

# DELIVERABLE

**Project Acronym:** FLAVIUS

**Grant Agreement number:** ICT-PSP-250528

**Project Title:** Foreign Language Versions of Internet and User generated Sites

---

## D4.5 SMT customization module through dictionary

**Revision:** 2.0

---

**Authors:**

Daniel Marcu (Language Weaver SRL)

Bogdan Giurgiu (Language Weaver SRL)

Madalina Georgescu (Language Weaver SRL)

---

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

## Revision History

Revision	Date	Author	Organisation	Description
1.0	March, 21st	Bogdan Giurgiu, Daniel Marcu	Language Weaver SRL	Version 1.0
1.4	August, 25 <sup>th</sup>	Bogdan Giurgiu	Language Weaver SRL	Document updated to include restuls of the Component tests
1.5	August 29 <sup>th</sup>	Bogdan Giurgiu	Language Weaver SRL	Document updated to include 2 case studies
1.6	September 12 <sup>th</sup>	Madalina Georgescu	Language Weaver SRL	Document updated to include 3 experiments
2.0	September 14 <sup>th</sup>	Bogdan Giurgiu	Language Weaver SRL	Version 2.0

### Statement of originality:

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

## Table of Contents

1. Objective & Scope .....	5
2. Requirements .....	5
3. SDL BeGlobal API Guide.....	6
3.1 Introduction.....	6
3.2 Overview.....	6
4. SDL BeGlobal API Details .....	9
4.1 Authorization Details.....	9
4.2 Get Language Pairs .....	11
5. Term and Brand Management .....	13
5.1 Create Custom Term List .....	13
5.2 Add/Delete Term(s) to/from Custom Term-List.....	15
5.3 Upload File to Custom Term-List (Import CSV) .....	16
5.4 Get Custom Term-List (Get the details/content of a specific Term List).....	18
5.5 Get List of Custom Term-Lists for Language Pair (for account).....	20
5.6 Delete Custom Term-List.....	21
5.7 Upload File to Brand List .....	22
5.8 Get Brand List .....	23
5.9 Edit Brand-List .....	25
6. Test coverage of the Term & Brand Management API .....	27
6.1 Component test - Overview .....	27
6.2 Component test - Create Custom TermList.....	28
6.4 Component test - Upload File To Custom Term List .....	30
6.5 Component test - Delete Custom TermList.....	31
7. Dictionary Customization - Case Studies.....	33
7.1 Introduction.....	33
7.2 Experiment 1: Test on Training .....	33
7.2.1 Case Study – LW Customer.....	33
7.2.2 Case Study – Overblog.....	42
7.2.3 Final conclusions Experiment 1: Test on Training .....	50

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

7.3	Experiment 2: Test on New Blind Set of Data .....	52
7.3.1	Case Study: LW Customer .....	53
7.3.2	Case Study: Overblog.....	56
7.3.3	Final conclusions Experiment 2: Test on New Blind Set of Data .....	60
7.4	Experiment 3: Test on Blind Set .....	62
7.4.1	Action plan.....	63
8.	Achievements to date .....	67
9.	Appendix A – WP4 – Term and Brand Management Guide .....	69
10.	Appendix B – WP4 – Customization through retraining .....	70

<b>Project co-funded by the European Commission within the ICT Policy Support Programme</b>	
<b>Dissemination Level</b>	
<b>C</b>	<b>Confidential, only for members of the consortium and the Commission Services</b>

# 1. Objective & Scope

The objective of this milestone is to deliver training capabilities of the Statistical Machine Translation engines with the help of dictionaries.

The Dictionary infrastructure has been developed to serve the second objective of the WP4, **Create interfaces to customize translation engines.**

# 2. Requirements

The original requirements have been detailed in the third task from WP4 listed below:

### Task 4.3: Customize engines through dictionaries (D4.5, D4.7)

In this task we will develop a dedicated module for users willing to customize the translation of their website for specific words or phrases. The module will enable to create a personal dictionary, add terms and phrases to it, and to connect it to an SMT engine.

The infrastructure will be accessed by the FLAVIUS platform with the use of an Application Programming Interface (API) developed by Language Weaver SRL as part of the same task.

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

## 3. SDL BeGlobal API Guide

### 3.1 Introduction

To meet the requirements presented in Section 2, one needs the following :

- Online access to an MT engine via secured API
  - BeGlobal API is a REST based application programming interface to SDL Language Weaver’s Automated Translation systems accessible over the Internet and utilizing existing technologies and protocols such as HTTP, SSL, and XML. API functions are accessed using a combination of URLs (resources) and HTTP requests types (actions)
  - Access to the MT engine via secured API has been provided to partners in June 2010 as part of the D4.1
- The ability to alter the behavior of the MT engine via a dictionary

To alter the behavior of the Machine Translation, several steps had to be considered :

- The ability to create a dictionary
- The ability to modify and manipulate the content of the dictionary
  - This also includes the upload of a terminology list
- The ability to create a brand list (the brands presented in the list will not be translated by the MT engine, eg. Apple, Language Weaver, World Trade Center, etc.)
- The ability to edit the content of the brand list
  - This includes the upload of a predefined brand list into the system

The above requirements and functionalities have been achieved with the creation of the Term and Brand Management API calls, presented in detail in Section 6 of this document.

### 3.2 Overview

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

To get started, all application developers will need three pieces of information from SDL Language Weaver: An API Key, a corresponding Account ID and an User ID. The API key is used to authenticate your access rights to the service.

FLAVIUS partners will be able to access the following language pairs: English to/and from French, Spanish, German, Swedish, Polish and Romanian.

The SDL BeGlobal API is hosted at <https://api.sdlbeglobal.com>. Any application using the API must have access to the Internet in order for the service to work.

Each API function is a combination of a specific URL (resource) and an HTTP request type (e.g. GET, POST, DELETE). GET requests are equivalent to query operations. POST requests are equivalent to creation operations and DELETE requests are equivalent to destroy/delete operations.

The following document covers the following requests you may perform using the SDL BeGlobal API

#### Term and Brand Management:

- Create Custom Term List
- Add/Delete Term(s) to/from Custom Term List
- Upload File to Custom Term List (Import CSV)
- Get Custom Term List (Get the details/content of a specific Term List)
- Get List of Custom Term Lists for Language Pair (for account)
- Delete Custom Term List
- Upload File to Brand List
- Get Brand List
- Edit Brand List

The SDL BeGlobal API utilizes HTTP as the communication protocol and all data is encrypted using SSL (via HTTPS). Every HTTP request is authenticated and must contain a string with your Account ID – User ID (please note the User ID is not mandatory to all calls) is and a unique “Signature” in the HTTP header. The Signature is generated by the client application using the API Key. The SDL BeGlobal API will authenticate every request by analyzing the Account ID – User ID and confirming the application supplied signature is correct.

**NOTE** - The API key is never sent in any HTTP request. Please do not share your API key with anyone else. HTTPS protects data transmitted from the client to the server from eavesdropping and other various attacks. Additional information on authentication can be found later in this document.

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

A typical SDL BeGlobal API function is a combination of a URL and an HTTP request type (GET, POST, DELETE etc). Most translation URLs take the form of the following template:

**https://api.sdlbeglobal.com/v2/<resource>/<source-language>.<target-language>/<possible-optional-parameters>/**

where:

- <resource> is one of the following (for the Translation Calls):
  - translation
  - translation-async
  - quality-translation-async
  - lpinfo
  - user
  - file
- <optional-parameters> may contain optional parameters for an API function. Each optional parameter is delimited by a '/' and typically provided in a key=value format. Optional parameters will vary depending upon the API function call.

The HTTP request types can be one of HTTP GET, HTTP POST, or HTTP DELETE. SDL BeGlobal API functions will support either all or a subset of the above request types.

Every HTTP request to SDL BeGlobal must contain the following HTTP headers:

- "LW\_Date": Provides the current date and time. (e.g. "LW-Date: Thursday, March 6, 2008 2:30 PM GST")
- "Authorization": Contains the SDL BeGlobal authorization string used to authenticate the request. The Authorization value takes the form of "LWA:<account\_id>-<user\_id>:<signature>" or "LWA:<account\_id>:<signature>" (e.g. "Authorization: LWA:1234-1000:1abc234def=")

Every API function will return an HTTP response code as well as an XML document. The XML document will provide the response data for the API function call if the HTTP response code is an HTTP 200 OK. The XML document will provide the error data for the API function call if the HTTP response code is an HTTP 40x.

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services



## 4. SDL BeGlobal API Details

The following section provides in-depth details of the API.

### 4.1 Authorization Details

Every HTTP request sent to the SDL BeGlobal API is authenticated using a custom HTTP scheme. Two additional HTTP headers are added to every request:

- LW\_Date – Contains a date/time string based on the request time
- Authorization – Contains a string made up of three strings (each separated by a colon):  
“LWA:<account\_id>:<signature>” or “LWA:<account\_id>-<user\_id>:<signature>”

**Important Note:** There is a clear distinction between the account\_id (unique for a client) and the user\_id (a client can have several users with different roles)

**Important Note:** There are calls presented in this document where both <account\_id> and <user\_id> are required, and there are calls that require only the <account\_id>.

#### Creating the LW\_Date

LW\_Date header should contain a string that represents the current request date and time. E.g. “Thu Mar 1 2008 12:00:01 GST”

#### Creating the Authorization

The Authorization header consists of three string segments – all separated by a colon. The first segment is always “LWA”. The second segment is composed by the <account\_id> and <user\_id>. The third segment is a unique signature.

The unique signature is generated using a keyed-HMAC (Hash Message Authentication Code) and a SHA1 digest. For more information on HMAC-SHA1, please refer to RFC 2104. To create the signature, first create a “message” using three elements from the HTTP header. The three elements used are:

- The HTTP request type (one of GET, POST, or DELETE). The request type should be in upper case format.
- The current date/time in string format. This MUST match the string used in “LW\_Date” header.

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

- The URI requested (e.g. /v2/translation/eng.spa/). Remember that the URI requested should be the path to the resource on the server. It should not contain "<https://api.sdlbeglobal.com>" in the variable.

The message is the concatenation of these three elements with a newline in between each string:

Message = HTTP request type + "\n" + date (string used in LW\_Date) + "\n" + URI

Each element must **not** have any whitespace before or after the element (use strip() or trim() on each string – depending on the programming language you are using). Make sure the message is in UTF-8 encoding.

“Sign” the message or calculate the HMAC on the message using your assigned API Key. Then calculate the Base64 encoding of the HMAC result to obtain the “signature”.

The signature is <account\_id> or combined with the <account\_id>-<user\_id> (as noted in the API call) and the string “LWA” to create the value for the Authorization header. The value will take the form:

“LWA:<account\_id>:<signature>”

or

“LWA:<account\_id>-<user\_id>:<signature>”

Here is a pseudo-code example of what needs to take place with every HTTP request:

```
String LWA_API_KEY = "1234abcd";
String LWA_Account_ID = "1234";
String LWA_User_ID = "1000"; //this is required only for calls where user_id is mandatory
// Construct the message to be signed
String szHttpRequestType = "GET"; // Make sure to change this if you are performing POST, DELETE
String szDate = GetCurrentDate();
String szURI = '/v2/lpinfo/eng.spa/';
String szMessage = szHttpRequestType.trim() + "\n" + szDate.trim() + "\n" + szURI.trim();
// Sign the signature and convert into Base64
String szSignature = Base64.encode(HMAC-SHA1(LWA_API_KEY, szMessage));

// Assume HTTP request object in request
// Add two custom HTTP request headers
Request.addheader("LW-Date", szDate);
Request.addheader("Authorization", "LWA:" + LWA_Account_ID + "-" + LWA_User_ID + ":" + szSignature);
// if the user_id is not mandatory than the above line from the header will look like this
// Request.addheader("Authorization", "LWA:" + LWA_Account_ID + ":" + szSignature);
```

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

Every HTTP request made to the SDL BeGlobal API must contain the above authorization steps. It's important to always use the current Date/Time on every request to ensure that the signature is unique from request to request.

## 4.2 Get Language Pairs

### URL Resource:

<https://api.sdlbeglobal.com/v2/user/>

### Description:

This resource returns a list of all the approved language pairs for an authenticated account\_id/API Key.

This resource supports only HTTP GET request types.

### Mandatory Parameters Description:

- /v2– specifies the version of the API
- /user – specifies the account\_id

### Optional Parameters Description:

- None

### Output Response:

Successful requests will receive an HTTP 200 response code as well as an XML document. Failed HTTP requests will result in an HTTP 400 response code along with an error message in an XML document.

For successful HTTP GETs, you will receive an XML document similar to the following:

```
<lwresponse>
  <service_version>v2</service_version>
  <requested_url>/v2/user/</requested_url>
  <request_type>GET</request_type>
  <request_time>Tue March 27 16:54:45 2008</request_time>
  <source_language></source_language>
  <target_language></target_language>
  <response_data type='user_get'>
    <user_id>1234</user_id> // This id represents the account_id
    <approved_language_pairs>
      <language_pair>
        <lpid>1</lpid>
        <source_language>eng</source_language>
        <source_language_name>English</source_language_name>
        <target_language>spa</target_language>
        <target_language_name>Spanish</target_language_name>
        <domain>IT</domain>
        <version>4.3</version>
      </language_pair>
    </approved_language_pairs>
  </response_data>
</lwresponse>
```

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

```

        <language_pair>
            <lpid>5</lpid>
            <source_language>eng</source_language>
            <source_language_name>English</source_language_name>
            <target_language>spa</target_language>
            <target_language_name>Spanish</target_language_name>
            <domain>baseline</domain>
            <version>4.3</version>
        </language_pair>
    </approved_language_pairs>
</response_data>
</lwresponse>

```

If the API request fails, the server will return an HTTP 40x response code along with an XML document. The XML document will look as follows:

- HTTP 400 Bad Request – Bad Request
- HTTP 401 Unauthorized – Authorization Failure
- HTTP 404 Not Found – Resource Not Found

**NOTE:**

*A Failure response sample is presented in the next call “Language Pair Information Request”.*

**Pseudo-code Examples:**

Ex. 1: Get the list of approved language pairs for account\_id “1234” and API Key “1234abcd”

- Create an HTTP GET request to: <https://api.sdlbeglobal.com/v2/user/>
- Create a string by concatenating the request type (GET), Current Date, request URI (/v2/user/) with each string separated by a newline character (“\n”). Sign the message using the API key “1234abcd” and then Base64 encode the resulting output. This is your “signature.” Include the signature and the account\_id “1234” in the HTTP “Authorization” header (e.g. “Authorization: LWA:1234:<signature>”). Add a second HTTP header called “LW-Date” which has the Current Date used to create the signature as the value.
- Submit GET request
- Receive HTTP response. If HTTP 200 OK, parse XML document response for the list of approved language pairs.

# 5. Term and Brand Management

## 5.1 Create Custom Term List

**URL Resource:**

<https://api.sdlbglobal.com/v2/language-pair/<lpid>/term-list/>

**Description:**

This API request allows a user to create a new term-list for a language pair specified by an LPID.

This resource supports only HTTP POST request types

**Mandatory Parameters Description:**

- /v2– specifies the version of the API
- <lpid> – This is an integer signifying the language pair ID (equivalent to the record ID in the TRANSLATION\_SERVERS table)
- term\_list\_name=[string] – This is passed as POST data, with the value being a URL-encoded string containing the desired name of the new custom term-list. (e.g. a new term-list called “Intel terms” would be passed in as “term\_list\_name=Intel+terms”)
  - NOTE – if the term-list name already exists, this API request will fail with an HTTP 40x response code and an error message. This call only creates new unique custom term-lists. It will not edit existing term-lists.

**Note:**

- For this call the signature should include both account\_id and user\_id id (See Section 1: Authorization Details)
  - account\_id=[accountID] –This is an integer identifying the account ID of the requester.
  - user\_id=[userID] – This is an integer identifying the user ID of the requester.

**Optional Parameters Description:**

Optional parameters are provided in a key-value notation. The following optional keys can be specified with this API:

- entries =[string] – This is passed as POST data with the value being a URL-encoded string containing a custom XML format. The XML data allows users to add terms to a newly created term-list immediately. If “entries” is not found in the input parameters, this API will simply create an empty custom term-list named after the “term\_list\_name” passed in.

The XML format that needs to be used is as follows

```

<entries>
  <entry>
    <source>../source>
    <target>../target>
    <comment>../comment>
  </entry>
  <entry>
    <source>../source>
    <target>../target>
    <comment>../comment>
  </entry>
</entries>

```

### Output Response:

Successful requests will receive an HTTP 200 response code as well as an XML document. Failed HTTP requests will result in an HTTP 400 response code along with an error message in an XML document.

If the External API request is successful, the server will return an HTTP 200 OK response along with an XML document indicating the new term-list ID created.

```

<lwresponse>
  <service_version>v2</service_version>
  <requested_url>/v2/language-pair/74/term-list/user_id=123</requested_url>
  <request_type>POST</request_type>
  <request_time>Wed Mar 3 14:55:51 2010</request_time>
  <response_data type='language_pair_term_list_post'>
    <account_id>1234</account_id>
    <user_id>1000</user_id>
    <lpid>74</lpid>
    <source_language>eng</source_language>
    <target_language>fra</target_language>
    <term_list_id>456</term_list_id>
    <term_list_name>blah</term_list_name>
    <comments>
      <comment>Term List Created</comment>
      <comment>Duplicated items were ignored</comment>
      <comment>...</comment>
    </comments>
  </response_data>
</lwresponse>

```

## 5.2 Add/Delete Term(s) to/from Custom Term-List

### URL Resource:

<https://api.sdlbeglobal.com/v2/language-pair/<lpid>/term-list/<termListID>/terms/>

### Description:

This API request allows a user (specified by the account ID and the user ID) to add terms to an existing custom term-list given a language pair ID and a term-list ID.

This resource supports only HTTP PUT request types

### Mandatory Parameters Description:

- /v2– specifies the version of the API
- <termListID> –This is an integer identifying the term list ID to add/delete terms to/from
- entries=[xml] – This is a URL-encoded string containing a predefined XML format allowing users to add/delete terms.

The proposed format is as follows:

```
<entries>
  <add_entry>
    <source></source>
    <target></target>
    <comment></comment>
  </add_entry>
  <add_entry>
    <source></source>
    <target></target>
    <comment></comment>
  </add_entry>
  <add_entry>
    <source></source>
    <target></target>
    <comment></comment>
  </add_entry>
  <add_entry>
    <source></source>
    <target></target>
    <comment></comment>
  </add_entry>
  <delete_entry>
    <source></source>
  </delete_entry>
</entries>
```

### Note:

- For this call the signature should include both account\_id and user\_id (See Section 1: Authorization Details)

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

- account\_id=[accountID] –This is an integer identifying the account ID of the requester.
- user\_id=[userID] – This is an integer identifying the user ID of the requester.

**Note:**

- Prior to executing this call, the user has to use the Get List Of Term-Lists for Language Pair call (listed below in section 6.5) in order to obtain the <termListID>=term\_list\_id

**Optional Parameters Description:**

- None

**Output Response:**

Successful requests will receive an HTTP 200 response code as well as an XML document. Failed HTTP requests will result in an HTTP 400 response code along with an error message in an XML document.

The External API will return an HTTP 200 OK response if the API request was successful along with the following XML output:

```
<lwresponse>
  <service_version>v2</service_version>
  <requested_url>/v2/language-pair/74/term-
list/456/terms/user_id=123</requested_url>
  <request_type>PUT</request_type>
  <request_time>Wed Mar 3 14:55:51 2010</request_time>
  <response_data type='language_pair_term_list_terms_put'>
    <account_id>1234</account_id>
    <user_id>1000</user_id>
    <lpid>74</lpid>
    <source_language></source_language>
    <target_language></target_language>
    <term_list_id>456</term_list_id>
    <term_list_name>blah</term_list_name>
    <comments>
      <comment>Terms Added</comment>
      <comment>Duplicated items were ignored</comment>
      <comment>...</comment>
    </comments>
  </response_data>
</lwresponse>
```

### 5.3 Upload File to Custom Term-List (Import CSV)

**Resource URL:**

[https://api.sdlbglobal.com/v2/language-pair/<lpid>/term-list/<termListID>/file/input\\_format=<inputFormat>/](https://api.sdlbglobal.com/v2/language-pair/<lpid>/term-list/<termListID>/file/input_format=<inputFormat>/)

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services



**Description:**

This API request allows users to upload a file containing terms to be added to an existing term-list given a language pair ID and a term-list ID. The file must be a comma-separated-value file at this time.

We have chosen the CSV file type because of its simplicity of creation and utilization. For most of the FLAVIUS customers, this format should be the easiest one to use, when creating and updating term-list.

This resource supports only HTTP PUT request types

**Mandatory Parameter Descriptions:**

- /v2– specifies the version of the API
- <lpid> – This is an integer identifying the language pair containing the custom term-list.
- <termListID> – This is an integer identifying the term-list ID to add a file to
- <inputFormat> – This is a string identifying the input file format. Currently it is only “csv”. Anything else is rejected with an HTTP 40x response. (e.g. “input\_format=csv”)
- “term\_file” – This POST data string identifies the file to be added to the custom term-list.
  - External API – This points to the actual multi-part form data passed by the user to the external API. The external API will take this data and then save it to a network location. This network location is then passed to the internal Whitney Web App API using the “term\_file” POST data key.
  - Internal Whitney Web App API – This is a POST data string containing the path to the location of the file on a network file share. (e.g. “term\_file=/home/giraffe/data/users/1/term-list/123/file.csv”)

**Note:**

- For this call the signature should include both account\_id and user\_id (See Section 1: Authorization Details)
  - account\_id=[accountID] –This is an integer identifying the account ID of the requester.
  - user\_id=[userID] – This is an integer identifying the user ID of the requester.

**Optional Parameters Descriptions:**

- None

**Output Response:**

Successful requests will receive an HTTP 200 response code as well as an XML document. Failed HTTP requests will result in an HTTP 400 response code along with an error message in an XML document.

If the External API request is successful, the server will return an HTTP 200 OK response along with an XML document. The XML document will look as follows:

```
<lwresponse>
  <service_version>v2</service_version>
  <requested_url>/v2/language-pair/74/term-
list/456/terms/file/input_format=lwxml/user_id=123</requested_url>
  <request_type>PUT</request_type>
  <request_time>Wed Mar 3 14:55:51 2010</request_time>
  <response_data type='language_pair_term_list_terms_file_put' >
    <account_id>1234</account_id>
    <user_id>1000</user_id>
    <lpid>74</lpid>
    <source_language></source_language>
    <target_language></target_language>
    <term_list_id>456</term_list_id>
    <term_list_name>blah</term_list_name>
    <input_filename>..</input_filename>
    <input_format>lwxml</input_format>
    <comments>
      <comment>Terms Added</comment>
      <comment>Duplicated items were ignored</comment>
      <comment>...</comment>
    </comments>
  </response_data>
</lwresponse>
```

## 5.4 Get Custom Term-List (Get the details/content of a specific Term List)

### URL Resource:

<https://api.sdlbeglobal.com/v2/language-pair/<lpid>/term-list/<termListID>/>

### Description:

This API request retrieves the details/content of a specific term-list (dictionary).

This resource supports only HTTP GET request types.

### Mandatory Parameters Description:

- /v2– specifies the version of the API
- <lpid> – This is an integer identifying the language pair to retrieve the custom term-list from
- <termListID> – This is an integer identifying the custom term list to retrieve

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

**Note:**

- For this call the signature should include both account\_id and user\_id (See Section 1: Authorization Details)
  - account\_id=[accountID] –This is an integer identifying the account ID of the requester.
  - user\_id=[userID] – This is an integer identifying the user ID of the requester.

**Optional Input Parameters:**

- None

**Output Response:**

Successful requests will receive an HTTP 200 response code as well as an XML document. Failed HTTP requests will result in an HTTP 400 response code along with an error message in an XML document.

If the API request is successful, the server will return an HTTP 200 OK response along with an XML document. The XML document will look as follows:

```

<lwresponse>
  <service_version>v2</service_version>
  <requested_url>/v2/language-pair/74/term-list/456/user_id=123</requested_url>
  <request_type>GET</request_type>
  <request_time>Wed Mar 3 14:55:51 2010</request_time>
  <response_data type='language_pair_term_list_get' >
    <account_id>..</account_id>
    <user_id>1000</user_id>
    <lpid>74</lpid>
    <term_list_id>456</term_list_id>
    <term_list_name>blah</term_list_name>
    <source_language>eng</source_language>
    <target_language>fra</target_language>
    <last_modified_time>...</last_modified_time>
    <entries>
      <entry>
        <source>..</source>
        <target>..</target>
        <comment>..</comment>
      </entry>
      <entry>
        <source>..</source>
        <target>..</target>
        <comment>..</comment>
      </entry>
    </entries>
  </response_data>
</lwresponse>

```

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

## 5.5 Get List of Custom Term-Lists for Language Pair (for account)

### URL Resource:

<https://api.sdlbeglobal.com/v2/language-pair/<lpid>/term-list/>

### Description:

This API request retrieves a list of custom term-lists (dictionaries) on a language pair (for a specific account) given the language pair ID.

This resource supports only HTTP GET request types

### Mandatory Parameters Description:

- /v2– specifies the version of the API
- <lpid> – This is an integer identifying the language pair to retrieve custom term-list from

### Note:

- For this call the signature should include both account\_id and user\_id (See Section 1: Authorization Details)
  - account\_id=[accountID] –This is an integer identifying the account ID of the requester.
  - user\_id=[userID] – This is an integer identifying the user ID of the requester.

### Optional Parameters Description:

- None

### Output Response:

Successful requests will receive an HTTP 200 response code as well as an XML document. Failed HTTP requests will result in an HTTP 400 response code along with an error message in an XML document.

If the API request is successful, the server will return an HTTP 200 OK response along with an XML document. The XML document will look as follows:

```
<lwresponse>
  <service_version>v2</service_version>
  <requested_url>/v2/language-pair/74/term-list/user_id=123</requested_url>
  <request_type>GET</request_type>
  <request_time>Wed Mar 3 14:55:51 2010</request_time>
  <response_data type='language_pair_term_lists_get' >
    <account_id>1234</account_id>
    <user_id>1000</user_id>
    <lpid>74</lpid>
```

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

```

<source_language></source_language>
<target_language></target_language>
<term_lists>
  <term_list>
    <term_list_id>456</term_list_id>
    <term_list_name>blah</term_list_name>
  </term_list>
  <term_list>
    <term_list_id>789</term_list_id>
    <term_list_name>foobar</term_list_name>
  </term_list>
</term_lists>
</response_data>
</lwresponse>

```

## 5.6 Delete Custom Term-List

### URL Resource:

<https://api.sdlbglobal.com/v2/language-pair/<lpid>/term-list/<termListID>/>

### Description:

This API request deletes a specific custom term-list (dictionary) from a language pair given a language pair ID and a term-list ID. The account owner can only delete a custom term-list (dictionary) they own.

This resource supports only HTTP DELETE request types

### Mandatory Parameters Description:

- /v2– specifies the version of the API
- <lpid> – This is an integer identifying the language pair to delete the term-list from
- <termListID> – This is an integer specifying the custom term-list to delete.

### Note:

- For this call the signature should include both account\_id and user\_id (See Section 1: Authorization Details)
  - account\_id=[accountID] –This is an integer identifying the account ID of the requester.
  - user\_id=[userID] – This is an integer identifying the user ID of the requester.

### Optional Parameters Description:

- None

### Output Response:

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

Successful requests will receive an HTTP 200 response code as well as an XML document. Failed HTTP requests will result in an HTTP 400 response code along with an error message in an XML document.

If the API request is successful, the server will return an HTTP 200 OK response along with an XML document. The XML document will look as follows:

```
<lwresponse>
  <service_version>v2</service_version>
  <requested_url>/v2/language-pair/74/term-list/456/user_id=123</requested_url>
  <request_type>DELETE</request_type>
  <request_time>Wed Mar 3 14:55:51 2010</request_time>
  <response_data type='language_pair_term_list_delete' >
    <account_id>1234</account_id>
    <user_id>1000</user_id>
    <lpid>74</lpid>
    <source_language></source_language>
    <target_language></target_language>
    <term_list_id>456</term_list_id>
    <term_list_name>blah</term_list_name>
    <comments>
      <comment>Term List Deleted</comment>
    </comments>
  </response_data>
</lwresponse>
```

## 5.7 Upload File to Brand List

### URL Resource:

[https://api.sdlbglobal.com/v2/brand-list/file/input\\_format=<inputFormat>/](https://api.sdlbglobal.com/v2/brand-list/file/input_format=<inputFormat>/)

### Description:

This API request allows users to add terms stored in a supported file to the account brand list. The file will be uploaded to the web service using multi-part form data format.

We have chosen the CSV file type because of its simplicity of creation and utilization. As in the case of the term-list, we consider this format the easiest one to use, when creating and updating a brand-list.

This resource supports only HTTP PUT request types

### Mandatory Parameters Description:

- /v2– specifies the version of the API
- <inputFormat> – This is a string identifying the input format of the file. Currently, it is a CSV file. (e.g. “input\_format=csv”)

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

- source\_file=[multi-part FORM data] – This is a POST form variable which contains multi-part form data for the file uploaded by the user.

**Note:**

- For this call the signature should include both account\_id and user\_id (See Section 1: Authorization Details)
  - account\_id=[accountID] –This is an integer identifying the account ID of the requester.
  - user\_id=[userID] – This is an integer identifying the user ID of the requester.

**Optional Parameters Description:**

- None

**Output Response:**

Successful requests will receive an HTTP 200 response code as well as an XML document. Failed HTTP requests will result in an HTTP 400 response code along with an error message in an XML document.

If the API request is successful, the server will return an HTTP 200 OK response along with an XML document. The XML document will look as follows:

```
<lwresponse>
  <service_version>v2</service_version>
  <requested_url>/v2/brand-
list/file/input_format=plain/user_id=123</requested_url>
  <request_type>PUT</request_type>
  <request_time>Wed Mar 3 14:55:51 2010</request_time>
  <response_data type='brand_list_file_put' >
    <account_id>1234</account_id>
    <user_id>1000</user_id>
    <input_format>plain<input_format>
    <input_filename>.</input_filename>
    <comment>Brand List Updated</comment>
  </response_data>
</lwresponse>
```

**Notes:**

The web service will only support a few file formats including:

- UTF-8 encoded plain text file with one term per line
- CSV UTF-8 encoded file with one term per line

## 5.8 Get Brand List

**URL Resource:**

<https://api.sdlbeglobal.com/v2/brand-list/>

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

**Description:**

This API request allows users to retrieve the account brand list as an XML document.

This resource supports only HTTP GET request types

**Mandatory Parameters Description:**

- /v2– specifies the version of the API

**Note:**

- For this call the signature should include both account\_id and user\_id (See Section 1: Authorization Details)
  - account\_id=[accountID] –This is an integer identifying the account ID of the requester.
  - user\_id=[userID] – This is an integer identifying the user ID of the requester.

**Optional Parameters Description:**

- None

**Output Response:**

Successful requests will receive an HTTP 200 response code as well as an XML document. Failed HTTP requests will result in an HTTP 400 response code along with an error message in an XML document.

If the API request is successful, the server will return an HTTP 200 OK response along with an XML document. The XML document will look as follows:

```

<lwresponse>
  <service_version>v2</service_version>
  <requested_url>/v2/brand-list/user_id=123</requested_url>
  <request_type>GET</request_type>
  <request_time>Wed Mar 3 14:55:51 2010</request_time>
  <response_data type='brand_list_get' >
    <account_id>1234</account_id>
    <user_id>1000</user_id>
    <entries>
      <entry>
        <term>.</term>
        <comment>.</comment>
      </entry>
      <entry>
        <term>.</term>
        <comment>.</comment?
      </entry>
    </entries>
  </response_data>
</lwresponse>

```

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services



## 5.9 Edit Brand-List

### URL Resource:

<https://api.sdlbglobal.com/v2/brand-list/>

### Description:

This API request allows a user to edit an existing brand-list for the account.

This resource supports only HTTP PUT request types

### Mandatory Input Parameters:

- /v2– specifies the version of the API
- “entries=[string]” – This is a POST DATA key/value pair where the value is a URL-encoded string containing and XML formatted document. The XML format is as follows:

```
<entries>
  <add_entry>
    <source>MSN</source>
    <comment>c1</comment>
  </add_entry>
  <add_entry>
    <source>Yahoo</source>
    <comment>t1_01</comment>
  </add_entry>
  <add_entry>
    <source>AMD</source>
    <comment>c2</comment>
  </add_entry>
  <add_entry>
    <source>IntelNew03</source>
    <comment>c3</comment>
  </add_entry>
  <delete_entry>
    <source>MSN</source>
  </delete_entry>
</entries>
```

### Note:

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

- For this call the signature should include both account\_id and user\_id (See Section 1: Authorization Details)
  - account\_id=[accountID] –This is an integer identifying the account ID of the requester.
  - user\_id=[userID] – This is an integer identifying the user ID of the requester.

**Optional Parameters Description:**

- None

**Output Response:**

Successful requests will receive an HTTP 200 response code as well as an XML document. Failed HTTP requests will result in an HTTP 400 response code along with an error message in an XML document.

If the API request is successful, the server will return an HTTP 200 OK response along with an XML document. The XML document will look as follows:

```

<lwresponse>
  <service_version>v2</service_version>
  <requested_url>/v2/brand-list/user_id=123</requested_url>
  <request_type>PUT</request_type>
  <request_time>Wed Mar 3 14:55:51 2010</request_time>
  <response_data type='brand_list_put' >
    <account_id>1234</account_id>
    <user_id>1000</user_id>
    <comment>Brand List Updated</comment>
  </response_data>
</lwresponse>

```

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

## 6. Test coverage of the Term & Brand Management API

### 6.1 Component test - Overview

The Term & Band Management API has been covered with integration tests to ensure the quality of the software delivered. The tests have been built using the TestNG framework, a framework inspired from JUnit and NUnit (<http://testng.org/doc/index.html>).

All tests have been run on the production environment, located at <https://api.sdlbeglobal.com>.

All functionality has been covered with integration tests and the tests have been focused on both the individual components as well as flows (that included two or more components). The details below cover the individual components that are part of the Term & Brand Management API :

- Create Custom Term List
- Add/Delete Term(s) to/from Custom Term List
- Upload File to Custom Term List (Import CSV)
- Get Custom Term List (Get the details/content of a specific Term List)
- Get List of Custom Term Lists for Language Pair (for account)
- Delete Custom Term List
- Upload File to Brand List
- Get Brand List
- Edit Brand List

The tests have been grouped in Happy Flow tests and Negative Flow tests.

A **Happy Flow** test- covers the case when all the details provided by the user are correct, and all the restrictions are met.

A **Negative Flow** test – covers the scenarios when the details provided to the API are incorrect or some of the restrictions clearly stated are not met.

All the integration tests created are being run with every new build to ensure the robustness of the already delivered functionality. The test can also be run on demand.

We have presented below only a sample of the Use cases considered during our testing process, the ones we felt have the highest relevance.

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

## 6.2 Component test - Create Custom TermList

This scenario covers the creation of a TermList for the English to French language pair, with the name “simpleDict”.

The Happy Flow test has successfully passed, and the response from the API is “Empty Term List Created” that confirms the creation of the TermList named “simpleDict”.

The Negative tests in this scenario included (but were not limited) to the following situations:

- Requesting the creation of a dictionary with a name that already existed for that account
- Requesting the creation of a dictionary with a name that contains special characters (listed in the attached Appendix A)
- Requesting the creation of a dictionary for a language pair that is not part of the available language pairs for the account

```
Response data= <?xml version="1.0" encoding="UTF-8"?>
<lwresponse>
  <service_version>v2</service_version>
  <requested_url>/v2/language-pair/1000/term-
list/account_id=10555/user_id=1599</requested_url>
  <request_type>POST</request_type>
  <request_time>Wed Aug 24 14:15:48 UTC 2011</request_time>
  <response_data type='language_pair_term_list_post'>
    <account_id>10555</account_id>
    <user_id>1599</user_id>
    <lpid>1000</lpid>
    <source_language>en</source_language>
    <target_language>fr</target_language>
    <term_list_id>474</term_list_id>
    <term_list_name>simpleDict</term_list_name>
    <comments><comment>Empty Term List Created</comment></comments>
  </response_data>
</lwresponse>
```

A similar scenario here for the same API call, will cover the more likely possibility of creating a new TermList with the name “newENFR” for the same English to French language pair that will not be created empty but directly with some terms.

The Happy Flow test has passed and the response of the API “Term List Created”, confirms the successful creation of the TermList

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

```

Response data= <?xml version="1.0" encoding="UTF-8"?>
<lwresponse>
  <service_version>v2</service_version>
  <requested_url>/v2/language-pair/1000/term-
list/account_id=10555/user_id=1599/</requested_url>
  <request_type>POST</request_type>
  <request_time>Wed Aug 24 14:32:00 UTC 2011</request_time>
  <response_data type='language_pair_term_list_post'>
    <account_id>10555</account_id>
    <user_id>1599</user_id>
    <lpid>1000</lpid>
    <source_language>en</source_language>
    <target_language>fr</target_language>
    <term_list_id>476</term_list_id>
    <term_list_name>newENFR</term_list_name>
    <comments><comment>Term List Created</comment></comments>
  </response_data>
</lwresponse>

```

### 6.3 Component test - Add/Delete Term(s) to/from Custom Term List

This scenario covers the addition of couple of terms to an existing TermList named “NewSpanish” for the language pair English to Spanish.

Happy Flow - covers the case when all the details provided are correct, and all the restrictions are met.

The following dictionary data was used to add/delete terms

```

<term_list_item row_id="1" enabled= "true" term_list_id = "7397" term_list_name =
"NewSpanish" source_language="eng" target_language="spa" description = "20766"
z_bad_tl_params="" term_list_upload_file ="" >
  <entry_item row_id="1" enabled= "true" source="feedback" target="comentarios"
comment="" description="" fk_term_list_id = "1"/>
  <entry_item row_id="2" enabled= "true" source="soul" target="alma"
comment="" description="" fk_term_list_id = "1"/>
  <entry_item row_id="3" enabled= "true" source="soul" target="alma"
comment="" description="delete" fk_term_list_id = "1"/>

```

The Happy Flow test has successfully passed, and we got the confirmation from the API with the response “Terms Added”. The API also confirmed the duplicates (found in the above sample) were ignored.

