximum difference of 5 words,
cannot be done with grep
 - grep does not allow any ranked retrieval

Document indexing

- Avoid linear scanning of documents
- The **index** helps to find which term occurs in which document
- **Term** = „word“. Depending on the preprocessing steps one may have:
 - All words
 - Lemma's
 - Keywords or lemma's of keywords
- One builds a **term-document incidence matrix**.
- The simplest form of such matrix is a binary one

IR

CLIR

EBMT

EBMT+CLIR

Term - Document incidence matrix

	Anthony And Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Anthony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0
...						

Term vector

Document vector

IR

CLIR

EBMT

EBMT+CLIR

Boolean Retrieval model

- Asks any query which is in form of a boolean expression of terms
- Allowed operators: **AND, OR, NOT**
- Such queries view documents as set of words.

IR

CLIR

EBMT

EBMT+CLIR

Brutus

AND

Caesar

AND

NOT

Calpurnia

Boolean retrieval model - Example

Anthony
And
Cleopatra

Julius
Caesar

The
Tempest

Hamlet

Othello

Macbeth

Anthony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0
...						
NOT Calpurnia	1	0	1	1	1	1

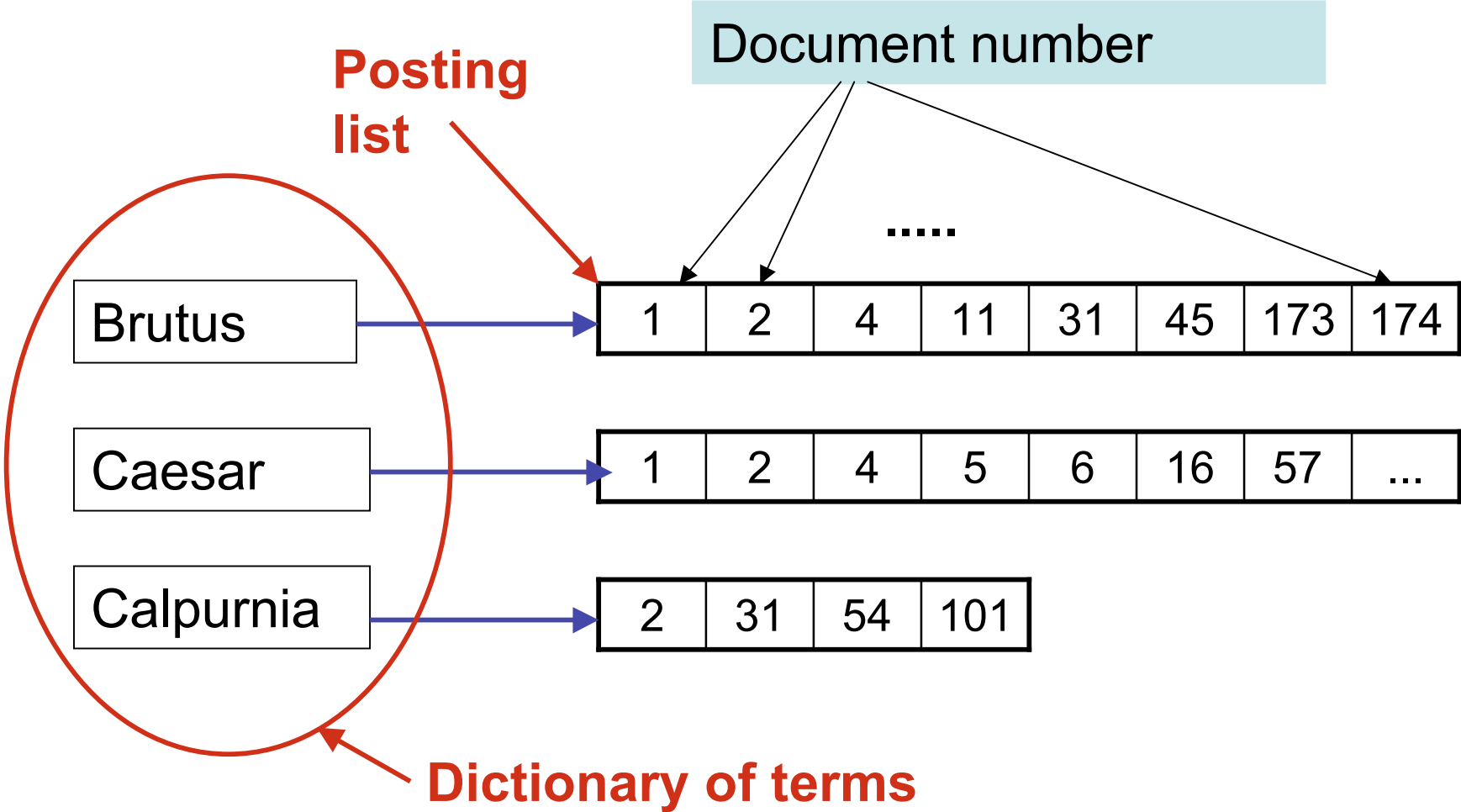
1:1 Term Matrix line index -Problems-

- A realistic collection of documents has $N = 1$ million documents
- Each document has at least 1K terms
- Each term occupies in average 6 bytes
- The corpus has approx 6GB
- If there are approx $m = 500\,000$ distinct terms we build a matrix of size $500K \times 1M = 1/2$ trillion 0 and 1
- BUT: the matrix is extrem sparse (most of the entries are 0)

Record only the „1“
positions

Inverted index

Inverted index - Example



Building an inverted index

1. Collect documents to be indexed. The result is a set of documents
2. Tokenize the text. Each document is a set of tokens
3. Linguistic preprocessing: Each document is a list of normalized tokens (stems or lemmas)
4. Create lexicon: set of distinct normalized tokens in all documents
5. Indexing: each term in the lexicon has attached a posting list

IR

CLIR

EBMT

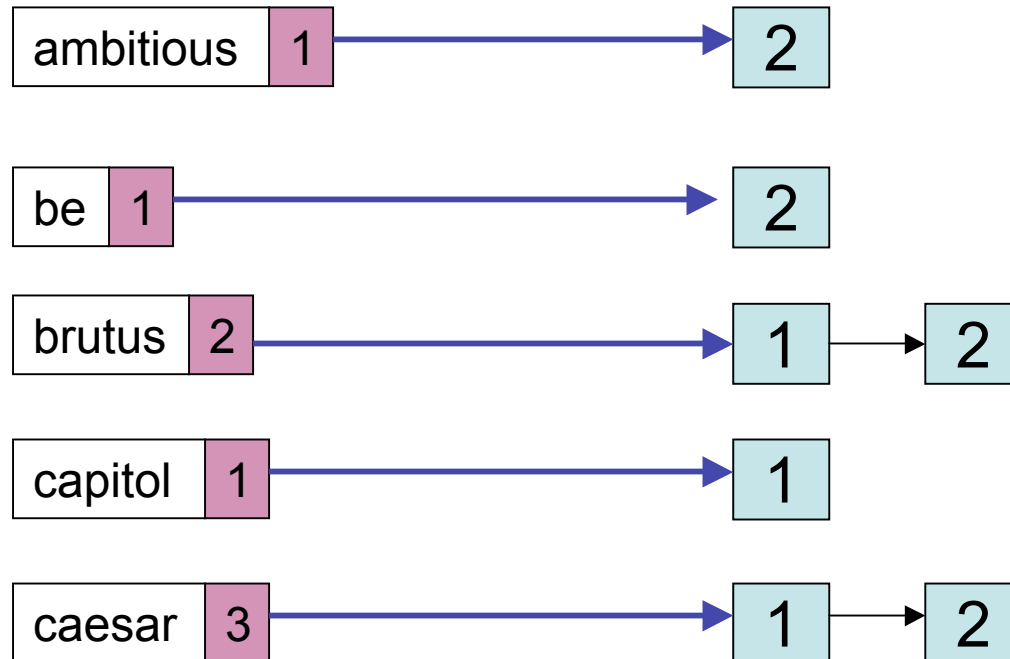
EBMT+CLIR

Extended inverted index

terms

frequencies

posting lists



Limitations of boolean models

A strict boolean expression over terms with an unordered results set is often too limited for the search needs like:

- It is often useful to search for compounds or phrased that denote a concept like „operating system“
- A boolean model only records presence or absence of terms, does not consider frequency of terms
- Boolean queries do not offer any possibility of ranking
- There is no possibility to include in the search synonym terms

IR

CLIR

EBMT

EBMT+CLIR

Normalising

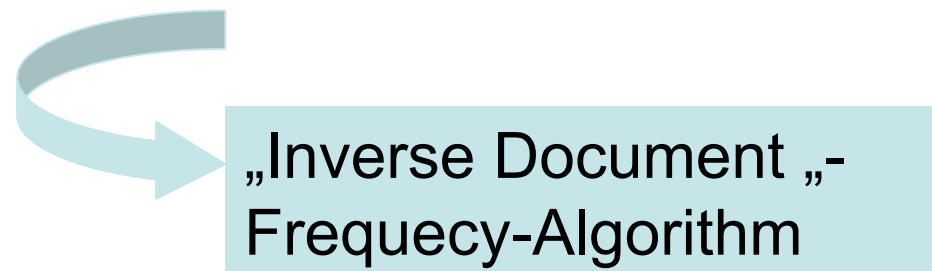
- One can define synonym lists for a specific domain
- E.g if we define HTML and XHTML as synonyms a document which contains XHTML will be indexed in the posting-lists for HTML

N-Gramms Index

- This technique use the hypothesis that many of the sequences of words in the query appear exactly in the same order in the text.
- This is especially relevant if we work at meta-level (PoS) <Noun1>...<Noun3>, <ART><NOUN>, <ADJ>. <NOUN>
- There are 2 methods:
 - Extracting of 2-gramms and 3-gramms and indexing or
 - Perform first PoS tagging and then index certain PoS sequences
- In the latter case the query will be translated into PoS and then boolean retrieval will be performed.

Limitations of Posting-Lists

- One may need linguistic tools which are not available for all languages
- The translation of query in boolean expression is not always possible
- All terms of the query are regarded as equal.



Inverse Document-Frequency -Principle

- The frequency of the term t in Document d , marked with tf_{td} is weighted through a log function:
- I.e.
 - $wf_{td} = 1 + \log tf_{td}$, if $tf_{td} > 0$ und
 - $wf_{td} = 0$ if $tf_{td} = 0$
- Up to now, also with this weighting, all terms in the query are equal important
- Therefore the Inverse-frequency is computed $(idf_t) = \log (N / df_t)$
 - where df_t = number of documents containing the term t
- The importance of the term for the document is a mixture of the two weights

$$(1) \text{Tf-idf}_{td} = \text{tft}_d \times \text{idf}_t$$

- The so called confidence value of a query is the sum of coefficients (1) which are calculated for every term in the query.

From Term Model to Vector model

- Even if the frequency of the terms is considered, term models cannot look into dependency between term inside a query.
- This dependency is the result of collocation of such terms in more documents.
- This phenomenon can be captured only by using vector models

Latent Semantic Analysis (LSA)

- LSA is an automatic statistical method which extracts on basis of a big corpus probabilities for lexical-semantic relationships .
- It makes use of high-dimension matrices
- LSA work without
 - Lexicon
 - Knowledge base
 - Semantic Network
 - Syntactic Parser
 - Morphology
- Home-Page: <http://lsa.colorado.edu/>

IR

CLIR

EBMT

EBMT+CLIR

Latent Semantic Analysis Example -1-

Usually keywords
are used

Text (Titles of technical reports):

c1: *Human* machine *interface* for *computer* applications

c2: A *survey* of *user* opinion of *computer system response time*

c3: The *EPS user interface* management *system*

c4: *System* and *human system engineering testing of EPS*

c5: Relation of *user* perceived *response time* to error measurement

m1: The generation of random, binary, ordered *trees*

m2: The intersection *graph* of paths in *trees*

m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering

m4: *Graph minors*: A *survey*

IR

CLIR

EBMT

EBMT+CLIR

Latent Semantic Analysis -Example -

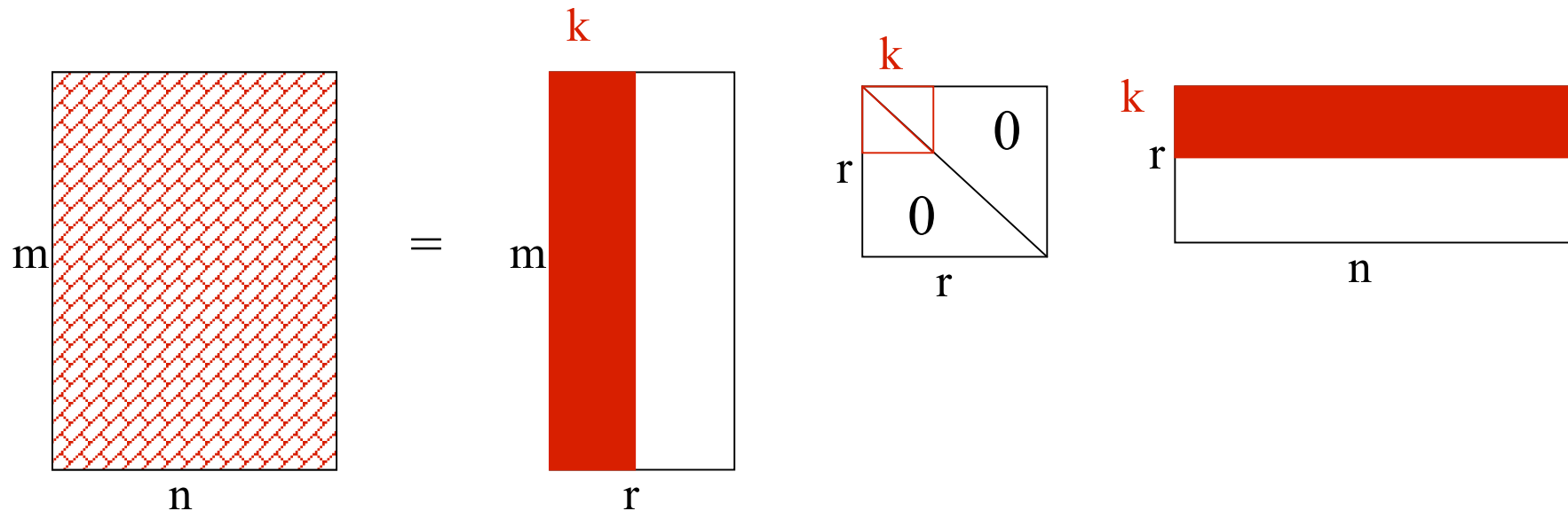
	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	1	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

Frequency

In real applications the frequency is normalised

Latent Semantic Analysis -Matrix transformation -

SVD - Single Value Decomposition



$$A = B \times I \times C$$

The bigger the matrix is the more complex is the computation

$$A_k = B_k \times I_k \times C_k$$

IR

CLIR

EBMT

EBMT+CLIR

Latent Semantic Analysis - Matrix Transformation -Example -

Matrix reconstruction for $k=2$

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

IR

CLIR

EBMT

EBMT+CLIR

“tree” appears not in m4, but in titles with “graph” and “minors”

$$\text{korr}(\text{human}, \text{user}) = 0.94$$

$$\text{korr}(\text{human}, \text{minors}) = -0.83$$

$$\text{korr}(\text{human}, \text{user}) = 0.38$$

$$\text{korr}(\text{human}, \text{minors}) = 0.29$$

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	1	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

Original

IR- Conclusions

- For not complex queries and relative low number of documents, the boolean model is suitable
- If the a ranking is necessary for the retrieved documents one may use the inverted index.
- Vector models help to capture lexical-semantic relationships between terms of the query . With such models one may find documents also when they do not contain the term.

Making IR to work with multilingual Queries and Documents

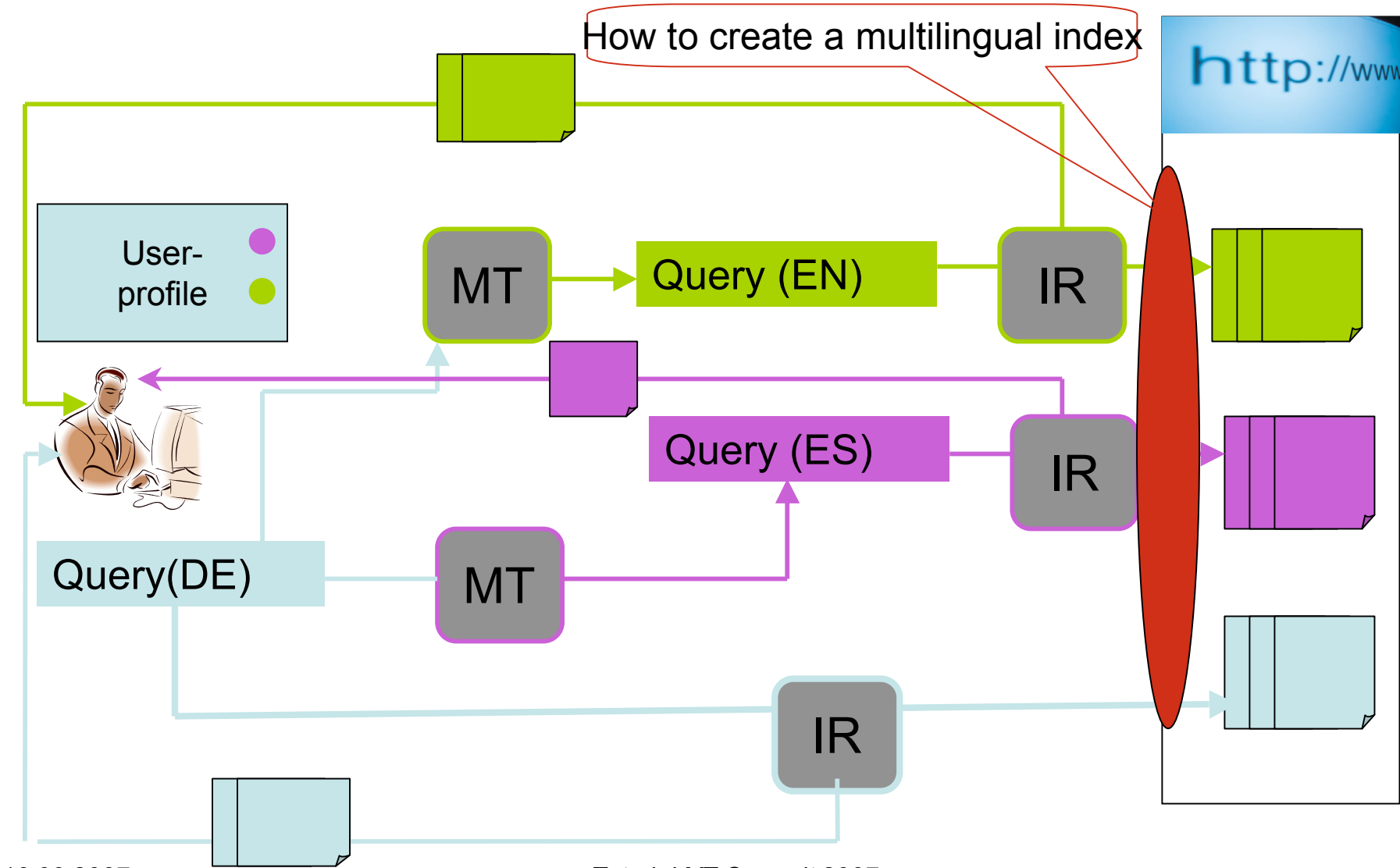
IR

CLIR

EBMT

EBMT+CLIR

Multilingual extension of IR



IR

CLIR

EBMT

EBMT+CLIR

Multilingual indexing

Options:

- Translate Texts:
 - Case 1 : into all relevant languages (afterwards we are in the monolingual IR case) - not feasible for large collections of documents
 - Case 2: into Pivot language - not very different from Case 1. The costs for further translations may be lower
- Translate index terms
- Translate query
 - Case 1: Query into n monolingual document pools
 - Case 2: Query into a multilingual document pool

Translation of index



- One idea is to analyse the query in the query language and to translate the terms.
- Problem: there is low chance that the translation of terms will match the index of the documents.
- Usually index contain only words and not multi-word terms
- How to solve ambiguity ?

•Fehler: mistake, fault, error, bug
•Nuclear: Kern, zentral, nuklear
•Power: macht, Kraft, Strom
•Plant; Pflanze, unternehmen



Nuclear power
plant
Zentrale
Kraftpflanze???

Query analysis

- Before it can be translated the query must be analysed:
 - Find key concepts and translatable units
- „drug and heroin dealers in the former German Democratic Republic“
- Resolve gapping (drug dealer /heroin dealer)
 - Remove function words after analysis (in, the, and) 
 - Identify multiword units 
 - Drug dealer / heroin dealer / German Democratic Republic
 - Remove noise words (former)

IR

CLIR

EBMT

EBMT+CLIR

Query Translation

Main problem:

- Short queries have few context for disambiguation
- MT-dictionaries must fit the domain
- 1:n translations must be disambiguated

Solution: link an ontology to the MT system

CLIR -conclusions

- CLIR inherits all existent and unsolved problems by IR
- Additionally the:
 - Query translation and disambiguation of terms
 - Multilingual index
- Increase the difficulties
- One need no perfect syntactic translation but adequate semantic translation
- An ontology is absolutely necessary to ensure the mapping between words and concepts of the query and to act as a kind of Interlingua

- EBMT-Principles
- Matching
- Alignment
- Recombination
- Adding linguistic information to EBMT

Basic principles and methods in Example Based Machine Translation

General Principles -Corpus based MT

- The linguistic phenomena in both languages as well as the transfer rules are no longer linguistically described but derived automatically from a parallel corpus.
- Next step is a training phase, in which are calculated the connections between elements in the source language as well as in the target language (sometimes the results are called „knowledge sources“).
- First an aligned corpus is built
- The translation is the result of 2 processes:
 - A search process (of elements in the source language)
 - A best-evaluated relation with a target expression
- There are 2 types of corpus-based MT systems
 - Example based MT - The translation of a source text is based of translation examples in the database
 - Statistical MT - the alignment information from the corpus is used for the training of a statistical translation model

