Let's MT! — A Platform for Sharing SMT Training Data

Jörg Tiedemann, Per Weijnitz

Department of Linguistics and Philology
Uppsala University
jorg.tiedemann@lingfil.uu.se, per.weijnitz@convertus.se

Abstract

In this paper we describe the LetsMT! platform for sharing training data for building user-specific machine translation models. We give an overview of the general structure of the data repository including the flexible internal storage format that will be used to access data via a transparent user interface. Several tools will be integrated in the platform that support not only uploading data in various formats but also the verification, conversion and alignment of translated documents. The shared resources can then be used within the platform to train tailored translation models using existing state-of-the-art technology that we will integrate in LetsMT! In this paper we show the potentials of such an approach by comparing a domain-specific system with the general purpose engine provided by Google Translate. Our results suggest that domain-specific models may lead to substantial gains even when trained on scarce resources.

1. Introduction

In recent years, statistical machine translation (SMT) has become the leading paradigm for machine translation. However, the quality of SMT systems largely depends on the size and appropriateness of training data. Training SMT models becomes a major challenge for less supported languages since parallel corpora of reasonable size are only available for a few languages. Furthermore, most parallel resources come from very restricted domains and models trained on these collections will always have a strong bias towards the domain of the training data.

To fully exploit the huge potential of existing open SMT technologies we propose to build an innovative online collaborative platform (LetsMT![1]) for data sharing and MT building. This platform will support the upload of public as well as proprietary MT training data allowing users to build multiple MT systems according to their selections of shared training data. Permissions to access uploaded content will be set by the users allowing them to define user groups to share the data with. We will stress the possibility of data privacy that will motivate professional users to use our platform but we hope to achieve a liberal sharing policy among our users.

The main goal of LetsMT! is to make SMT technology accessible for anyone and to enable every-day users to build tailored translation engines on their own and user-contributed data collections without worrying about technical requirements. Initial data sets and baseline systems will be made available to show the potentials of the system and to motivate users to upload and share their resources.

In this paper we describe the general structure of the data repository and the internal storage format that we will use. Finally, we also include a test case illustrating the benefits of domain-specific SMT models compared to general purpose translation using state-of-the art MT provided by Google.

[1]LetsMT! is a ICT PSP PB Pilot Type B project from the area CIP-ICT-PSP.2009.5.1 Multilingual Web: Machine translation for the multilingual web.

2. The LetsMT! Data Repository

One of the key functions of the LetsMT! platform is to provide the possibility to train domain-specific SMT models tailored towards specific needs of its users. For this appropriate data resources are required. LetsMT! is based on data sharing and user collaboration. We will allow data uploads in a variety of formats and store all resources in a unified internal storage format.

The LetsMT! data repository will be based on a robust *version-controlled file system*. We will use a simple and clear file structure to store parallel and monolingual data. Each corpus identified by a unique name (parallel or monolingual) will be stored in a separate version-controlled repository. The name of the corpus will be used as the name of this repository and may contain arbitrary numbers of documents. Repositories can be created by any user but each user will only have access to his/her own branch inside this repository that will be set up during creation time. Each LetsMT! user can then work with a copy of existing corpora through branching (of course only if permissions allow that). In this way we create a space-efficient and flexible environment allowing users to share data and even to apply changes to their copy without breaking data integrity. This will allow us to integrate on-line tools for personal data refinement, for example, tools for adjusting sentence alignments. These refinements can again be shared between users. Another benefit of version-control systems is that changes can be traced back in time. Specific revisions can be retrieved on demand and data releases can be defined.

Inside each repository we will keep the original uploads in their raw format in order to allow roll-back functionalities. Furthermore, pre-processed data in our internal corpus format will be stored together with their meta-data. We will use ISO 639-3 language codes to organize the data collection in appropriate subdirectories. Meta-data will also be stored in a central database allowing users to quickly browse and select training data according to their needs.