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

```

Response data= <?xml version="1.0" encoding="UTF-8"?>
<lwresponse>
  <service_version>v2</service_version>
  <requested_url>/v2/language-pair/1000/term-
list/7397/terms/account_id=20766/user_id=1906/</requested_url>
  <request_type>PUT</request_type>
  <request_time>Thu Aug 04 15:59:07 UTC 2011</request_time>
  <response_data type='language_pair_term_list_terms_put'>
    <account_id>20766</account_id>
    <user_id>1906</user_id>
    <lpid>1000</lpid>
    <source_language>en</source_language>
    <target_language>sp</target_language>
    <term_list_id>7397</term_list_id>
    <term_list_name>NewSpanish</term_list_name>
    <comments>
      <comment>Terms Added</comment>
      <comment>Duplicated items were ignored</comment>
    </comments>
  </response_data>
</lwresponse>

```

## 6.4 Component test - Upload File To Custom Term List

This scenario covers the upload of a CSV file named “TermFileExample.csv” to the English to French dictionary named “simpleDict”.

The Happy Flow test has passed and we can see from the response below (“Terms Added”) the API response confirming the addition of the new terms.

```

<lwresponse>
  <service_version>v2</service_version>
  <requested_url>/v2/language-pair/1000/term-
list/474/file/input_format=csv/account_id=10555/user_id=1599/</requested_url>
  <request_type>PUT</request_type>
  <request_time>Thu Aug 25 13:26:24 UTC 2011</request_time>
  <response_data type='language_pair_term_list_terms_put'>
    <account_id>10555</account_id>
    <user_id>1599</user_id>
    <lpid>1000</lpid>
    <source_language>en</source_language>
    <target_language>fr</target_language>
    <term_list_id>474</term_list_id>

```

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

```

<term_list_name>simpleDict</term_list_name>
<input_filename>/home/.../TranslationProj/source/TermFileExample.csv</input_file
name>
    <input_format>csv</input_format>
    <comments><comment>Terms Added</comment></comments>
</response_data>
</lwresponse>

```

The Negative tests in this scenario included (but were not limited) to the following situations:

- Uploading a bad formatted csv file (the response is listed below for a file named “TermFileExample\_25.csv”)
  - This will be the most common scenario and has been covered from several angles that included: ANSI and not UTF-8 formatting, having a CSV with a different structure than the one LW is recommending.
  - The frequency of this scenario will be reduced by providing to the FLAVIUS users access to the Appendix A as a guideline for the creation of the TermLists and BrandList
- Uploading a file with terms that contained restricted characters (listed in the Appendix A)

```

<?xml version="1.0" encoding="UTF-8"?>
<lwresponse>
    <service_version>v2</service_version>
    <requested_url>/v2/language-pair/1000/term-
list/474/file/input_format=csv/account_id=10555/user_id=1599/</requested_url>
    <request_type>PUT</request_type>
    <request_time>Thu Aug 25 14:27:53 UTC 2011</request_time>
    <error_messages>
    <error_message>Operation could not be completed.</error_message>
    </error_messages>
</lwresponse>

```

## 6.5 Component test - Delete Custom TermList

This scenario covers the deletion of the English to French dictionary named “simpleDict” that was created at one of the previous steps, with all its contents.

The Happy Flow test has passed and we can see from the response below “Term List Deleted” the successful deletion of the dictionary.

```

<?xml version="1.0" encoding="UTF-8"?>
<lwresponse>
    <service_version>v2</service_version>

```

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

```

<requested_url>/v2/language-pair/1000/term-
list/474/account_id=10555/user_id=1599/</requested_url>
<request_type>DELETE</request_type>
<response_data type='language_pair_term_list_delete'>
  <account_id>10555</account_id>
  <user_id>1599</user_id>
  <lpid>1000</lpid>
  <source_language>en</source_language>
  <target_language>fr</target_language>
  <term_list_id>474</term_list_id>
  <term_list_name>simpleDict</term_list_name>
  <comments><comment>Term List Deleted</comment></comments>
</response_data>
</lwresponse>

```

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services



# 7. Dictionary Customization - Case Studies

## 7.1 Introduction

Term and Brand management represents the solution which ensures that certain terms and/or brands are correctly and consistently used throughout translations. Term and Brand lists are created with the purpose of improving the quality of the machine translated output, by eliminating (totally or partially) recurrent problems found in the translations. However, the improvement in the quality of the translations is strictly dependent on how the lists of terms and brands are created.

In order for a Term or Brand list to be reliable, it must be composed of words and/or phrases with high impact in the data set that needs to be translated.

The next chapter will be a demonstration of how Term and Brand lists can improve the quality of the machine translation output. To demonstrate this, two cases of successful implementation will be described in a step-by-step approach. Each step that will be presented is an important part of the process that leads to the creation of reliable Term and Brand lists.

## 7.2 Experiment 1: Test on Training

This experiment has followed a scenario we envision some of the FLAVIUS customers will likely follow as well. This experiment was based also on our experience with some of our commercial customers.

### 7.2.1 Case Study – LW Customer

#### 7.2.1.1 Type of data used

This case study presents how a LW Customer who does not have the means to opt for a trained system (as he has little to no parallel data to build such a system) can take advantage of the dictionary customization.

#### Customer data characteristics:

- The small – medium-sized customer data consisted of:
  - a monolingual set of 1,444 segments (24,232 words)

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

- a small bilingual set of 5229 segments (26,552 words)
- The source language of the data is German
- The language pair needed for translation: German>English
- The data is composed of concise product descriptions (watches)
- The segments are not real sentences, but product ads consisting of enumerations of single words and/or short phrases and short customer reviews in the same format. No segmentation was done as we wanted to preserve as unit of analysis an entire product ad.
- The domain is restricted to watch descriptions (a very narrow domain)
- The words are mostly unambiguous in context and most of the times there is one single translation variant that can be used
- The descriptions are highly similar, there are many words repeated throughout the data set

### 7.2.1.2 Initial translation of the data

---

Before any Term or Brand list was created, the source language data was first translated from German into English with the machine translation engine, without customization (baseline translation). This first step ensured the analysis of the translated output, with the aim to identify frequent translation problems.

### 7.2.1.3 Analysis of the initial translation and identification of recurrent translation issues

---

Once the source language data was translated with the machine translation engine without applying any list of terms or brands, the next step was to take a close look at the translated output. The purpose was to identify frequent problems in the data, but also to see what measures can be taken to solve these problems.

As a result of the analysis, the following translation problems were identified:

- words and phrases remained **non-translated**
- words and phrases were **translated incorrectly** into the target language
- certain **brands were translated**, while they were supposed to be kept as such in the target language

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

- **spelling mistakes** in the source data resulted in those misspelled words not being recognized by the system, therefore they were kept as such in the English translation

#### 7.2.1.4 Selection of a relevant set for applying Term and Brand lists

---

In order to test the Term and Brand list functionality, a set of 140 sentences was selected from the initially translated data, with a focus on the above-mentioned categories of problems. The selected set included examples from each category of mistakes that appeared frequently in the translated data.

#### 7.2.1.5 Creation of reliable Term and Brand lists

---

Once the test set was selected, the next step was to extract the problematic words and phrases belonging to that data set and separate them in two categories:

**Note:**

*Only terms and phrases that presented translation problems were considered eligible for term/brand list entries. If terminology was correctly rendered by the baseline system, we no longer considered those words and phrases as potential dictionary candidates as there was no need for additional customization.*

- **terms to be included in the Term list**
  - the terms were chosen for the Term list based on their frequency in the data set. First of all, the list included those words and phrases with the highest frequency and which were either non-translated or mistranslated.

**Note:**

*The frequency of terms was assessed with a simple Search and Find All in a text editor*

It is important to **sort the terms by their frequency** in the data set before adding them to a Term list. Priority should always be given to highly frequent terms, because the higher the frequency, the bigger the impact of the list on the translated output.

- Phrases composed of 2 or more words were also added to the Term list

It is important to include **short sentences** in the Term list, because this will ensure that their meaning is correctly and consistently rendered. If single terms already included in the dictionary are also part of larger phrasal constructions, you have to make sure to include them in the Term list for optimal results.

**Note:**

*The search and replace function will always begin with the larger structures, so adding these to the list will help improve the quality of the translation.*

- **terms to be included in the Brand list**

- the Brand list included the brands (names of companies or products) which should have been kept as such in the machine translation output, while they were either partially or completely mistranslated thus affecting customer and brand voice .

**Term list**

The table below contains some examples for each category of mistakes found in the test set and included in the Term list. The structure of the table is the following:

- **Source term** column: contains examples of terms extracted from the original data set (in German), which were included in the dictionary.
- **Target term** column: contains the translations into English of the source terms included in the dictionary.
- **Source text** column: contains the original context in which the term was found.
- **Translation without Term list** column: contains the translation of the original segment, to which no Term list was applied.
- **Translation with Term list** column: contains the translation of the original segment, to which the Term list was applied.

Source term	Target term	Source text	Translation without Term list	Translation with Term list
<b>Non-translated terms</b>				
Zifferblatt	dial	Zustand: gebraucht; leichte Gebrauchsspuren; Herstellungsjahr: 60er; Funktionen: Chronograph; (Minuten, Sekunden);	condition: used; easy tracks; Year of manufacture: 1960s; functions: Chronograph; (minutes, seconds);	condition: used; slight signs of use; production year: 1960s; functions: Chronograph; (minutes, seconds);
Handaufzug	movement	<b>Zifferblatt:</b> blau; Gehäuse: stahl; Maße: 37x37x13mm; Werk: <b>Handaufzug;</b> Landeron51; Band: Leder; Lieferzeit: ca. 1 Woche; Garantie:; 12 Monate	<b>Zifferblatt:</b> blue; Housing: steel; dimensions: 37x37x13mm; work: <b>Handaufzug;</b> Landeron51; Band: Leather; DELIVERYTIME: approx. 1 week; guarantee:; 12 months	<b>dial:</b> blue; Case: Steel; dimensions: 37x37x13mm; <b>movement:</b> handwind; Landeron51; strap: Leather; time of delivery: approx. 1 week; warranty:; 12 months
Taschenuhrkette	pocket watch chain	<b>Taschenuhrkette,</b> 585er <b>Gelbgold,</b> Länge 42,5 cm, Gewicht 14,5 g, sehr guter Zustand	<b>Taschenuhrkette,</b> 18K <b>Gelbgold,</b> length 42,5 cm, Weight 14,5 g, very good condition	<b>pocket watch chain,</b> 18K <b>yellow gold,</b> length 42,5 cm, Weight 14,5 g, very good condition
Gelbgold	yellow gold			
<b>Mistranslated terms</b>				
Band	strap	Herrenarmbanduhr, Ref. 6500.12, limitierte Serie für F.A. Porsche weltweit 1999 Stück. Gehäuse und <b>Band</b> Titan, Kaufdatum 2002, Garantie.	Herrenarmbanduhr, Ref. 6500.12 , limitierte series for F. A. Porsche worldwide 1999 piece. Housing and <b>volume</b> Titan, Kaufdatum 2002, guarantee.	men's wrist watch, Ref. 6500.12 , Limited series of F. A. Porsche worldwide 1999 piece. Titanium case and <b>strap,</b> date of purchase 2002, warranty.
Uhr	watch	Extrem seltene Kompass <b>Uhr</b> in Titan. Perfekter Zustand. IWC <b>überholt</b>	Extremely rare compass <b>hours</b> in titanium. Perfect condition. IWC <b>outdated</b>	Extremely rare compass <b>watch</b> in titanium. Perfect condition. IWC <b>overhauled</b>
überholt	overhauled			
<b>Phrases (composed of at least 2 words) partially mistranslated/not translated</b>				

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

beidseitig drehbare Lünette	bidirectional rotating bezel	Herrenarmbanduhr, Ref. 1680, <b>beidseitig drehbare Lünette</b> , um 1977, Garantie.	Herrenarmbanduhr, Ref. 1680, <b>beidseitig rotating bezel</b> , 1977, guarantee.	Men's wrist watch, Ref. 1680, <b>bidirectional rotating bezel</b> , 1977, warranty.
Limitierte Serie	Limited series	Perfekte Uhr mit allen Unterlagen. <b>Limitierte Serie</b> 300 Stck.	perfect hours with all the documents. <b>Limitierte series</b> 300 pcs.	perfect watch with all documents. <b>Limited series</b> 300 pcs.

### Brand lists

The table below contains some examples of brand names which were mistranslated in the initial machine translated output. Therefore, it was decided that they should be included in the Brand list to ensure consistent and reliable brand and customer voice. The structure of the table is the following:

- **Brand** column: contains some of the brands which were mistranslated before a Brand list was applied.
- **Source text** column: contains the original context in which the brand was found.
- **Translation without Brand list** column: contains the translation of the original segment, to which no Brand list was applied.
- **Translation with Brand list** column: contains the translation of the original segment, to which the Brand list was applied.

Brand	Source text	Translation without Brand list	Translation with Brand list
Fliegeruhr Mark	<b>Fliegeruhr Mark</b> , guter Zustand, kaum getragen; letzte Reinigung 2010; Zustand: gebraucht; leichte Gebrauchsspuren; Herstellungsjahr: ca 80 jahre alt; Zifferblatt: gelb-weiß, etwas verfärbt; Gehäuse: Stahl, Kristallglas; Maße: Durchmesser ca. 32mm; Werk: Handaufzug; eigenes Werk, Kal. 581; Band: Leder; Lieferzeit: 1	<b>Pilot's watch Mark</b> , good condition, hardly worn; last year of manufacture: ca 80 years old; Zifferblatt: cleaning 2010; Condition: used; easy tracks; yellow-white, something discoloured; Housing: steel, crystal; dimensions: diameter approx. 32mm; work: Handaufzug; own work, kal. 581; Band: Leather; DELIVERYTIME: 1 to 3	<b>Fliegeruhr Mark</b> , good condition, hardly worn; last cleaning 2010; Condition: used; slight signs of use; production year: ca 80 years old; dial: yellow-white, something discoloured; Case: Steel, crystal glass; dimensions: diameter approx. 32mm; movement: handwind; own movement, kal. 581; strap: Leather; time of delivery: 1 to 3 days;

	bis 3 Tage; Gewährleistung: keine	days; ensure: no	warranty: no
Glashütte Original	Herrenarmbanduhr <b>Glashütte Original</b> , Automatic mit Großdatum, Stahl mit Krokolederarmband Dornschließe, Saphirglasboden, Gehäusedurchmesser 39 mm, ca. 2 Jahre alt, Orig. Schachtel und Papiere	Herrenarmbanduhr <b>glassworks original</b> , Automatic with big date, steel with Krokolederarmband buckle, Saphirglasboden, diameter body 39 mm, approx. 2 years old, orig. Boxes and papers	men's wrist watch <b>Glashütte Original</b> , Automatic with large date, steel with alligator strap buckle, sapphire crystal back, Case diameter 39 mm, approx. 2 years old, orig. Boxes and papers

For the LW Customer, having bilingual text proved to be helpful in the sense that we could create a terminology list from the previously translated material which translated into better results obtained in a timely manner. Therefore, by using the terms extracted from the monolingual data set and the parallel one, we managed to create the following Term and Brand List:

Types of entries	Number of entries
Term list entries:	454
Brand list entries:	281
Total	735

### 7.2.1.6 Assessing the impact of the Term and Brand lists

The actual impact of applying Term and Brand lists to a text that needs to be translated can be measured if the two translations (one without dictionary and the other with dictionary) are analyzed in parallel.

This analysis was done by means of a human evaluation called sentence evaluation. The evaluation consisted of the following steps:

- The original text, together with the two translations was set up as a sentence evaluation job containing 140 source segments (product ads).
- Two persons were asked to evaluate the translations using a **blind-evaluation** methodology: the display of the two translations was set up such that the identity of the engines that produced the translations was both hidden and randomized from one screen to the next.

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

- Each evaluator had to read each segment (the original text plus the 2 translations) and assign to each translation a score from 1 to 5.
- The scale used for this evaluation was the Likert scale, on which 1 is the lowest score and 5 is the highest score.

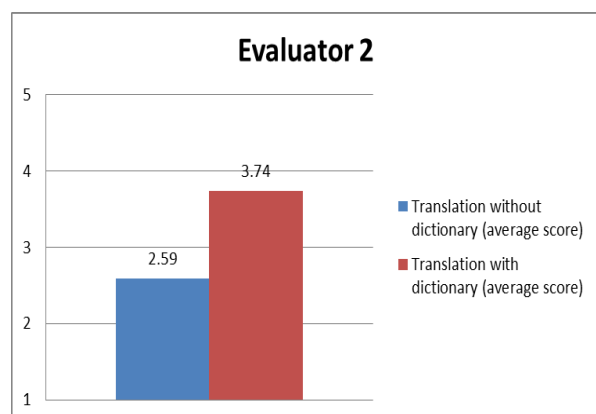
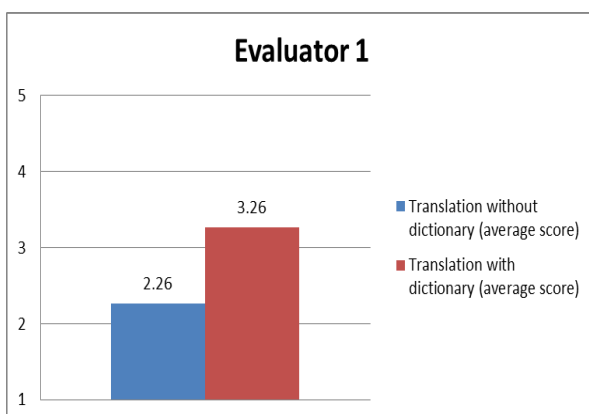
**Scoring Guidelines:**

- **5** - The document is **understandable** and **actionable**, with **all critical information accurately transferred**. **Nearly all of the text** is well translated.
- **4** - The document is **understandable** and **actionable**, with **most critical information accurately transferred**. **Most of the text** is well translated.
- **3** - The document is **not entirely understandable** but it is **actionable**, with **some critical information accurately transferred**. The text is stylistically or grammatically odd. **Some of the text** is well translated.
- **2** - The document is **possibly understandable** and **actionable** given enough context and/or time to work it out, with **some information accurately transferred**.
- **1** - The document is **not understandable** and it is **impossible to understand** the information it contains.

- The score assigned had to reflect the level of usability of the translation (i.e. if the translation could be useful to someone who only speaks the language that the text was translated into).
- Once the evaluations were finished, the results obtained were then analyzed to see if the quality of the translations had been improved with the help of the Term and Brand lists.

The graphs below show the results of the evaluation performed by the 2 evaluators. One chart has been generated per evaluator, displaying the average of the scores assigned to the translations with and without dictionary entries.

***Evaluator 1 & Evaluator 2 results:***



Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services



- The blue column shows the average score assigned by the evaluators to the translations without Term and Brand lists. The average for the set of 140 sentences is **2.26 (Evaluator 1)** and **2.59 (Evaluator 2)** on the 1 to 5 Likert scale.
- The red column shows the average score assigned by the evaluators to the translations with Term and Brand lists. The average for the set of 140 sentences is **3.26 (Evaluator 1)** and **3.74 (Evaluator 2)** on the 1 to 5 Likert scale.
- The difference in the average scores assigned to the two translations is **+1 point (Evaluator 1)** and **+1.15 points (Evaluator 2)** in favor of the customized translation. Both evaluators considered that there is a quality gap between the baseline translation and the dictionary translation of 1.08 points on average. This quality gap can be better understood in terms of the human evaluators preferring one system’s translations versus the other: for a +1.08 Likert difference between systems A and B, human evaluators indicate that, without doubt the translations of system A are considerably better than the translations of system B.

### 7.2.1.7 LW Customer: Conclusions

---

The results of the LW Customer experiment clearly show the positive impact of using Term and Brand lists. The average scores assigned to the translations with dictionary are considerably higher than the ones without dictionary. Both evaluators (who worked independently from each other and without know what engine produced what segments) considered that the quality of the translations with dictionary was on average 1.08 points higher.

However, this considerable improvement should not be regarded as completely independent of a series of conditions which made it possible:

- The original data was of such nature that Term and Brand lists were quite easily created
- Since the data consisted of enumerations of product features, most of the terms were context-independent.
- There were many repetitions in terms of descriptions, which mean that the dictionary entries had a higher weight, thus leading to a more noticeable improvement of the translation quality.

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

## 7.2.2 Case Study – Overblog

### 7.2.2.1 Type of data used

---

**The Overblog data characteristics:** The type of data used for translation in the context of the Flavius project has the following characteristics:

- The medium-sized towards high customer data set consisting of:
  - a monolingual set of 18,641 segments (266,614 words)
- The source language of the data is English
- The language pair needed for the translation is English > French
- The data consists of financial articles posted on blogs. The articles were segmented as the unit of analysis in our case was the sentence.
- The language is specialized, due to the specific terminology used
- Since the data set is highly specialized, the words are unambiguous in context, generally there is one single translation variant that can be used for a specific term

### 7.2.2.2 Initial translation of the segmented data

---

The financial articles were first segmented, thus the Customer data consisted of segmented data for ease of analysis.

Before creating a Term or Brand list, the data set was first translated from English into French with the machine translation engine, without customization (baseline translation). This first step allowed for an efficient analysis of the translated output, with the aim to identify frequent translation problems.

### 7.2.2.3 Analysis of the initial translation and identification of recurrent translation issues

---

After the data was translated with the machine translation engine without applying any list of terms or brands, the next step was to analyze closely the machine translation output. The purpose of this analysis was to identify frequent translation problems, but also to see what measures can be taken to solve these problems.

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

As a result of the analysis, the following translation problems were identified:

- words and phrases remained **non-translated**
- words and phrases were **translated incorrectly** into the target language
- certain **brands were partially or completely translated** into French, although their original form should have been preserved
- **spelling mistakes** in the source data resulted in those misspelled words not being recognized by the translation system, therefore they were kept as such in the French translation

#### 7.2.2.4 Selection of a relevant set for applying Term and Brand lists

---

In order to test the Term and Brand list functionality, a set of 100 sentences was selected from the initially translated data. The selection included examples from each category of mistakes that appeared frequently in the translated data.

#### 7.2.2.5 Creation of reliable Term and Brand lists

---

Once the test set was selected, the next step was to extract the problematic words and phrases and separate them in two categories:

- **terms to be included in the Term list**
  - problematic terms with high frequency in the data set (non-translated, mistranslated and misspelled terms)
  - phrases composed of at least 2 words (translated incorrectly)
  - context-independent terms, which can be safely used throughout the data set
- **terms to be included in the Brand list**
  - names of financial institutions and products, which were either partially or completely mistranslated

**Note:**

*Only terms and phrases that presented translation problems were considered eligible for term/brand list entries. If terminology was correctly rendered by the baseline system, we no longer*

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

considered those words and phrases as potential dictionary candidates as there was no need for additional customization

### Term lists

The table below contains some examples for each frequent type of mistakes found in the test set and which in the end was included in the Term list. The structure of the table is the following:

- **Source term** column: contains examples of terms extracted from the original data set (in English), which were included in the dictionary.
- **Target term** column: contains the translations into French of the source terms included in the dictionary.
- **Source text** column: contains the original context in which the term was found.
- **Translation without Term list** column: contains the translation of the original segment, to which no Term list was applied.
- **Translation with Term list** column: contains the translation of the original segment, to which the Term list was applied.

Source term	Target term	Source text	Translation without Term list	Translation with Term list
<b>Non-translated terms</b>				
teletracking	la traçabilité du client	<b>Teletracking</b> is performed to see if a customer has defaulted on a payday loan elsewhere.	<b>Teletracking</b> est effectué pour voir si un client a manqué à un prêt sur salaire ailleurs.	<b>La traçabilité du client</b> est effectuée pour voir si un client a manqué à un prêt sur salaire ailleurs.
creditcheck	vérification de crédit	Credit reference agencies are consulted for a <b>creditcheck</b> .	Organismes de référence de crédit sont consultés pour un <b>creditcheck</b> .	Organismes de référence de crédit sont consultés pour une <b>vérification de crédit</b> .

home mortgage	hypothèque résidentielle	What are the best Halifax <b>home mortgage</b> deals in the UK?	Quelles sont les meilleures halifax <b>home mortgage</b> traite au Royaume-Uni?	Quelles sont les meilleures halifax <b>hypothèque résidentielle</b> traite au Royaume-Uni?
ATM	GAB	An automated teller machine, or <b>ATM</b> allows financial transactions to be made by users.	Un guichet automatique, ou <b>ATM</b> permet aux transactions financières à être effectués par les utilisateurs.	Un guichet automatique, ou <b>GAB</b> permet aux transactions financières à être effectués par les utilisateurs.
<b>Mistranslated terms</b>				
bad credit mortgage	hypothèque mauvais crédit	One of the financial offerings which is helping many people, is a <b>bad credit mortgage</b> .	L'une des offres financières qui est d'aider de nombreuses personnes, est une <b>mauvaise de crédit hypothécaire</b> .	L'une des offres financières qui est d'aider de nombreuses personnes, est une <b>hypothèque mauvais crédit</b> .
low APR loan	prêt à TEG réduit	Getting a <b>low APR loan</b> should be on top of every loan taker's agenda.	Obtention d'un <b>faible avr prêt</b> devrait être à la tête de chaque emprunteuse l'ordre du jour.	Pour obtenir un <b>prêt à TEG réduit</b> devrait être à la tête de chaque preneur de prêt de l'ordre du jour.
<b>Phrases (composed of at least 2 words) translated incorrectly</b>				
no credit check auto loan	financement automobile sans vérification de crédit	A <b>no credit check auto loan</b> suits consumers who have a poor credit history.	<b>Aucune vérification de crédit un prêt auto</b> suits les consommateurs qui ont un mauvais dossier de crédit.	Un <b>financement automobile sans vérification de crédit</b> convient les consommateurs qui ont un mauvais dossier de crédit.
educational savings account	compte d'épargne éducation	Provident independent individual retirement accounts (IRA), <b>educational savings account</b> (ESA) and health savings accounts (HSA) deliver various	Indépendants de prévoyance comptes individuels de retraite (IRA), à l' <b>éducation compte d'épargne</b> (ESA) et comptes d'épargne santé (HSA) fournir	Indépendants de prévoyance comptes individuels de retraite (IRA), <b>compte d'épargne éducation</b> (ESA) et comptes d'épargne-santé (HSA)

		returns to their respective account holders.	divers renvoie à leurs titulaires de compte respectifs.	fournir divers renvoie à leurs titulaires de compte respectifs.
<b>Misspelled terms</b>				
shortern	réduire	In order to determine whether you can afford any of the low cost mortgages which are available, it is essential that an online calculator is used to determine how much should be borrowed and whether it is necessary to <b>shortern</b> or extend the term of the agreement.	Afin de déterminer si vous pouvez vous le permettre tout du faible coût hypothèques qui sont disponibles, il est essentiel qu'un calculateur en ligne est utilisé pour déterminer quel devrait être emprunté et s'il est nécessaire de <b>shortern</b> ou étendre la durée de l'accord.	Afin de déterminer si vous pouvez vous le permettre tout du faible coût hypothèques qui sont disponibles, il est essentiel qu'un calculateur en ligne est utilisé pour déterminer quel devrait être emprunté et s'il est nécessaire de <b>réduire</b> ou de proroger la durée de l'accord.

### Brand lists

The table below contains some examples of brand names which were either partially or completely mistranslated in the non-customized translation of the data. Since these brand names should not have been translated, they were added to the Brand list in order to be rendered correctly during the translation process.

The structure of the table is the following:

- **Brand** column: contains some of the brands which were mistranslated in the translation without a Brand list.
- **Source text** column: contains the original context in which the brand was found.
- **Translation without Brand list** column: contains the translation of the original segment, to which no Brand list was applied.
- **Translation with Brand list** column: contains the translation of the original segment, to which the Brand list was applied.

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

Brand	Source text	Translation without Brand list	Translation with Brand list
JLL Partners	It was rescued by its parent company, <b>JLL Partners</b> , who pumped in \$100 million, and brought it back to life.	Il a été sauvé par sa société mère, <b>Jll partenaires</b> , l'OMS pompé dans 100 millions de dollars, et il l'a ramenée ensuite à la vie.	Il a été sauvé par sa société mère, <b>JLL Partners</b> , qui pompée à 100 millions de dollars, et le ramena à la vie.
Mountain West Bank	Rush now and open an account with <b>Mountain West Bank</b> within a span of ten minutes to enjoy its multitudinous benefits.	Rush maintenant et ouvrir un compte de <b>banque ouest de montagne</b> dans une période de dix minutes une multitude de jouir de ses avantages.	Rush maintenant et ouvrir un compte avec <b>Mountain West Bank</b> dans une période de dix minutes une multitude de jouir de ses avantages.
First National Bank (FNB)	All about: <b>First National Bank (FNB)</b> online	Tous les propos de : <b>première banque nationale (FNB)</b> en ligne	Tout savoir sur: <b>First National Bank (FNB)</b> en ligne

For Overblog the Term and Brand list created had the following distribution:

Number of Term List entries	64
Number of Brand List entries	13
Total entries	<b>77</b>

### 7.2.2.6 Assessing the impact of the Term and Brand lists

Just like in the case of the LW customer project, the impact of the Term and Brand lists on the machine translated data from the Flavius project was measured by means of a sentence evaluation.

The evaluation consisted of the same steps:

- The original text (in English), together with the two translations (into French) was set up as a sentence evaluation job containing 100 source sentences.
- Again, 2 persons were asked to evaluate the quality of the translations using a **blind-evaluation** methodology: the display of the two contrasting translations was set up such that the identity of the engines that produced the translations was both hidden and randomized from one screen to the next.

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

- Each evaluator had to read each segment (the original text plus the 2 translations) and assign to each translation a score from 1 to 5.
- The scale used for the evaluation was the Likert scale, on which 1 is the lowest score and 5 is the highest score.

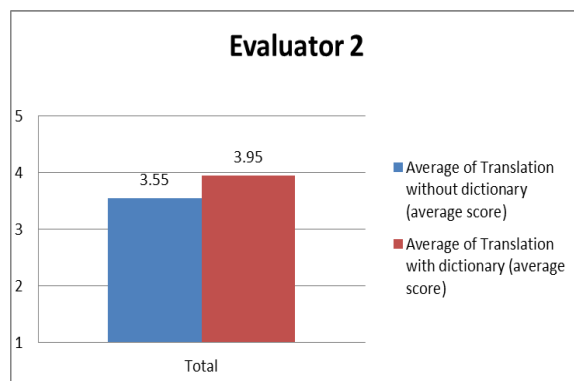
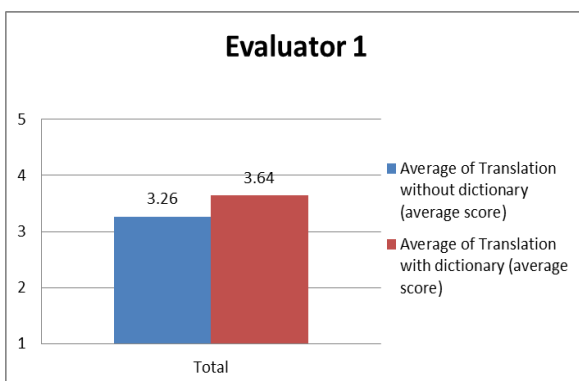
**Scoring Guidelines:**

- **5** - The document is **understandable** and **actionable**, with **all critical information accurately transferred**. **Nearly all of the text** is well translated.
- **4** - The document is **understandable** and actionable, with **most critical information accurately transferred**. **Most of the text** is well translated.
- **3** - The document is **not entirely understandable** but it is **actionable**, with **some critical information accurately transferred**. The text is stylistically or grammatically odd. **Some of the text** is well translated.
- **2** - The document is **possibly understandable** and **actionable** given enough context and/or time to work it out, with **some information accurately transferred**.
- **1** - The document is **not understandable** and it is **impossible to understand** the information it contains.

- The score assigned had to reflect the level of usability of the translation (i.e. if the translation could be useful to someone who only speaks French).
- Once the evaluations were finished, the results obtained were then analyzed to see if the quality of the translations had been improved or not with the help of the Term and Brand lists.

The graphs below show the results of the evaluation performed by the 2 evaluators. One chart has been generated per evaluator, displaying the average of the scores assigned to the translations with and without dictionary entries.

***Evaluator 1 & Evaluator 2 results***



Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services



- The blue column shows the average score assigned by the evaluators to the translations without Term and Brand lists. The average for the set of 100 sentences is **3.26 (Evaluator 1)** and **3.55 (Evaluator 2)** on the 1 to 5 Likert scale.
- The red column shows the average score assigned by the evaluators to the translations with Term and Brand lists. The average score for the set of 100 sentences is **3.64 (Evaluator 1)** and **3.95 (Evaluator 2)** on the 1 to 5 Likert scale.
- The difference in the average scores assigned to the two translations is **+0.38 points (Evaluator 1)** and **+0.40 points (Evaluator 2)** in favor of the customized translation. This quality gap can be better understood in terms of the human evaluators preferring one system’s translations versus the other: for a +0.3 Likert difference between systems A and B, human evaluators indicate that, on average, the translations of system A are better or equal than the translations of system B **75%** of the time.

#### 7.2.2.7 Overblog: Conclusions

---

The results of the test case based on data from the Flavius project show that Term and Brand lists lead to an improvement of the translation quality. The average scores assigned to the translations with dictionary are clearly higher than the ones without dictionary. From the graphs presented above it is obvious that both evaluators (who evaluated the data independently from each other) considered that the quality of the translations with dictionary was indeed better by a difference of 0.39 points on the Likert scale on average.

Nevertheless, the quality improvement was lower than in the case of the LW Customer. There are several reasons which explain the difference in the gap of quality improvement:

- The Overblog data consisted of complex sentences, in which context played an important role. By replacing a number of words with the dictionary entries, grammatical structures can be affected, at times causing the language to sound unnatural in the case of customized translations
- There were considerably fewer translation problems in terms of structures and words/phrases as opposed to LW Customer. It was therefore more difficult to identify the terms with the highest impact on the translation quality to include them in the

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

term/brand list. The data from Overblog belonged to the financial domain – a domain which is part of our baseline training data – and this translated into an overall good quality of the baseline translations (as indicated by both evaluators to be on average at 3.41 on the Likert scale).

**Note:**

*Language Weaver baselines are conventionally trained on data belonging to the following ten domains: Blog, Business, ChatSms, Ecom, Entertainment, Health, Regional News, SciTech, Sport, World News*

- The data was less rich in repetitions as compared to the case of the previous customer.

### 7.2.3 Final conclusions Experiment 1: Test on Training

After the description of the two use cases, the following conclusions can be drawn:

The two experiments were very similar in terms of workflow: almost the same steps were followed in order to assess the impact of the customized translation with Term and Brand as compared to the baseline translation with no customization.

However, the difference in the results obtained was determined by the type of data which had to be translated. The type of data used for translation is therefore a decisive factor that should be always taken into consideration when considering to opt for dictionary customization.

For LW customer we had a sub-set of parallel data from which we could collect parallel terms to include in the Term and Brand List, which optimized the dictionary creation process as opposed to Flavius where we only worked with a monolingual set of data.

The success of the LW customer experiment was determined by the structure of the data that was translated: short product descriptions consisting of almost the same terminology; as a result, dictionary entries were safely used, since it was very unlikely that they affect the structure of the segment.

On the other hand, the experiment based on Flavius data showed a slightly lower improvement in the translation quality with Term and Brand lists. This can be explained by the fact that the data was highly complex and unique: fewer words and phrases with high frequency as compared to LW Customer.

In conclusion, the Term and Brand list functionality can improve considerably the quality of the translations, as long as a number of factors are taken into consideration.

For best results, Term list entries should consist of:

- words with only one translation equivalent and with high frequency ;
- adding sentences to the dictionary (composed of at least 2 words) is always preferable;
- if possible, dictionary entries should be context-independent (adding context-dependent words to the dictionary might result in mistranslations).

The Brand list should contain proper nouns which are relevant for a specific customer/domain.

**Shortcomings of the current experiment: Test on training**

The shortcoming of the current approach was that we tested on data that we had analyzed as well, without isolating a blind set prior to the analysis step, to see how the created dictionary would handle fresh data from the same domain.

The goal of the current experiment was to see if in a very controlled manner, we can get a preliminary assessment of the impact of the customized translation as opposed to the baseline translation. Even without being able to assess the impact of the currently created dictionaries on a blind data set, the results of the current approach have that advantage of making proof of the highest quality threshold that can be attained when using customized dictionary translation for these 2 particular customers.

**Maximum quality threshold of dictionary customized translation:**

We have thus learned that for LW Customer, we can achieve up to 1 point of quality improvement on the Likert scale. From an average of 2.5 (possibly useful text) for LW Customer in the case of baseline translation, we made a quality jump to 3.5 (understandable text, mostly well translated and generally useful), thus to publishable quality (without any further human intervention) when using customized dictionary translation.

For Flavius, we can expect based on this first experiment a quality improvement of up to half a point on the Likert scale, 0.40 to be more precisely. As opposed to LW Customer, the baseline translation of Overblog data in the Flavius experiment was already much better (already at quality 3) which did not require the degree of customization that LW Customer needed.

We can also conclude that the approach of human evaluation of baseline translation with a goal to assess overall quality and catch possible translation problems and create dictionary entries out of these can work for small-medium data sets but cannot be scaled to larger volumes of data.

To correct the shortcomings of the current experiment, we decided to run a second experiment in which we want to assess the impact of the dictionary created in the first experiment on a new blind set of data.

### 7.3 Experiment 2: Test on New Blind Set of Data

The second experiment has as main goal to measure the impact of the currently created dictionaries on a new set of data for both customers described in the previous experiment.

We received for both customers a set of fresh data, different from the one analyzed in Experiment 1 but from the same domain and on the same subject matter.

In order to measure the impact of the dictionary, we have used an internal tool to assess the coverage of the dictionary created in Experiment 1 on the newly received data. This tool was created in-house to measure term/brand list coverage on the customer data so as to be able to assess utility of the dictionary. The tool was designed specifically for these projects but we plan to use it as a preliminary analysis of customer-generated dictionaries to assess the utility of those dictionaries on the customer data sets.

**Note:**

*The “Dictionary Coverage tool” measures the number of hits from the dictionary list that were found in the new set of data. We therefore obtain a report with the frequency/term and the numbers of segments in which the terms were found.*

The measurements show that there is coverage for both customers but with significant differences which can be accounted to the intrinsic differences between the two customers: volume of data analyzed, the number of terms and brands created and the type of segment structure for each customer data.

### 7.3.1 Case Study: LW Customer

In this first case study we have reused the Term and Brand List created in Experiment 1.

For LW Customer the dictionary created in the previous experiment had the following distribution:

Number of Term List entries	454
Number of Brand List entries	281
Total entries	<b>735</b>

Nothing was added or subtracted from the dictionary created in the previous experiment.

#### 7.3.1.1 Test set: new data

---

From the new data received we selected a sub-set which consisted of 200 segments with a total word count of 5789 words. The data consisted of product ads of different watches and customer descriptions of watch products.

The unit of analysis was preserved in this second experiment, namely the segment consisting of one product ad.

#### 7.3.1.2 Dictionary coverage test

---

The “Dictionary Coverage tool” executed a look-up of each term in the new data set and counted the number of segments that had at least one dictionary hit per segment. The tool identified a number 196 unique segments which had at least one hit per segment from the originally created dictionary.

In the table below you can see that the dictionary created in Experiment 1 has a coverage rate of 98% in the case of the new data from the same customer. Preliminary results looked promising but this does not necessarily mean that the customized translation has improved significantly, that is why the next step of this experiment was to set up a human sentence evaluation as described in Experiment 1 to assess the quality of the translation.

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

Type of data	Number of sentences
New set of data	200
Dictionary coverage	196
Coverage rate of Term and Brand List	98%

### 7.3.1.3 Assessing the impact of the Term and Brand lists

The actual impact of applying Term and Brand lists was assessed as in the previous experiment with the 2 translations analyzed in parallel. From the set of 196 segments with dictionary hits we selected 150 segments that were packaged as a human sentence evaluation job. The evaluation consisted of the following steps:

- The original text, together with the two translations was set up as a sentence evaluation job containing 150 segments (product ads).
- Two persons were asked to evaluate the translations using a **blind-evaluation** methodology: the display of the two translations was set up such that the identity of the engines that produced the translations was both hidden and randomized from one screen to the next.
- Each evaluator had to read each segment (the original text plus the 2 translations) and assign to each translation a score from 1 to 5.
- The scale used for this evaluation was the Likert scale, on which 1 is the lowest score and 5 is the highest score.

#### Scoring Guidelines:

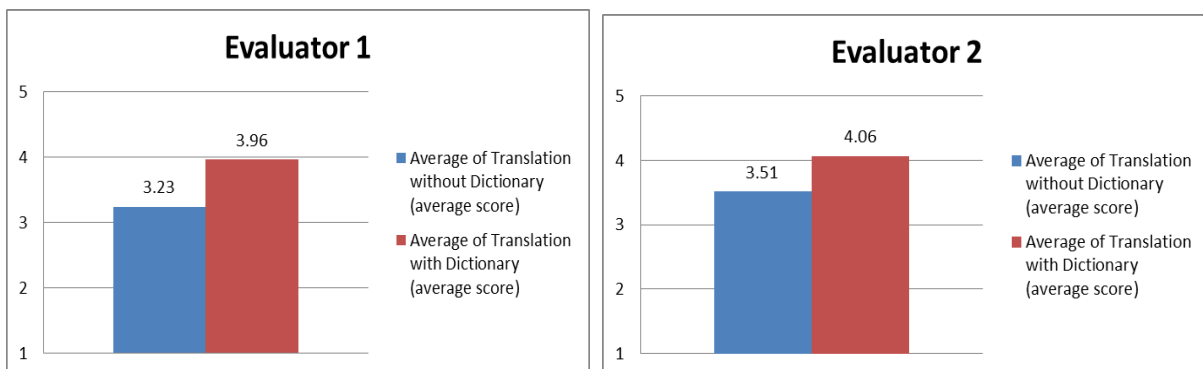
- **5** - The document is **understandable** and **actionable**, with **all critical information accurately transferred**. **Nearly all of the text** is well translated.
- **4** - The document is **understandable** and **actionable**, with **most critical information accurately transferred**. **Most of the text** is well translated.
- **3** - The document is **not entirely understandable** but it is **actionable**, with **some critical information accurately transferred**. The text is stylistically or grammatically odd. **Some of the text** is well translated.
- **2** - The document is **possibly understandable** and **actionable** given enough context and/or time to work it out, with **some information accurately transferred**.
- **1** - The document is **not understandable** and it is **impossible to understand** the information it contains.

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

- The score assigned had to reflect the level of usability of the translation (i.e. if the translation could be useful to someone who only speaks the language that the text was translated into).
- Once the evaluations were finished, the results obtained were then analyzed so as to see if the quality of the translations had been improved with the help of the Term and Brand lists.

The graphs below show the results of the evaluation performed by the 2 evaluators. One chart has been generated per evaluator, displaying the average of the scores assigned to the translations with and without dictionary entries.

### Evaluator 1 & Evaluator 2 results



- The blue column shows the average score assigned by the evaluators to the translations without Term and Brand lists (baseline translation). The average for the set of 150 sentences is **3.23 (Evaluator 1)** and **3.51 (Evaluator 2)** on the 1 to 5 Likert scale.
- The red column shows the average score assigned by the evaluators to the translations with Term and Brand lists. The average for the set of 150 sentences is **3.96 (Evaluator 1)** and **4.06 (Evaluator 2)** on the 1 to 5 Likert scale.
- The difference in the average scores assigned to the two translations is **0.72 point (Evaluator 1)** and **0.55 points (Evaluator 2)** in favor of the customized translation. Both evaluators considered that there is a quality gap between the baseline translation and the dictionary translation of **0.64** points on average. This quality gap can be better understood in terms of the human evaluators preferring one system's translations versus the other: for a +0.6 Likert difference between systems A and B, human evaluators indicate that, on average, the

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

translations of system A are better or equal than the translations of system B 85% of the time.

### 7.3.1.4 LW Customer: Conclusions

---

The results of the LW Customer experiment show a positive impact of using Term and Brand lists on a new fresh set of data. The average scores assigned to the translations with dictionary are higher than the ones without dictionary. Both evaluators (who worked independently from each other and without know what engine produced what segments) considered that the quality of the translations with dictionary was on average **0.64** points higher on the Likert scale.

However, we can note that the improvement is less significant than the one of the initial experiment, as it was to be expected, but it seems that humans do feel inclined to prefer the customized translation to the detriment of the baseline for this particular customer. We can notice an increase from an average of 3.37 on Likert for the baseline translation to a quality of 3.92 on Likert scale for the customized translation.

### 7.3.2 Case Study: Overblog

In this case study we have reused the Term and Brand List created in Experiment 1.

For Overblog the dictionary created in the previous experiment had the following distribution:

Number of Term List entries	64
Number of Brand List entries	13
Total entries	<b>77</b>

Nothing was added or subtracted from the dictionary created in the previous experiment.

#### 7.3.2.1 Test set: new data

---

From the new data received we selected a sub-set which consisted of 5,836 segments with a total word count of 97,194 words. The data consisted of financial articles posted on Overblog.

The unit of analysis was preserved in this second experiment as well, namely the segment consisting of one sentence.

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services



### 7.3.2.2 Dictionary coverage test

---

The “Dictionary Coverage tool” executed a look-up of each term in the new data set and counted the number of segments that had at least one dictionary hit per segment. The tool identified a number 268 unique segments which had at least one hit per segment from the originally created dictionary.

In the table below you can see that the dictionary created in Experiment 1 has a very poor coverage rate of only 5% in the case of the new data set from the same customer. Preliminary results look less promising but this does not necessarily mean that the customized translation has not improved in the cases where the dictionary had coverage that is why the next step of this experiment was to set up a human sentence evaluation as described in Experiment 1 to assess the quality of the translation.

Type of data	Number of sentences
New set of data	5836
Dictionary coverage	268
Coverage rate of Term and Brand List	5%

### 7.3.2.3 Assessing the impact of the Term and Brand lists

---

The actual impact of applying Term and Brand lists was assessed as in the previous experiment with the 2 translations analyzed in parallel. From the set of 268 segments with dictionary hits we randomly selected 150 segments that were packaged as a human sentence evaluation job. The evaluation consisted of the following steps:

- The original text, together with the two translations was set up as a sentence evaluation job containing 150 segments (sentences).
- Two persons were asked to evaluate the translations using a **blind-evaluation** methodology: the display of the two translations was set up such that the identity of the engines that produced the translations was both hidden and randomized from one screen to the next.

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

- Each evaluator had to read each segment (the original text plus the 2 translations) and assign to each translation a score from 1 to 5.
- The scale used for this evaluation was the Likert scale, on which 1 is the lowest score and 5 is the highest score.

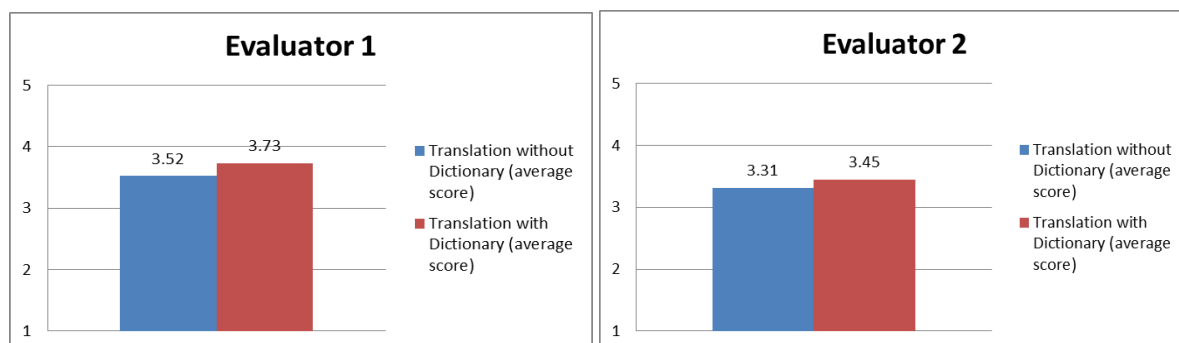
**Scoring Guidelines:**

- **5** - The document is **understandable** and **actionable**, with **all critical information accurately transferred**. **Nearly all of the text** is well translated.
- **4** - The document is **understandable** and actionable, with **most critical information accurately transferred**. **Most of the text** is well translated.
- **3** - The document is **not entirely understandable** but it is **actionable**, with **some critical information accurately transferred**. The text is stylistically or grammatically odd. **Some of the text** is well translated.
- **2** - The document is **possibly understandable** and **actionable** given enough context and/or time to work it out, with **some information accurately transferred**.
- **1** - The document is **not understandable** and it is **impossible to understand** the information it contains.

- The score assigned had to reflect the level of usability of the translation (i.e. if the translation could be useful to someone who only speaks the language that the text was translated into).
- Once the evaluations were finished, the results obtained were then analyzed to assess the impact of the dictionary customized translation.

The graphs below show the results of the evaluation performed by the 2 evaluators. One chart has been generated per evaluator, displaying the average of the scores assigned to the translations with and without dictionary entries.

**Evaluator 1 & Evaluator 2 results:**



- The blue column shows the average score assigned by the evaluators to the translations without Term and Brand lists (baseline translation). The average for the set of 150 sentences is **3.52 (Evaluator 1)** and **3.31 (Evaluator 2)** on the 1 to 5 Likert scale.
- The red column shows the average score assigned by the evaluators to the translations with Term and Brand lists. The average for the set of 150 sentences is **3.73 (Evaluator 1)** and **3.45 (Evaluator 2)** on the 1 to 5 Likert scale.
- The difference in the average scores assigned to the two translations is **0.21 point (Evaluator 1)** and **0.13 points (Evaluator 2)** in favor of the customized translation. Both evaluators considered that there is a quality gap between the baseline translation and the dictionary translation of **0.17** points on average. This quality gap can be better understood in terms of the human evaluators preferring one system’s translations versus the other: for a +0.1 Likert difference between systems A and B, human evaluators indicate that, on average, the translations of system A are better or equal than the translations of system B 66% of the time.

#### 7.3.2.4 Overblog: Conclusions

---

The results of the human sentence evaluation for the type of data analyzed in the Flavius experiment shows a positive impact of using Term and Brand lists on a new blind set of data. The average scores assigned to the translations with dictionary are slightly higher than the ones without dictionary. Both evaluators (who worked independently from each other and without knowing what engine produced what segments) considered that the quality of the translations with dictionary was on average 0.17 points higher on the Likert scale

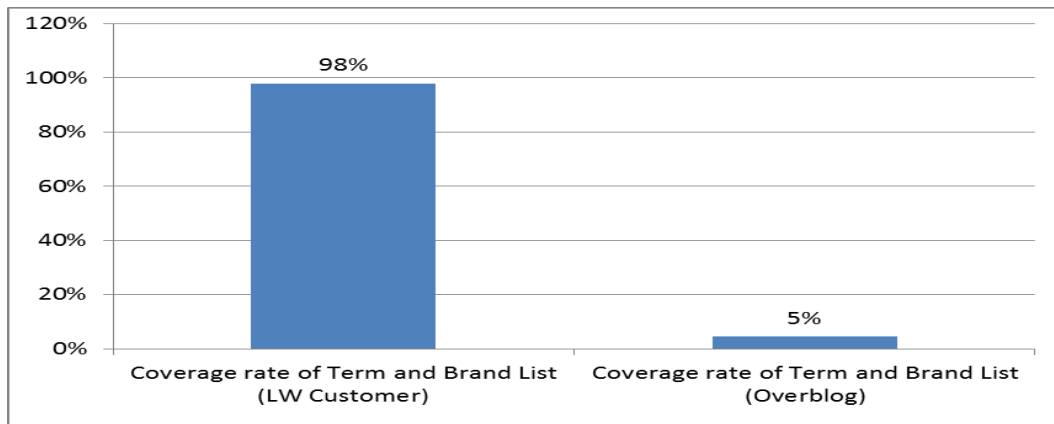
We can note that the improvement is less significant than the one of the initial experiment, as it was to be expected, but it is positive as it seems that humans do feel inclined to prefer the customized translation to the detriment of the baseline for this particular customer.

