FP7-ICT-2011.4.2
Contract no.: 288342
www.xlike.org

Deliverable D3.3.1

Early machine translation based semantic annotation prototype

Editor:	Marko Tadić, UZG
Author(s):	Marko Tadić, UZG; Božo Bekavac, UZG; Željko Agić, UZG; Matea Srebačić, UZG; Daša Berović UZG; Danijela Merkler, UZG.
Deliverable Nature:	Prototype (P)
Dissemination Level: (Confidentiality)	Public (PU)
Contractual Delivery Date:	M21
Actual Delivery Date:	M24
Suggested Readers:	All partners using the XLike Toolkit
Version:	1.0
Keywords:	machine translation, semantic annotation, Cyc ontology

Disclaimer

Full Project Title:	XLike– Cross-lingual Knowledge Extraction
Short Project Title:	XLike
Number and Title of Work package:	WP3 – Cross-lingual Semantic Annotation
Document Title:	D3.3.1 Early machine translation based semantic annotation prototype
Editor (Name,Affiliation)	Marko Tadić, UZG
Work package Leader (Name, affiliation)	Achim Rettinger, KIT
Estimation of PM spent on the deliverable:	5 PM

Executive Summary

The main goal of the XLike project is to extract knowledge from multi-lingual text documents by annotating statements in sentences of a document with a cross-lingual knowledge base. The purpose of the early machine translation based semantic annotation prototype described here, is to investigate the whether the SMT systems could be used to translate from natural language into a formal language. This translation would then be used as the semantic annotation of a natural language sentence. We have described the experiment using the Moses SMT system suite and presented the evaluation of results.

Table of Contents

List of Figures

List of Tables

Abbreviations

SL	Source Language
TL	Target Language
IL	Interlingua
NL	Natural Language
FL	Formal Language
NL2FL	Natural Language to Formal Language
MT	Machine Translation
SMT	Statistical Machine Translation
L	Language
TM	Translation Model
LM	Language Model

Definitions

Parallel Corpus	Parallel corpus consists of documents that are translated directly into different languages.
Comparable Corpus	Comparable corpus, unlike parallel corpora, contains no direct translations. Overall they may address the same topic and domain, but can differ significantly in length, detail and style.
Source language	Language of the text that is being translated.
Target Language	Language of the text into which the translation is being done.
Formal language	Artificial language that uses formally defined syntax.
Language pair	Unidirectional translation from the SL to TL. Translation from L_a to L_b is one language pair and from L_b to L_a is another language pair.

1 Introduction

In this deliverable we are presenting the results of research leading to the early machine translation based semantic annotation prototype. This part of the project was envisaged and covered by the research plans situated in the WP3, namely T3.3.

1.1 Motivation

The main goal of the XLike project is to extract knowledge from multi-lingual text documents by different means and treating the documents at all possible levels: from the document collection, over documents as unique entities, up to individual paragraphs and sentences that occur in these documents. The knowledge can be formally represented as statements in a formal language, resembling a formal logic calculus or any other semantically rich format (e.g. RDF triples), or as mappings from any of the mentioned levels of processing to a desired conceptual space (e.g. Cyc ontology, Wikipedia, Dbpedia, Linked Open Data, etc.).

Different work packages and the respective tasks within the XLike project examine different approaches to this problem, while the task T3.3 covered in this deliverable is trying to initially investigate how the machine translation techniques could be exploited for cross-lingual semantic annotation.

Then main idea behind this task is to investigate how the use of statistical machine translation (SMT) techniques could facilitate obtaining the mappings between text and its semantic representation(s). The development of this early prototype started from a very simple idea: would it be possible to train a SMT-system to translate from natural language as a source language into a formal language as a target language. The work presented here has been conducted as a proof of concept, i.e. whether this idea, that could be applicable in theory, once turned into a real SMT-system, really produces results usable by humans and/or machines for further processing. In this early prototype we were using the basic capabilities of SMT-systems to train a translation model and target language model, while advanced approaches (factor based) are left for the final version of this approach.

At this point we are investigating the processing and results only at the level of individual, isolated sentences, while the level of translating the whole document will be investigated further in this task and be presented in D3.3.2.

2 Statistical Machine Translation techniques

2.1 General Framework

Starting from the milestone paper by IBM team [1], over the development of GIZA++ tools [2], and the complete SMT system Moses [3], the SMT has gained a serious momentum in the last decade. The availability of open source SMT system Moses facilitated the spread of research in SMT. Also, the availability and collecting of more parallel data (bitexts), predominantly from the web, contributed largely to the increase in maturity of SMT systems.

Unlike the early MT systems that were primarily rule-based and thus highly dependent on the languages involved, translation direction and the quality of SL analysis and TL generation, the current SMT systems are cheaper in demand for human effort and offer broader scale of automation. In their basic incarnation they do not need all levels of linguistic analysis and generation, thus do not include many person months of work by highly skilled experts in order to build the moderate MT system for only one language pair, but in the first run, SMT systems require large amounts of parallel data (bitexts).

General SMT scenario involves collecting the parallel data, aligning them at the sentence level, using that data for training the SMT systems and building a Translation model (TM) for transfer of words and phrases from SL into TL. In order to select between different probable translations and to use the most appropriate (often also more natural) TL text, very large Language Models (LM) are used for the final SMT system output. In Figure 1 a general SMT process is presented as a diagram.

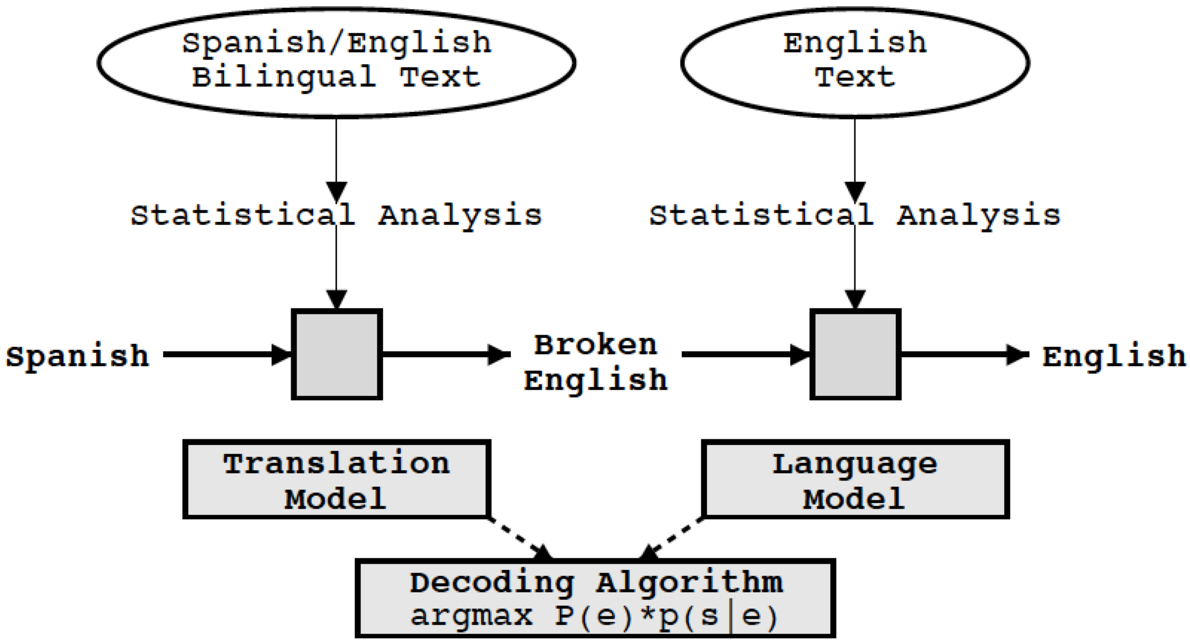


Figure 1. General diagram of a SMT system (from [4])

Generally, the mentioned scenario involves natural language (NL) as both, SL (Spanish) and TL (English). Training is performed using a large Spanish-English parallel corpus and TM is being built. Large English monolingual corpus is used to build (train) LM. Decoder applies the Decoding Algorithm to all TM outputs and uses LM to select as the final output the most probable translation of sentence **s** in TL. This is a SMT process described in a nut shell and all SMT systems so far (including Moses) were adapted for NL as TL.

2.2 Early prototype: proof of concept

The idea behind this task is simple. Since the main goal of XLike project is to build technology for extracting and representing knowledge from the text cross-lingually in a language independent (common) format, preferably formally defined, we suggested that this representation could be written in a formal language. From the Semantic Web community the representation of basic relations in the form of RDF triples has become common way of representing knowledge involving concepts from a conceptual space or an ontology. However, population of conceptual spaces and ontologies with relations from the texts has been complicated, demanding and involved a lot of human effort if it is entirely rule-based (for introduction to semantic knowledge management and procedures of populating ontologies see [5,6]). Wouldn't it be possible to apply analogous shift in methodology here, like it was applied with the change from rule-based MT to statistical MT?

This would involve usage of SMT techniques for automatic translation from natural language into formal language. Theoretically, FL should be easier to generate (or to select between possible translations) since it has:

1. fixed word order: the notorious problem in SMT are TLs with free word order;
2. formal syntax: no syntactic irregularities that usually appear in NL texts, no phrases in TL that have to be treated as single units;
3. no NL morphology: often errors in inflectional endings contribute to lower fluency of TL, they are results of data sparsness problem introduced by the fact that in inflectionally rich languages words appear in different word-forms all belonging to the same lemma and that SMT systems are not always sensitive enough to select the right word-form for a given co-text.

Here we present the first attempt to check whether it would be possible to use SMT system, trained on a parallel corpus consisting of large set of aligned sentence pairs, where one side of the pair is a NL sentence and the other side of the pair is FL "sentence", i.e. statement in a FL. This SMT system should be able to translate from NL sentence into FL "sentence", that in this turn can serve as the knowledge representation of NL sentence. For those who are familiar with older MT systems, this may look like a half-way of the MT system based on interlingua (IL), i.e. like translation from SL to IL only.

However, this theoretical starting point had to be proved, but this was possible only by collecting a large parallel corpus with specific requirements (NL as SL and FL as TL) and by adapting the existing SMT frameworks for a particular TL, namely non-NL output.

An additional argument for such an experiment can be the following. Although the usage of SMT output is primarily intended for humans, it is not the rare case to use the SMT output not just by humans, but also by machines that include the translation obtained in this way in their pipelines for further processing.[1] Even if the result of NL2FL SMT would no be acceptable for immediate usage by humans, it could be usefull to further processing steps by machines.

[1] In the project Let'sMT! the SMT output was used in an industrial case when the SMT translation of Polish, Czech, Slovak stock market reports into English was used by a system that was automatically extracting information on events at these stock markets. See more details at http://www.letsmt.org.

3 CycL as a Target Language

In order to train the NL2FL SMT system we needed a large parallel corpus of aligned NL and FL sentences. Manual annotation of an English monolingual document collection large enough for this purpose, where annotations of NL sentences would be statements in FL, was not an option due to enormous human effort that would have to be invested. However, such a parallel corpus could have been generated from FL side, i.e. from an ontology. We learned that Cyc ontology is capable of generating valid English sentences out of its stored relations, so we asked the consultants from Cyc and partners from IJS to couple these English sentences with their formal representations. Cyc ontology itself uses CycL so the choice of CycL as the FL for our experiment was clear choice.

3.1 Characteristics of CycL

CycL is an ontology language closely connected to Cyc ontology which in turn is the part of Linked Open Data. CycL is the FL used for representing knowledge in Cyc ontology and it is defined as a declarative language based on classical first-order logic (relationships), with additional modal operators and elaborated quantificators. The concept names in CycL are constants and are always denoted by **#$** prefix. Constants cover:

- individuals: such as different NEs (**#$BarackObama**, **#$Australia**);
- collections: **#$Fruit-TheFood** with members of collections appearing as their instances;
- truth functions: they return only true/false answers and can be broken down into logical operators (**#$and**, **#$or**, **#$not**, etc.), quantifiers (**#$forAll**, **#$thereExists**, etc.) and predicates (**#$isa**, **#$genls**, etc.);
- functions: they return individuals or collections (**#$FruitFn** returns fruits from collection of plants provided as an argument).

The following examples illustrate the general characteristics of CycL:

(#$isa #$BarackObama #$UnitedStatesPresident)

Figure 2. Example if **#$isa** predicate

(#$genls #$BabyOil #$BabyToiletrySubstance)

Figure 3. Example if **#$genls** predicate

(#$capitalCity #$Croatia #$Zagreb)

Figure 4. Example of **#$capitalCity** predicate

There were attempts in the Cyc community to translate from text into CycL by using simple syntactic patterns, but the lack of overall precise syntactic analysis of English prevented the large scale application of such approach. We wanted to give it a try with another approach, using the SMT system.

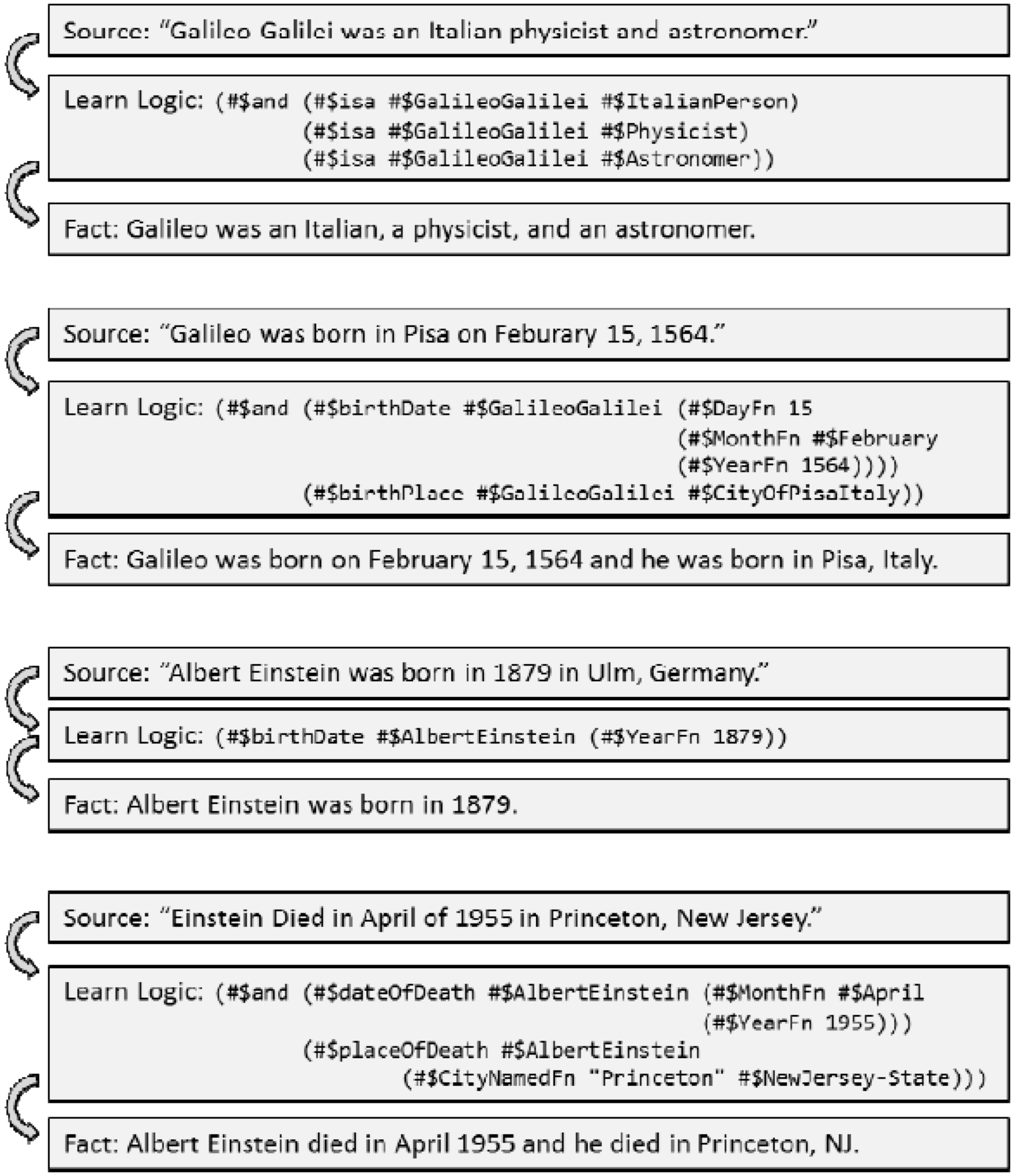


Figure 5. Example of an early approach on how to translate text into logic description using simple syntactic patterns.

3.2 Preparing the training data

Generation of English sentences aligned with FL "sentences" was done by partners from IJS since they operate Cyc ontology as a whole. The first generation run provided 50,000 of aligned English-CycL sentence pairs to check whether the output generated in this format would be suitable for further processing. We noticed that a lot of English sentences were referring to relations between two concepts denoted by their IDs instead by terms in plain English, so we had to filter this output. Also, this amount of sentence pairs was not enough for the experiment since the training data for SMT systems usually run in an order of magniture more.

This filtering process was applied within the second generation run and it yielded 650,000 clean English-CycL aligned sentence pairs. This amount of data represented an acceptable quantity of parallel data for a decent SMT experiment, particularly having in mind the monotonous nature of CycL as TL. The training data were prepared in TMX format, an open XML industry standard format for exchanging parallel data.

```
<tu>
 <tuv xml:lang="en">
  <seg>Zagreb, Croatia's longitude is 16 degrees</seg>
 </tuv>
 <tuv xml:lang="se">
  <seg>(#$longitude #$CityOfZagrebCroatia (#$Degree-UnitOfAngularMeasure 16.0))</seg>
 </tuv>
</tu>
```

Figure 5. Example of training data in TMX format

Out of the prepared 650,000 sentence pairs, a test set of 10,000 sentence pairs was set aside for evaluation purposes.

4 Using Moses

The SMT system we used for this experiment was an open source SMT systems suite Moses[2]. Our initial plan was to use Let'sMT! platform[3], the already existing platform for generating SMT systems out of your own parallel data.

UZG team was a partner in the ICT-PSP project Let'sMT! and there it gained experience not just in training and using SMT systems, but also in putting together the user-oriented platform that allows users to build their own SMT systems out of their own (even private) data and to use these systems publicly or with restricted access.

The Let'sMT! platform provides an open access service for building, runing and using the user's own custom machine translation systems. In practice, the Let'sMT! platform is a web front-end to Moses SMT system suite that allows users to avoid all complicated parametrizations required for tailoring Moses to one's needs. The Let'sMT! platform is also a repository of tranied public or private SMT systems that can be used freely or with restricted access. The Let'sMT! platform is running in an flexible manner on Amazon Grid. This makes it extremely useful because it is the platform that provides the fastest open access facility for training large Translation and Language Models, and all that accessible in an user friendy way.