Internally all uploaded documents will be converted to a simple XML format which is easy to process and convert. Basically, we will add appropriate sentence boundaries to

the textual contents with unique identifies within each document. Sentence alignments will be stored in separate files with pointers referring to sentences in the corpus. An example is given in figure 1.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE cesAlign PUBLIC "-//CES//DTD XML
          cesAlign//EN" "">
<cesAlign version="1.0">
  <linkList><linkGrp targType="s"
fromDoc="Europarl/xml/eng/ep-00-01-17.xml"
  toDoc="Europarl/xml/fre/ep-00-01-17.xml">
      <link xtargets="1;1" />
      <link xtargets="2;2" />
      <link xtargets="3;3 4" />
```

Figure 1: Sentence alignments in LetsMT!

One of the main advantages of this approach is that alignment can be changed easily without the need of changing anything in the original corpus files. Various alignment versions can be stored and several languages can be linked together without repeating corpus data. Furthermore, corpus selection can be done using the same format. Several parallel corpora or only parts of certain corpora can be selected without the need of explicitly concatenating the corresponding corpus data. These selections can then be stored space-efficiently in the repository. They can be shared and revised easily. A simple procedure can then be used off-line to extract the actual data from the repository when training is initiated.

3. User-Tailored SMT Models

The largest benefit of the LetsMT! platform will be the support of user-specific SMT engines. Users of the platform will have full control over the selection of data resources which will be used for training a system. The potentials of such an approach can be seen in the test case described below.

We took data from the medical domain in order to show the impact of domain-specific data on SMT training. In particular we used the Swedish-English portion of the publicly available EMEA corpus which is part of OPUS (Tiedemann, 2009). This corpus covers a very specific domain including documents published by the European Medicines Agency. We extracted non-empty sentence alignments with a maximum of 80 tokens per sentence from the corpus in order to create appropriate training data for standard phrase-based SMT. Table 1 lists some statistics of the data.

	English	Swedish
sentences	898,359	898,359
tokens	11,567,182	10,967,600
unique sentence pairs		
sentences	298,974	298,974
tokens	4,961,225	4,747,807

Table 1: Training data extracted from EMEA

The EMEA corpus contains a lot of repetition as we can see from the numbers in table 1. The number of unique sentence pairs is much lower than the count for the original corpus. Naturally, we want to test the SMT model on unseen data only also to make a fair comparison to general-purpose machine translation. Therefore, we merged multiple occurrences of identical sentence pairs in order to create a set of unique sentence pairs and randomly selected 1000 of them for tuning and another 1000 for testing. The remaining sentence pairs are used for training. We trained standard phrase-based SMT models in both directions on that data using the target language side of the parallel training corpus for training the 5-gram language model. We basically used standard settings of the Moses system (Koehn et al., 2007) including lexicalized reordering and minimum error rate tuning.

For comparison we translated the same test set of 1000 example sentences using the current on-line system of Google Translate (date of the run: 28 August 2010) and measured lower-case BLEU scores for both systems. The results are shown in table 2.

	Google	Moses-EMEA
English-Swedish	50.23	59.29
Swedish-English	46.57	65.42

Table 2: Translation quality in terms of BLEU scores

The gain that we achieved by using in-domain training data is more impressive than we actually had expected. In the general case data of such a small size would not be sufficient for training appropriate SMT models. Not only the parallel data used for training the translation model is very little but especially the monolingual target language data used for the language model is much smaller than otherwise recommended. However, due to the domain specificity and especially the translation consistency in our data reasonable results can be achieved with this tiny amount of training data. Furthermore, we can see that general purpose translations do not reach the same quality even though they are trained on vastly larger amounts of data. It might even be possible that our training and test data is part of the collection used by Google as these documents are publicly available on the web. This, however, is beyond our control and we can only speculate about the resources used to train Google's translation engine.

4. References

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *ACL '07: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180, Morristown, NJ, USA.

Jörg Tiedemann. 2009. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia.