SEVENTH FRAMEWORK PROGRAMME
THEME 3
Information and communication Technologies

PANACEA Project

Grant Agreement no.: 248064

Platform for **A**utomatic, **N**ormalized **A**nnotation and
Cost-**E**ffective **A**cquisition
of Language Resources for Human Language Technologies

D8.1
Analysis of Industrial User Requirements

Dissemination Level:	Public
Delivery Date:	July 16, 2010
Status – Version:	Final
Author(s) and Affiliation:	Gr. Thurmair, V. Aleksic (Linguatec)
Contributors:	DCU, ELDA, UPF

Table of contents

0 Introduction

0.1 Outline

The PANACEA tools are supposed to create resources for Language Technology (LT) applications. One of the goals of the project is that the combination of PANACEA tools in the form of a factory can also support the process of the creation of Language Resources (LR) in contexts of practical importance.

First, the paper discusses the question of who are the users / types of users of PANACEA, and which workflows will need to be supported.

Part A describes the user **requirements** which need to be met if practical use of the PANACEA tools should be envisaged. In WP 8, the target industrial application has been selected to be Machine Translation, both in the context of a rule-based and of a statistical system.

The requirements will be divided into three main groups:

- Requirements to the production of the LRs, i.e. the PANACEA **factory**. The factory must be usable and functional, and must provide all functionality required to fulfil the task of LR creation.
- Requirements to the **tools** which are offered in PANACEA. Tools must meet requirements in accessibility, output quality, format compliance etc.
- Requirements to the output of the tools and their quality for **Language Resources**. This group of requirements focuses on required content, e.g. the required annotations of extracted dictionaries, the data quality for SMT creation, etc.

The context is the application of MT to a new specific domain, i.e. a tuning and adaptation task. It is understood that the quality of MT can be improved by adapting it to specific domains in which the customers operate.

Part B describes **evaluation** criteria. It again is divided into three chapters:

- Evaluation of the factory, following the requirements set up. (This, however, will be done in WP 7 where the factory is one of the focal elements).
- Evaluation of the single tools, from a ‘final evaluation’ tool-based perspective (WP 8.2), evaluating the outputs / results of combinations of tools
- Evaluation of the PANACEA system in a task-based environment (WP 8.3), i.e. tuning of an MT system to a particular domain.

Finally, part B gives a detailed **work plan** for all tasks defined in WP 8.

0.2 Terminology

0.2.1 Definitions

AAI [Stanica 2006]
Authentication and Authorization infrastructure
An infrastructure that provides Authentication and Authorization Services. The minimum service components include Identity and Privilege Management with respect to users and resources.

Factory
The set of the platform and the NLP tools used to produce LR.

Metadata [Guenter 2004]
Structured information that describes, explains, locates, and otherwise makes it easier to retrieve and use an information resource.

Metadata registry [Guenter 2004]
A formal system for the documentation of the element sets, descriptions, semantics, and syntax of one or more metadata schemes.

Platform
The set of tools (registry, workflow editor, etc.), software, documentation (closed vocabularies, format definitions, etc.), which combined define the PANACEA interoperability space. The NLP tools used as web services are not considered to be part of the platform.

Provenance data
Information that provides a traceable record of the origin and source of a resource

Registry
Repository focused on the needs of SOA environments typically used to publish, search and retrieve a wide variety of technical documents and information as WSDL location, documentation, schemas, service descriptions, business process design models, policy documents and so on.

Resource [Berners-Lee 2005]
The term "resource" is used in a general sense for whatever might be identified by a URI. Familiar examples include an electronic document, an image, a source of information with a consistent purpose (e.g., "today's weather report for Los Angeles"), a service (e.g., an HTTP-to-SMS gateway), and a collection of other resources. A resource is not necessarily accessible via the Internet; e.g., human beings, corporations, and bound books in a library can also be resources. Likewise, abstract concepts can be resources, such as the operators and operands of a mathematical equation, the types of a relationship (e.g., "parent" or "employee"), or numeric values (e.g., zero, one, and infinity).

Repository [CiTER]
Facility that provides reliable access to managed digital resources.

SOA [Mackenzie 2006]
Service Oriented architecture
A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations.

SP [Stanica 2006]
Service provider

An entity that provides access to a service.

Web service [Brown 2004]
A web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format.

Workflow [Wulong 2001]
Workflow is a term used to describe the tasks, procedural steps, organizations or people involved, required input and output information, and tools needed for each step in a business process.

1 Users and Use Cases

When setting up a system like PANACEA factory, it is worthwhile to define the target user group, and to think support typical use cases in an optimal way.

1.1 Types of Users

1.1.1 End users

End users would not be the typical user of PANACEA. End users do not have linguistic skills; they may want to look up dictionaries, to translate some text by a web based translation engine, at most to produce KWIC type concordances or similar corpus analysis results.

As opposed to other activities like Language Grid, PANACEA is not designed to support this type of applications; by definition it creates resources for such systems which end users would possibly want to use.

1.1.2 Linguistic administrators („users“)

This is the typical use case for PANACEA factory. PANACEA assumes that some linguistically trained person collects resources to improve or extend their applications (new domains, languages etc.).

Such people need to have skills in computational linguistics, some programming experience, but they would not be ‚hard-core‘ programmers. They would be the main *users* of the factory.

This group of users is called ‚users‘ henceforth.

1.1.3 Technical administrators („administrators“)

This user group would be able to configure the factory, redefine the web service interfaces, worry about throughput and scalability, etc. They would have very good programming skills, but maybe less skills in computational linguistics. They would be the main *administrators* of the factory.

The overall system will have to factor out the task of technical administration, as it does not just refer to the administartion of the factory but also to the services provided. Support on the SP side will be part of the services offered.

This group of users is called ‚administrators‘ henceforth.

1.2 General Use Cases

Typical use cases exist for these user groups, some of them are sketched here. These use cases could be converted into test cases for testing the PANACEA system.

1.2.1 Use cases for users (linguistic administrators)

The following use cases could be imagined for PANACEA factory users:

Corpus Tasks

Such tasks could comprise activities like

- Find a corpus by web crawling
- Process a corpus: sentence-segmentise it, tokenise / lemmatise / tag it

- Align two corpora: on document level, on paragraph level, on sentence level

Dictionary tasks

Dictionary tasks would be:

- Create a dictionary from corpus data (general purpose or domain specific)
- Enlarge a dictionary with corpus-extracted information (on entry level, on annotation level (additional annotations), on transfer level (additional translations)
- Search corpora for new / unknown words (to identify dictionary gaps)
- Trace word occurrences over time ('word of the day')

Extraction tasks

Such tasks could comprise:

- Extract information items from corpora (named entities, or just key terms)
- Build Alerting system (do texts match the alerting profile?)
- Topic assignment (create classifiers for a list of topics, assign topics to a corpus of incoming texts)
- Opinion mining (extract opinions about persons / products / product features)

Translation Tasks

Such tasks would be:

- Collect / add corpus data for SMT creation
- Create Language Model, for specific language, and / or for specific domain
- Create Translation Model (new language direction, new specific domain)
- Create / Adapt an (R)MT dictionary (with translations, with linguistic annotations (monolingual, transfer))

The standard case will be that users of PANACEA will already have resources available, and want to update / merge them with new material. So while the first set of services would be relevant to create them, in later development phases services to compare and merge resources will become important.

1.2.2 Use cases for administrators

Administrators of the PANACEA platform typically would need to be supported by the following activities:

User maintenance

This is relevant as PANACEA will have registered users. Activities would be:

- Users contact administration board for participation (by email)
- Admin updates user DB and sends access data (ID and password) to users email

(same for changes / updates and deletions of user records).
This use case would have to be extended in cases where access to PANACEA is not for free.

Service maintenance

PANACEA services must be maintained:

- adding / registering new services (or giving guidelines how to do this)
- removing services no longer available
- monitoring service availability and service access

Resource maintenance

PANACEA resources must be administered:

- adding / deleting specific resource to a service
- validating resources
- editing resources

System maintenance

The system infrastructure must be administered:

- maintaining the technical infrastructure
- maintaining the business logic (GUI, service and workflow configuration, logging, etc.)

1.3 Use cases for Industrial evaluation

PANACEA WP 8 aims at evaluating the system in industrial contexts. As the project intends to provide tools for several industrial use cases, two of them will shortly be sketched.

1.3.1 Alerting System

An alerting system would inform a user on developments happening in the internet: Mentioning of persons in newspapers, alerting users for new business developments, opinion mining on the features of a newly launched product etc.

Such a system would essentially require the following building blocks:

- a web crawler, to identify relevant documents (like newspaper articles, blog contributions etc.)
- a normalisation component to extract good text from the web documents

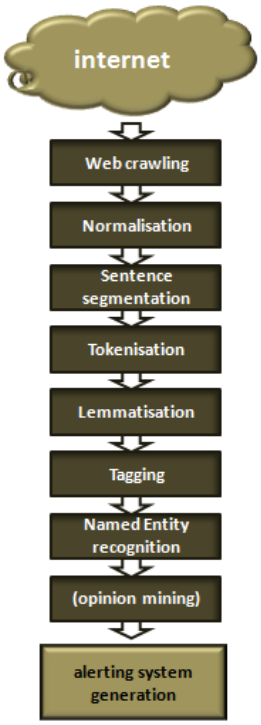


Fig. 1-1: Possible workflow for opinion mining

- a segmentation of the text into sentences and tokens
- lemmatisation, to find the lexical items
- tagging for a shallow syntactic description of the input sentences
- named entity recognition, to identify the information objects of interest (persons, products etc.), and possibly an opinion mining component which detects the opinions of the users on the identified objects.

In case the result matches a given alerting scheme the users are informed about this new document.

Such a system could be extended into the multilingual domain, meaning that many monolingual applications would run in parallel. It could be imagined that PANACEA tools would be used to extend the language coverage of such an existing application.

Such a workflow should be supported by PANACEA; however it will not be systematically evaluated in WP 8[1].

1.3.2 Machine Translation System

PANACEA WP 8 defines a specific use case for evaluation, which is the adaptation of an MT system to a specific / specialised domain.

This a very complex use case, however is does not cover all PANACEA tools, nor all PANACEA languages. In turn, it has practical relevance, as the production of MT systems is one of the major industrial applications of language technology.

Details and requirements for this use case are given below (Chapter 3). It will imply:

- Web crawling, in search for a corpus of parallel documents for a particular special domain
- Normalisation of these documents; removal of boilerplates, normalisation of character codes, hyphenation, etc.
- Sentence segmentation, breakdown of texts into sentences
- Sentential alignment; handling of non-alignable segments
- Tokenisation, both for RMT and SMT usage

With this toolset, the input for one type of MT systems can be generated.

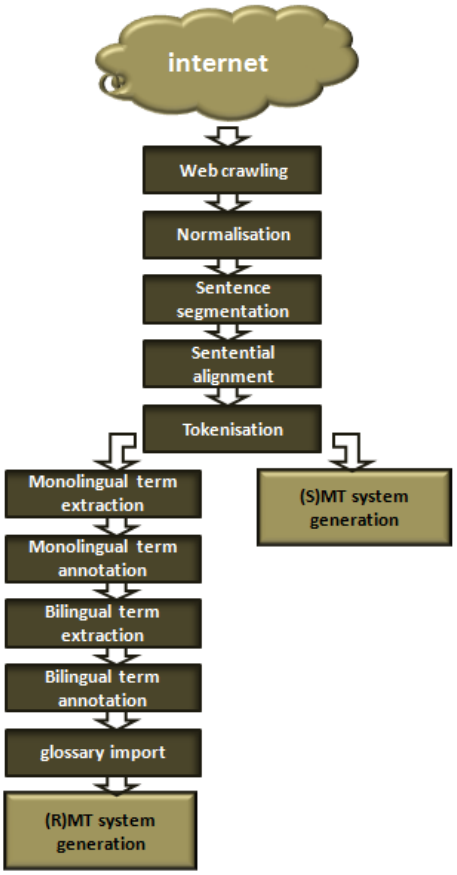


Fig. 1-2: Possible workflow for MT production

For RMT systems, additional tools are required which produce the glossaries describing the domain-specific terminology. These tools are:

[1] Evaluation of an opinion mining application is not intetnded for PANACEA; it is just mentioned to show that machine translation is not the only application which PANACEA is able to support.

- Monolingual term extraction, identifying the source terms (both single and multiwords)
- Monolingual term annotation, producing the entry annotations required by the MT system; both for the source and later for the target side entries
- Bilingual term extraction, identifying translation candidates for a given source term
- Bilingual term annotation which defines transfer conditions for lexical selection in case of for 1:n translations
- Glossary term input, to merge the domain specific terminology with the already existing terms[2]
- Named entity recognition for proper names, which must be protected from being translated, or added to the dictionaries as proper names

In a factory-like workflow, these tools should be concatenated in (maybe two) series of workflows, to be called 'General-MT-adaptation' and 'RMT-adaptation' respectively.

It should be noted that some of these tools, like term extraction, named entity recognition etc., themselves can be workflows, consisting of several elementary steps like dictionary lookup, tagging etc.

[2] To do this, the tools of existing systems will be used; the PANACEA lexicon merging tools in WP 6 would have a different focus and language coverage.

PART A: Requirements

2 Factory Requirements

The first group of requirements comprises the functionality, usability, and operability of the PANACEA factory. It forms a subset of all factory requirements (collected in the WP 3 specifications[3]), written from a users' point of view. Such a point of view does not differentiate between requiremetns to be fulfilled by the *platform* software, and requirements to be fulfilled by the provided services. For them it is just one infrastructure. Therefore all such requirements are collected here, in in case there are different addressees for some requirements this is mentioned there.

2.1 Functional Requirements

This section defines the functionality of the factory.

2.1.1 Requirements to running workflows for LR creation

Req-FCT-001: Inspect available services

Users must be able to get an overview of the possibilities on the PANACEA platform: this would be the first action after joining PANACEA ("what can I do here?"). The overview should list the services, their availability, the languages covered, the resources used, accessibility / copyright status, and other relevant information.

It should be sufficient to determine if the PANACEA service portfolio could meet the users' needs.

Req-FCT-002: Run a service

Users must be able to run a service: Connect to PANACEA platform, launch a service, download the result.

Req-FCT-003: Edit service parameter values

Users may want to give parameters for a service, e.g.: language, domain, dictionaries to be used, corpora to be accessed. Parameters depend on the services; services must specify which parameters they support.

Req-FCT-004: Inspect input/output data

Users should be able to inspect input and output files of a service. Each step of the PANACEA platform must be understandable for its users.

(E.g. some corpus files are too big to be opened in an editor).

Req-FCT-005: Inspect resources

Users should be able to find documentation on the resources used by the services, in order to decide if the service matches their requirements. (E.g.: what type of corpus data is used? Does the domain fit? etc.)

[3] PANACEA D3.1 'Requirement Analysis of the Platform'

Req-FCT-006: Upload user resources

Users may want to upload their own resources for a given service, like: user-specific dictionaries; customer-specific parallel texts, etc.

Service providers may decide to offer uploading possibilities for users, and to merge those additional LRs into their LR bases, after having validated them.

Req-FCT-007: Validate resources

Users should be able to ask for a validation of some resources before they launch them into the PANACEA platform. Validation refers to technical (compliance to specific formats) as well as linguistic (availability of certain annotations, like POS) elements. At least technical validation should be provided. (This is a requirement to the service providers rather than the platform: They should be accepted as PANACEA service only if they provide the appropriate tools).

Req-FCT-008: Configure services into workflows

Users must be able to create configurations of services, i.e. combine services into workflows which perform several services in a row.

The platform should support users in this configuration task.

Req-FCT-009: Run workflows

Users must be able to run such configurations, and stay in control of this process (monitor, inspect results etc.).

2.1.2 Requirements for workflow administration

In cases users have to perform the same workflow repeatedly (e.g. periodic dictionary updates), it would be helpful to be able to store a certain workflow configuration, and re-run it periodically. The requirements then would be:

Req-FCT-101: Store a workflow configuration

This stores a certain workflow configuration, under a user-selected name. The configuration is linked to a user record; this user owns the configuration.

Req-FCT-102: Delete a workflow configuration

This deletes a stored workflow configuration. Only the owner of the configuration can delete it.

Req-FCT-103: Show workflow configurations

This shows all existing workflow configurations which a user owns.

2.1.3 Requirements of resource administration

The PANACEA tools use themselves Language Resources, e.g. corpora, dictionaries etc., for instance to create phrase tables for SMT. PANACEA factory users should be in a position to define which resources should be used by a given web service; e.g. if they want a specific domain dictionary they may want to exclude general-purpose corpus data. Therefore there should be flexibility for each tool to specify which resources it should use in a given workflow (e.g. ‘Create a translation table but without using Europarl data’) (this could be told to the respective web service by means of a parameter list).

A central PANACEA storage of all resources is not envisaged; it is in the responsibility of the respective service *providers* to expose possibilities for resource administration.

Req-FCT-111: Add LR to service

The idea is that users can upload corpus data or dictionaries or whatever resources, to be used by the respective service. This requires that the LR is valid (formally and contentwise).

Req-FCT-112: Delete LR from service

Only the LRs which a user has uploaded can be deleted by them.

Req-FCT-113: Validate LR

In case new resources enter the PANACEA factory universe there should be some validation check, to prevent a service from crashing due to ill-formed LRs. Validation should be done at least on the level of formal correctness (compliance to formatting guidelines).

Req-FCT-114: Edit LR

While editing of LRs should not be supported (as one user may correct something which another user decided not to be wrong), there should be a way of controlled corrections of wrong entries etc., and an infrastructure for this should be available.

All these requirements refer to the *services* provided for the platform, not to the platform

2.1.4 Requirements for the Registry

PANACEA factory creates an open environment, and it is expected that new services may want to join the platform, e.g. a new tagger for Polish wants to be integrated. Such a service must be registered and made know to everybody; and later in the lifecycle it may be deregistered again.

Req-FCT-121: Register a web service

Registering a service includes checking it for compliance with the factory interfaces, promoting it to the users, integrating it into the presentation and monitoring tools, etc.

It should be the task of technical administrators, not of PANACEA users, to register services.

The registration will ask for a specific set of metadata (a Basic Metadata Description (BAMDES), cf. Parra et al. 2010).

Req-FCT-122: Deregister a web service

Administrators should be able to deregister a service, e.g. if it is not longer supported.

Req-FCT-123: Announce a web service

Users of the PANACEA platform need to be informed when new services are added / existing services are disabled etc.; in the simplest case by sending emails.

Req-FCT-124: List web services

Users should be able to easily browse a list of web services.

Req-FCT-125: Search web services

Users should be able to make searches based on some metadata, tags or others.

Req-FCT-126: Documentation and annotation of web services

Web services should be able to be well documented and annotated in the Registry. These annotations should follow some closed vocabularies or metadata guidelines if necessary (PANACEA metadata).

2.1.5 Requirements for user administration

It was decided that the PANACEA system will be freely available but will have some registration procedure. This means that there must be support for administration of users. These requirements do not have priority for the central functionality of the platform; they may become relevant in later versions, and in the deployment phase.

Req-FCT-131: Add a user record

This creates a new user record. A minimal approach is to have user-id, password, and email as elements of a user-record. There will always be an action for an administrator to confirm the new user record / admit a new user.

Req-FCT-132: Edit a user record

E.g. allow for changing the password, or the email. If users should be able to edit their own records they need a GUI to do so.

Req-FCT-133: Delete a user record

It needs to be decided how users will be treated; automatic deletion would be envisaged e.g. in cases where users are admitted only with certain time limits.

2.2 Usability Requirements

Talking about usability implies to have a clear view on the target users of the PANACEA system. Two main types of users have been defined above: linguistic administrators, as ‚users' of the PANACEA factory, and technical administrators of the factory. Both types of users need to be supported.

2.2.1 Requirements for Users

Req-FCT-201: Users need an overview of available services

Such an overview should at least contain: a list of the services / their availability / the languages they support / the LRs which they use / accessibility status / <additional information>.

Users should be able to sort the services according to these criteria.

Req-FCT-202: Users must be able to configure a workflow

a. There must be an editor which allows users to configure, edit and safe workflows for particular tasks, by selecting the appropriate services.

b. The editor must verify the validity of the selection (e.g. detect and propose missing steps in the workflow);

c. It should also verify if the selected service is available.

d. Designed workflows should be able to be shared in a portal.

Req-FCT-203: Users must be able to run a workflow configuration

Users may either compose a workflow ad-hoc, using the editor, or select a pre-defined workflow.

Users could have the option to run it all-in-one-step, or stepwise, with the possibility to inspect intermediate results.

Req-FCT-204: Users must be able to monitor the progress

As many things can happen during such a workflow (e.g. non-availability of services, empty files, etc.), and as some workflows may be very time-consuming, it is necessary to inform users about the status of their workflow, by showing how the progress is, and by sending meaningful error messages in cases the system has problems.

Req-FCT-205: Users must be able to inspect results

This holds both for the final results (e.g. a dictionary, a named entity list etc.), as well as for intermediate results, to verify that the system behaves as expected. System must provide a suitable displaying functionality (e.g. by launching the user's editor).

Users should also be able to edit intermediate results if possible.

Req-FCT-206: Other user activities

For other user activities (e.g.: delete workflow from workflow list; get / change access data etc., cf. reqs FCT-121ff. above), some documentation should be available.

2.2.2 Requirements for administrators

Req-FCT-250: Administrators' Documentation

No special GUI will be developed in the first version of the PANACEA factory for administrators. Instead, there will be documentation how the different tasks described above (management of users, of services, of resources etc.) will have to be performed.

This is relevant as we want other researchers / groups to offer their services in the PANACEA platform; they need clear technical advice how to do this.

2.3 Operational Requirements

This group of requirements talks about the internal state of the platform, and how its users would be informed. Users will not use the platform if its internal behaviour is not transparent.

Req-FCT-301: Availability of services

As this is a key issue in a web service environment, the Registry must have information on the availability of its services at any time: accessibility, proper functioning etc. In order to communicate this to the users, the Registry will need a possibility, provided by the services, to learn about their status.

Req-FCT-302: Speed / Waiting times

The workflows launched by users should be able to finish in a reasonable amount of time. No one would expect instant responses, however excessive waiting times will reduce the level of accessibility significantly.

Req-FCT-303: Scalability

In case of bottlenecks in processing, the platform, or rather the respective services, should provide additional resources to guarantee acceptable throughput; alternatively some upper limit threshold should be communicated to users.

Req-FCT-304: Logging

For each workflow executed by the factory, there should be a log file, stating when it was started and finished, intermediate steps, parameters used (e.g. languages), error messages of the different components, maybe statistics (e.g. sentences processed), etc.

This is very helpful for users and essential for administrators in cases where surprising results are delivered.

Req-FCT-305: Error Handling

The system needs a smart error recovery strategy, depending on the type of errors. From the delivery of informative information of the users about errors to automatic restart of services, there needs to be an error -> action list so that users have a good view what happens in which case.

Req-FCT-306: Validity Checks

There is a special kind of errors which needs to be reported; this is in case resources used by the services (either at registration time, or at runtime) are not well-formed. This is a very frequent source of errors: empty lines in parallel corpora, malformed dictionary entries etc.

Resources which are not valid should not be able to be registered, and terminate a workflow (e.g. if the malfunction of a service creates ill-formed output). This is a requirement for the services, however, not really for the factory.

The services should provide an option to run validity checks according to the respective resource, so that the platform can report errors, which should be as explicit as possible (line number / entry number / type of error etc.).

2.4 Sustainability Requirements

This set of requirements refers to a state where the PANACEA platform is running and needs to be sustained. Aside from organisational questions (how will the PANACEA platform be maintained after the end of the project?), there are also technical issues.

In addition to the administration requirements described above (new users, new services etc.), the following requirements could make sense.

Req-FCT-401 Bug reporting: services

There must be a mechanism by which errors in running the platform and its services can be reported (e.g.: service produces empty output). These bug reports refer to the software functionality.

Req-FCT-402 Bug reporting: resources

There must be a mechanism by which users can inform the administrators of services about bugs in their resources (wrong dictionary entries, missing abbreviations etc.). Owners of the services may want to be informed about the quality of their resources, and profit from improvement proposals.

Req-FCT-403 Versioning

The PANACEA platform must be developed in versions, with release notes specifying the difference to the previous versions, the fixes, new features etc.

3 Requirements for the PANACEA Tools

The tools to be produced in PANACEA will have to meet certain requirements. In the current work package, the requirements referring to the tools focus on end-to-end workflows, and consider the functionality / usability of the tools only insofar they are linked to the overall purpose.

Detailed evaluations of the single tools, and evaluations of their improvements within the PANACEA development cycle, are not part of this analysis; they will be dealt with in WP 7 of the PANACEA work plan. WP 8 uses a black box rather than a glass box evaluation.

3.1 General requirements

There is a set of requirements which holds for all tools in the PANACEA factory.

Each tool must meet the following requirements:

3.1.1 Technical requirements

Among the technical requirements are the following ones:

Req-TOL-001: The tool must be implemented as a Web Service

Following the architecture as defined in WP 3, all tools in the PANACEA factory will be accessed as web services. So each tool must be accessed as a web service.

Req-TOL-002: The tool must be accessible by the PANACEA platform

Tools participating in PANACEA must expose their accessibility towards the PANACEA registry, and must register their service in the way specified in the WP 3 specifications.

3.1.2 Integration requirements

Req-TOL-003: The tool must be compliant with the input / output interfaces

PANACEA will define interfaces for input / output, parameter settings etc. of the tools participating in the factory. The tools must be compliant with these definitions.