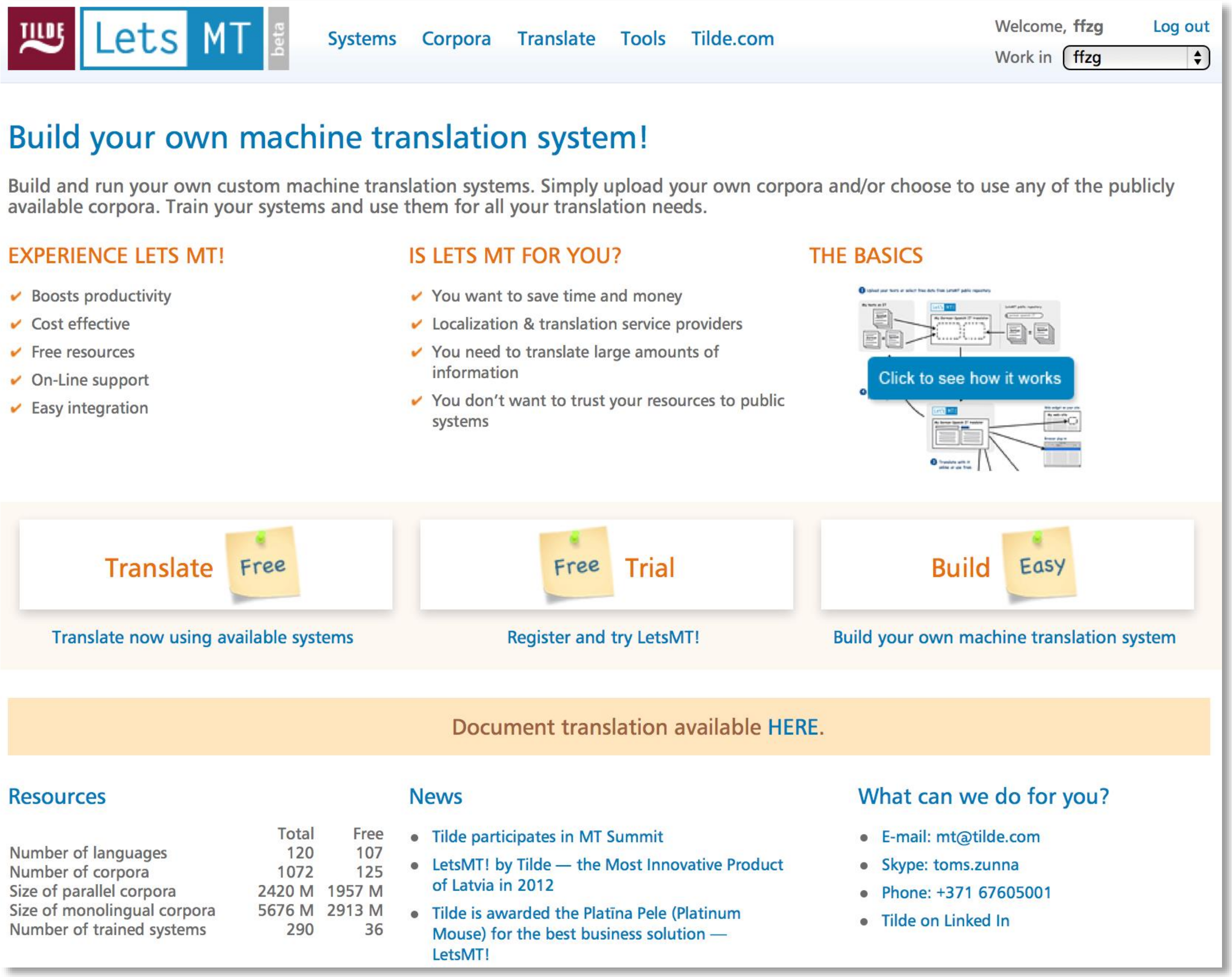


Figure 6. Let'sMT! platform homepage

[2] http://www.statmt.org/moses/

[3] http://www.letsmt.eu, see also about the Let'sMT! project at http://www.letsmt.org.

4.1 Training Moses

The prepared training data were fed into the Let'sMT! platform as a parallel corpus. The system required the setting of language labels for SL and TL from the closed list of existing NLs, so no FLs could have been introduced as TL. Therefore, we had to choose a highly improbable language as a label for CycL and we selected Sami (Northern) with ISO 639-2 code "se". This explains the usage of "se" in TMX and in Let'sMT! platform.

Since the training data were prepared in accordance with the Let'sMT! platform's specification, the platform processed, ingested and stored our parallel corpus regularly.

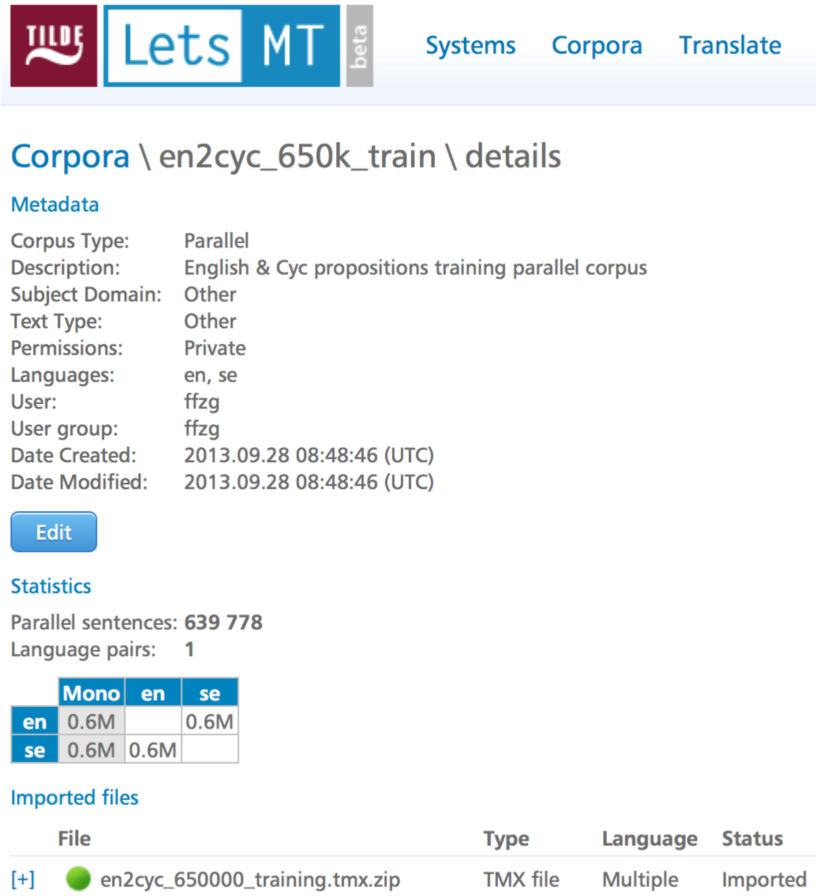


Figure 7. The training data uploaded as a parallel corpus to Let'sMT! platform

However, the initial training attempts were not successful. The training processes were interrupted always at the same point. We had several dozens of attempts to adapt the training data by introducing the smaller number of sentence pairs, or to use different input format (Moses text files instead of TMX). All our attempts were producing errors in training. This lead to a delay of handing out this deliverable from M21 to M24 in agreement with the coordinator and PO.

It was only after the UZG team installed its own instance of Moses, that it was noticed where the source of error originated from. The internal Moses tokenizer was treating the CycL constants as multi-word units. Namely, the string of characters **#$i sa** was tokenized internally as three tokens: #, **$**, and **i sa**. This character sequence separated as two tokens turned # and **$** into the most frequent tokens in the FL part of training data and this resulted in training errors. Also, we detected some errors in TMX formatting (although Let'sMT! platform accepted them) and corrected that as well.

