A Distributed Resource Repository for Cloud-Based Machine Translation

Jörg Tiedemann[1], Dorte Haltrup Hansen[2], Lene Offersgaard[3], Sussi Olsen[4], Matthias Zumpe[5]
[1,5] University of Uppsala, Sweden, [2,3,4] University of Copenhagen, Denmark
E-mail:[1] jorg.tiedemann@lingfil.uu.se, [2]dorteh@hum.ku.dk, [3]leneo@hum.ku.dk, [4]saolsen@hum.ku.dk, [5] matthias.zumpe@lingfil.uu.se

Abstract

In this paper, we present the architecture of a distributed resource repository developed for collecting training data for building customized statistical machine translation systems. The repository is designed for the cloud-based translation service integrated in the Let'sMT! platform which is about to be launched to the public. The system includes important features such as automatic import and alignment of textual documents in a variety of formats, a flexible database for meta-information using modern key-value stores and a grid-based backend for running off-line processes. The entire system is very modular and supports highly distributed setups to enable a maximum of flexibility and scalability. The system uses secure connections and includes an effective permission management to ensure data integrity. In this paper, we also take a closer look at the task of sentence alignment. The process of alignment is extremely important for the success of translation models trained on the platform. Alignment decisions significantly influence the quality of SMT engines.

Keywords: machine translation, resources, alignment

1 Introduction

The LetsMT! project is an EU-project with the aim of building a platform for user tailored machine translation and online sharing of training data.[1] The system requires a repository to store and handle data in a safe and functional way. For this purpose, we have built a distributed resource repository (henceforth RR) that allows the users to upload their data collections as either public, shared or private data. In order to ensure security of uploaded content, we explicitly do not allow any browsing and downloading of data. However, all data sets are described by extensive metadata information including access permissions. Using this information, it is possible to select appropriate data sets for user-tailored SMT training tasks. Another important feature of the resource repository is the integrated conversion and alignment of documents in a variety of formats. RR takes care of validation, conversion, text extraction and sentence alignment (if necessary) in order to prepare data for training phrase-based SMT engines. Another important property of the RR is that it has to fit into the distributed cloud-based architecture of the LetsMT! platform. It, therefore, implements several WebService API's using secure HTTP requests. Details of the RR architecture will be described in section 3. Let us first have a look at the general RR design according to a the needs of potential users.

2 Functional Design

The RR functionality design is based on recommendations from a user survey (D1.1 2010). Potential users from the localization industry, translation agencies and academic institutions provided initial feedback. This section lists the most important user requests, all handled by the current implementation of the RR.

The users' willingness to upload texts to the RR completely depends on a secure repository with the option of specifying access permissions. For many potential users it is important to keep private data or to limit access to a selected user group. Concerning the upload functionalities, most users (especially from the localization industry) stated that TMX (Translation Memory eXchange format) is the most important upload format. Another localization standard format mentioned is XLIFF (XML Localisation Interchange File Format). However, import of other document types (MS Word, PDF and also plain text) was also requested. These formats require special handling as they need to be converted and, in the case of translated documents, aligned. The current RR handles all these formats and additional import modules can easily be added. More details will be given in section 6.1 below.

An important feature of the RR is to let users share corpora for SMT system training. In order to select appropriate corpora for the training process, the users need information about the corpus e.g. language pairs, subject domain, text type, corpus size, data provider and a general description of the corpus. These metadata records are saved in the RR for each data collection, and, as the type of metadata information can change over time, we decided to use a schema-less database management system that can easily be extended with additional information. The web frontend facilitates the user's classification of data uploads by providing fixed lists of subject domains and text types. However, the RR implementation is flexible and changes in the future are supported.

[1] LetsMT! is supported by the European Commission's ICT Policy Support Programme and is running from Mar. 1st 2010 until Aug. 31 2012.