Req-TOL-004: The tool must have an interface validation

The single tools should reject input which is not compliant with its interface definitions (e.g.: wrong character code; empty lines etc.), and validate their output for their interface definition compliance. In a chain of tools, error analysis looks at the right tool, not at a crash in a tool caused by some previous errors.

3.1.3 Quality requirements

Req-TOL-005: The tool must have a reasonable functional quality

I.e. the tool must 'do its job': tokenise a text, collect parallel web sites, etc. This is a very general requirement, and will be specified in detail below for each tool. The criterion for quality compliance is the usability of the tool in the global evaluation workflow.[4]

[4] There will be much more detailed criteria, stated in the documents produced by WP 7.

3.1.4 Usability requirements

The tool must be usable by the users of the PANACEA factory. This includes some requirements beyond the pure accessibility.

Req-TOL-006: The tool must have a reasonable software quality

This is a rather general statement about expectations of software used:

- It should be robust
- It should do its job in reasonable time
- It should give some feedback on its progress
- It should communicate any problems encountered

Details depend on the single tools, and for different tools, different aspects of software quality are relevant.

Req-TOL-007: The tool must have documentation

The tool must have been properly registered. Registration includes documentation. The documentation for a given tool must at least contain information on the following topics:

- How to access and use it: Input/output formats; interfaces; parameters; error handling
- Linguistic content of the tool, e.g.: tagset used, type of syntactic output produced, dictionary information expected, etc.
- Language resources used (dictionaries, corpus data)
- Availability information (licensing etc.)

There is a procedure for tool description, developed in PANACEA WP 4. The final documentation of a tool should contain at least the information documented there.

3.2 Specific additional requirements for the single tools

This section describes which additional requirements must be met by the single PANACEA tools, both on the side of quality and on the side of functionality / usability etc.

It considers requirements for such tools in the context of the main WP 8 workflow, namely to create resources for MT in specialised domains, for both rule-based and statistical approaches, and related to German <-> English directions (cf. fig. 1-2, repeated here).

In a factory-like workflow, these tools should be concatenated in (maybe two) series of workflows, to be called ‘General-MT-adaptation’ and ‘RMT-adaptation’ respectively.

The *General-MT-adaptation* workflow comprises web crawling, document normalisation, sentence segmentation, sentence alignment and tokenisation. From there either rule-based or data-driven MT systems can be built.

The *RMT-adaptation* workflow contains in addition monolingual term extraction and annotation, bilingual term extraction and annotation, glossary import and named entity recognition.

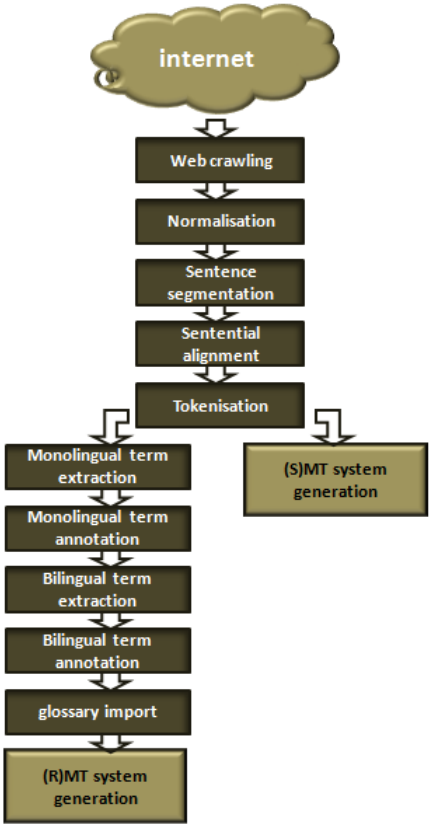


Fig. 1-2: Possible workflow for MT production

In order to be usable in an industrial context, these tools must meet some general requirements; these requirements are described in the following sections.

3.2.1 Requirements for the General-MT-adaptation workflow

This workflow must produce a list of tokenised aligned sentences.

Req-TOL-100: All required tools must exist for the language direction in question

This is quite obvious. Some tools are more language-dependent (normalisers, sentence-segmentisers), others are less, depending on the implementation.

Req-TOL-110: The crawl must produce sufficiently many bilingual comparable, document-aligned texts for extraction of parallel sentences

It should be noted that the scenario is a domain-specific adaptation of a given system; so it is assumed that some general bilingual training data are already available, and 'only' domain-relevant texts need to be found.

It needs to be found out what the minimum amount of training data is, required for domain adaptation.

Req-TOL-111: The crawler must be adaptable to certain domains

E.g. by allowing users to provide seed lists of key terms, by allowing them to restrict the search (e.g. only to BMW motorbikes site), etc. Good data selection is a key success factor for good MT results.

Req-TOL-120: The normalisation must provide documents in text form

This relates to the deformatting task, converting files from HTML / PDF etc. into a text representation, removing inline formatting, interpreting table structures, headings and other layout-related information[5].

This task is critical as some of the available tools (e.g. pdf converters) produce defective output which deteriorates the data material.

The output of the normalisation should be given in some standardised format, to be defined by WP 3.

[5] For the time being, rich text formats (.doc, .xls, .pdf, .ps etc.) will not be supported.

Req-TOL-121: The normalisation must eliminate / correct deficient strings

This task has a language-independent part (e.g. boilerplate removal from HTML documents, character conversion), but also language specific parts (de-hyphenation, possibly spelling correction, punctuation treatment etc.

Result of the normalisation must be strings, which, after tokenisation, can count as 'meaningful' words or symbols of the language to be processed.

(An open issue is whether multi-language documents must be assumed, i.e. bilingual texts where one column is in German, the other in English. This case requires a language identifier, and special treatment in alignment).

Req-TOL-130: The sentence segmentation must identify correct sentence boundaries

This task is language-specific. Typical cases are sentence-final abbreviations, punctuations mixed with digits, quotes, insertions with parentheses etc.

Wrong sentence segmentation leads to alignment and parsing errors; not more than 5% of the 'clear'[6] sentence boundaries should be erroneous.

Req-TOL-140: The sentence alignment must produce 'meaningful' correspondences

The better aligned the sentences are, the better the MT output quality will be. The result of the alignment should at least represent the state of the art.

Req-TOL-150: The tokenisers must produce 'meaningful' tokens

While most of the cases are clear, there are configurations which depend on the target system, like German hyphenation ('*XML-Beschreibung*': one or three tokens?[7]), treatment of number-digit combinations (*1.5, 2.5:2.5*), measure units (*€5,20, 7.2m/sec²*) and the like.

Tokenisers must be explicit in their decisions about such cases, so that users can react, e.g. by inserting appropriate conversion tools into the workflow.

Req-TOL-190: The General-MT-adaptation workflow must produce 'good' aligned material

Overall, the output will be aligned sentences, each sentence consisting of lists of tokens.

- Alignment must be such that the MT tools have a chance to find word correspondences.
- Tokenisation must be such that MT tools can make use of the tokens (by aligning them, by doing dictionary lookup, etc.).

Errors in tokenisation and alignment deteriorate the MT quality, and therefore must be minimised.

3.2.2 Requirements for the RMT-adaptation workflow

While lists of aligned and tokenised sentences can be used as input to tools like GIZA++, preparation of the data for an MT glossary requires an additional workflow, following the

[6] There are always uncertain and unsolvable cases, e.g. sentences crossing paragraph boundaries in enumerations. They are not considered.

[7] For a language model, *one* token should be assumed; for a morphological analyser, *three* tokens are easier to handle.

General-MT-adaptation workflow. It starts from aligned tokenised sentences and produces MT glossaries.

While some tasks of the RMT-adaptation workflow are also relevant for SMT (like word level alignment), the focus here is on producing glossaries for RMT systems. In this respect, the set of tools must support the following requirements:

Req-TOL-200: All required tools must exist for the language direction in question

This is quite obvious, like in the other workflow. Again, some tools are more language-dependent, others are less, depending on the implementation. Missing tools always require manual intervention.

Req-TOL-210: Monolingual term extraction must produce vocabulary for the whole domain

It must identify dictionary gaps, both for single words but also for multiwords (even where all multiword parts are known to the system). A special case is monolingual word sense disambiguation; it is unclear if this helps in translation[8].

As dictionary gaps usually lead to parse failures, the coverage must be as complete as possible (not more than 3% error rate).

The output should contain unknown terms as well as known ones (as they could have new translations in the narrow-domain context), presented as lemmata.

Monolingual term extraction should be done for both languages involved, because both sides need annotations (part of speech, morphology etc.), and because they need to be aligned later on.

Req-TOL-220: Monolingual term annotation must produce entries with the necessary annotations

The lemmata of the term extraction need to be annotated to be useful in machine translation.

Which annotations are needed in a monolingual entry is described below. From the workflow point of view there are two options:

- If annotations should be used *without* user checking, the error rate must be 3-5% of the entries (this is the error rate accepted for humans).
- If this is not achievable then the annotation results must be presented in a way which makes users more productive if they use the PANACEA tools, as opposed to doing the annotations themselves.[9]

Adding annotations to a dictionary entry is a time-consuming task, and productivity gains here are most relevant.

Req-TOL-230: Bilingual term extraction must offer translations for all domain terms

The task here is to find translations for the monolingual source term lists.

[8] For this discussion, cf. Vickrey et al. 2005, Lee at al., 2007, Specia at al. 2006

[9] For instance, presentations of alternatives usually make users slower. Interactive tools to help them in making annotations (like concordances) are useful but not in the focus of PANACEA.

Possible outcomes are:

- No translations could be found; this should be reported in the log file.
- The source entry is known, the target is known: The transfer is known, and applies also to the new domain
- The source entry is known, the target is new: There is an additional translation for the new domain, to be marked e.g. by a domain tag.
- The source entry is new, with 1 translation: This new entry could be domain-specific, and marked with a domain tag[10]
- The source entry is new, and has several translations: In this case, additional disambiguation information must be provided, beyond the domain tag.

Req-TOL-240: Bilingual term annotation must specify lexical selection

Object of this annotation effort is the transfer entry, not the monolingual entry. The annotation consists in tests and actions defining under which conditions a given transfer should be selected.

Req-TOL-250: The resulting domain-specific glossary must be exchangeable

While translation dictionaries usually have idiosyncratic representation, the resulting glossary should follow a general representation and adhere to standards of dictionary material exchange.

Req-TOL-251: The resulting domain-specific glossary must be imported into the MT system

Import requires merging of the new glossaries into the existing (general-purpose) dictionary. Merging strategies, and side-effects to other translation domains need to be considered.

Import requires a compiler which transforms the new glossary into a MT-system-specific format so that it can be imported.

Req-TOL-260: Named entities must be recognised

Named entities usually must not be translated[11]; therefore they receive a special treatment in MT. There are two ways of dealing with named entities in RMT:

- There is an NE recognition component at runtime which marks up named entities, either as 'do not translate'[12], or with a semantic label like 'person', 'place' etc. which can be interpreted by the MT system[13]
- There is a NE recognition in the term extraction process which identifies names, and adds them to the system dictionary, with itself as translation (en '*Clinton*' -> de '*Clinton*').

In both cases, NE recognition and semantic labelling is required; the difference is how such a component is used. (NE development is not part of PANACEA, so existing components will be used in the task-based evaluation context).

[10] However, it also could be a simple dictionary gap …
[11] except for some place names.
[12] Babych/Hartley 2003
[13] Thurmair 2006

4 Requirements for the Language Resources for the MT task

This section describes the requirements for the content of the LRs to be produced, and their quality. Of course, these requirements are application-specific, and they are language-pair-specific. In PANACEA the work plan is to use machine translation as the application, and German<->English as language pair, so the requirements will reflect these decisions.

4.1 Language Resources for rule-based MT systems

In case of RMT systems, the domain adaptation task mainly refers to creating a domain-specific dictionary, merging it with the system dictionary, and loading it for the domain-specific translations.

The PANACEA tools need to provide some basic information for MT dictionaries if they want to be a useful support. The following requirements define which information items / annotations are needed by MT systems to run additional dictionaries successfully.

Of course different MT systems use different forms of dictionary contents, and also different annotation schemas and formats. To avoid to be biased towards a particular system, the requirements contain a general description of the necessary annotations, without asking for a particular formalism: For instance, all systems somehow use inflectional information; however the way this information is stored and processed differs largely among the systems. Therefore a generic representation (some feature-value- based approach) will be defined, from which the different MT systems could generate their idiosyncratic representations, by running compilers.

In turn, information which no current MT system uses is dropped from the requirements list.

As a practical limitation, the requirements only comprise open word classes. Function words undergo very idiosyncratic coding in the different MT systems, and will not be subject to domain-specific adaptations anyway.

4.1.1 Monolingual Information

This set of requirements defines the information needed to process (analyse or generate) monolingual information. Not all MT systems use all information items; however there is a common understanding what current RMT systems would expect. The information provided by PANACEA must be described so that its representations can be validated and accessed.