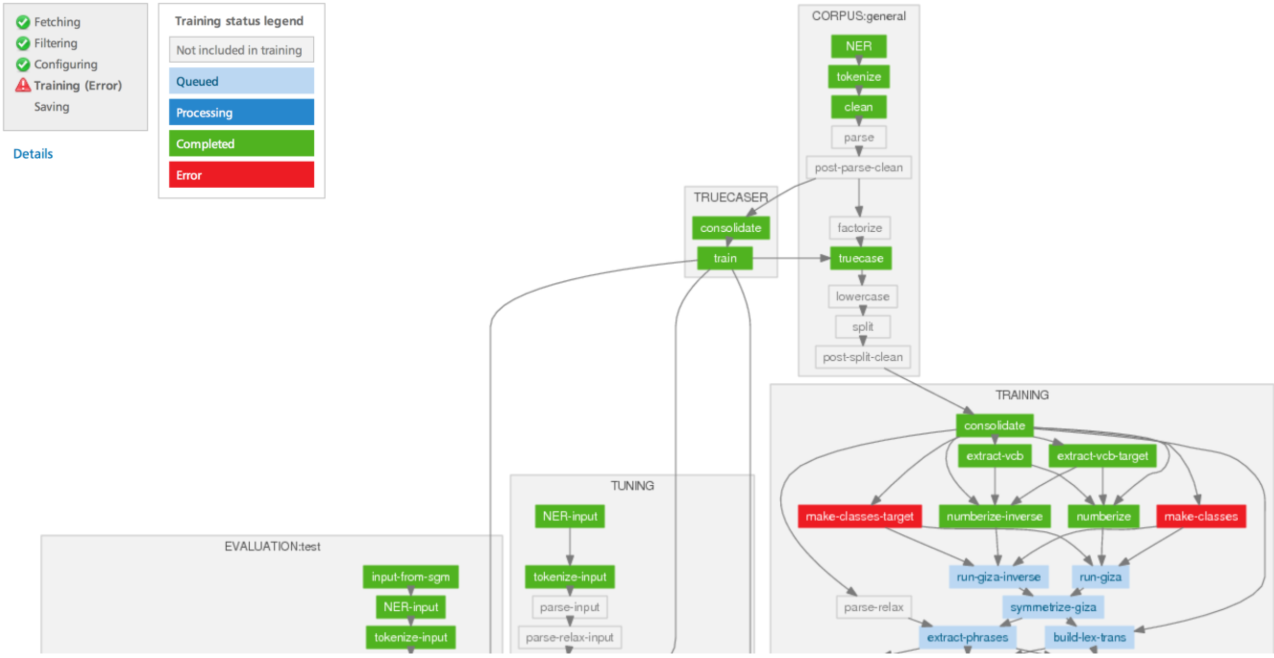


Figure 8. Example of errors in the initial training seen in the training chart

The solution was to adapt the training data instead of redefining the Moses tokenizer. So we simply replaced the sequence **#$** with two UTF-8 characters (**áé**) that were not detected in FL part of training data. After that correction the training went smoothly and yielded the complete **En-EnSemRep-Model02** SMT system capable of translating from English into CycL. However, some of the sentence pairs were discarded by the system during training and we still have to investigate the reason for this.

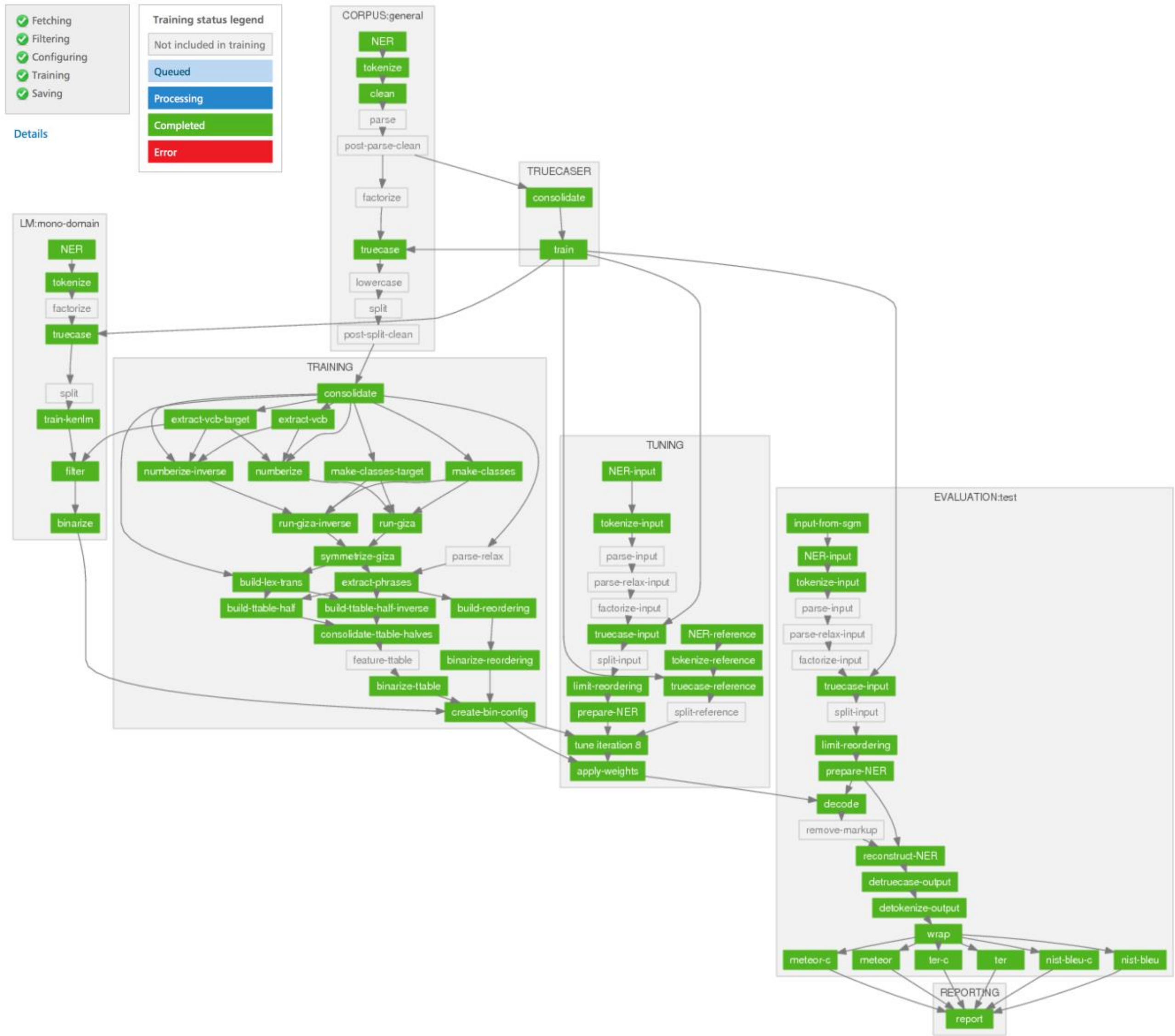


Figure 9. The completed training chart for En-EnSemRep-Model02 SMT system

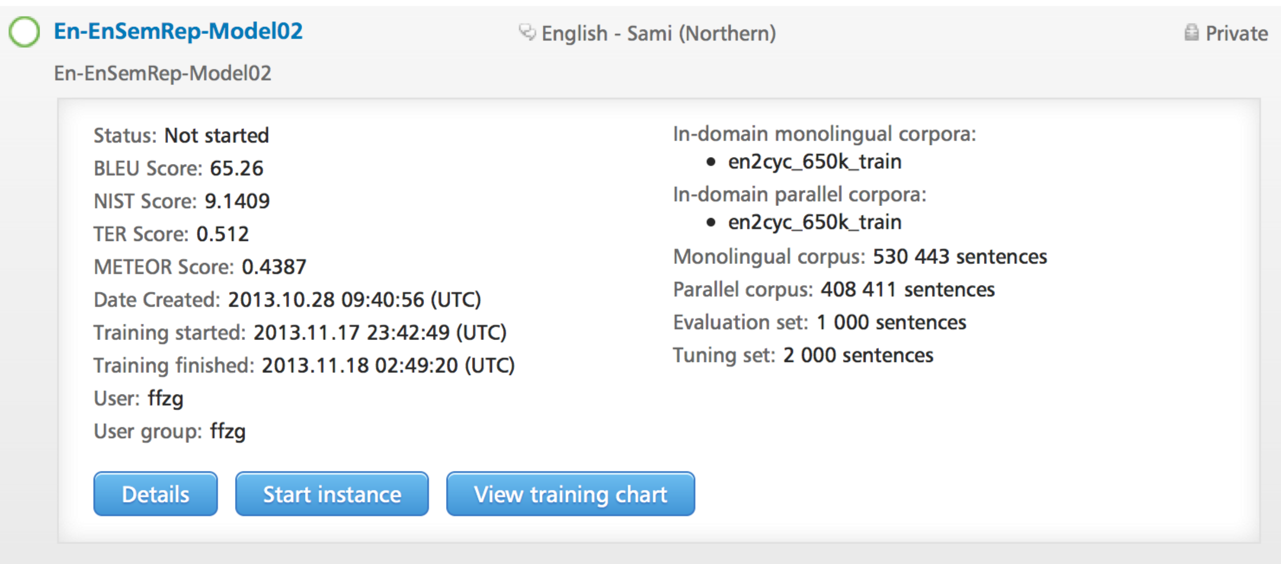


Figure 10. The En-EnSemRep-Model02 available for translation at Let'sMT! platform

5 Evaluation of Translation

This section describes the translation and evaluation of the translation produced by **En-EnSemRep-Model02** SMT system.

5.1 Translation from English into CycL

The trained SMT systems can be used through the Let'sMT! platform service Translate only if they are in the running state. Similar to Google Translate, the Let'sMT! platform opens a SL box (where user pastes the SL text) and a TL box (where selected running SMT system provides the translation). Entire SL files can be translated without this web interface as well.