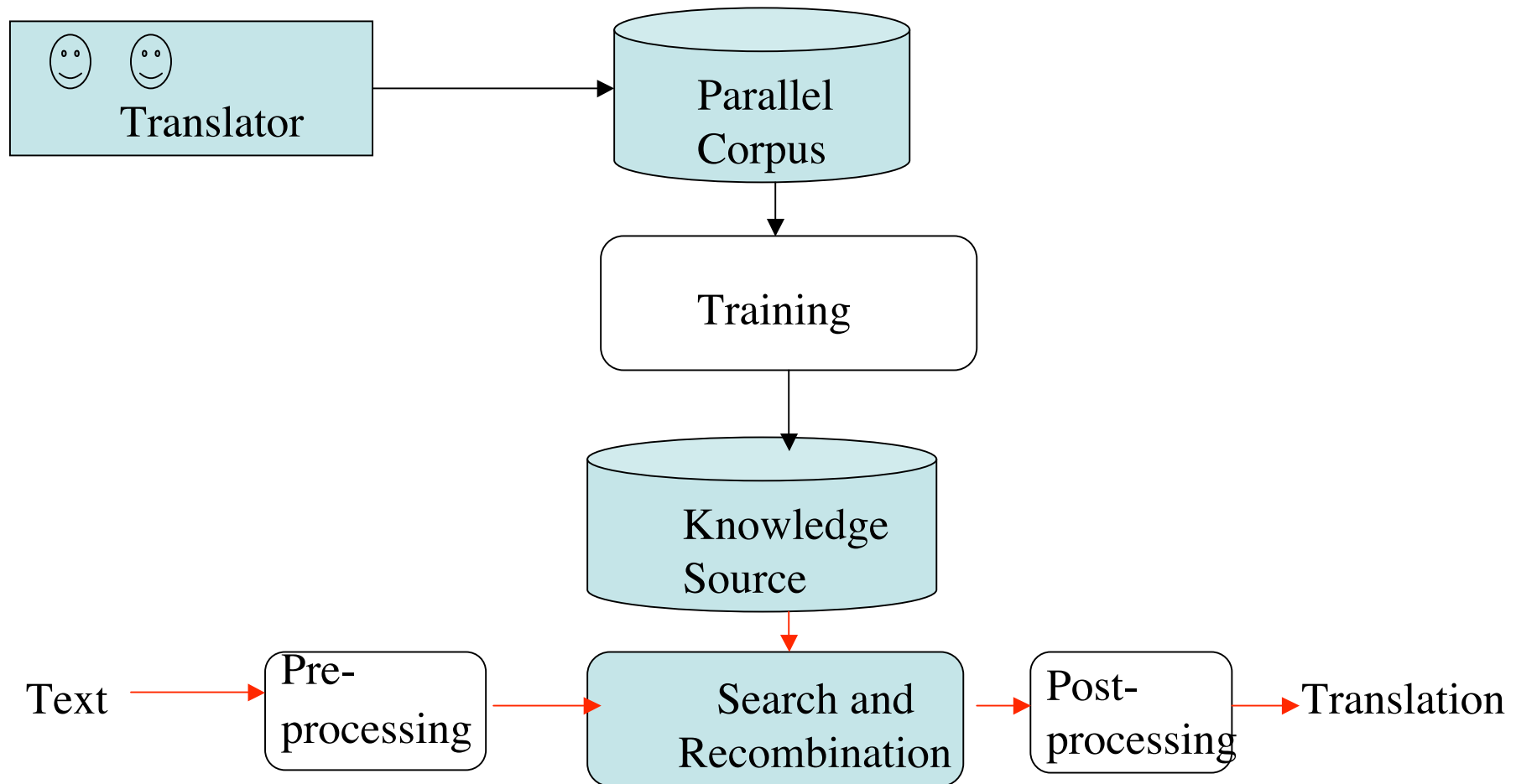
IR

CLIR

EBMT

EBMT+CLIR

Generic Architecture of a corpus-based MT-system



IR

CLIR

EBMT

EBMT+CLIR

EBMT Sources: Theory of Translation

A new translation may use as much material as possible from old translations (produced within the same domain, time, etc.).



Advantages of this approach:

- saves time
- ensures the terminological and stylistic consistency



Many human translations are revisions, improvements, changes of previous translations.

EBMT sources: cognition science

- Human translations are mostly not the result of deep linguistic analysis but more of an appropriate,
 - Division of the sentence in chunks followed by
 - Translation of the components as well as
 - Combination of these components.
- The translation of the components is done through analogy with previous existent translations.

Analogy principle (Nagao, 1984)

EBMT source: MAHT

- Translators use often big databases with translation examples (Translator's workbenches / Translation memories).
- E.g. TRADOS - a TM-system for 12 European languages
- The system searches in the database all entries in the source language similar with the input and shows their translations
- The human translator identifies the pieces which he needs, and performs their recombination.

IR

CLIR

EBMT

EBMT+CLIR

General Principles of EBMT

- A parallel Corpus is used
- Part of the input text are compared with source chunks in the corpus
- The translation of the retrieved parts are put together and form the translation.

Or

- The most similar sentences to the input in the SL corpus are retrieved (a distance is defined)
- The corresponding translations are combined to form an output

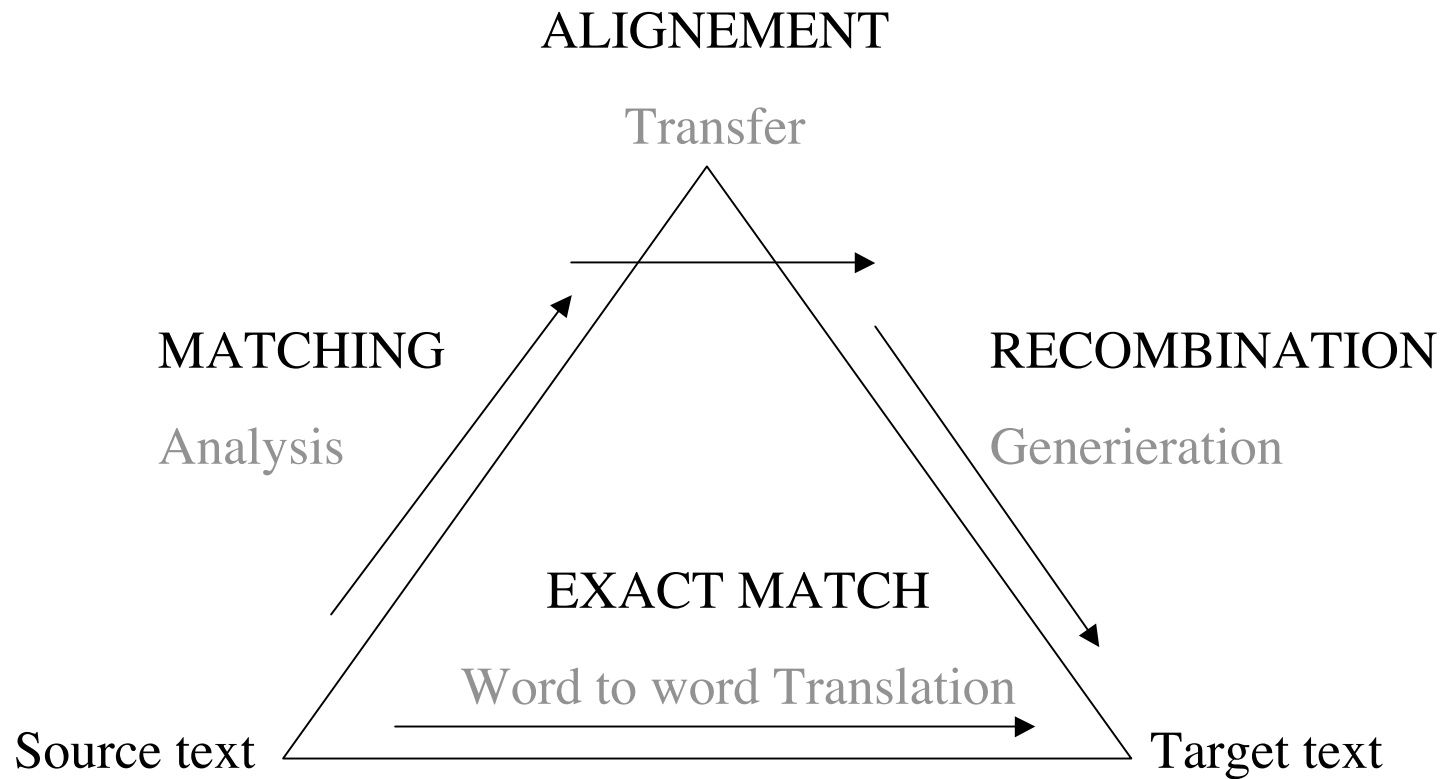
IR

CLIR

EBMT

EBMT+CLIR

Translation pyramid for EBMT



Functionality of an EBMT-System

- Relevant examples from a parallel corpus are extracted and saved in a database
- The input is compared with entries in the database(matching-phase).
 - Either the system looks for the identity of (parts of the) input with the database entries or
 - a distance between the input and the database entries is computed, and the database entry with the minimal distance to the input is chosen.
- Further on, in the alignment phase, the corresponding parts in the target language are retrieved (this is trivial when the whole identical input is found in the DB)
- The corresponding chunks in the target language are recombined and build the output

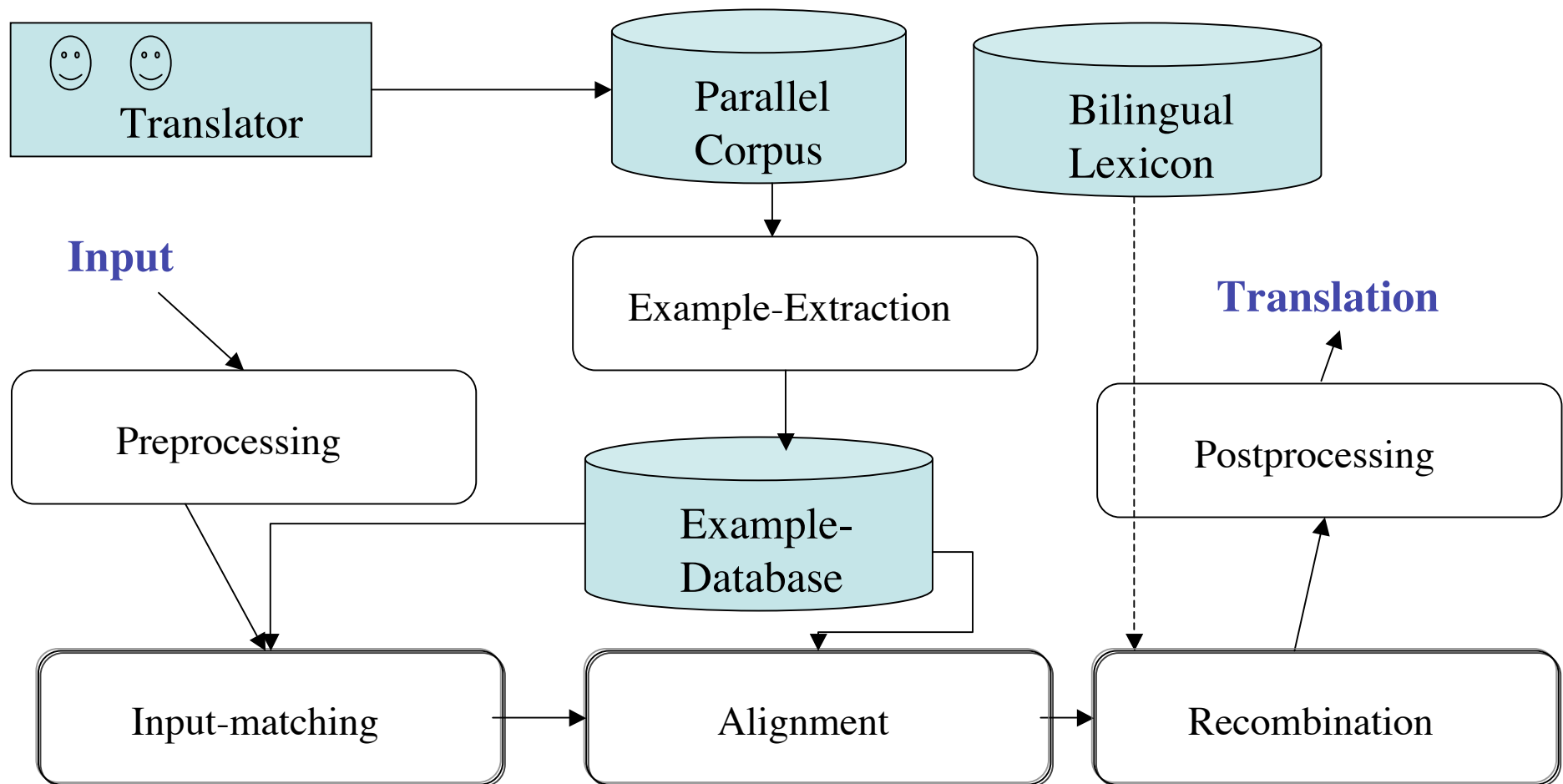
IR

CLIR

EBMT

EBMT+CLIR

Architecture of an EBMT-System



Important decisions when building a database of translation examples

- Size: How many examples have to be stored?
- Length of entries: how long should be the translation examples ?
- Annotations: Do we need additional information?
- Data: What do we store (Strings, grammatical structures)?
- How do we store in order to retrieve easily

Relevant Examples?

- For a good lexical coverage:
 - a lot of domain relevant words
 - As much as possible with co-occurrences (reflexiv, particle verbs, etc.)
- For a good syntactic coverage:
 - Structures containing main and relative clauses
 - Active and passive voice sentences
 - questions
 - Sentences with embedded structures, e.g attribute sentences, conjunction sentences

Length and Size of Examples

- The *size* of the example database varies between some hundreds and 800.000 sentences.
- The bigger the database, the better the system works
- There is no ideal *length* for the examples:
 - The longer the examples, the lower the chance for a match
 - The shorter the example the bigger the chance to have some ambiguities
- Usually the standard *unit* for the examples is a sentence

EBMT - Example

- Input: *Ungeeigneter Kraftstoff kann zu Motorschäden führen*
- the translation database contains:
 - *Starke Motorbelastung kann zu Motorschäden führen - High engine loading can cause engine damage*
 - *Ungeeigneter Kraftstoff darf nicht benutzt werden.- Unsuitable fuel must not be used*
- Following chunks are identified
 - *kann zu Motorschäden führen - can cause engine damage.*
 - *Ungeeigneter Kraftstoff - Unsuitable fuel*
- The translation is then:
 - *Unsuitable fuel can cause engine damage*

Corpus-Tagging for EBMT -1-

- It is possible to mark in the corpus words or morphemes, which delimit a clear co-text: like quantifiers, conjunctions, pronouns, question markers, etc.
- E.g.
<QUANT> all uses
<QUANT> tous usages

Corpus tagging for EBMT Example Gaijin System

Phrasal segmentation using Marker Hypothesis

- Psycholinguistic constraint on grammatical structure
- States that natural languages are marked for grammar by a closed set of lexemes and morphemes
- Gaijin exploits such markers as signals for beginning and end of a phrasal segment:
 - Prepositions: in, out, on, with,...
 - Determiners: the, those, a, an,....
 - Quantifiers: all, some, many,....
- Markers not considered to start a new segment if previous/next segment would consist entirely of marker words

How to organise the database

- There is no „best solution“
- The easiest way: all the words which exist in the database are stored and for each word a list with the id of the sentences where they appear is provided
- In the matching phase a threshold is fixed and only sentences containing at least the threshold number of words are compared with the input.

Input for Matching

- The problem is to find out, which parts of the input can be retrieved in the database
- This is done through a combination of string-based, statistical-based methods (e.g. big probability for multi-word lexemes), and help of additional linguistic knowledge.
- String-based matching approaches:
 - Edit distance
 - Angle of similarity
 - Semantic similarity

String-based Matching

- The similarity is measured between the input string and each string in the database. Following distances are used:
 - “longest common sequence”
 - “Edit distance”: how many operations (Insert, Delete, Replacement) are necessary to transform the input string into an entry in the Database
- These methods can be implemented easier through greedy algorithm, or dynamic programming

Database-Search (Alignment) -1-

- In the best case one example in the database is identical with the input
- Usually only parts of the input are found in the database
- The simplest is the organisation of the database (no indexing, no markers, no syntactic structures), the more difficult is the retrieval both of
 - the identical parts in the SL
 - Their translation equivalents

Database-Search (Alignment) -2-

- There are elaborated statistical procedures to align the segments. They are based on statistical models of the SL and TL.
- Easier: the syntactical structures in both languages (at least for some problematic chunks) are stored and links between the SL and TL structures are provided.
- Another option is to mark at least words which delimit unambiguous parts of the sentence (see marker hypothesis).

Recombination

- Without grammatical structures , or at least some markers , is is very difficult,
- When syntactical structures are provided , the procedure relies on tree unification
- Quite often one provide a set of basic transformation rules (like N Adj --> Adj N)
- Additionally one may use statistical measures on a large corpus (Internet) or even a target language model to determine the most probable combination of chunks
- For strong inflected languages is usually a morphological post-processing necessary.

EBMT with morphological/lexical knowledge

- Use only the stems when measuring the distance between input and entries in the database
- Mark in the database words with unambiguous function (e.g conjunctions)
- Whenever possible align fixed expressions
- When measuring the Edit distance look at the PoS of the words

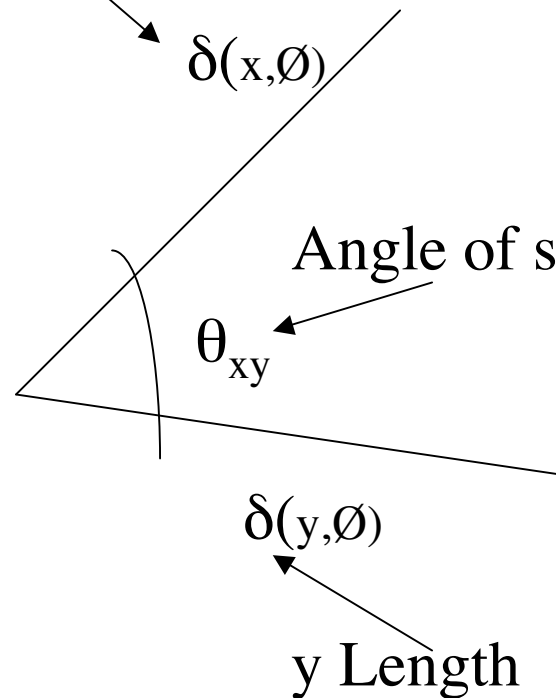
Word-based Matching: "Angle of similarity"

- 1 -

- A trigonometrical distance is computed.
- The distance between 2 sentences corresponds to a difference function δ .
- This difference function works similar as the string-based matching (the number of operations is calculated)
- The operations are weighted, e.g. the insertion of a comma has a smaller weight than the absence of an adjective.
- The weights are defined according to the system and the translation domain

Word-based Matching: "Angle of similarity" - 2 -

x Length



Distance between
sentence x and sentence y

$$\sin \frac{\theta_{xy}}{2} = \frac{\delta(x, y) - |\delta(x, \emptyset) - \delta(y, \emptyset)|}{2 \times \min\{\delta(x, \emptyset), \delta(y, \emptyset)\}}$$

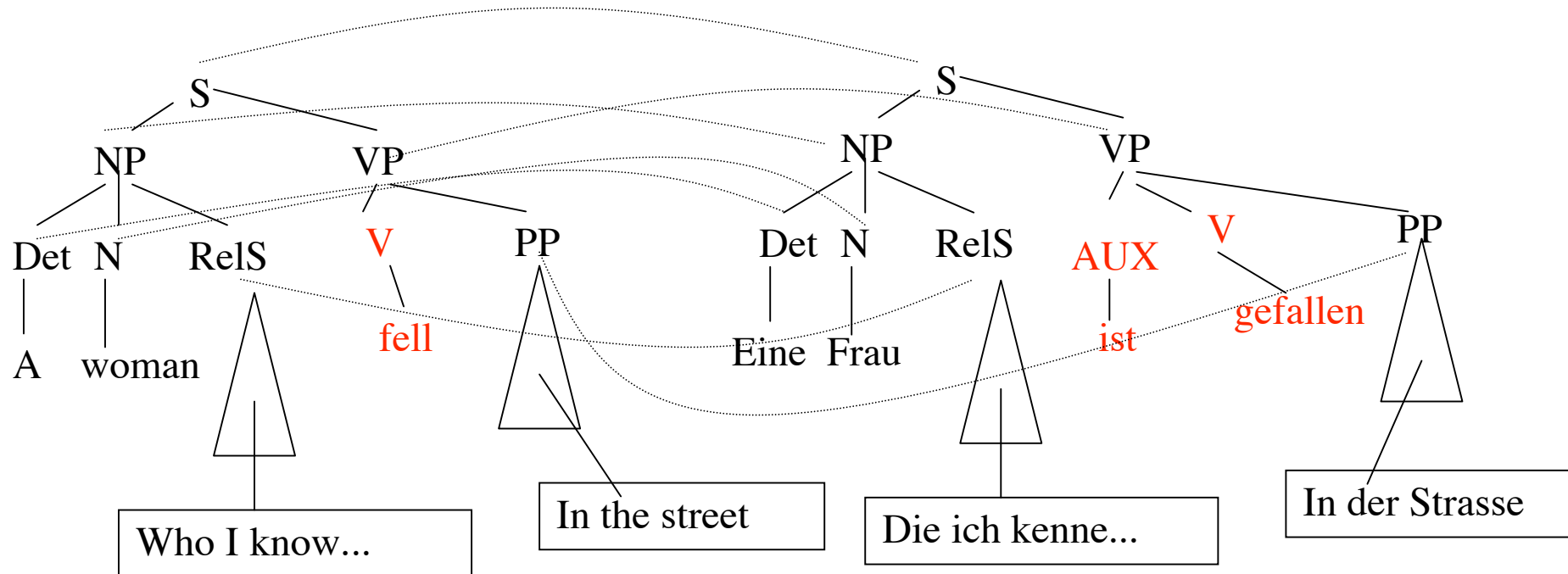
Word-based Matching - "Angle of similarity" Example

1. *Lesen Sie Seite 3 im Kapitel "Benzin"*
 2. *Lesen Sie Seite 3 im Kapitel "Bremsen" und Seite 5 in Kapitel "Länderspezifische Bemerkungen"*
 3. *Lesen Sie Seite 4 im Kapitel "Bremsen".*
- String-based matching gives a closer similarity between sentence 1 and sentence 3 because they differ only by 1 word.