In the survey, it was also requested to add features for rating corpora based on the experience with those resources. In general, such information can easily be added to any resource in the RR using our flexible metadata DB. It will be the task of the web frontend to provide appropriate interfaces to give such valuable feedback within the platform. User-provided quality assessment can be combined with automatic validation procedures. In section 6.3, we look at some possibilities that can help to evaluate corpus and alignment quality

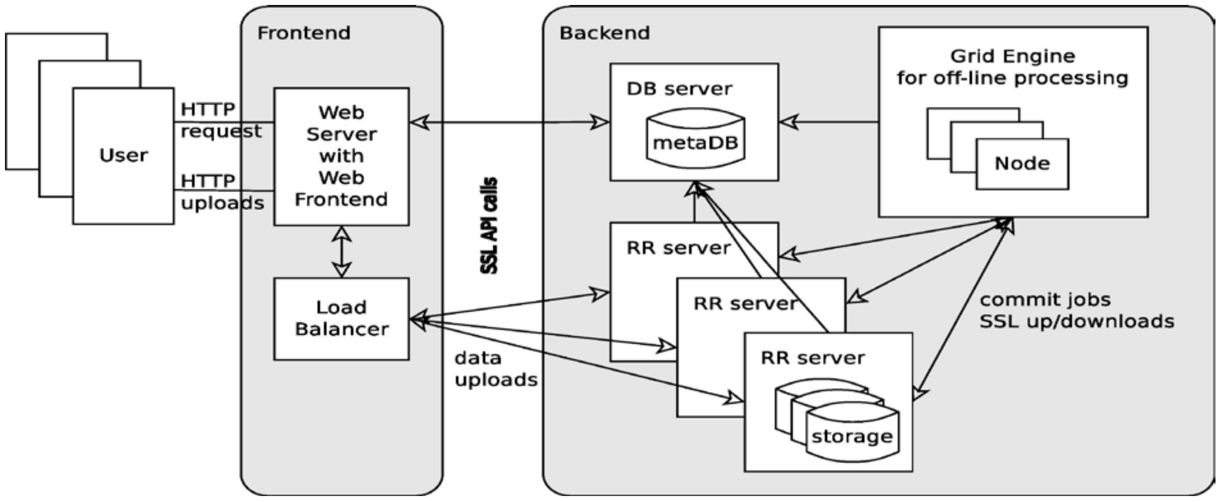


Figure 1: The general architecture of the LetsMT! resource repository

3 Architecture and Implementation

The resource repository is designed to be highly modular to allow a maximum of flexibility and scalability. Figure 1 illustrates the general architecture of the repository software. The communication with the web frontend of LetsMT! is handled via secure HTTP requests. The repository software provides several web service APIs to perform tasks at the backend and to obtain information from the resource collection. The backend contains a central database server (DB server), which manages data permissions, group settings and the entire collection of meta information available for each resource in the repository. The database is connected to all resource repository servers (RR server) via a client-server architecture.

The database uses a schema less key-value store, which makes it possible to extend the database content very easily without changing any settings or internal data structures. In our current implementation, we use TokyoCabinet (FAL Labs), a modern implementation of a database management system that supports the storage of arbitrary data records without pre-defined typed data structures. We use the table extension of TokyoCabinet that makes it possible to attach records of key-value pairs to each resource identifier (which we use as unique table key). The software also supports flexible search operators on automatically indexed key-value pairs, which makes the access to the data extremely fast. For the connection to the database server, we use TokyoTyrant, a network interface to TokyoCabinet. This server runs as a multi-threaded daemon on the DB server and TokyoTyrant clients connect from other RR servers via dedicated network ports.

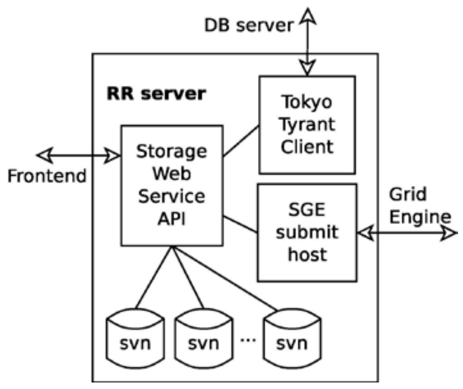


Figure 2: Resource Repository Servers in LetsMT!