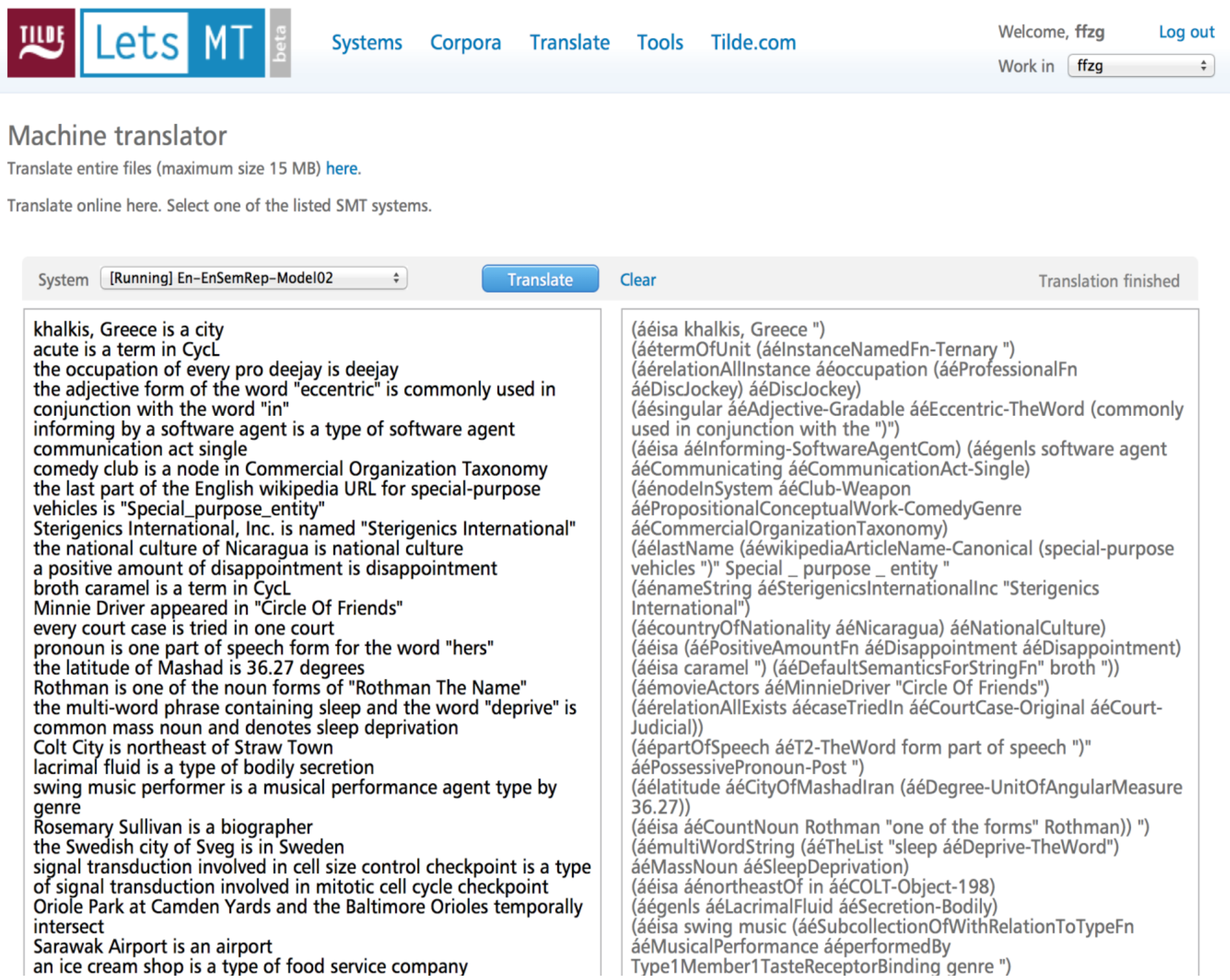


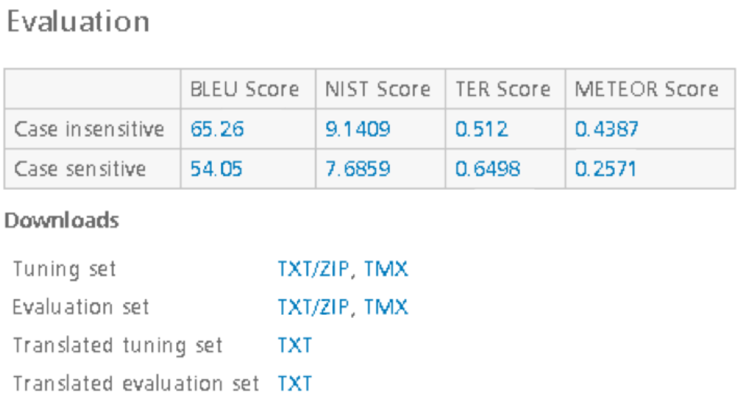
Figure 11. Example of the translation from English to CycL

5.2 Evaluation of the current translation quality

In MT community there are two basic types of evaluation of the MT quality: automatic and human.

5.2.1 Automatic evaluation

At the end of the training process the Let'sMT! platform produces automatic evaluation of the trained SMT system using the standard automatic evaluation measures such as BLUE, NIST, TER and METEOR scores.


Evaluation

	BLEU Score	NIST Score	TER Score	METEOR Score
Case insensitive	65.26	9.1409	0.512	0.4387
Case sensitive	54.05	7.6859	0.6498	0.2571

Downloads

Tuning set	TXT/ZIP, TMX
Evaluation set	TXT/ZIP, TMX
Translated tuning set	TXT
Translated evaluation set	TXT

Figure 12. Automatic evaluation of translation quality for **En-EnSemRep-Model02** SMT system

The values of these automatic evaluation scores turned out to be surprisingly good, so we expected usable translations by humans also, and not just the machines. However, this kind of evaluation should not be always considered realistic, particularly when the translation should be used by humans. In our case, we were not limited to this type of usage (human) exclusively, but we still organized the manual evaluation in order to check the usefulness of this approach to the semantic annotation of texts at the very basic level.

5.2.2 Human evaluation

For the human evaluation in this early prototype of SMT used for semantic annotation, we used 1,000 sentences from the test set of 10,000 sentence pairs that was set aside previously (see Section 3.2). This human evaluation set of 1,000 was translated using **En-EnSemRep-Model02** SMT system and result was submitted to the evaluation process. The human evaluation was performed by three evaluators, each covering one third of human evaluation set (i.e. 2x333 and 1x334 sentences).

The software used for human evaluation was Sisyphos II, an open source MT human evaluation package produced by a Munich-based LT company Linguatec within the ACCURAT project[4], as a part of ACCURAT Toolkit [7]. This suite of programs written in Java enable three different human evaluation scenarios: Absolute evaluation, Comparative evaluation and Postediting evaluation.

Since the CycL was the TL in this case, the Postediting evaluation was not applicable since it was meant to measure number of corrections done by humans on MT output errors. Forcing humans into correcting CycL statements and then measure their performance, wouldn't really give us a realistic evaluation of translation quality.

Comparative evaluation would be useful if we had to compare outputs of two different systems or a MT system output with a human translation. In our case we had only one version of output, so our clear choice for evaluation was Absolute evaluation scenario.

However, this scenario was targeted for human judging the quality of MT output using two categories with several possible values: Adequacy and Fluency. It is clear that both categories are easily applicable to judge the quality of translation into a NL, but are not so easily applicable for judging the quality of translation into

[4] http://www.accurat-project.eu

a FL. We had to determine several simple rules when is the translation into CycL adequate fully or not adequate at all, and when is fluency CycL-grammatical or not fluent in CycL at all.