However: Sentence 2 is actually a better choice as sentence 1 is contained entirely. This choice is made by the "angle distance".

EBMT with syntactic knowledge

- The Translation patterns are not words, but syntactical structures in both languages with corresponding links
- A “semantic network“ is used additionally; in this semantical network the distances between words express semantic similarity.



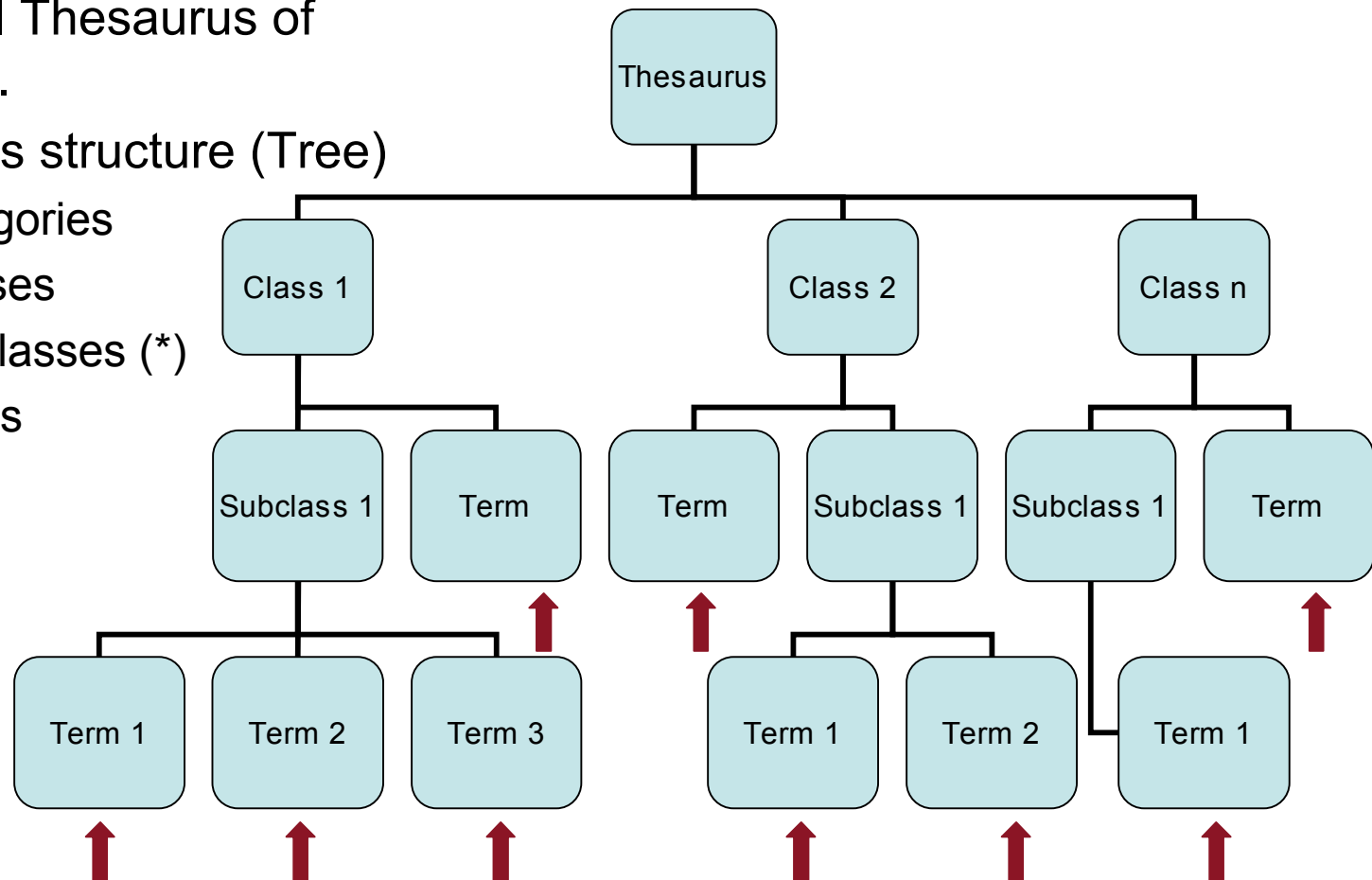
Word-based matching - 1 -

- For example for the following entries in the DB:
 - Der *Abstand* zwischen den Kontrollen soll 2 Jahre nicht überschreiten
↔ The *interval* between 2 general checks should not exceed 2 years.
 - Der *Abstand* zwischen den Nebelleuchten ist x cm.
↔ The normal distance between fog-lights is x cm.
- The input : *Wo finde ich den Abstand zwischen den Rädern?*
 - *Räder* in the semantic network is closer to *Nebelleuchten*, therefore *Abstand* is translated by *distance*,
although the edit distance between *Räder* and *Kontrolle* is smaller than the edit distance between *Räder* and *Nebelleuchte*.

Construction of the Semantic Network (I)

- Bilingual Thesaurus of NOUNS.
- Elements structure (Tree)
 - Categories
 - Classes
 - Subclasses (*)
 - Terms

NOUNS



Construction of the Semantic Network (II)

- Spanish Culture
 - Entertainment
 - Fashion
 - Sports
 - Religion
 - Dietary Habits
 - Mediterranean Diet
 - Typical Food
 - Tapas
 - Art
 - Monuments
 - Mosque
 - Museum
 - Monastery
 - ...
- Spanish Geography
 - Territories (“map”)
 - Autonomous Region
 - City
 - Province
 - Town
 - ...
 - Geographical Quirks (“geo”)
 - Mount
 - Mountain
 - Mountain Range
 - River
 - Ocean
 - ...
 - Cardinal Points

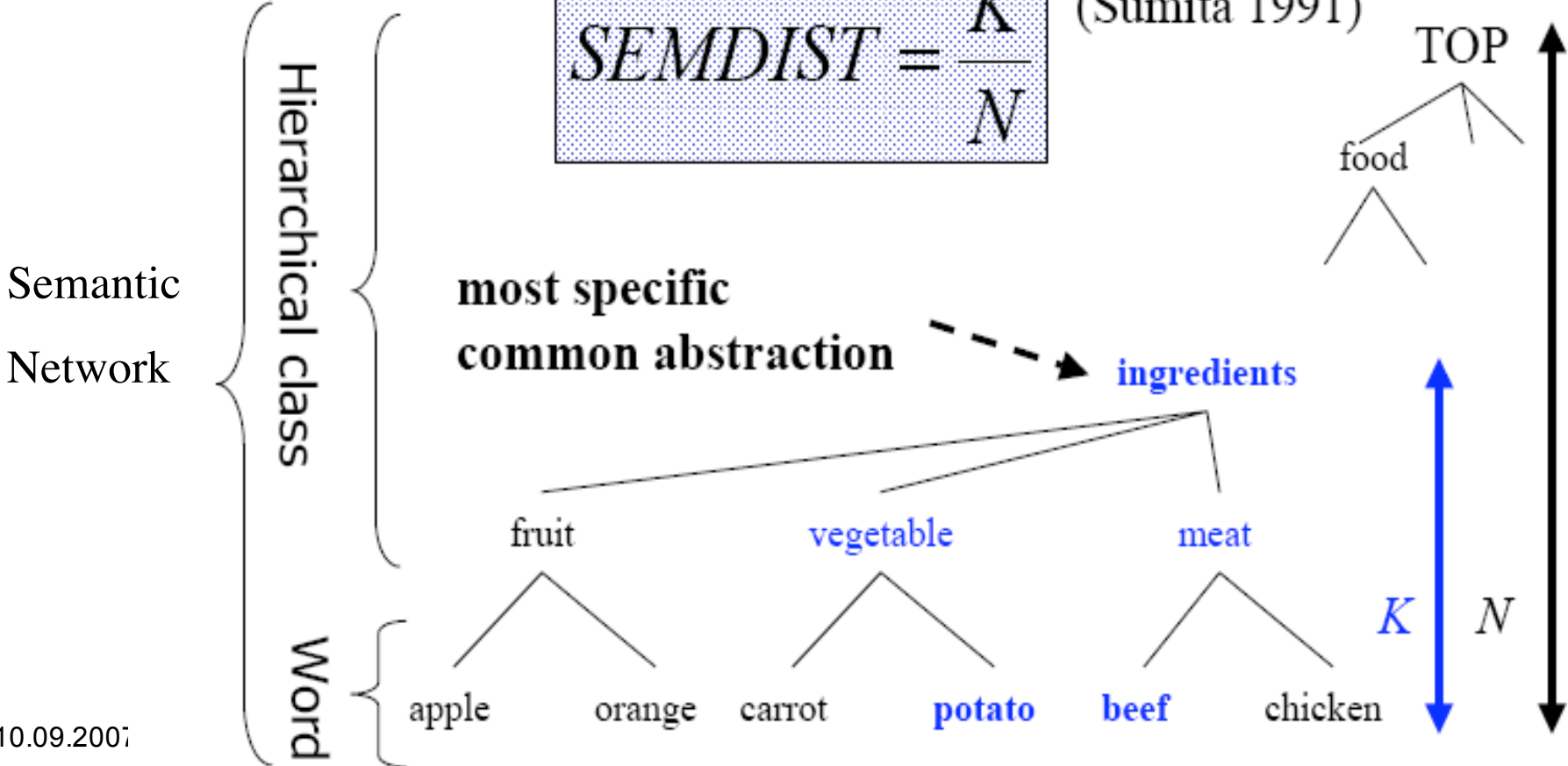
Measuring the Distance (I)

$$dist = \frac{I + D + 2 \sum semdist}{L_{input} + L_{example}}$$

- Distance

$$SEMDIST = \frac{K}{N}$$

(Sumita 1991)



Measuring the Distance (II)

- Semantic Distance
 - If two words are in the same subclass -> Semantic Distance = 0. Maximal Similarity.
 - Sea – Mountain -> SD = 0
 - If they are in different categories -> Semantic Distance = 1. Completely Dissimilar.
 - Sea – Museum -> SD = 1

Measuring the Distance. Sample

- Initial sentence manipulation (lexicon):
 - INPUT: “I have seen the Alhambra of Granada”


↓
“see the monum of map”


- CORPUS : “You will see the Mosque of Cordoba”

↓
“see the monum of map”

0 insertions 0 deletions 0 substitutions
dist = 0

Measuring the Distance. Sample

- Initial sentence manipulation (lexicon):
 - INPUT: “The autonomous region of Andalusia lies in the south of Spain”


“The region of map lie in the cardinal point of map”
 - CORPUS : “The gulf of Almeria lies in the east of Andalusia”


“The gulf of map lie in the cardinal point of map”

0 insertions 0 deletions 1 substitutions
 semdist (region, gulf) = 0.5
 dist = $(0+0+2*0.5) / (11+11)$

IR

CLIR

EBMT

EBMT+CLIR

EBMT-Conclusions

- The building of the example database is very important for the system.
- Semantic similarity should be used not as stand alone but in connection with a string measure.
- Recombination , although the most complicated i EBMT has lower relevance for CLIR

Putting all together EBMT and CLIR

IR

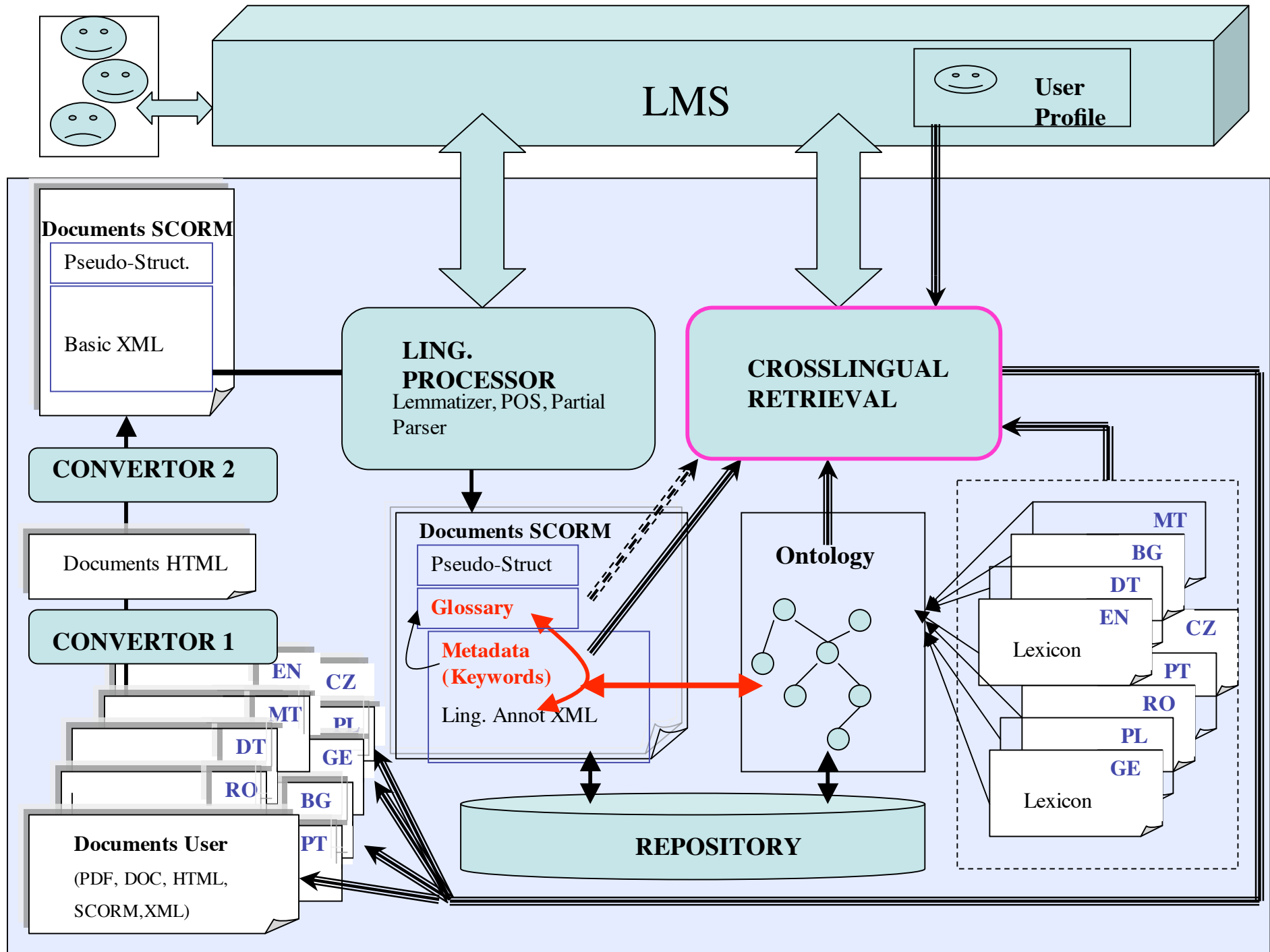
CLIR

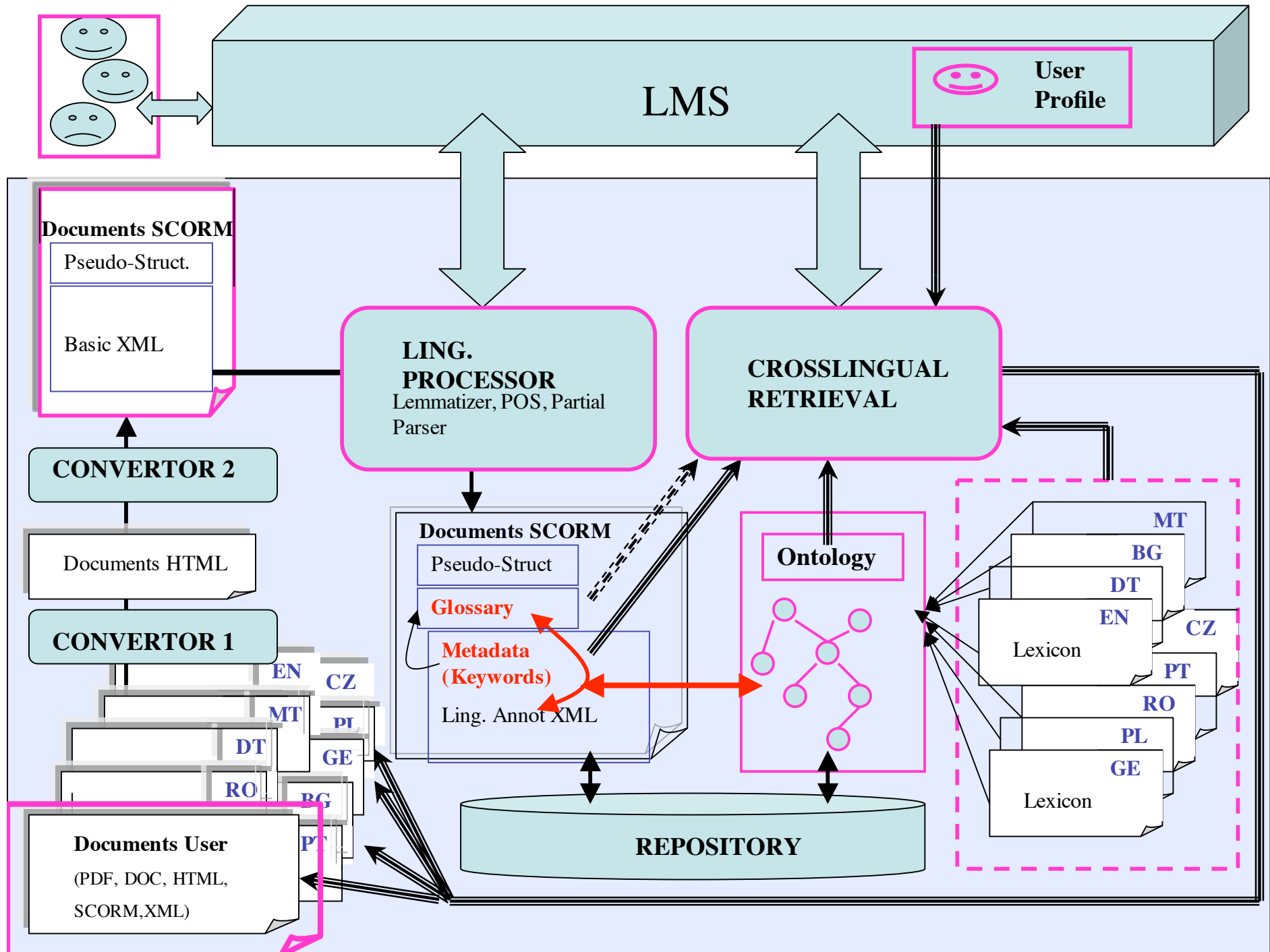
EBMT

EBMT+CLIR

Framework

- EU-Project LT4eL: Language Technology for eLearning (www.lt4el.eu)-12 Partners
- Learning objects : 8 (9) Languages
- eLearning Test-System : open source platform ILIAS (www.ilias.de/)
- Domain: Computer Science for non CS specialists





IR

CLIR

EBMT

EBMT+CLIR

Ontology creation

707 classes linked through **is_a** relation

Formalize in OWL concept definition

Provide a definition for the concepts and attach a label

Cluster keywords into concepts.

Select keywords relevant for CS and translate them into English (also through NPs)

1000 Keywords extracted with KWE in each language
(for German 36 Documents)

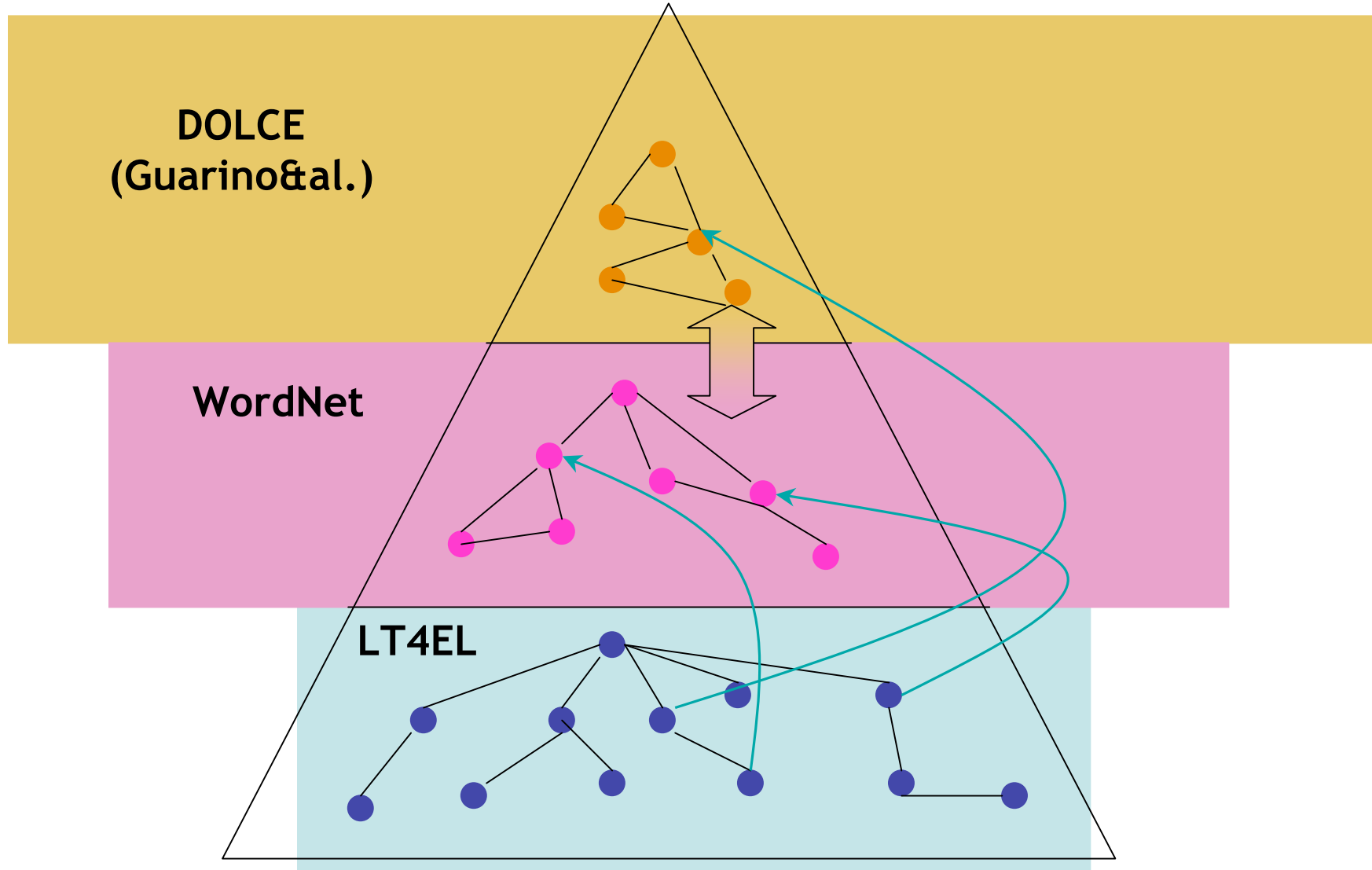
IR

CLIR

EBMT

EBMT+CLIR

Connection with other ontologies



IR

CLIR

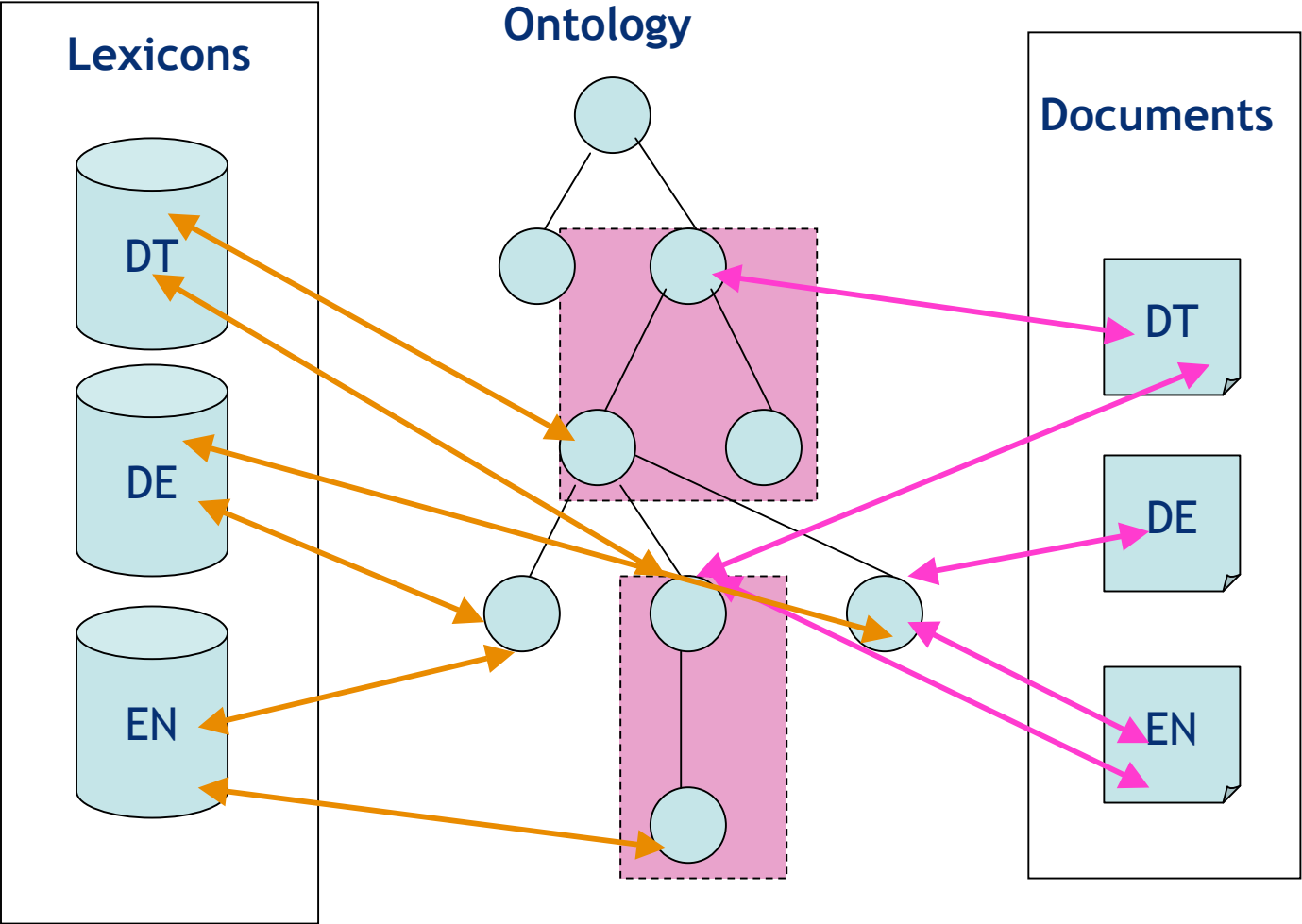
EBMT

EBMT+CLIR

Ontology-Example

```
<owl:Class rdf:about="http://www.lt4el.eu/CSnCS#IndexDataStruct">
  <rdfs:comment>A data structure that enables sublinear-time
  lookup.</rdfs:comment>
  <rdfs:comment>Hyper OWN: http://www.loa-
  cnr.it/ontologies/WordNet/OWN#DATA\_STRUCTURE</rdfs:comment>
  <rdfs:comment>Hyper WN20: ENG20-05396271-n</rdfs:comment>
  <rdfs:comment>ID: id918-3</rdfs:comment>
  <rdfs:subClassOf>
    <owl:Class rdf:about="http://www.lt4el.eu/CSnCS#DATA_STRUCTURE">
    </owl:Class></rdfs:subClassOf>
</owl:Class>
```

Ontology and multilingual data



IR

CLIR

EBMT

EBMT+CLIR

Mapping of multilingual lexicons -1-

```
<entry id="id543">
  <owl:Class rdf:about="http://www.lt4el.eu/CSnCS#Presentation">
    <rdfs:comment>Presentation is the process of presenting the content of a topic to an
audience.</rdfs:comment>
    <rdfs:comment>Equal OWN: http://www.loa-
cnr.it/ontologies/WordNet/OWN#PRESENTATION__PRESENTMENT__DEMONSTRATION</rdfs:com
ment>
    <rdfs:comment>Equal WN20: ENG20-00496521-n</rdfs:comment>
    <rdfs:comment>ID: id1307</rdfs:comment>
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://www.loa-
cnr.it/ontologies/WordNet/OWN#SHOW_1"></owl:Class></rdfs:subClassOf></owl:Class>
    <def>Presentation is the process of presenting the content of a topic to an audience.</def>
    <termg lang="de">
      <term shead="1">Darstellung</term>
      <term>Präsentation</term>
    </termg>
  </entry>
```


IR

CLIR

EBMT

EBMT+CLIR

Mapping of multilingual lexicons -2-

- German 939 entries mapped on 707 classes in Lt4eL ontology
- The difference corresponds to terms mapped on more than one concept.

Are there terms which cannot be related to an existent concept in the ontology?

Domain too narrow to study more complicated phenomena

IR

CLIR

EBMT

EBMT+CLIR

Multilingual, semantic document retrieval using an ontology

- Starting points
 - A multilingual document collection
 - An ontology including a domain ontology on the domain of the documents
 - Concept lexicalisations in different languages
 - Annotation of concepts in the documents
- Resources and first Application Scenario: LT4eL

IR

CLIR

EBMT

EBMT+CLIR

Goals of the approach

- 1. Improved access to documents
 - Find docs that would not be found by simple text search
- 2. Multilinguality
 - One implementation for multiple languages
- 3. Crosslinguality
 - Retrieve languages in languages other than language for query or ontology presentation

Outline of search procedure

1. **User submits a free text query**
2. **Query is tokenised analysed and translated (using EBMT and the ontology).**
3. **See document list**

A list of documents is displayed with some meta information, for example:

 - title;
 - length;
 - original language;
 - keywords and concepts that are common to both the query and the document;
 - other keywords and concepts that are related to the document but not to the query.
4. **See concepts for refining search**
 - Concepts related to the search query → starting point for browsing
 - No related concepts from search query → root of ontology starting point

IR

CLIR

EBMT

EBMT+CLIR

Example result

English documents

- “Introduction to Access Database”
- “Linux Doc Reference”
- “Introduction to Word”

Bulgarian documents

- Запознанство с Word for Windows
- “Външен вид на слайдовете и специални ефекти”

<input type="checkbox"/>	Application Program	Show related concepts
<input checked="" type="checkbox"/>	WordProcessing	Show related concepts
<input type="checkbox"/>	MicrosoftWord	Show related concepts

IR

CLIR

EBMT

EBMT+CLIR

Outline of search procedure

4. **View documents**
User looks into the documents from the list.
5. **Browse ontology**
6. **Select concepts**
7. **Select search option**
User selects an option about how to use the ontology fragments for search.

Outline of search procedure

8. **See new document list**
A new list of documents is displayed, based on only ontological search.
9. **See updated concept browsing units**
Concepts that are common to the found documents
Example:
 - Concept “Report”
 - Some documents about academic writing
 - Concept “Publication”
10. **Repeat steps from step 5 (Browse ontology)**
User selects another set of related concepts and submits it as the search key, etc.

IR

CLIR

EBMT

EBMT+CLIR

Outline of query translation

Input typed in the User Interface



Tokenize input; extract each word for the sentence



Extract the sentences in the DB which contain at least „X“ words from the input.



Measure **Angle of Similarity + semantic distance** between input and these sentences and extract the best match



Extract the longest common sequence between input and the best match and detach it from the input



Find in the DB the translation equivalents for the matched chunks



Recombine the translated chunks

Lemmatize input and DB-entries; measure distance also with lemmatized versions

Repeat the operation with rest of the input

Use „marker hypothesis“ in the DB

IR

CLIR

EBMT

EBMT+CLIR

Required parameters

- Possible languages of search query (in which lexicons should we look?)
- Retrieval languages
- Language for User Interface
- Show concepts that are shared in at least N of the found documents.
- If less than N documents are found for a certain concept: try with superconcept and subconcepts

IR

CLIR

EBMT

EBMT+CLIR

Free text query

- Why start with a free text query?
 - User wants results fast (as in Google)
 - Compete with fulltext search and keyword search
 - Find starting point for ontology browsing

Search functionality comprises:

1. Find terms in lexicons that reflect search query.
2. Find corresponding concepts for derived terms.
3. Find relevant documents for concepts.
4. Create ranking for set of found documents.
5. Create ontology fragment containing necessary information to present concept neighbourhood
6. Find "shared concepts": concepts occurring in 50% of the documents

1: Query -> Terms

- Free text
 - Tokenise → lemmatise → create combinations for multiword terms (e.g. "space bar"), or:
 - Automatic substring match, or:
 - Substring match followed by manual selection of terms
- Take into account: diacritics (é -> e)

2: Term -> Concept

Not always 1:1 mapping.