Some of these requirements are language-independent, some are language dependent. As the PANACEA WP 8 is on German <-> English, only these two languages are considered. Other languages may require other annotations.

4.1.1.1 Basic information

Basic information defines an entry. Entries have lemma, part-of-speech, and possibly reading number. The requirements are:

Req-RMT-001: Every entry must have a lemma.

The lemma gives the entry in its canonical form. This form needs to be defined (e.g. if it is a multiword lemma and contains inflected parts).

Req-RMT-002: Every entry must have a part-of-speech-tag.

The POS information must refer to the dictionary specification, and contain a legal value.

Req-RMT-003: An entry may have a reading number

In case there are entries of the same <lemma,POS> information, a reading number may be used to differentiate readings within such a <lemma,POS> description.[14]

Req-RMT-004: Entries need an entry type

It defines if a lemma is a single word, a compound (in the German agglutinated form), or a multiword. Depending on the entry type, different dictionary annotations must be present.

4.1.1.2 (Single word) Morphology

Most RMT systems have elaborate morphology sections. In general, they use inflectional patterns, gender, number etc. to describe the entries.

Req-RMT-101: Nouns need a markup of inflectional class

This holds for both German and English nouns. Inflection must also cover umlauts (‘*Haus – Häuser’*) or irregular forms (‘*child – children’*). The reference inflection paradigm has to be defined.[15]

Req-RMT-102: German nouns need gender information

Double gender annotation sometimes points to different readings (de ‘*der Hut*’ (en ‘*hat’*) / ‘*die Hut*’ (en ‘*caution’*)); pluralia tantum cannot be marked for gender.

Req-RMT-103: Verbs must mark separable parts

Again this holds for German (‘*kauft … ein’*) and for English (‘*let … down’*, ‘*pass … on’*). The remaining part may have to be stored as well for analysis purposes (e.g. ‘*kaufen’* while the lemma is ‘*einkaufen’*).

Req-RMT-104: Verbs need a markup of inflectional class

For German, and for English. Like for nouns, the list of possible inflectional classes must be defined beforehand. There is a dependency to the existing dictionary content: In case all exceptions / strong verbs etc. are already in the dictionary, the inflection extraction can focus on the ‘easy‘ but frequent cases of weak verb inflection.

Req-RMT-105: German Verbs need a markup of their auxiliary in perfect tense

(*haben / sein*: ‚*er hat geschrieben*‘ – ‚*er ist geschwommen*‘).

Req-RMT-106: Adjectives must mark comparative/superlative behaviour

For German, and for English. Some adjectives can form comparative / superlative forms, others cannot. In English, we must know if comparative form is done by suffix (*-er/est*) or by particle (*more/most*), or both.

[14] To my knowledge, however, no current RMT system could make use of this information.
[15] In order to be system-independent, in OLIF (cf. www.olif.net) the inflectional class is given by an example: *Inflects_like …*; then each system knows how to make use of this information.

Req-RMT-107: Adjectives need a markup of inflectional class

For German, and for English. Like for nouns, the list of possible inflectional classes must be defined beforehand. Again, not the full paradigm needs to be specified in cases where irregular forms are already in the dictionary, and need not be extracted from corpus data.

4.1.1.3 Multiword morphology

Monolingual dictionaries will contain compounds or multiword terms, i.e. lemmata which have non-compositional meaning. For analysis, and generation, these entries need additional descriptions. This holds for German and English, and all parts of speech.

In an extension of the approach taken by (Grégoire 2009, Deksne et al. 2008), multiwords need a sequence of lemmata and parts of speech, and the head information, as minimal information items.

Req-MT-111: Multiword entries must mark a head

The head could be the number of the head element in the multiword expression.

Req-RMT-112: Multiwords need a list of parts-of-speech tags they are composed of.

E.g. ‚*analoges Signal*' would be 'adj-noun', while ‚*Recht auf Arbeit*' would be 'noun-prep-noun'. This is relevant as e.g. adjectives need to be inflected in agreement with the noun. This also holds for German compounds.

Req-RMT-113: Multiwords need a list of the lemmata of their parts

In the example, this would be ‚*analog,signal*' and ‚*recht,auf,arbeit*' respectively. Needed to identify multiword in analysis. This also holds for German compounds, the adjective ‚*versicherungsfremd*' would have ‚*versicherung,fremd*' as parts.

Req-RMT-114: German compounds need to specify the compounding infix

In the case just mentioned, the infix would be the '*–s'*. Would be needed if compounds are supposed to be generated (i.e. in generation in an English->German system).

4.1.1.4 Syntactic information

Syntactic information is not really standardised; however, most systems use some of the following annotations:

Req-RMT-121: Position

For German and English (adjectives and) adverbs: before / after verb, sentence-initial etc.; the value sets for analysis and for generation may differ as analysis may need a broader coverage than generation.

Req-RMT-122: Subcategorisation

Verbs, nouns and adjectives need to be subcategorised (common / proper noun, count / mass noun, passivation of verbs, etc.).

Req-RMT-123: Argument structure

Argument structures should be given, in terms of roles (obligatory and optional), role types (subject, direct object, indirect object), filler categories (nominal, adjectival, verbal,

prepositional, clausal), filler syntax (e.g. case marking), filler semantics (on simple feature level: human, animate).

4.1.1.5 Semantic information

Req-RMT-131: Semantic type

As current MT systems do not make much use of semantics, a simple type schema would be sufficient. It should also cover the named entities (place, person, product, company, time etc.).

4.1.2 Transfer Information

RMT systems use special resources (dictionaries) which link source and target resources, and add source language tests and (target language) actions. The link usually is based on a directed bilingual link of <lemma,POS> from source to target.

Standard lexical transfer replaces the source by the target term[16]. Complex lexical transfer is required in cases where 1:n transfers must be considered, i.e. where the right transfer must be selected from several target candidates. This is done by tests and actions.

There are different types of tests:

- **Domain tags** are the simplest way. Some entries are marked such that they are preferred if the text is from a specific domain.
- **Grammatical tests** usually refer to underspecified tree configurations, either local nodes (some feature value tests, like number) or partial trees (like a verb and its direct object).[17]
- Semantic / **conceptual tests** investigate the conceptual context in transfer selection; systems supporting this feature[18] usually rely on a larger context (paragraphs instead of contexts).

Actions are usually linked to tests, and take care of specific constellations for a given transfer, e.g. argument switching, idiosyncratic translations of prepositions, insertion / deletion of lexical material, and the like.

MT systems differ widely in the way such information is expressed and used, and in the formalisms how it is described; therefore, a representation of this information must be found which is sufficiently expressive, and as generic as possible. Single MT systems would have to compile it from there into their idiosyncratic environments.

4.1.2.1 Target Basic information

Req-RMT-201: Transfer link

Each transfer entry needs to provide <sourcelemma, sourcePOS> - <targetlemma, targetPOS> link.

In case the monolingual dictionaries contain readings, then the respective reading numbers will have to be provided as well.

[16] note that this can be multiword entries on either side
[17] It is a known problem for RMT systems that such tests do not work if the underlying structure is not built properly, i.e. if the parse fails. Parse failure affects both analysis and transfer.
[18] Thurmair 2006, Mihăltz 2005

4.1.2.2 Transfer tests

Req-RMT-211: Domain markup

In case the transfer is specific for a domain, this domain should be specified. Several domains per transfer are possible.[19]

Such markers could be set automatically, by a topic classification component[20].

Req-RMT-212: Grammatical tests

The dictionary must provide grammatical tests for lexical selection. Such tests can be local (on current_node) or configurational.

Req-RMT-213: Conceptual tests

The dictionary should provide conceptual tests for lexical selection, i.e. concepts which indicate (with a certain probability) a certain transfer selection.

4.1.2.3 Transfer actions

Req-RMT-221: Transfer actions

Some transfers require actions in the target language, like insertion / deletion of lexical material, argument switching, preposition subcategorisation, number switching, conjunctions requiring subjunctive case etc. Inasmuch such actions depend on lexical material they need to be specified in the transfer action part of these lexicons.

4.2 Language Resources for statistical MT systems

In order to generate an SMT system from parallel and monolingual corpora, resources need to be prepared from which the translation tables and language models can be generated. These resources must satisfy the following requirements:

4.2.1 Parallel corpora

Req-SMT-001: Parallel corpora must be of sufficient translation quality

SMT is able to work properly with parallel corpora, whereas comparable corpora are not adequate.

Req-SMT-002: Parallel corpora must be of sufficient size

Only sufficient amounts of parallel data can result into a good SMT system.

Req-SMT-003: Parallel corpora must be accurately tokenised

The tokens in the corpora must be identified and separated by blank spaces.

[19] Domain codes are not too helpful. Some can be assigned on monolingual level, some on transfer level; however even in specific domains, the general-purpose readings of a term can occur, and vice versa.
[20] cf. Thurmair 2006

Req-SMT-004: Parallel corpora must be aligned at sentence level

All sentences in the source language of the parallel corpora must be aligned to their counterpart translations in the target language (and vice versa).

Req-SMT-005: Parallel corpora must be in appropriate format

The parallel corpus must consist of two files (one for the source language and one for the target language) each containing the same number of lines. Each line must contain one (or more) tokenized sentences so that the ith line in one file is aligned to the ith line in the other one.

4.2.1.1 EBMT

The EBMT component of MaTrEx / Marclator requires marker files for source and target languages.

Req-SMT-011: Marker files available for the source and target languages

There should exist files with appropriate coverage of marker words for the source and target languages.

Req-SMT-012: Marker files are compliant with Marclator / MaTrEx marker file format

The marker files should follow the input format expected by Marclator.

4.2.1.2 Factored SMT

For factored SMT (optional) the following requirements must be satisfied:

Req-SMT-021: Parallel corpora must be lemmatised and PoS tagged

Each token of the corpus should be assigned both a lemma and a PoS tag.

Req-SMT-022: Parallel corpora format should be compliant with Moses factored format

Each token in the factored format consists of multiple factors delimited by "|". Tokens are delimited by a blank space as usual.

4.2.1.3 Syntax based MT

We might want to have additional information for syntax based MT models. These reqs will be defined in a next version of this document.

4.2.2 Monolingual corpora

Req-SMT-101: Monolingual corpora must be of sufficient linguistic quality

Minimum of spelling errors and other noise (words in other languages) should be present in the monolingual corpora.

Req-SMT-102: Monolingual corpora must be of sufficient size

Only sufficient amounts of monolingual data can result into a good language model. Languages with rich morphology and free word order require more LM training data than those with fixed word order simple morphology.

Req-SMT-103: Monolingual corpora must be accurately tokenised

The tokens in the corpora must be identified and separated by blank spaces.

PART B: Evaluation

5 Evaluation procedures of PANACEA factory tools, and resources

5.1 PANACEA Factory

The PANACEA **factory** will be developed within the PANACEA development cycles and therefore evaluated there (cf. WP 7). No additional evaluation activity will be performed in WP 8; it will be assumed that the factory works as specified, and covers the functionality required for the industrial evaluation task.

The aspects of the PANACEA factory most relevant for Industrial Evaluation are:

- Can the platform be configured for the task described in the use case 1.3 above?
- Can the platform be used to launch all the tools required for this use case? Can intermediate results be inspected?
- Can usable industrial-type systems be built with its results? (Input for MaTrEx and Personal Translator)

These questions will be answered in the WP 7 development cycle tests.

5.2 PANACEA tools

The same holds for the PANACEA **tools**. The tools will be developed as part of the development cycles. Within these cycles, evaluation and test procedures have been defined, which the tools need to undergo.

It should be noted that the industrial evaluation covers only a fraction of the PANACEA tools, both in functionality and in coverage; therefore the development of PANACEA tools has its own test and evaluation work, which is laid down in the specifications of PANACEA WP 7.

Relevant questions would be:

- Are the required tools available for the required languages?
- Is the output quality sufficient to support the building of an industrial system?

These questions will be further detailed in the evaluation of the development cycles in WP7. However, there will also be a look at the tools from a final version point of view; this will be described in the next section.

5.3 PANACEA resources

Also in terms of **language resources**, only a minor part of the PANACEA developments will be considered in WP 8, which focuses on Machine Translation resources for German-English language directions. PANACEA covers much more languages and resources, therefore evaluation of such resources will be done in WP 7 again.

The same evaluation principles as for the tools also hold for the language resources (availability, quality), so no specific evaluation task is planned in WP8.

6 Tool-based evaluation of PANACEA

This section describes the evaluation to be performed for the final version of the tools, after they have left the development cycle.

The tools will have been extensively tested inside of the development cycles. For the final tests, only their contribution to workflows will be investigated.