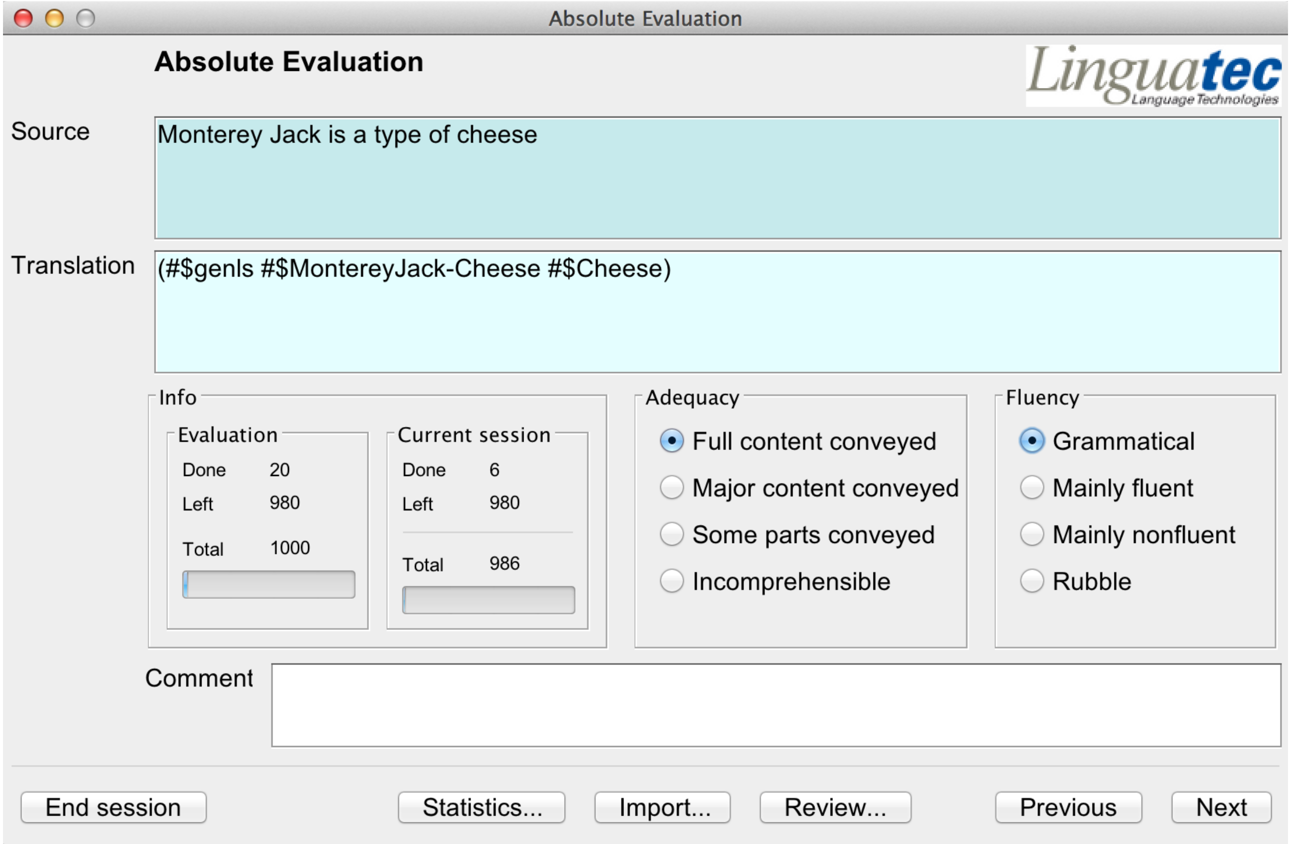


Figure 13. Sisyphos II screen with Absolute evaluation scenario

Cumulative results of human Absolute evaluation are given in the Table 1.

Category	Value	Occurences	Percentage
Adequacy	Full content conveyed	209	20.9%
	Major content conveyed	289	28.9%
	Some parts conveyed	270	27.0%
	Incomprehensible	232	23.2%
Fluency	Grammatical	212	21.2%
	Mainly fluent	137	13.7%
	Mainly non fluent	244	24.4%
	Rubble	407	40.7%

Table 1. Results of the human evaluation of translation quality of 1000 English sentences translated into CycL by **En-EnSemRep-Model02** SMT system

Interpretation of results from the Table 1 show that human evaluation scored the translation quality of **En-EnSemRep-Model02** SMT system much lower than automatic evaluation. The average Adequacy would fall into value *Some parts conveyed* (but very close to *Major content conveyed*), while Fluency would fall into value *Rubble* (almost 41% of all translations are CycL-nonfluent, thus breaching its syntactic rules (mostly due to the mismatching parenthesis). This means that good part of content from English sentences is conveyed into CycL, but it is not done following the strict formal syntax of this FL. This also means that translation from English into CycL, as it is performed by this SMT system, is not immediately applicable for usage where statements with clean and regular CycL syntax are expected.

5.3 Future SMT systems and evaluation

This early prototype of usage of SMT for semantic annotation was based on the simplest type of SMT systems, namely, word- and phrase-based translation model. What is planned further in this task is to experiment with usage of larger training set (preferably more million sentence pairs) and more complicated translation models, e.g. factor based, that will use results obtained by WP2 linguistic processing pipelines.

The automatic and human evaluation that have been performed, belong to an **intrinsic evaluation**, i.e. we were evaluating how the SMT system performs when applied to isolated parts of documents (in this case sentences only). In the continuation of this task also an **extrinsic evaluation** will be performed, i.e. how the results of this SMT system can be used in further processing steps and how would its usage boost the performance of the whole XLike toolkit.

This evaluation scenario is yet to be designed, but some spots/tasks can be envisaged already at this stage, e.g. for detecting/disambiguating Named Entities, for detecting/disambiguating general concepts, for detecting/disambiguating relations (particularly generic ones like **#$isa** or **#$genls**).

Also, human evaluation can be enhanced by using more evaluators to evaluate the same parts of translations and then the overlapping (using e.g. kappa factor) can be calculated and a proper average obtained. In this way we will receive more inter-personally relevant results.

6 Conclusion

With this deliverable we have reported on the first experiment that attempted to use SMT system for translation from NL as SL into FL as TL. The CycL was the FL of our choice because the training material could have been produced in a non-expensive way by generating from Cyc Ontology a set of aligned pairs of English sentences with their respective CycL "sentences" as counter parts.

This parallel corpus served as a training material for Moses based SMT system, with which we had some problems in running up, but eventually it provided translations.

Judging by automatic evaluation procedure, the scores of four standard automatic MT evaluation metrics (BLEU, NIST, TER and METEOR) could guarantee high quality translation. However, human evaluation applied intrinsically in absolute evaluation scenario yielded lower results, but still acceptable to certain degree.

The final decision whether this approach could be useful in the XLike processing platform or it should be discarded as invaluable, will have to be done after the extrinsic evaluation of usefulness of this SMT-based semantic annotation within the whole processing pipeline. Particularly, it might be shown that the usability of this approach is more applicable at the level of the paragraphs, whole document or sets of documents instead of lower levels of sentences.

References

[1] Brown, P. F.; Della Petra, S. A.; Della Pietra, V. J.; Mercer, R. L. (1993) The Mathematics of Statistical Machine Translation: Parameter Estimation, *Computational Linguistics*, volume 19, number 2, pp. 263-311.

[2] Och, F. J.; Ney, H. (2003) A Systematic Comparison of Various Statistical Alignment Models, *Computational Linguistics*, volume 29, number 1, pp. 19-51.

[3] Koehn, P.; Hoang, H.; Birch, A.; Callison-Burch, C.; Federico, M.; Bertoldi, N.; Cowan, B.; Shen, W.; Moran, C.; Zens, R.; Dyer, C.; Bojar, O.; Constantin, A.; Herbst, E. (2007) Moses: Open Source Toolkit for Statistical Machine Translation, Annual Meeting of the Association for Computational Linguistics (ACL2007), demonstration session, Prague, Czech Republic.

[4] Knight, K.; Koehn, P. (2003) What's New in Statistical Machine Translation (Tutorial slides), University of Southern California, pp. 4 (http://people.csail.mit.edu/people/koehn/publications/tutorial2003.pdf, accessed 2013-12-20).

[5] Davies, J.; Grobelnik, M.; Mladenić, D. (eds.) (2009) Semantic Knowledge Management, Springer, Berlin-Heidelberg.

[6] Buitelaar, P.; Cimiano, P. (eds.) (2008) Ontology Learning and Population: Bridging the Gap between Text and Knowledge, IOS Press, Amsterdam.

[7] Pinnis, M., Ion, R., Ştefănescu, D., Su, F., Skadiņa, I., Vasiļjevs, A., & Babych, B. (2012) ACCURAT Toolkit for Multi-Level Alignment and Information Extraction from Comparable Corpora. Proceedings of the ACL 2012 System Demonstrations, ACL, Jeju, South Korea, pp. 91–96.