- Corresponding concept is missing from ontology
 - LT4eL: not in lexicon
 - Unique result: term is lexicalisation of one concept
 - Multiple concepts from one domain, e.g.:
 - Key (from keyboard)
 - Key (in database)
 - Concepts from more domains:
 - Window (graphical representation on monitor)
 - Window (part of a building)
 - Different concepts for different languages:
 - “Kind” (English: sort/type)
 - “Kind” (German: child)
- Let the user choose: present multiple browsing units

3: Find relevant documents for concepts

- Simplest:
 - Disjunctive search:
 - For each concept, each document that is annotated with it is returned
- Use super/subconcepts
- Further possibilities
 - Conjunctive search:
 - Combination of concepts must occur in a document
 - Context search:
 - Combination of concepts must occur in a paragraph or sentence
 - Word & Concept search combined:
 - Document must contain concepts as well as certain words

3: Find relevant documents for concepts (continued)

- Is a superconcept of the document topic really relevant for the document too?
 - Negative example: `It4el:Subroutine` \subset `It4el:Software`.
Other children of `Software` are e.g.: `Shareware`,
`AuthoringLanguage`
 - Positive example: `GraphicalUserInterface` \subset `UserInterface`
- How useful is it, to find documents that treat a subconcept?
 - `It4el:Program` has 93 subconcepts, e.g.:
 - `ApplicationProgram`
 - `Computervirus`
 - `Driver`
 - `Unzip`

IR

CLIR

EBMT

EBMT+CLIR

4: Ranking

- Annotation frequency: number of times that concepts for search are annotated in the document
 - Normalise: divide by document length
- Superconcepts and subconcepts of search concepts have lower weight
 - A factor determines their weight

IR

CLIR

EBMT

EBMT+CLIR

Evaluation

- Within ILIAS we will have also a keyword search engine (LUCENE)
- Results on ontological search will be compared with the keyword search (precision/recall)
- Still in discussion is the qualitative evaluation

IR

CLIR

EBMT

EBMT+CLIR

Goals of the Evaluation

- Within an eLearning system we may prove:
 - Not only that the user retrieves the appropriate documents
 - But also that the learning process is improved through ontological search
- In comparison with crosslingual retrieval in web:
 - Domain in well defined
 - Only under specific conditions multilingual material is available
 - The degree of user's knowledge in another language may influence the validation results

Validation -Scenario-

1. student selects languages A,B (C) as search languages
2. student introduces keywords in language A -->applies keyword search --->Result: Set 1 of documents
3. student introduces keywords in language A ----> with the help of ontology retrieves the term equivalents in language A, B (C) ----> performs keyword search in each of the languages ----> Result: Set 2 of documents
4. student introduces keywords in language A----> performs ontological search (as blackbox process) --> Result: Set 3 of Documents (hopefully in more than language A)
5. student introduces keywords in language A --> browsing and search of the ontology ----> Result: set 4 of documents (probably in language A and some other languages)

IR

CLIR

EBMT

EBMT+CLIR

Further work

- Final Implementation and testing of the cross-lingual search engine. Integration with ILIAS
- Evaluation of the ontology and search results
- Compare pure ontological cross-lingual search with EBMT results
- Design particular scenarios appropriate to the existent content and available multilingual test-users

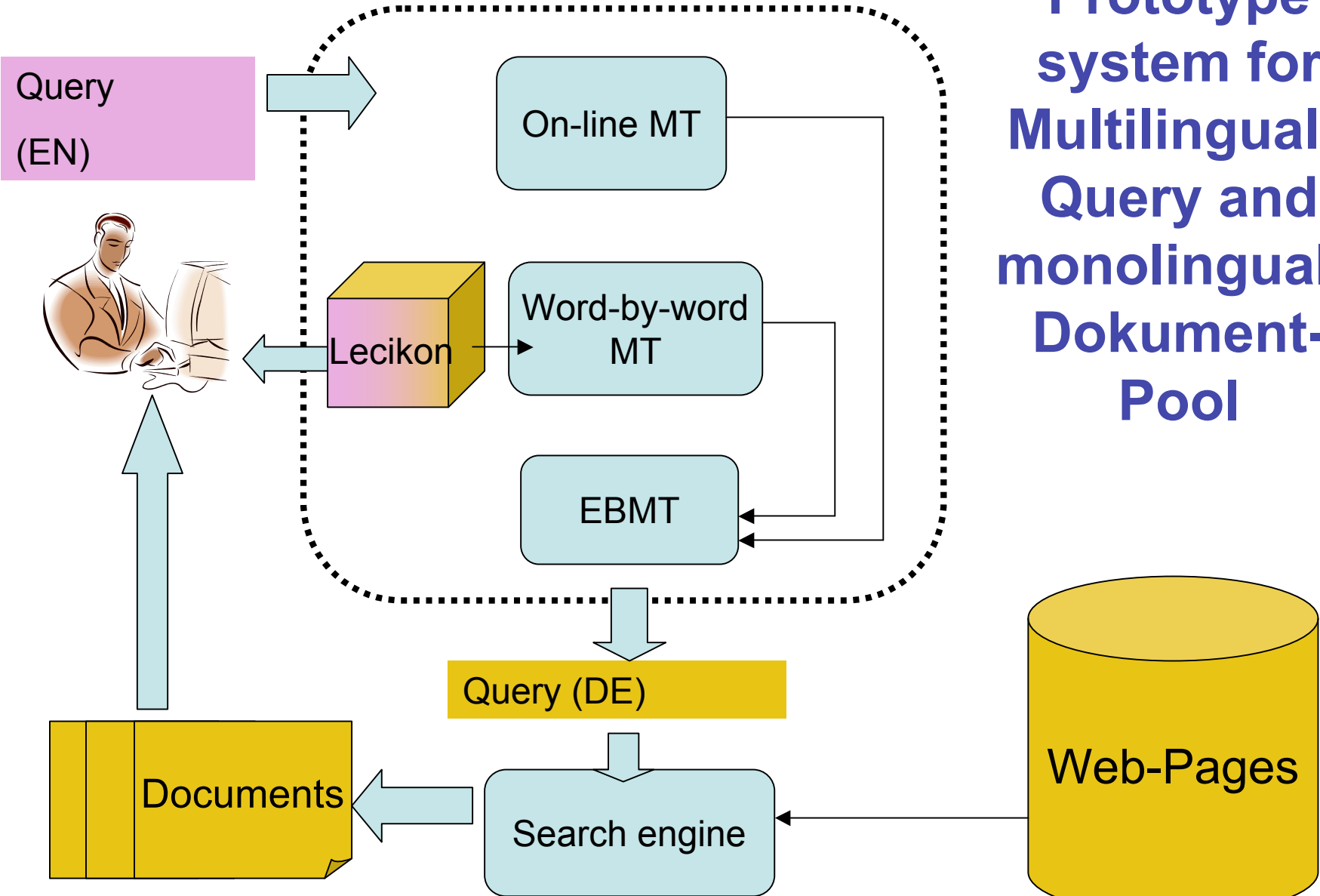
IR

CLIR

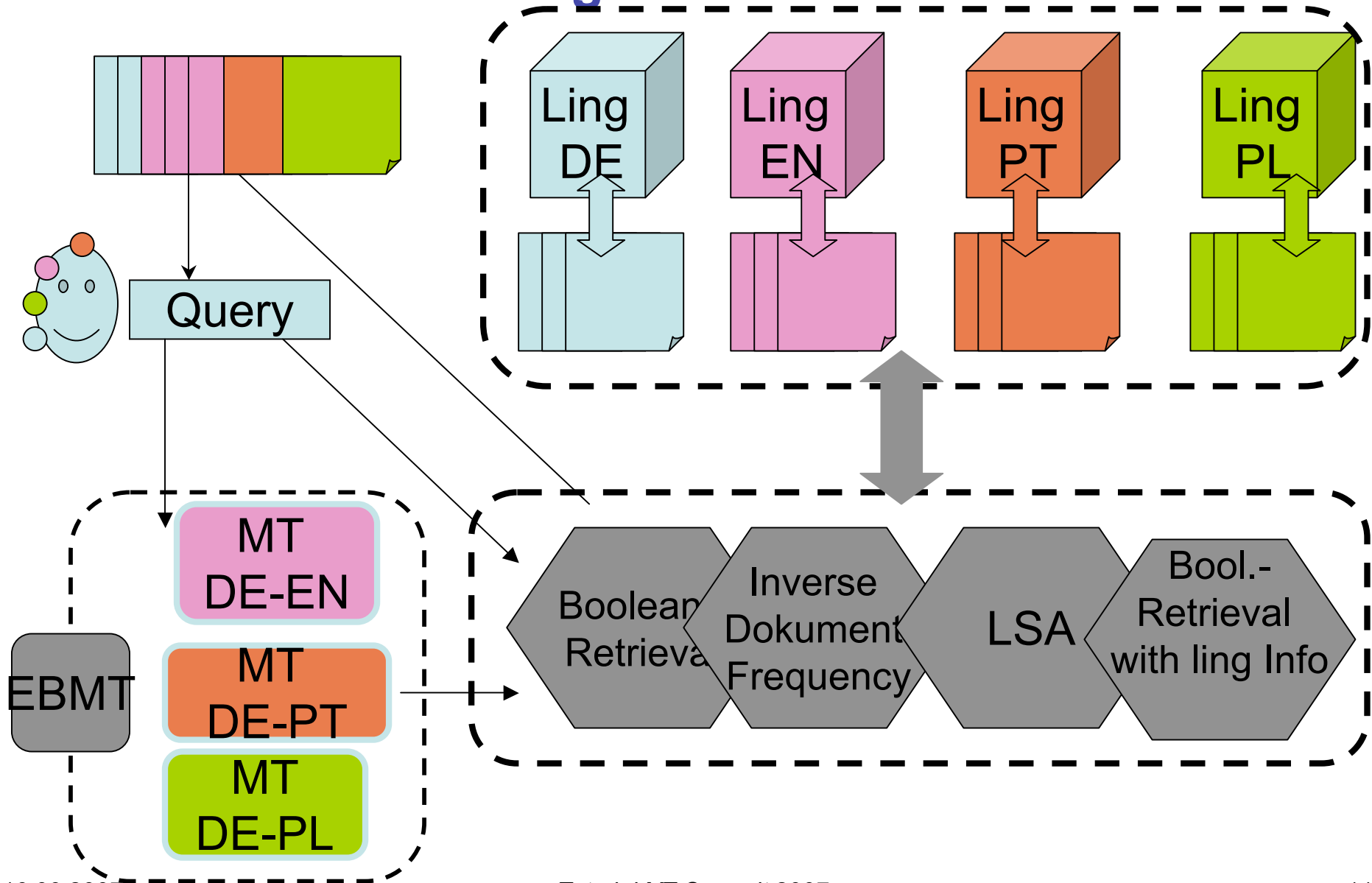
EBMT

EBMT+CLIR

Prototype system for Multilinguale Query and monolinguale Dokument-Pool



Prototype System for testing IR Methods in Multilingual environment



Conclusion

- EBMT , even without added linguistic knowledge can be used successfully for crosslingual search.
- However without an ontology backbone , only the use of EBMT does not overcome the gap between the terms of the query and the concepts the user had in mind.

Literature

- An introduction to Information Retrieval , Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, Draft Version 2006, Cambridge UP
- Crosslingual Information Retrieval, Slides by Gr. Thurmair, Pisa 2004
- Recent Advances in Example Based Machine Translation, A. Way and M. Carl (Eds.), Kluwer Academic Publishers, 2003