Several RR servers (see figure 2) can run in parallel and each of them connects with the central database server via the TokyoTyrant client interface. Finally, each RR server will be configured as a submit host for the grid engine. We use the Oracle Grid Engine (formerly Sun Grid Engine, SGE), a standard open-source implementation for Grid Computing. In this way, each RR server may submit jobs to the central SGE server that distributes task to the grid execution hosts in the high-performance computer cluster. Typical jobs are data import and validation or SMT training processes. Heavy data processing jobs can thus be distributed via the grid engine, which can easily be scaled according to the load of the system. Furthermore, a load balancer at the frontend handles the distribution of resources to appropriate storage servers. This reduces the load due to intensive data transfers on individual machines.

The modular design of the platform supports highly distributed setups. Each server can run on different machines in a cloud-based installation. New RR servers and SGE execution hosts can be added on demand. The database can also be replicated on remote servers. TokyoTyrant natively supports this functionality.

4 General Workflow

One of the main principles of the LetsMT! platform is to maintain user-provided content for building machine translation engines from scratch using state-of-the art technology in statistical machine translation. Potential users may, therefore, sign-up for this service and create their own data repositories and translation systems. A user may view the metadata and extend existing data collections (if permissions allow) or create new ones.

The first step of creating a new resource is to make a new “corpus” that will be used as the slot containing all documents and alignment information. A user will be asked to fill in appropriate metadata to describe the resource and may then proceed with uploading data.

Documents may be uploaded in a variety of formats as described earlier. The user needs to select the format from the list of supported formats and may upload single documents or entire archives. For now we support zip and tar archives that may contain an arbitrary number of documents (the current upload limit is 2GB). Documents will be converted automatically by the backend after uploading to the RR. This will be done by scheduling appropriate import jobs at the Grid engine. The user also needs to specify the language for monolingual data uploads which will be validated using automatic language detection (for the languages included in the detection software). In future, we may also include automatic detection of languages for user uploads without explicit specification of the source language. Language information for aligned data in TMX and XLIFF files is taken directly from the input documents.

During the automatic import, there will be a test whether certain monolingual documents need to be aligned. The system automatically looks for files with identical names but different language specifications. Matching file names are then automatically sentence aligned by the system in the backend process. The alignment procedure presumes that user uploads apply appropriate naming conventions for parallel documents. In future, we may include additional procedures for matching potentially parallel documents.

Users who upload data collections may decide how to share them with others. Permissions are defined per slot, i.e. for an entire corpus. Only the owner of a resource (the user who creates the slot) may change its permissions. The underlying system supports private data (not shared with anyone), public collections (readable by everyone) or shared collections (readable by members of a certain group of users). The latter type is not yet supported by the LetsMT! frontend but fully implemented by the backend RR software. Users may create and maintain groups in order to give members of those groups permissions to certain resources in the repository.

Finally, users may use resources from the repository for creating task-specific translation systems. The repository can efficiently be queried using the flexible metadata store. Language filters, domain filters and other specific query parameters can easily be combined to narrow down search results even in large data collections. Naturally, only the data sets, for which a specific user has read permissions, will be visible.