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

### 7.3.3 Final conclusions Experiment 2: Test on New Blind Set of Data

The dictionary coverage indicated that there is a significant difference in terms of coverage per customer as shown in the table below:

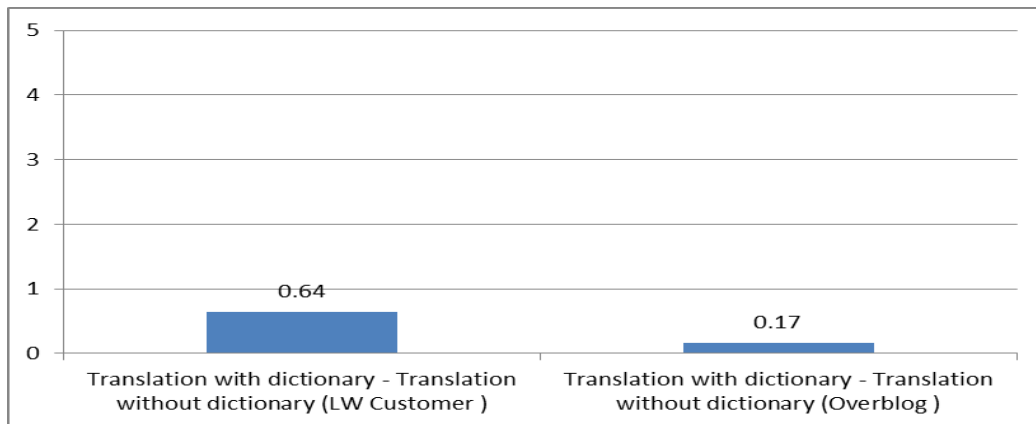
**Coverage rate of Term and Brand List for LW Customer and Overblog:**



While for the LW Customer we had 98% dictionary coverage, for Overblog we only had 5% coverage. The reason for this gap is mainly due to the difference in type of data the customer uses: LW Customer uses small product descriptions rich in repetitions, while the other makes use of entire sentences, with few repetitions. The difference in coverage can also be due to the size of the dictionary: for the LW Customer we had 735 terms and brands (in part collected from baseline translation analysis and in part from the parallel data), for Overblog we only had 77 entries. Due to the high degree of internal segment repetitions in the data of LW Customer, the rate of coverage is significantly higher than for Overblog where incorrect terminology was less frequent and thus we only managed to have a 5% coverage.

The quality gap as indicated by humans is also different between the 2 cases: while for LW Customer, the quality gap in terms of the utility and actionability of the document is 0.64, for Overblog, the perceived quality gap only seems to be at 0.17.

**Quality improvement for LW Customer and Overblog:**



We can also notice that the results in terms of quality improvement on this new set of data do not rise to the maximum threshold identified in the test on training (Experiment 1), nevertheless, the results are positive. We can see that evaluators evaluating sentences from a blind set in both cases scored higher the customized translation as compared to the baseline translation.

As a direct result of these two experiments and the two different case studies, and based on our previous experience in working with Term and Brands, we have crystalized a document with guidelines to be followed by the users of the FLAVIUS platform, in order to achieve best results when using these 2 features. Please see the attached “**Appendix A – WP4 - Term and Brand Management Guide**”.

Taking into consideration all these items will ensure the best and safe use of the Term and Brand list functionality. The document will continue to be improved once more experiments are completed. The details provided by LW in the attached document will be merged with additional guidelines provided by Softissimo that will reflect rules and restrictions from the FLAVIUS interface perspective and the final document will be available for FLAVIUS customers.

**Note:**

***Any termlist uploaded should be a termlist made for the purpose of the MT system. A found termlist is probably contraproductive.***

**Shortcomings of current experiment:**

The results of the 2<sup>nd</sup> experiment show that the increase in quality measured on the test on training between the baseline translation and the dictionary customized translation, the upper bound, was not achieved on a new set of data belonging to the same customer/domain as expected.

Furthermore, we can see that the method of dictionary creation (test on training) does not ensure sufficient coverage for a customer that does not have highly repetitive data as it was the case of LW Customer, nor does it seem to be efficient in the case of a large volume of data.

The lower impact on blind data is also due to the fact that the dictionary entries were created uninformed by dictionary managers having any knowledge of what words are not part of the knowledge that is already used by the MT engine.

Such a method of analysis can be safely used in cases of small sets of data or at a very preliminary stage of the dictionary customization to ensure a good understanding of the customization process. However, in case of customers with higher volumes of data, such as Overblog, we think it is important to start the dictionary creation effort equipped with a set of tools that will ensure increased coverage for improved results.

This is why we have currently started a 3<sup>rd</sup> experiment in which we will use terminology extraction tools in the dictionary creation effort to analyze words and phrases that are eligible for dictionary entries from a customer data and the resulting dictionary would be put to the test on a blind set isolated upfront from the customer data.

## 7.4 Experiment 3: Test on Blind Set

This scenario will follow a use case for some of the medium to large FLAVIUS customers, that will have to automate, as much as possible the process of creation the Term and Brand Lists, due to large amounts of data that have to be processed.

Experiment status: *work in progress*.

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

### 7.4.1 Action plan

#### 7.4.1.1 Consolidate the customer data set

We will start the analysis from a consolidated data set for both customers as we have received a new set of data. From the consolidated set, we will isolate a set of segments: product ads for LW customer and sentences for Overblog.

The consolidated data set distribution is presented in the table below:

Customer data	Number of Segments	Number of words
<b>Consolidated Customer data set (Overblog)</b>		
<i>Monolingual data</i>	21559	315103
<b>Consolidated Customer data set (LW Customer)</b>		
<i>Monolingual data</i>	1933	35028
<i>Bilingual data</i>	2714	34953 (word count for both languages)

#### 7.4.1.2 Isolate from the customer data a blind test set

	Number of Segments	Number of words
<b>Isolated blind set from Consolidated Customer data set (Overblog)</b>	2918	48525
<b>Isolated blind set from Consolidated Customer data set (LW Customer)</b>	200	5721

This project is characterized by the fact that we isolate a blind test set from the customer data set upfront. We have also adopted a different working methodology for a use case of a customer who has a large volume of data and who opts for dictionary customization. We will try to have a semi-automated/statistical approach to data analysis to ensure that we can efficiently generate customer/domain- driven dictionaries from larger volumes of data.

### 7.4.1.3 Unk extraction from the baseline translation of Customer data

---

The unk extraction will be done with an in-house tool which detects the words that remained untranslated (UNKs) and which have a severe impact on the actionability and utility of any segment for a potential target language user. The unk detection tool also identifies the frequency of the unks in translated text which will ensure a correct prioritization of workload of dictionary creation.

Therefore, we will give priority to higher frequency unks so that we can ensure they will become part of the brand and term list and thus get translated correctly.

### 7.4.1.4 Bilingual Term extraction and analysis on high frequency terms and phrases (if parallel data is available)

---

This set will only be done for the LW customer as we have a small set of new parallel data from which we can collect already translated terms.

We cannot do the same for Overblog we do not have parallel data from the financial domain, however, as observed in the previous experiments, the advantage of Overblog is that the data to be translated is part on one of the domains on which the baseline was trained and thus takes profit of an improved quality baseline which would result in less of a dictionary customization effort.

As for the UNKs detection process, priority will be given to high frequency words and phrases.

### 7.4.1.5 Monolingual Term extraction and analysis on high frequency terms and phrases

---

The tools that we plan to use for both the monolingual and bilingual term extraction are:

- **SDL MultiTerm Extract** (a tool that makes use of a statistical approach, based on term frequency). The advantage of using SDL MultiTerm Extract is that it does an automatic mono and bilingual terminology candidate identification and extraction. A percentage score is attributed to each extracted term to indicate the probability of utility of that term as a dictionary/glossary entry.

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services



- **SDL PhraseFinder:** Linguistic approach, based on rules built within the tool. This tool is also equipped with frequency metadata: it logs and reports the frequency of the exported terms enabling the user to generate a corpus-driven dictionary.

Both tools have a concordance search window which enables the Linguist/Language Expert to see the term in context. Morphological inflations are also listed, thus making sure that you can capture all occurrences of a word which can later be turned into dictionary entries.

Both tools enable the user to save hours of time through automated terminology extraction, enabling you to quickly create terms and customer data-driven glossaries/term lists.

There are other freely available tools on the market which perform term extraction and which can be using in a similar way.

We plan to run a:

- Word extraction
- Phrase extraction (min 1 term – max. 5 terms) and analyze the top N frequent ones

to assess if these are good dictionary candidates.

#### 7.4.1.6 Baseline translation analysis of in context extracted terms

We will take the top N extracted words and phrases and X context sentences in which they appear. The context sentences will be translated using BeGlobal baseline translation and a quality assessment will be done on the translated word/phrase in the context of a particular sentence. If the phrase is translated consistently well, we dismiss it as a potential term candidate. However, if we notice that the phrase is not well translated, it becomes an eligible candidate for a term dictionary.

We will be working with reports like the one below:

Extracted term	Target translations	Score/Frequency
source phrase 1	target translation	99%
source phrase 2	target translation	208 occurrences

Once we have a dictionary built from term and UNK extraction, we are ready to do the analysis on the blind set to measure impact.

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

By using the above mentioned approach we hope to obtain better results in a timely manner and to improve coverage, accuracy, usability and overall consistency of translations.

#### 7.4.1.7 Assessing the impact of the Term and Brand lists

We will do a blind human sentence evaluation of the baseline translation, compared to the customized translation.

We will try to do repeated experiments that use the top 100, 200, 400, 800, dictionary entries sorted by their frequency and measure the impact of each this dictionary.

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

## 8. Achievements to date

From of the original three objectives of the Work Package 4 presented in the FLAVIUS Description of Work document on page 49, the first one has been delivered by Language Weaver in June 2010. The delivery of the second objective is a shared effort between Language Weaver (infrastructure) and Softissimo (integration of the infrastructure into the User Interface):

- **Create interfaces to customize translation engines**

The first step in the delivery of the infrastructure for the customization of the translation engines has been achieved by the Delivery 4.5 of the FLAVIUS project:

### **D4.5 SMT customization module (through dictionary)**

- SMT systems can be customized through dictionaries.

The research we have performed and detailed in this document has confirmed the utility of the dictionaries in customizing the translation output by:

- Improving the quality of the SMT output from +0.1 up to +1.08 points on Likert scale.
- Not degrading the MT output, if used correctly. Guidelines are presented in

#### **Appendix A.**

The component tests performed ensure the robustness of our delivered production ready systems. We build for scale and reliability and our systems, both hardware and software, are ready to handle the needs of the FLAVIUS customers. More tests will be performed as the FLAVIUS platform will continue to integrate our systems with the other components.

This milestone is part of the larger “MS4 - FLAVIUS Platform offers customization interfaces”. This objective and milestone has been achieved with the delivery of the Term and Brand Management API.

The team at Language Weaver has been actively working on the Delivery 4.6 (part of the same MS4 listed above):

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

#### D4.6 SMT customization module (through retraining)

We have performed research in this direction to validate the utility of the trained engines in the customization of the translation output. Our research is detailed in **Appendix B** and will be part of the D4.6.

The research confirmed the following:

- With a small set of data (about 100,000 words) the trained system is producing an improved translation measured by a +0.1 increase on the Likert scale and +1.86 increase in BLEU points
- With a larger data set (about 500,000 words) the trained engine is producing an improved translation measured by a +0.3 increase on Likert scale and +4.60 increase in BLEU points

The work for this delivery will continue to focus on implementing the infrastructure needed for the training customization.

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

## 9. Appendix A – WP4 – Term and Brand Management Guide

Complete details and best practices to follow in creating a Term list or a Brand list can be found in the attached document. These guidelines will be merged with the restrictions and guidelines provided by Softissimo from the User Interface perspective and the resulted document will be available for the FLAVIUS customers online.

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services

## 10. Appendix B – WP4 – Customization through retraining

A research study has been performed on Customization through retaining with the data provided by TVTrip, one of the FLAVIUS partners. The research results will be included with the Delivery 4.6 - SMT customization module (through retraining).

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
C	Confidential, only for members of the consortium and the Commission Services