6.1 Evaluation target

The target of the tool-based evaluation will be to make sure that the tools required as parts of larger workflows function as expected in such contexts. Two main criteria will be evaluated:

- **Availability**: Are the tools available to support the different workflows in which they should be used? Do they support the required languages? Can they be accessed from the PANACEA platform? Do they run with reasonable performance? etc.
- **Quality**: Do the tools produce output in the quality required by the applications for which they are intended?

These questions are subject of the tool-based evaluation in WP8.

6.2 Evaluation object

The tool-based evaluation looks at the *output* of (clusters of) PANACEA tools, so the object of the evaluation will be *results* of PANACEA tools. The tools themselves are supposed to be evaluated in WP7; here, only the point of view of a final inspection is taken, and this inspection is based on the output which the tools produce.

This approach implicitly answers also the question of tool availability, as no output can be inspected if the tools are not available.

6.3 Evaluation criteria

The criteria of availability and quality comprise several of the requirements specified in chapter 3 above.

6.3.1 Availability criteria

By running the tools, answers can be given to requirements that the tools are accessible from the PANACEA platform (Req-TOL-002) as a Web Service (Req-TOL-001), and that the PANACEA interfaces are supported (Req-TOL-003).

Also, evaluation of the software quality (Req-TOL-006) and support of the required languages (Req-TOL-100, Req-TOL-200) can be evaluated. Also, the documentation requirement (Req-TOL-007) will be tested from an end-user point of view.

Availability also refers to the possibility to adapt the tools to a particular task, e.g. the crawler (Req-TOL-111), or the possibility to inspect intermediate processing result, and adapt / modify the language resources used by the tools.

6.3.2 Quality criteria

It will be assumed that the single tools will have been evaluated with respect to their quality in the component tests of the development cycles (WP 7). Therefore the tool evaluation focuses on

'strategic' output objects in the work flows, which integrate the results of several PANACEA tools. Such objects will be:

- aligned and tokenised sentences (level 1)
- annotated bilingual dictionaries (level 2)

The **sentence level evaluation** will answer the question if there are sufficiently many data, if they are normalised, sentence-segmentised and tokenised properly, and if the alignment produces meaningful results.

Evaluation will use human inspection of some parts of the aligned corpora, and counting errors of the tools which contributed to the output: Errors in normalisation, segmentation, tokenisation, and alignment (on sentence level) will be counted[21]. Details will be coordinated with the WP 7 component evaluation task.

The **dictionary level evaluation** will comprise four main tools: monolingual and bilingual term extraction, and monolingual and bilingual entry annotation. These tools create an annotated bilingual dictionary, to be used by MT systems in the task-based evaluation.

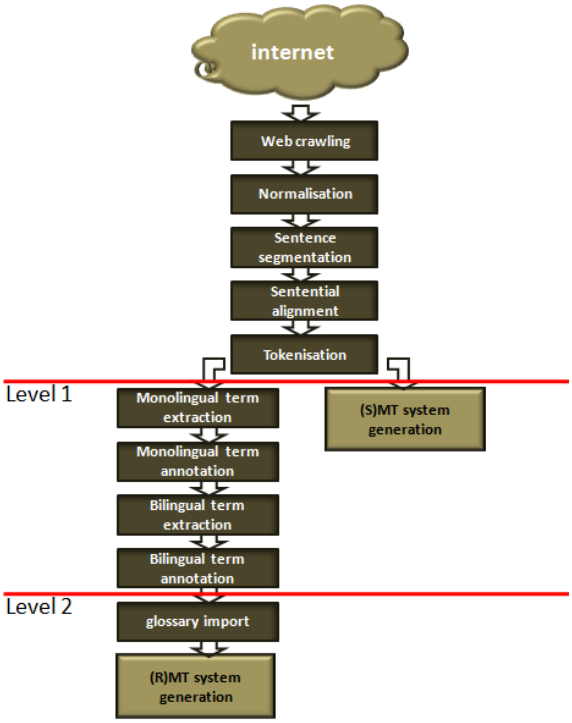


Fig. 6-1: Evaluation levels for tool evaluation

The dictionary will be evaluated according to the following sets of criteria:

- Formal criteria / Validation: Wellformedness of the produced entries, presence of obligatory annotations, size
- Quality criteria: Correctness of proposed translations, using a test sample
- Annotation criteria: Proper annotation of the entries, also using a test sample

Errors will be collected, and traced to one of the four components which have built the dictionary.

6.4 Evaluation Setup

The tool-based evaluation will consist of the following steps:

6.4.1 Corpus collection

Two bilingual corpora will be collected, using the parallel web crawler. They should be in a special domain, like software manuals. The size of the corpus should be such that it can support the requirements of the different extraction tools, in order to allow them to show their full capacity.

The languages will be determined depending on the progress of the different PANACEA tools, but two different language pairs will be involved.

[21] only ‚clear' errors are relevant here, unclear cases are less important in a final evaluation round.

6.4.2 Sentence level evaluation

a. The corpus will be processed by the PANACEA tools for normalisation, sentence segmentation, and tokenisation. For each of the languages, a monolingual test sample of 10 times 50 tokenised sentences will randomly be collected.

b. The monolingual parts of the aligned sentences will be manually checked for normalisation, tokenisation and sentence segmentation errors; the errors will be classified according to which component produced them, and an error rate will be computed.

The challenge here is to define what such errors are, e.g. what a tokenisation error is. This definition will be done in cooperation with the tool evaluation in WP7.

c. The monolingual corpora will undergo sentence-alignment. From the result, a random sample of 10 times 50 aligned segments will be collected, for both language pairs.

d. The alignment part is evaluated. The criterion is 'alignment precision' (Moore 2002, following Brown et al. 1991)[22], i.e. the number of correct 1-1 sentence alignments. Manual evaluation of the alignments of the test sentences will add a level of correctness to the overall error rate, for the alignment used.

6.4.3 Dictionary level evaluation

a. The aligned corpus will be used as data source to run the dictionary creation and annotation tools. From the result, a random sample of bilingual entries will be inspected, in the size of 500 to 1000 entries.

b. The entries will be evaluated according to the criteria explained above. The evaluation dimensions will be:

Validation / Formal criteria:

- Is the character code correct? Does the file only contain legal characters?
- Are all obligatory annotations available?
- Do the annotations contain legal values? (data type; correct values for member-typed annotations; correct spelling / lemma presentation for string-typed values)

These criteria can be evaluated by a validation program. There should not be any obvious errors in the data[23].

Monolingual annotation criteria:

- Are the monolingual entries properly annotated? E.g. are all nouns annotated with '*feminine*' really feminine? Are the inflectional patterns correctly assigned? Are the parts of speech right? etc.

Annotation errors are: wrong values, incomplete or missing values.

[22] the Alignment Error Rate (AER) is defined on *word* level, not on *sentence* level, cf. e.g. Frazer/Marcu 2007.
[23] There can be unclear cases e.g. in spelling; they should not count as error.

Translation quality criteria:

- Are the proposed translations correct?

This will be evaluated by searching entries in other dictionaries. Errors mean that the claimed translation cannot be found anywhere.

Bilingual annotation criteria:

- Are the bilingual entries properly annotated? E.g. Do entries with more than one translation have transfer selection conditions? Do all entries have a default translation? Is the alignment information correct in cases of multiword entries?

Annotation errors are: wrong values, incomplete or missing values.

The evaluation will be done by a manual check of the translations and annotations provided, divided into the evaluation for monolingual and for bilingual entries and their annotations in the defined languages.

An error rate of less than 5% for annotations is usually considered to be acceptable.

c. An error analysis will be performed, to clarify why which component produced an incorrect entry or assignment.

6.4.4 Collection of results, evaluation report

The results of the different activities in the tool-based evaluation will be collected and presented in an evaluation report, which will form the deliverable D8.2.

6.5 Evaluation result

The evaluation activity should answer the following questions:

A Are the PANACEA tools available for building task-oriented workflows?

This refers to web service integration, software quality, documentation, availability etc.

B Can such workflows be built for several languages?

This refers to language coverage.

C1. Is the tool quality acceptable, on sentence level evaluation?

There will be some intrinsic evaluation but ‘acceptable’ really refers to the translation quality of the systems which use the sentence level results as input.

C2. Is the tool quality acceptable, on dictionary level evaluation?

Again there should be an intrinsic quality of such dictionaries (error rates of above 5-7% are usually not acceptable), and again there is an extrinsic criterion, namely if an MT system can

use the dictionaries and produce superior MT output quality. This, however, cannot be answered in a *tool*-based approach but will be evaluated in the *task*-based evaluation.

6.6 Acceptance criteria

The evaluation and acceptance criteria for the single tools will be defined in WP 7.

There is an issue in the tool-based evaluation, however, related to the *combination* of tools into whole workflows. There are two extremes:

- the errors of the single tools accumulate, with the result that after three or four steps the output of the workflow becomes unusable
- the errors even out, and later tools are robust enough to cope with non-perfect input.

These questions would be investigated in the tool-based evaluation procedure. In total, the fuzzy criterion of 'usability for an intended task' would have to be applied, which leads to an extrinsic evaluation strategy on tool level.

For human inspection, experience shows that error rates of 3-5% would be acceptable, both on the level of aligned sentences, and on the dictionary level. Higher error rates require special justification.

7 Task-based Evaluation of PANACEA

WP 8 of PANACEA, called 'Industrial Evaluation', aims at the evaluation of the usability of the PANACEA platform for an industrial development. The use case which will be evaluated is Machine Translation, i.e. the development of resources for MT systems.

Two systems will be considered:

- adaptation of the MaTrEx system (statistical MT, provided by DCU)
- adaptation of the Personal Translator system (rule-based MT, provided by Linguatec)

Languages will be German – English (both directions), adaptation domain will be automotive.

7.1 Evaluation target

There are two evaluation criteria:

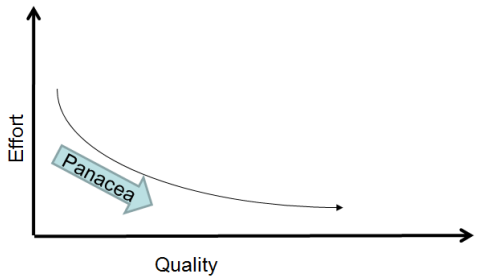
- **Productivity**: Can language resources be built with less effort than conventional techniques by using the PANACEA factory?

  This question relates mainly to the rule-based MT system creation which required significant manual effort to create the dictionary resources. To answer it, a comparison of the effort required to adapt a system to a new domain between a *non*-PANACEA-based and a PANACEA-based development task will be performed.

- **Quality**: Can language resources be built which lead to improved quality as compared to a baseline system?

  The objective of the evaluation is to determine if PANACEA tools can lead to an improved quality with more efficient and less costly production. So, evaluation is planned focusing on *one* system at a time. Neither a comparison of system quality of different systems, nor an evaluation of the 'absolute' translation quality is aimed at. The objective of the evaluation is to compare the quality of an *untuned* with the one of a *tuned* system.

These criteria contribute to the **effort / quality** relationship. The combination of the two criteria just mentioned will enable the project to answer the 'industrial' question of investment and return-of-investment (measured in terms of quality[24]), i.e. 'How much quality gain can we get with which cost, using PANACEA tools?'



[24] which is not completely correct, as improved quality does not automatically mean improved return of investment.

So the interest is to prove that PANACEA tools enable industrial users to have better quality systems with less effort.

7.2 Evaluation object

7.2.1 Workflow

There are two evaluation objects, which are interrelated:

- The first evaluation object is a **workflow**, namely the production of language resources for an MT system. This workflow is divided into two sections following the two main approaches towards MT system creation, namely data-driven (SMT) and knowledge-driven (RMT) approaches. Within the RMT approach, again two workflows are evaluated: conventional techniques, and PANACEA-based techniques.

  As a result, three workflows need to be evaluated. This is shown in fig. 7-1.

- The second evaluation object is a comparison of the **translation quality** produced by the respective workflows.

  This will imply, for each workflow, a comparison between the baseline system (*before* adaptation) and the adapted system.

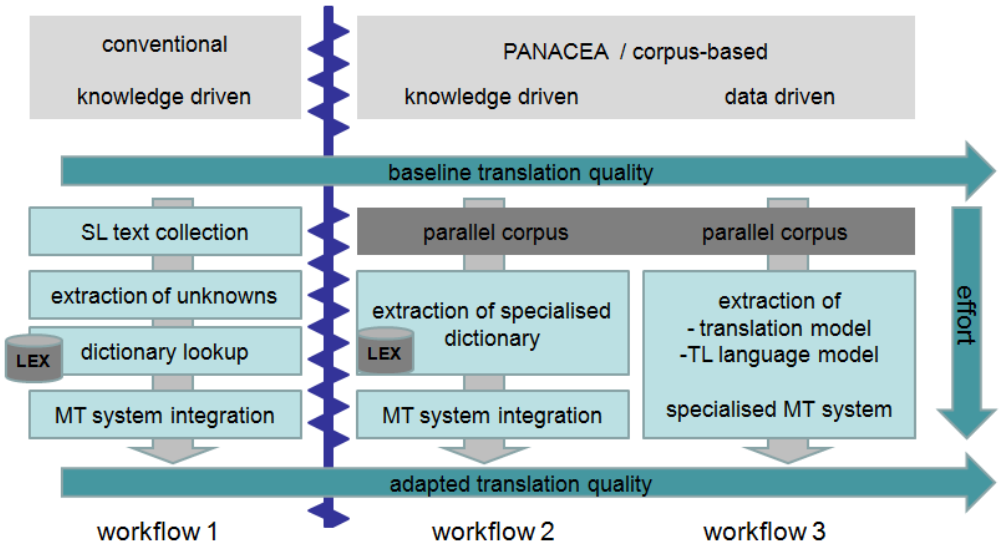


Fig. 7-1: MT evaluation – overview

Based on the outcome of these two evaluations, a ratio of effort and quality gain can be determined.

7.2.2 Test systems

In order to evaluate the workflows, two test systems will be used into which the PANACEA evaluation will be embedded:

- We use as an SMT system: MaTrEx (DCU)
- We use as an RMT system: Personal Translator 14 (LT)

Baseline systems will be produced, from which the evaluation starts; these systems will then be adapted to the automotive domain, and effort and quality will be evaluated.

7.3 Evaluation criteria

The evaluation will imply two sets of criteria.

7.3.1 Productivity criteria

For productivity comparison, the relevant criterion is the **person-hours** needed. So all activities during the execution of the workflow will be measured in terms of person-hours.

It could be imagined that *several* persons should be involved in the evaluation, in order to form an average of productivity. However, as the evaluation is only *relative* (i.e. non-PANACEA workflow compared to PANACEA workflow), the absolute ration of productivity is less important, so the number of testers does also not really matter[25].

What is intended is to have an *indication* of the *relative* productivity of both approaches.

7.3.2 MT quality criteria

Finding criteria for MT evaluation is a challenge in itself. The following section gives just a sketch, and proposes conclusions for the PANACEA task at hand.

7.3.2.1 Automatic evaluation measures

a. The first automatic measures were n-gram based, with WER, BLEU and NIST as the most important representatives. They calculate the distance of some MT output to a (set of) reference translations, and they claimed to mirror human intuition on translation quality: "The BLEU score correlates highly with human judgements" (Papineni et al. 2002).

There has been a long debate since then, and there is consensus among researchers in the evaluation field about the following issues:

- They depend on the reference translations (Popescu-Belis 2008), and tend to favour low-quality human translations (Culy 2003[26])
- These scores to not correlate to human intuition about translation quality (Callison-Burch et al. 2009, Zhao et al. 2009, Hamon et al. 2006).
- They are sensitive for MT system architecture, and penalize rule-based systems as such systems produce (often acceptable) variance in lexical selection and constituent ordering
- They are less discriminative in areas of very low and very high quality (Babych/Hartley 2008)

[25] There are other factors which influence productivity, like availability of data etc.; so there is no point in too sophisticated settings here.

[26] "The only professional translator got worse scores than the translations of all seven non-professionals … This is because the non-professional translations tended to be fairly literal and stayed as close to the source text as possible."

- They are not suitable for error analysis

b. There were many proposals to improve these metrics, by taking additional information into account.

Scores like METEOR (Banerjee/Lavie 2005) try to improve over the pure n-gram based methods, e.g. by allowing synonyms based on WordNet. Use of entailment information has been proposed by Padó et al. 2009. Weighting the n-grams according to information load has led to metrics like d-score and s-score (Babych/Hartley 2004), adaptations for the treatment of agglutinative languages like Turkish have been proposed (Tantug et al. 2008), and more syntactically oriented measures have been developed recently in (Giménez/Màrquez 2008, 2009, Owczarzak et al. 2007) which was found to be closest to human intuition in the MT workshop 2009 (Callison-Burch et al. 2009).

Also, in the Chinese MT evaluation campaign, a method of 'linguistic checkpoints' is added to the standard (BLEU-based) metrics (Zhao et al. 2009, Zhou et al. 2008), to reach a more objective evaluation result[27].

Such elaborate metrics tend to be closer to the human intuition, but they have the disadvantage that they are language-dependent, and they have internal errors (like parse failures) which reduce the objectivity of the measure.

c. There have been proposals to base the evaluation on semantic criteria, like semantic role mapping (Lo/Wu 2010), which could be learned from a semantically tagged corpus, word similarity (Wong 2010, Apidianaki 2008), or contrastive lexical evaluation (Max et al., 2010). Approaches which automatically measure the (semantic) adequacy of a translation would be a clear improvement as carrying meaning is the core of translation.

d. The conclusion for PANACEA is that automatic scores should not be the *only* means to evaluate the systems, for two reasons:

- It is known that the scores obtained by automatic measures do not always correlate with real good/bad translations, and thus their reliability is limited. However they have shown to be usable in the evaluation of development progress for a given system.
- In industrial practice, they don't play a role, as usually reference translations are not available.

However, to have an indication of quality, automatic measures will be computed, like BLEU, TER, METEOR, or DCU-dependency[28]

7.3.2.2 Human Evaluation methods

Human evaluation has always been a method for measuring MT system performance, starting from things as simple as eye-tracking (Doherty/O'Brien 2009). The basic problem for human evaluation is cost (in time and effort, Przybocki et al. 2010), and the problem of subjectivity: Inter-rater agreement is a special aspect to be considered here (cf. Hamon 2010)

[27] They state „that the higher BLEU score doesn't always mean higher translation adequacy".
[28] Owczarzak et al. 2007. It could also be made usable for German texts.

a. Many human evaluation efforts follow the basic translation criteria of adequacy and fluency as stated in the FEMTI framework (King et al. 2003, Estrella et al. 2008), following multi-point scales between three and seven points ('very good' -> 'very bad'). The challenge is to improve objectivity, which seems to be less difficult for adequacy than for fluency (cf. the design of the TAP-ET tool, Przybocki et al. 2010). An alternative is just to measure preference of one MT output compared to (one or several) other ones, which seems to be easier than to judge an MT output for adequacy / fluency (Callison-Burch et al. 2009).

b. In a shift from technology-oriented evaluation to task-based evaluation (Popescu-Belis 2008, Babych/Hartley 2008), the evaluation strategy was redefined. While tasks like identifying named entities (Voss/Tate 2006, Babych/Hartley 2008) focus only on specific aspects of translation, the work of (White 2000, Reeder/White 2003) tries to identify a correspondence between levels of text understanding and required levels of translation quality, such that lower levels (gisting etc.) can be achieved with less sophisticated MT tools.

c. However, the most natural task for a task-based evaluation is to produce proper translations from a MT translation, i.e. post-editing.

Two methods of post-editing are in use:

- **HTER** (Snover et al. 2006, 2009) (human-mediated translation edit rate) is a distance measure which creates the reference translation as the task of human post-editing. Like TER it computes insertions, deletions, substitutions and shifts, on a word-basis; however the reference translation does not pre-exist but the post-editor is told to produce a translation as close to the MT output as possible.
    - In practical contexts, however, HTER has two drawbacks, both of which are due to the requirement to produce output as similar to the MT system as possible:
    - It hampers the productivity of human post-editors, as they spend a part of their time to calculate the effect of their post-editing on the closeness to MT
    - It reduces the overall MT output quality as a close-to-MT translation is usually lower in quality than a freer and more fluent translation (Culy 2003)

    However, as there is human intervention in the PANACEA evaluation, the result can be used to create (H)TER scores, to have an indication of the differences in output quality.

- **Postediting time**. Postediting time is the most fundamental performance measure in machine translation, as one of its original goals was to increase the overall translation productivity: The claim was that with MT, more text could be translated in the same, or even shorter, time. In localisation industry, productivity is still the most prominent criterion of (economic) success.

  Postediting has attracted new interest recently. Plitt/Masselot 2010 showed how MT can improve postediting speed[29]; correlation between automatic scores and postediting was researched by Tatsumi 2009; integration of MT post-editing into a translation workflow was described by Groves/Schmidtke 2009, and He et al. 2010.

[29] They found in tests with Autodesk that postedited MT increases productivity by 20 – 131%, and even increases translation quality.

Being an industrial type evaluation, PANACEA will use post-editing time as task-based evaluation criterion for the MT output. This time could be related to a purely human translation of the test data which could be used as a baseline, and as a first reference translation (if resources of this task permit it).

d. The conclusion for PANACEA is to use post-editing time as the main evaluation criterion. As with this method, well-formed translations are created, these translations can be used as references, to compute (H)TER and BLEU scores[30]. They will be biased towards MT translation; however as both RMT and SMT output will be postedited this may not be too severe a handicap.

Again, as PANACEA WP 8 does not intend to perform an evaluation campaign but just wants to compare baseline translations with adapted translations of the same system, the problem of inter-rated agreement is less severe than in other contexts. The project will use two professional translators, experienced also in MT, to do the post-editing[31].

7.4 Evaluation setup

The evaluation will be organised in different phases, which can be described as follows:

7.4.1.1 Preparation Phase

In this phase, the test systems will be prepared for a baseline evaluation:

Test systems: The following systems will be prepared using 'out-of-the-box' technology

- Baseline DCU-MT system: general de-en, general en-de.
- Baseline LT-MT system: general en-de, general de-en.

Test data: Quality tests will be performed by collecting the following test suite:

- Texts of general domain (different domains, like news, economy, sports, business, software etc.). They should cover the whole spectrum of what usually is translated by MT. This set should comprise about 2000 sentences, for each German and English.
- Texts of automotive domain, different genres (product descriptions, parts information, maintenance texts etc.). This set should comprise about 1000 sentences, for each German and English.

These test data will be translated by both systems in both language directions, resulting in 4 sets of translations. The result will be about 6000 test sentences overall.

[30] On the correlation of different automatic scores and post-editing speed, cf. Tatsumi 2009

[31] Of course post-editing speed can be increased by software tools (good editors, good dictionary lookup tools etc.); however this will not be researched in the present context.

Reference translations:

Two reference translations will be produced during evaluation, as result of the post-editing effort.

Evaluation

Evaluation will be done in three ways:

- Postediting time. The MT outputs will be postedited by human translators, and the time required will be measured for both participating systems. The postedited texts will be used as a reference translation in the evaluation phase.
- Postediting time will be measured for general domain texts and automotive texts separately.
- Using automatic measure (BLEU, TER), based on these reference translations (possibly including a non-MT reference translation)
- (Optional) manual evaluation, in two lines:
    - Rate the MT outputs along a three-point scale ('good – understandable – bad)'
    - Compare the two MT outputs (better – equal) (this task is optional)

The evaluation results will be used as a basis for comparison with the tuned systems.

7.4.1.2 Adaptation phase

In the adaptation phase, the systems will be tuned for the automotive domain. For the SMT system this means to build a new system with extended resources, adding automotive texts to the text base. For the RMT systems, two adaptation strategies will be followed:

- A 'non-PANACEA-RMT' strategy, using conventional adaptation means (mainly dictionary)
- A 'PANACEA-RMT' strategy, using PANACEA tools

The result of the adaptation phase therefore will consist in *three* systems (cf. fig. 7-1 above). The adaptation effort for each of these systems will be measured, for later comparison.

A. For **PANACEA-SMT** (DCU) the tasks are:

This workflow creates an SMT system using the PANACEA factory. The following functionality of the factory is required:

- Collect a parallel corpus of the automotive domain for the translation model, by using web crawling for parallel corpora, and alignment on sentence level, and tokenisation
- Collect a monolingual corpus of the automotive domain for the target language model, by using web crawling
- Run the TM and LM creation tools

- Integrate the new resources into the system, and adapt the decoder

As with the other workflows, efforts will be logged.

B. For **PANACEA-RMT** (LT)

This workflow creates an RMT system using the PANACEA factory. The following functionality of the factory is required:

- Collect a parallel corpus of the automotive domain, by using web crawling for parallel corpora, and alignment on sentence level
- Extract bilingual terms from the parallel corpus, using term extraction
- Annotate the terms with MT relevant information, using PANACEA tools for corpus and term annotation
- Integrate the resulting dictionary resource as a user dictionary into the MT system for the test runs.

Again, efforts will be logged to allow for a productivity comparison.

C. For **non-PANACEA-RMT**: (LT)

A reference system will be built, based on conventional technology (cf. workflow 1 in fig. 7-1). It consists of the following steps:

- Collect a (monolingual) corpus of source language texts (for de, and for en). This corpus need not be parallel or even comparable.
- Run an 'unknown word search' with the Personal Translator, to identify unknown words
- Code each unknown word, doing a lookup in one of the available technical dictionaries, in the form of a special-domain user dictionary
- Integrate this user dictionary for the test runs.

As the comparison of efforts is one of the relevant evaluation criteria, logging of the efforts will be performed.