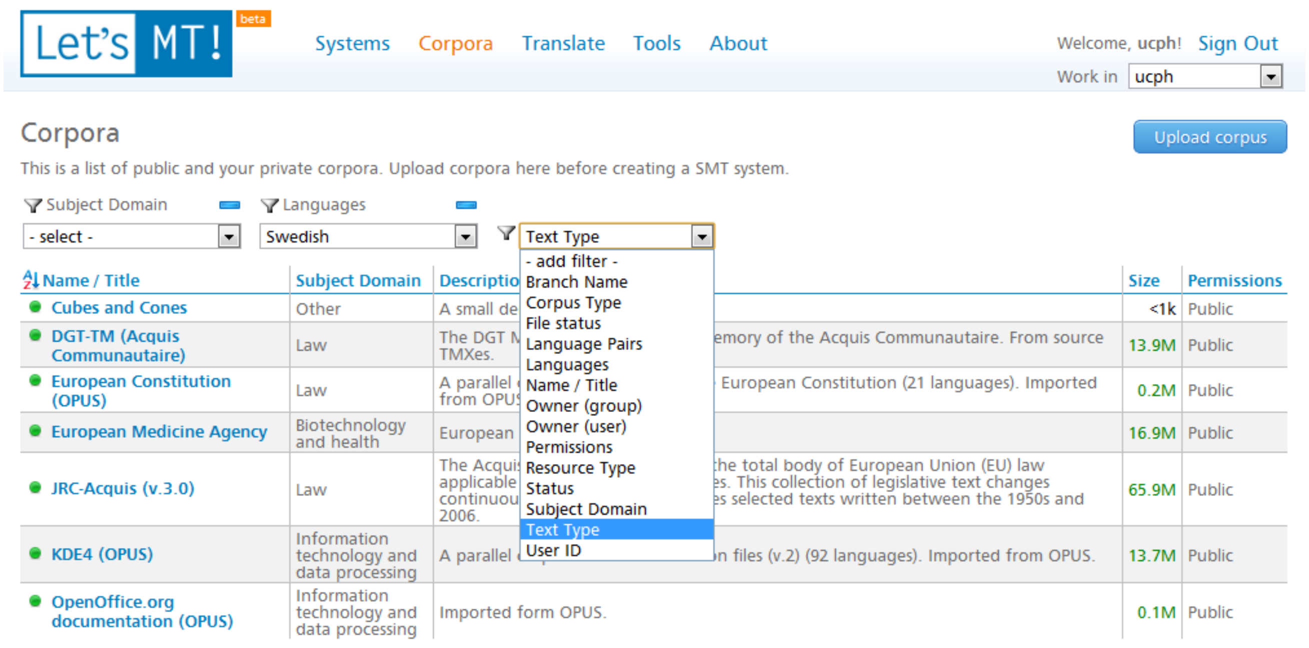
Selected resources may then be used to train statistical machine translation systems. For this, grid engine clients interact with the RR to fetch selected data and the RR software takes care of converting them to the format needed by the training procedures. In the LetsMT! Platform the training process is handled by the Moses SMT system.[2] Moses basically requires sentence-aligned parallel resources for training translation models and monolingual data for training target language models. Details of the SMT training process are beyond the scope of this paper.

5 Data in the LetsMT! RR

The LetsMT! RR has already been populated with resources for a variety of language pairs with special emphasis on the languages of the new members of the European Union. Apart from the well-known public resources from the EU e.g. European Parliament Proceedings, other resources mostly come from academic research projects or localization data collected by the commercial partners of the project. Almost all the project partners have now contributed to the repository and although the amount of available parallel data for some of the languages is rather sparse, the number of resources for all the partner languages is still growing. Most resources are public but several private corpora have been uploaded as well.

The subject domains represented in the public corpora for the time being are: law, finance, information technology and data processing, biotechnology and health, national and international organizations and affairs, education and other. “Other” is used for resources where the domain is mixed or undefinable. The number of languages represented in the repository is large (78). More important is the fact that the ten under-resourced languages in focus of the project are quite well represented. At the moment more than 120 M aligned sentences have been uploaded to the RR.

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


Let's MT! beta

Systems Corpora Translate Tools About

Welcome, ucph! Sign Out

Work in ucph

Corpora

Upload corpus

This is a list of public and your private corpora. Upload corpora here before creating a SMT system.

Subject Domain: - select -

Languages: Swedish

Text Type

- add filter -
- Branch Name
- Corpus Type
- File status
- Language Pairs
- Languages
- Name / Title
- Owner (group)
- Owner (user)
- Permissions
- Resource Type
- Status
- Subject Domain
- Text Type
- User ID

Name / Title	Subject Domain	Descriptio	Size	Permissions
● Cubes and Cones	Other	A small de	<1k	Public
● DGT-TM (Acquis Communautaire)	Law	The DGT M TMXes. emory of the Acquis Communautaire. From source	13.9M	Public
● European Constitution (OPUS)	Law	A parallel from OPUS European Constitution (21 languages). Imported	0.2M	Public
● European Medicine Agency	Biotechnology and health	European	16.9M	Public
● JRC-Acquis (v.3.0)	Law	The Acqui applicable continuou 2006. the total body of European Union (EU) law es. This collection of legislative text changes es selected texts written between the 1950s and	65.9M	Public
● KDE4 (OPUS)	Information technology and data processing	A parallel n files (v.2) (92 languages). Imported from OPUS.	13.7M	Public
● OpenOffice.org documentation (OPUS)	Information technology and data processing	Imported form OPUS.	0.1M	Public

Figure 3: The LetsMT! corpora site showing the currently uploaded Swedish corpora

Data collection is still in progress and some of the data being prepared for the repository right now will strongly increase the amount of resources. Furthermore, after the first public release, we hope that end-users will rapidly start uploading their own data sets leading to a fast increase of training material available in the RR.

Figure 3 shows a screen dump of the web interface of the RR, showing Swedish resources currently available. The metadata shows that there are corpora from three different domains. Each corpus has a short description and the size is stated. Furthermore it can be seen whether the resource is public or private. Private resources can however only be seen by the owner or by users who have read permissions for this resource and do not appear on this screen dump. As can be seen in the drop down list, lots of other filtering possibilities exist, e.g. text type, owner, provider, file status, etc. These filtering options facilitate the user's selection process when he/she wants to compile precisely the corpus that he/she needs for SMT training.

6 Data processing in RR

The uploaded documents are automatically converted into an internal XML format needed for training SMT system. We decided to take the burden of every-day users to prepare data sets in an appropriate corpus format and to facilitate the creation of parallel and monolingual data sets. In this section we describe the steps in the automatic import processes in addition to tests done on alignment quality and an experiment on data filtering procedures.

6.1 Data Import

The support of different input formats is not a simple task. Currently, we support two types of documents: (1) pre-aligned data sets in TMX, XLIFF and plain text format and (2) monolingual documents in popular formats like MS Word, PDF and plain text. Each of these formats has its challenges. None of them can be seen as a static, fully-specified document type that can easily be converted to our needs. Even open formats like TMX and XLIFF come in different flavours.

```
<xliff version="1.2">
 <file original="Graphic Example.psd"
  source-language="en-US" target-language="ja-JP"
  tool="Rainbow" datatype="photoshop">
  <header> <phase-group>
    <phase phase-name="extract" process-name="extraction"
     tool="Rainbow" date="20010926T152258Z"
     company-name="NeverLand Inc." job-id="123"
     contact-name="Peter Pan" contact-email="ppan@example.com">
     <note>Make sure to use the glossary I sent you yesterday. Thanks.
      </note></phase></phase-group> </header>
<body>
   <trans-unit id="1" maxbytes="14">
    <source xml:lang="en-US">Quetzal</source>
    <target xml:lang="ja-JP">Quetzal</target>
   </trans-unit>
   <trans-unit id="3" maxbytes="114">
    <source xml:lang="en-US">An application to manipulate and
     process XLIFF documents</source>
    <target xml:lang="ja-JP">XLIFF 文書を編集、または処理
     するアプリケーションです。</target>
   </trans-unit>
   <trans-unit id="4" maxbytes="36">
    <source xml:lang="en-US">XLIFF Data Manager</source>
    <target xml:lang="ja-JP">XLIFF データ・マネージャ</target>
   </trans-unit>
</body> </file></xliff>
```

Figure 4: An example XLIFF document (from http://en.wikipedia.org/wiki/Xliff)

Each import job, therefore, starts with a validation phase. Here, we try to check essential properties in order to proceed with the actual conversion. These checks highly depend on the given format and tools we have available. For example, TMX and XLIFF are checked according to their standard DTD's or XML schemas. We also use a fall-back validation for standalone XML files in case DTD validation fails. Slightly corrupted files are repaired with tidy (http://tidy.sourceforge.net/) if possible.