The result of the adaptation phase should be three systems, each tuned to the automotive domain; for each system the tuning effort (in person hours) will have been measured.

7.4.1.3 Evaluation phase

In the evaluation phase, the gains in quality will be measured, and related to the efforts needed to achieve them.

Test systems:

- Tuned DCU-SMT system, De>En, tuned DCU-SMT system, En>De
- Tuned non-PANACEA RMT system De>En, tuned non-PANACEA-RMT system En>De
- Tuned PANACEA RMT system De>En, tuned PANACEA-RMT system En>De

Test data:

The same test data as for the baseline evaluation will be used (they will not be part of any development effort).

Reference translations:

The translations produced for the baseline evaluation will be used as reference also for the adapted systems.

Evaluation:

Evaluation will be done according to

- Quality evaluation: comparing the quality achievement of each system with the baseline system
- Effort evaluation: relating the quality improvement of each system needed to the effort needed to achieve it.

There will be five sets of data available for evaluation for each translation direction, containing both domain-sentences and out-of-domain sentences:

- baseline SMT
- baseline RMT
- adapted SMT (workflow 3: SMT with PANACEA tools)
- adapted RMT2 (workflow 2: RMT with PANACEA tools)
- adapted RMT1 (workflow 1: RMT non-PANACEA, conventional way)

7.5 Evaluation result

Evaluation should answer the following questions:

A. To which extent has the quality improved for the adaptation domain?

For this purpose, the difference between the baseline automotive texts and the adapted automotive texts will be evaluated:

- by creating automatic scores (BLEU, TER) for the baseline translations and for the adapted translations, for each of the three test configurations
- by manual evaluation of the translation differences according to a three-point scale (better – similar – worse) for each of the three test configurations

This will also allow us to compare the quality of the PANACEA- vs. non-PANACEA-RMT.

B. To which extent has the original translation quality deteriorated?

Again, the difference between the baseline general texts before and after the adaptation will be evaluated:

- by creating automatic scores (BLEU, TER) for the baseline translations before and after the adaptation, for each of the three test configurations
- by manual evaluation of the translation differences according to a three-point scale (better – similar – worse) for each of the three test configurations

The idea is to identify side-effects, or overfitting effects of the domain tuning.

C. What is the effort to create the adapted versions?

This question is intended to help to decide if tuning towards a narrow domain can be done with reasonable effort. If the tuning effort is too high then it is commercially not viable.

The effort will be collected by evaluating the person-hours for the three workflows.

D. How much does PANACEA increase the productivity, i.e. reduce the development effort?

This question should evaluate the productivity improvement to be achieved by the PANACEA toolbox. It compares the efforts in Workflow 1 (conventional, non-PANACEA) to the efforts in Workflows 2 and 3 (PANACEA).

The hypothesis is that PANACEA significantly reduces the development efforts.

E. What is the postediting effort for the adapted texts?

The sentences of the automotive domain will be postedited, and compared to the postediting effort (for the automotive texts) of the baseline system. This will be done for each of the three test configurations.

The hypothesis is that postediting effort will be lower than for the baseline automotive texts.

F. What is the relation of effort and quality improvement?

We relate adaptation effort and quality gain for the three systems, to be able to compare the ratios of the three configurations.

This question relates quality improvement and effort, and indicates the productivity gain which can be achieved. It should answer the question with which effort how much quality can be achieved.

As effort and quality are incommensurable (except in their monetary form), only a verbatim comparison will be given, and a tentative statement ('is it worth the effort?') can be expected.

(*G. Optional: What is the 'best' output?*

This question compares the outputs of the different systems (workflow 1, workflow 2, workflow 3).

Comparison is done on a sentence-basis. The three outputs are offered to human evaluators for ranking (from <best,second,third> to <all_the_same>).

However this is not in the focus of PANACEA industrial evaluation.)

7.6 Acceptance criteria

The work package is successful if it can be shown that PANACEA tools can produce better quality translation than the baseline with less effort than a conventional system.

8 Tasks and work plan

The evaluation task in work package 8, as described above, requires a series of interdependent tasks, which also are interlinked with the other tasks in PANACEA. They are described as follows:

8.1 Task list

The work package needs the following tasks to be executed:

8.1.1 Tool-based Evaluation

8.1.1.1 Preparation Phase: Corpus data collection

Preparation consists in collecting the corpora. We take two language pairs (e.g. en<>de and en<>es), and a specific domain (e.g. software; yet to be decided, depending on the availability of data).

1. Collect parallel corpora, specific domain, de and en
2. Collect parallel corpora, specific domain, en and es

8.1.1.2 Sentence level evaluation

3. Create monolingual sentence level evaluation object en, by running normalisation, tokenisation, sentence segmentation
4. Create monolingual sentence level evaluation object de, by running normalisation, tokenisation, sentence segmentation
5. Create monolingual sentence level evaluation object es, by running normalisation, tokenisation, sentence segmentation
6. Select 500 English output sentences
7. Select 500 German output sentences
8. Select 500 Spanish output sentences
9. Evaluate accuracy for English test set
10. Evaluate accuracy for German test set
11. Evaluate accuracy for Spanish test set
12. Create bilingual sentence level evaluation object en-de, by aligning the two mono sides
13. Create bilingual sentence level evaluation object en-es, by aligning the two mono sides
14. Extract test set en-de (500 segments)
15. Extract test set en-es (500 segments)
16. Evaluate accuracy for English-German test set
17. Evaluate accuracy for English-Spanish test set

8.1.1.3 Dictionary level evaluation

18. Create a tool for validation checking
19. Create annotated monolingual dictionary English, by running monolingual term extraction and annotation tools
20. Create annotated monolingual dictionary German, by running monolingual term extraction and annotation tools
21. Create annotated monolingual dictionary Spanish, by running monolingual term extraction and annotation tools
22. Select 500-1000 random English mono entries

23. Select 500-1000 random German mono entries
24. Select 500-1000 random Spanish mono entries
25. Validate English mono entries
26. Validate German mono entries
27. Validate Spanish mono entries
28. Evaluate annotation accuracy for English mono entries
29. Evaluate annotation accuracy for German mono entries
30. Evaluate annotation accuracy for Spanish mono entries
31. Create annotated bilingual dictionary English-German, by running bilingual term extraction and annotation tools
32. Create annotated bilingual dictionary English-Spanish, by running bilingual term extraction and annotation tools
33. Select 500-1000 random German-English mono entries
34. Select 500-1000 random English-Spanish mono entries
35. Validate German-English mono entries
36. Validate English-Spanish mono entries
37. Evaluate annotation accuracy for German-English mono entries
38. Evaluate annotation accuracy for English-Spanish mono entries

8.1.1.4 Reporting

39. Create report and deliverable D8.2

8.1.2 Task-based Evaluation

8.1.2.1 Preparation Phase

This phase can be done independent of the other PANACEA developments, and start as soon as the reference systems are available.

System preparation

1. Collect parallel corpora, general domain
2. Align and tokenise the general domain data
3. Create baseline MaTrEx system
4. Create baseline PT system

Test data collection

5. Collect 2000 sentences general domain de > en
6. Collect 2000 sentences general domain en > de
7. Collect 1000 sentences automotive de > en
8. Collect 1000 sentences automotive en > de

Reference creation

9. Translate the test collection with MaTrEx
10. Translate the test collection with Personal Translator
11. Postedit 2000 sentences general domain de > en, MaTrEx output, measure effort

12. Postedit 2000 sentences general domain en > de, MaTrEx output, measure effort
13. Postedit 1000 sentences automotive de > en, MaTrEx output, measure effort
14. Postedit 1000 sentences automotive en > de, MaTrEx output, measure effort
15. Postedit 2000 sentences general domain de > en, PT output, measure effort
16. Postedit 2000 sentences general domain en > de, PT output, measure effort
17. Postedit 1000 sentences automotive de > en, PT output, measure effort
18. Postedit 1000 sentences automotive en > de, PT output, measure effort
19. Compute automatic scores for all postedited translations of each postedited package

8.1.2.2 Adaptation Phase

The adaptation phase contains tasks for the three workflows as described in fig. 7-1. The main tasks are:

Adaptation MaTrEx (workflow 3)

20. Collect parallel corpus for the automotive domain, using PANACEA tools. Measure effort
21. Cleanup, align, and tokenise the corpus data. Measure effort.
22. Collect monolingual corpus data for LM creation, using PANACEA tools. Measure effort.
23. Cleanup and prepare the monolingual data for LM creation. Measure effort.
24. Train MaTrEx for the automotive domain, create a de > en system. Training will affect both translation model and language model data. Measure effort.
25. Train MaTrEx for the automotive domain, create an en > de system. Training will affect both translation model and language model data. Measure effort.

Adaptation Personal Translator with PANACEA tools (workflow 2)

This task can re-use the parallel corpus data collection and cleanup (tasks 20, 21) from workflow 3. Then the following tasks are required:

26. Run monolingual term extraction from automotive corpora, de > en
27. Run monolingual term extraction from automotive corpora, en > de
28. Run monolingual term annotation from automotive corpora, de
29. Run monolingual term annotation from automotive corpora, en
30. Run bilingual term extraction from automotive corpora, de > en
31. Run bilingual term extraction from automotive corpora, en > de
32. Run bilingual term annotation from automotive corpora, de > en
33. Run bilingual term annotation from automotive corpora, en > de
34. Import glossaries as special-domain additional dictionaries into Personal Translator

Adaptation Personal Translator without PANACEA tools (workflow 1)

This task will be independent of the PANACEA toolbox, and use conventional means to adapt an RMT system to a new domain. Tasks will be:

35. Collect a monolingual corpus with automotive texts, de
36. Collect a monolingual corpus with automotive texts, en
37. Run tool to extract unknown words from text, de > en
38. Run tool to extract unknown words from text, en > de

39. Code unknown words in the de > en system, using available dictionaries, into a special additional dictionary
40. Code unknown words in the en > de system, using available dictionaries, into a special additional dictionary

8.1.2.3 Evaluation Phase

The evaluation phase will have the following tasks to do:

Productivity evaluation

This includes the following tasks:

41. Compute effort for workflow 1 (non-PANACEA RMT)
42. Compute effort for workflow 2 (PANACEA RMT)
43. Compute effort for workflow 3 (PANACEA SMT)
44. Compare efforts for workflow 1 and 2. This will show the productivity increase. This answers the questions C and D.

Quality evaluation

This includes the following tasks:

45. Postedit 1000 new automotive de > en, MaTrEx output, measure effort
46. Postedit 1000 new automotive en > de, MaTrEx output, measure effort
47. Postedit 1000 new automotive en > de, PT non-PANACEA output, measure effort
48. Postedit 1000 new automotive de > en, PT non-PANACEA output, measure effort
49. Postedit 1000 new automotive en > de, PT PANACEA output, measure effort
50. Postedit 1000 new automotive de > en, PT PANACEA output, measure effort
51. Compute BLEU and TER for each postedited package
52. Compute effort for postediting for the three workflows, and compare to postediting for the automotive texts of the baseline systems. This should answer question E.
53. Compare adapted MaTrEx – baseline MaTrEx
54. Compare adapted PT (non-PANACEA) – baseline PT
55. Compare adapted PT (PANACEA) – baseline PT
56. Compute improvements for the adaptation quality. This answers question A.
57. Do a manual check of (a part of) the differences in the translations of the general domain texts for the three systems[32]. This should answer question B.

Reporting

58. Create an evaluation report containing all the evaluation results, forming the deliverable D8.3.

[32] Doing a full postediting for the general domain tasks is considered to be too much effort for this question. Instead, inspection of the diff files would be adequate.

8.2 Task dependencies and timelines

This section describes the dependencies of the tasks of WP 8 from the rest of the packages of PANACEA. Internal dependencies are rather obvious: A test set cannot be evaluated before it has been created etc.; but for time lines it is important to know which PANACEA tools must exist to execute certain evaluation tasks.

8.2.1 Tool-based evaluation

This task depends on the availability of the PANACEA tools; they are planned to be available in T30. The only subtask which can start earlier is the creation of the dictionary validation tool (task 18). This is shown in fig. 8-1.