Another example is the plain text format. Character encoding is here an important issue. We, therefore, include BOM[3] detection (also for TMX/XLIFF files which may come in various encodings) and automatic character encoding identification based on chared (http://code.google.com/p/chared/). For language detection we apply textcat (http://www.let.rug.nl/vannoord/TextCat/) and our own language models. Other formats that require the extraction of texts (MS Word, PDF) will also go through this pipeline.

```
<tmx version="1.4b">
 <header creationtool="XYZTool" creationtoolversion="1.01-023"
  datatype="PlainText" segtype="sentence"
  adminlang="en-us" srclang="en"
  o-tmf="ABCTransMem">
 </header>
 <body>
  <tu><tuv xml:lang="en">
    <seg>Text in<bpt i="1">&lt;B&gt;</bpt>bold
          <ept i="1">&lt;/B&gt;</ept>.</seg>
   </tuv>
   <tuv xml:lang="fr">
    <seg>Texte en <bpt i="1">&lt;B&gt;</bpt>gras
          <epti="1">&lt;/B&gt;</ept>.</seg>
   </tuv>
</tu>
 </body>
</tmx>
```

Figure 5: TMX with inline annotation (an example from http://en.wikipedia.org/wiki/Translation_Memory_eXchange)

After successful validation, files are converted to our internal XML format. In this step, we extract plain texts from each document and apply necessary pre-processing steps. The latter include text normalization (for example removal of double white space characters, transformation of ligatures etc.) and sentence boundary detection. We use generic software for the last step together with language specific parameters and resources (for example, non-breaking prefixes). We intend to add further language and domain-specific tools to the pipeline in future developments.

Finally, after converting all uploaded documents into our internal corpus format, the system automatically aligns all possible language pairs in case of multi-parallel documents. Information about validation and conversion is added to the metadata store to make it possible to see the status of each resource together with detailed meta-information.

6.2 Sentence alignment

Although, sentence alignment is a well-known task and often considered to be solved in the SMT research community, it has a deep impact on the translation quality that can be achieved by systems trained on the parallel data. The main challenges arise with noisy data sets of possibly incomplete translations. In our case, where we support arbitrary user uploads and create parallel corpora on the basis of identical file names, it is an important challenge to tackle possible alignment problems. We, therefore, include a discussion about alignment quality using standard algorithms here.

Earlier work (Singh & Husain, 2005) has shown that different sentence aligners yield varying degrees of alignment accuracy for different corpora and language pairs. Here, we revisit two standard tools: a vanilla Gale&Church (Gale and Church, 1994) aligner and HunAligner (Varga, 2005).

Since the HunAligner initially performs a sentence-length based alignment similar to the Gale&Church algorithm and then builds an automatic dictionary based on this alignment and realigns the text, the best results are expected from the HunAligner.

	G&C	**HunAlign**
Rapid Press Releases		
Recall	54.8	92,6
Precision	56.7	89.8
Annual reports		
Recall	82.8	80.8
Precision	83.3	77.9

Table 1: Aligner performance for two text types.

In table 1, the output of each aligner is compared with alignments in a gold corpus (English-Danish press releases and annual reports). Both precision and recall are calculated. For press releases, HunAligner's precision and recall are much better than G&C's, for annual reports G&C's precision and recall are better, but the difference is less significant. As expected the performance for the HunAligner is better in general.

Sentence aligners estimate 1:1-alignments, 0-alignments and other types of alignments. 1:1-alignments are aligments of one sentence in one language to exactly one sentence in the other language. The aligners are unable to come up with crossing alignments, i.e. segments x followed by y in one language will not be aligned by two alignments to segment y' x' in the other language, but will end up as a 2:2 alignment. 0-alignments can be filtered

[3] Byte order mark

out when creating parallel resources as they are not useful for training statistical translation models. 1:1-alignments usually refer to the most reliable data sets for SMT training, as they state a one-to-one mapping of sentences.

	G&C	**HunAlign**	**Gold Corpus**
Rapid Press Releases			
1:1 alignments	69.9 %	93.4 %	89.9 %
0-alignments	4.7 %	3.8 %	2.2 %
Other alignments	25.4 %	2.8 %	7.9 %
Sentence length			22.9
Annual reports			
1:1 alignments	79.7 %	81.2 %	79.1 %
0-alignments	0 %	5.1 %	3.3 %
Other alignments	20.3 %	13.7 %	17.6 %
Sentence length			15.5

Table 2: Alignment types for the 2 text types.