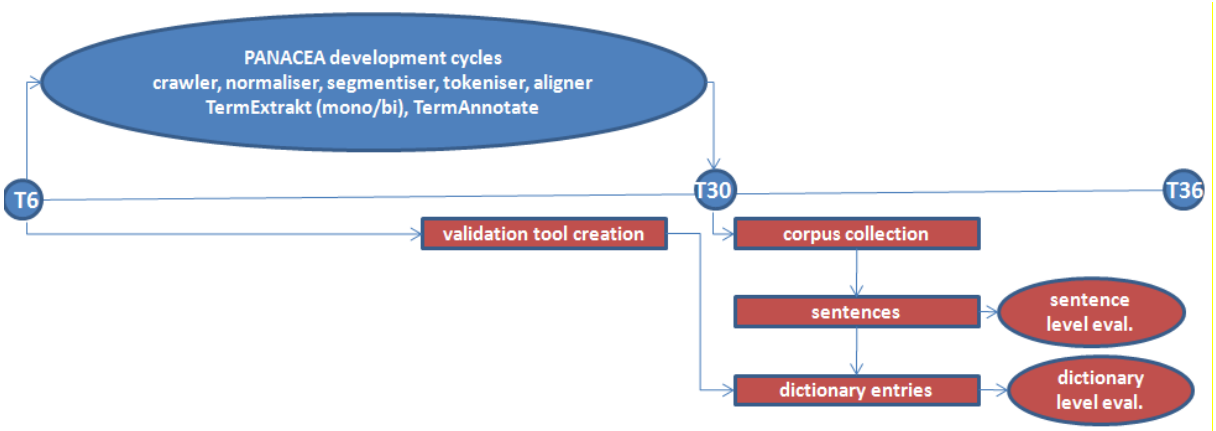


Fig. 8-1: Tool-oriented evaluation

Corpus collection precedes sentence level evaluation, which comes before dictionary evaluation.

8.2.2 Task-Based evaluation

This task contains some subtasks which are planned to be available before T30, or can even be carried out independently of the PANACEA tool development, cf. fig. 8-2.

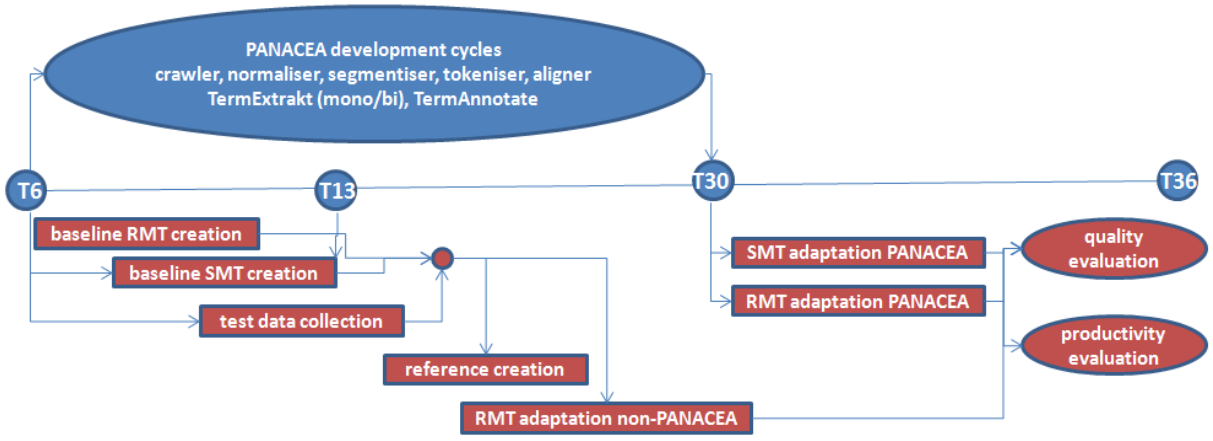


Fig. 8-2 Task-oriented evaluation

In particular,

- the setup of the baseline systems (for MaTrEx planned for T13)
- the collection of test data
- the baseline translation and reference creation
- the adaptation of the RMT without PANACEA tools

can be done independent of the PANACEA toolbox.

At T30, the adaptation tasks can start, using the PANACEA tools, followed by the evaluation tasks.

Deliverables for Tool-Based (D8.2) and Task-based (D8.3) evaluation are due in T36.

9 Citations

- Apidianaki, M., 2008: Translation-oriented Word Sense Induction Based on Parallel Corpora, Proc. LREC, Marrakech.
- Babych, B., Hartley, A. 2008: Sensitivity of Automated MT Evaluation Metrics on Higher Quality MT Output: BLEU vs Task-Based Evaluation Methods. Proc LREC Marrakech
- Babych, B., Hartley, A., 2003: Improving Machine Translation Quality with Automatic Named Entity Recognition. Proc. EACL/EAMT Budapest
- Babych, B., Hartley, A., 2004: Extending the BLEU MT Evaluation Method with Frequency Weightings. Proc. ACL 2004, Barcelona
- Banerjee, S., Lavie, A., 2005: METEOR: An Automatic Metric for MT Evaluaiton with Improved Correlation with Human Judgments. Proc. ACL Workshop on intrinsic and extrinsic evaluation measures for MT and/or Summarisation, Ann Arbor
- Berners-Lee, T, et al., 2005: "Uniform Resource Identifier (URI): Generic Syntax", IETF RFC 3986,January 2005, http://tools.ietf.org/rfc/rfc3986.txt
- Brown, A. and Haas, H., 2004: Web Services Glossary. *W3C working group note*. http://www.w3.org/TR/ws-gloss/
- Brown, P., Lai, J., Mercer, R., 1991: Aligning sentences in parallel corpora. Proc. 29th ACL, Berkeley
- Callison-Burch,Chr., Koehn, Ph., Monz, Ch., Schroeder, J., 2009: Findings of the 2009 Workshop on Statistical Machine Translation. Proc 4th Workshop on SMT, Athens
- Chan, Yee Seng, Ng, Hwee Tou, Chiang, D., 2007: Word Sense Disambiguation Improves Statistical Machine Translation. Proc. 45th ACL, Prague
- CiTER 2008: Citation of Electronic Resources, ISO Draft
- Culy, Chr., Riehemann, S.,2003: The Limits of N-Gram Translation Evaluation Metrics. Proc. MT Summit New Orleans
- Deksne, D., Skadiņš, R., Skadiņa, I., 2008: Dictionary of Multiword Expressions for Translation into Highly Inflected Languages. Proc. LREC Marrakech
- Doherty, St., O'Brien, Sh., 2009: Can MT Output Be Evaluated Through Eye Tracking? Proc. MT Summit XII, Ottawa
- Estrella, P., Popescu-Belis, A., King, M., 2008: Improving Contextual Quality Models for MT Evaluation Based on Evaluators' Feedback. Proc. LREC Marrakech
- Fraser, A., Marcu, D., 2007: Measuring Word Alignment Quality for Statistical Machine Translation. in: Computational Linguistics 33,3, p.293-303
- Giménez, J., Màrquez, L., 2008: Towards Heterogeneous Automatic MT Error Analysis. Proc. LREC Marrakech
- Giménez, J., Màrquez, L., 2009: On the Robustness of Linguistic Features for Automatic MT Evaluation. Proc. EAMT, 4th Workshop of Statistical MT
- Grégoire, N., 2009: Untangling Multiword Expressions, A study on the representation and variation of Dutch multiword expressions. Utrecht (LOT)
- Groves, D., Schmidtke, D., 2009: Identification and Analysis of Post-Edititing Patterns fro MT. Proc MT Summit XII, Ottawa
- Guenther,R., 2004: (Library of Congress), "PREMIS - Preservation Metadata Implementation Strategies Update 2: Core Elements for Metadata to Support Digital Preservation" RLG DigiNews: December 2004 http://www.rlg.org/en/page.php?Page_ID=20492#article2
- Hamon, O., 2010: Is my Judge a good One? Proc. LREC Malta

- Hamon, O., Popescu-Belis, A., Choukri, K., Dabbadie, M., Hartley, A., W. Mustafa El Hadi, W., Rajman, M., and Timimi, I., 2006. CESTA: First Conclusions of the Technolangue MT Evaluation Campaign. Proc. LREC Genova, Italy.
- He, Y., Ma, Y., van Genabith, J., Way, An., 2010: Bridging SMT and TM with Translation Recommendation. Proc. ACL Uppsala
- Hovy, E., King, M., Popescu-Belis, A., 2002: Principles of Context-Based Machine Translation Evaluation. in: Machine Translation, **16**, pp. 1-33
- King, M., Popescu-Belis, A., Hovy, E., 2003: FEMTI: creating and using a framework for MT evaluation. Proc MT Summit New Orleans
- Lo, Chi-kiu, Wu, Dekai, 2010: Evaluating Machine Translation Utility via Semantic Role Labels. Proc. LREC Malta
- MacKenzie C.M., Laskey K., McCabe F., Brown P.F., Metz R., Hamilton B.A., 2006: OASIS Reference Model for Service Oriented Architecture 1.0, August 2006, http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf
- Max, A., Crego, J.M., Yvon, F., 2010: Contrastive Lexical Evaluation of Machine Translation. Proc. LREC Malta
- Miháltz, M., 2005: Towards a hybrid approach to word sense disambiguation in Machine Translation. Proc. RANLP, Borovets
- Miller, K., Vanni, M., 2005: Inter-rater Agreement Measures and the Refinement of Metrics in the PLATO MT Evaluation Paradigm. Proc. MT Summit Phuket
- Moore, R.C., 2002: Fast and Accurate Sentence Alignment of Bilingual Corpora. Proc. AMTA
- Owczarzak, K., Graham, Y., van Genabith, J., 2007: Using F-structure in Machine Translation Evaluation. Proc. LFG07 Conf., (CSLI Publications)
- Owczarzak, K., van Genabith, J., Way, A., 2007: Dependency-Based Automatic Evaluation for Machine Translation. Proc. AMTA Workshop on Syntax and Structure in Statistical Translation, Rochester
- Padó, S., Galley, M., Jurafsky, D., Manning, Chr., 2009: Robust machine translation evaluation with entailment features. Proc. ACL Singapore
- Papineni, K., Roukos, S., Ward, T., Zhu, W., 2002: BLEU: a Method for Automatic Evaluation of Machine Translation. Proc. ACL, Philadelphia
- Parra, C., Villegas, M., Bel, N., 2010: The BAsic Metadata DEScription (BAMDES) and TheHarvestingDay.eu: Towards Sustainability and Visibility of LRT. Proc. LREC Malta, Workshop on Language Resources.
- Plitt, M., Masselot, Fr., 2010: A Productivity Test of Statistical Machine Translation Post-Editing in a Typical Localisation Context. in: The Prague Bulletin of Mathematical Linguistics 93, p.7-16
- Poibeau, Th., Messiant, C., 2008: Do we Still Need Gold Standards for Evaluation? Proc. LREC Marrakech.
- Popescu-Belis, A., 2008: Reference-based vs. task-based evaluation of human language technology. Proc. LREC
- Przybocki, M., Peterson, K., Bronsart, S., 2010: Translation Adequacy and Preference Evaluation Tool (TAP-ET). Proc. LREC Malta
- Reeder, Fl., White, J., 2003: Granularity in MT Evaluation. Proc. MT Summit IX, New Orleans
- Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J., 2006: A Study of Translation Edit Rate with Targeted Human Annotation. Proc. of AMTA-2006

- Snover, M., Madnani, N., Dorr, B., Schwartz, R., 2009. Fluency, Adequacy, or HTER? Exploring Different Human Judgments with a Tunable MT Metric. In Proc. of WMT09
- Specia, L., Das Graças Volpe Nunez, M., Castello Branco, R.G., Stevenson, M., 2006: Multilingual versus Monolingual WSD. Proc Workshop ‚Making Sense of Sense', Trento
- Stanica M., Wiberg T., Wierenga K., Winter S., Rauschenbach J., 2006: JRA5 Glossary of Terms - Second Edition- update of DJ5.1.1
- Tantug, A.C., Oflazer, K., El-Kahlout, I.D., 2008: BLEU+: a Tool for Fine-Grained BLEU Computation. Proc. LREC Marrakech
- Tatsumi, M., 2009: Correlation Between Automatic Evaluation Metric Scores, Post-Editing Speed, and Some Other Factors. Proc. ;MT Summit XII, Ottawa
- Thurmair, Gr., 2006: Using Corpus Information to Improve MT Quality. in: Proc. Workshop LR4Trans-III, LREC Genova
- Vickrey, D., Biewald, L., Teyssier, M., Koller, D., 2005: Word-Sense Disambiguation for Machine Translation. Proc. HLT/EMNLP, Vancouver
- Voss, C., Tate, C., 2006: Task-based Evaluation of Machine Translation (MT) Engines: Measuring How Well People Extract Who, When, Where-Type Elements in MT Output. Proc. LREC Genova
- White, J., 2000: Toward an Automated, Task-Based MT Evaluation Strategy. Proc. LREC 2000, Athens, Workshop on Evaluation of Machine Translation
- Wong. B., 2010: Semantic Evaluation of Machine Translation. Proc. LREC Malta
- Wulong T., 2001, http://searchcio.techtarget.com/sDefinition/0,,sid182_gci213384,00.html
- Zhao, H., Xie, J., Liu, Q., Lü, Y., Zhang, D., Li, M., 2009: Introduction to China's CWMT2008 Machine Translation Evaluation. Proc. MT Summit XII, Ottawa
- Zhou, M., Wang, B., Liu, Sh., Li, M., Zhang, D. Zhao, T., 2008: Diagnostic evaluation of machine translation systems using automatically constructed linguistic check-points. Proc. COLING, Manchester