Table 2 shows the distribution of alignment types for the 2 text types investigated. We assume that knowledge of the alignment types and the distribution of alignment scores can be used to indicate how well the parallel texts are suited for SMT training tasks.

6.3 Experiments with data filtering

When documents are converted automatically noise arises from different sources: the files might be broken or have different content than expected, the translations might not be completely parallel, the layout might have destroyed the text in the initial XML conversion etc. These factors consequently lead to bad sentence-alignment. This is especially severe for domain-specific settings we focus on in LetsMT! with only little training data available.

In this section we describe experiments with filtering of data on the basis of alignment scores and types. The results have not yet been implemented in the RR.

Statistical alignment algorithms often produce a score that can be interpreted as some kind of link certainty. The range of this scores depends of the alignment algorithm, an example of the format and scores from the HunAligner can be seen in figure 6.

```
<linkGrp targType="s" toDoc="en/8065.xml" fromDoc="da/8065 ">
        <link certainty="7.14777" xtargets="s1.1;s1.1" id="SL2" />
        <link certainty="5.4769" xtargets="s2.1;s2.1" id="SL4" />
```

Figure 6: Sentence alignment produced by HunAligner (as integrated in Uplug)

The references in the *xtargets* attribute (see figure 6) point to source and target languages sentences in the aligned documents.

As filter we used the alignment types and alignment scores from the HunAligner. The 0-alignments were removed and the average scores calculated for each document. From the average alignment scores we have done manual inspection of documents with a low average score (< 2). It seemed, however, that this wasn't a sufficient clue for alignment quality. In Pecina et al. (2011) an absolute score of 0.4 was used to filter out bad alignment. Our observations are, however, that especially positive low scores are not reliable while negative scores, high positive scores and average scores for the entire document are more useful. We therefore investigated documents with a high per cent of negative alignments (> 10%). In this case all parallel documents were of a bad quality. We also inspected documents without negative alignments. Absence of negative alignments can either indicate a perfectly parallel translation, an English-English ”translation” (the same file) or empty files. Finally we searched for the English word *the* in the non-English documents to spot false translations or language pairs being swopped.

Annual reports	Swedish	Danish	Dutch
Av. score, all reports	2.92	3.1	3.57
Av. score < 2	17 %	8 %	14 %
Neg. scores > 10%	13.5 %	7.7 %	7.4 %
Neg. scores = 0%	4 %	3.5 %	14.8 %
% of documents with mixed languages	1.6 %	4.2 %	2.8 %
% of documents filtered out	16.1 %	14.7 %	19 %

Table 3: Parameters for data filtering.

The filtering is done by 1) deleting the documents with more than 10% negative scores from the sentence alignments, 2) deleting documents with 0% negative scores that are either near to empty or empty and 3) deleting documents with large English text parts in the documents expected to be non-English (the groups can be partly overlapping). These filtering actions resulted in filtering out 14.7% to 19% of the documents. We expect this filtering approach to delete texts that are only partly parallel or files where the file conversion has been so erroneous that the alignments will be of low quality.

We suggest that high quality data in terms of being parallel and in-domain, can be filtered based on negative alignment scores from the HunAligner. Our findings are that positive alignment scores are less reliable than negative scores and that the average percentage of negative scores is a very good indicator for the alignment quality of the document and therefore of the data quality. It is very difficult to set a fixed cut-off limit but our manual investigations showed that a threshold of around 10 % negative alignments per document was the upper limit.

7 Future work

There are several plans for future developments of our system. As mentioned earlier, we consider using the integrated language detection for automatic classification of text uploads. In this way we can avoid the explicit specification of the document language and can further support the upload of large document collections with diverse contents. Automatic language verification can also be useful for TMX and XLIFF imports in order to validate the information given in those files. Another feature that we like to include is a flexible tool that can discover potentially parallel documents in large collections without relying on specific name conventions. Finally, we also need to improve language and domain-specific tools in our data processing pipeline.

We would also like to add to the RR the possibility to inspect alignment results and to re-run conversion and alignment processes if necessary, possibly with other tools and parameters. However, we still like to keep the abstraction level implemented by the current transparent system that avoids confusing non-technical users with internal data structures and formats. It is a challenge to maintain a system with intuitive and simple interfaces that still allows detailed adjustments by advanced users.

In the experimental phase of data filtering we would like to test other filtering options e.g. to delete all alignments inside each document with a negative alignment score, and to keep all positive alignments. This approach might, however, not lead to better results as the alignments around a negative alignment might still be suffering from the texts being out-of-sync concerning parallel sentences, but experiments will show.

8 Conclusion

This paper describes the distributed resource repository developed for the collaborative SMT platform Let'sMT! Our modular system supports the import of documents in a variety of standard formats and includes procedures for the automatic conversion and alignment of these documents to the formats necessary for training SMT models. The repository software uses version controlled file systems for storing data collections in a distributed environment. Arbitrary meta-information can be added to the data sets using a flexible key-value store. Backend jobs can be distributed over several clients in a scalable grid engine. All services can be used via RESTful Web-API calls using secure and encrypted connections.

The system also integrates essential pre-processing software including language identification, character encoding detection, sentence boundary detection and sentence alignment. The latter is crucial to build parallel corpora used for training statistical translation models.

In this paper, we also investigate the quality of standard alignment algorithms when applied to typical data sets uploaded to the repository. Furthermore, we explore options to filter out bad alignments from parallel corpora by removing documents with negative alignments over a certain threshold, and documents with large part of un-translated text. These filtering possibilities could also be added to the resource repository, for un-experienced users indicating the quality of their parallel corpora and for more advanced users as tuning options for the alignment process.

The resource repository is currently populated with more than 60 M aligned sentences for over 44 languages. The main focus is however on 10 under-resourced European languages and 7 different subject domains. In the future the amount of data is expected to grow since both the project and the future end-users will upload more parallel data.

9 References

Apache Subversion*:* "Enterprise-class centralized version control for the masses", *Information and software available from http://subversion.apache.org/*, website accessed 2011/10/13

Gale, W.A. and K.W. Church. 1994. A program for aligning sentences in bilingual corpora. *Computational linguistics*, 19(1), pp. 75-102.

Oracle Grid Engine: A distributed resource management (DRM) system, *Information available at http://www.oracle.com/us/products/tools/oracle-grid-engine-075549.html*, accessed 2011/10/13

Pecina,P et.al. Towards UsingWeb-Crawled Data for Domain Adaptation in Statistical Machine Translation. *15th EAMT conference*, 2011.

Public project report LetsMT! D1.1 *Report on requirements analysis*, 2010, http://project.letsmt.eu

Singh, A. K. and Husain, S. Comparison, Selection and Use of Sentence Alignment Algorithms for New Language Pairs. In *Proceedings of ACL 2005 Workshop on Parallel Text.* Ann Arbor, Michigan. June 200.

Tiedemann, J. Uplug - a modular corpus tool for parallel corpora. In *Parallel Corpora, Parallel Worlds,* pages 181-197, Rodopi, 2002.

TokyoCabinet: A lightweight database library. *Information and software available from http://fallabs.com/tokyocabinet/*, Fal Labs, accessed 2011/10/13

TokyoTyrant: The network interface of Tokyo Cabinet. *Information and software available from http://fallabs.com/tokyotyrant/*, Fal Labs, website accessed 2011/10/13

Vasiļjevs, A., Skadiņš, R., & Tiedemann, J. (2011). LetsMT!: Cloud-Based Platform for Building User Tailored Machine Translation Engines. In *Proceedings of the 13th Machine Translation Summit* (pp. 507-511). Xiamen, China

Varga, D., L. Németh, P. Halácsy, A. Kornai, V. Trón, V. Nagy (2005). Parallel corpora for medium density languages. In *Proceedings of the RANLP 200,* pp. 590-596.