Facilitating the Multilingual Single Digital Market: Case Studies in Software Containerization of Language Technologies

Dewi Bryn Jones, Delyth Prys, Stefano Ghazzali, Patrick Robertson

Bangor University, Bangor, Wales, UK
{d.b.jones, d.prys, s.ghazzali, p.robertson}@bangor.ac.uk

Abstract. If language technology tools and resources are to be used to facilitate the multilingual single digital market in Europe, they need to be simplified and packaged so that they may be adopted by developers and ICT providers who are not themselves experts in LT. This is especially important for less-resourced language communities as it will empower them to create products and services in their own language rather than depend on the altruism of those outside the language community. This paper presents case studies of the use of containerization technologies with Moses-SMT, HTK and META-SHARE so as to demonstrate how containerization can make such software more accessible and usable for non-specialists and as a result promote their widespread dissemination and adoption.

Keywords: language technology· containerization· Docker· Moses-SMT· HTK· META-SHARE

1 Empowering Local Digital Communities

META-NET's vision for an innovative, technically advanced multilingual Europe which is integrative and inclusive hopes to see language barriers overcome by 2020, enabling the single digital market through the use of language technology (META_NET Strategic Research Agenda for Multilingual Europe 2020). This paper aims to contribute to this goal by demonstrating techniques for making language technology resources more accessible to software developers, webmasters, translators and others who are not LT experts but who would nevertheless benefit if multilingual resources, services, and support were simple and straightforward to add to their products and services.

Simple and accessible methods of facilitating language technology resources in diverse digital environments are of particular concern to smaller language communities. Without a high level of expertise in developing LT, they may be reliant on external centres of research. This situation does nothing to empower these language communities or improve their own knowledge and skills. This is analogous to the idea of empowerment developed in social sciences, where giving individuals or groups the means to control their own destiny helps them to help themselves and maximizes the quality of their lives (Adams, 2008). Instead of low self-esteem, marginalization and reliance on others, such communities are enabled to develop resources and tools to answer their own priorities and needs. This is not an argument for fragmentation and withdrawal of transnational cooperation and coordination, as empowering local language communities can only happen within the framework of a common shared strategy, infrastructure and international standards. It is the establishment of a common purpose and methods of working that enables the inclusion of the various communities, in a supportive environment where communities are encouraged to realize the single digital market together, rather than passively rely on others to do it for them.

Empowering language communities can also strengthen the digital economy, especially in peripheral areas, where developing "an economy based on knowledge, research and innovation…fostering high employment to deliver social and territorial cohesion" is central to EU's strategy for smart, sustainable and inclusive growth (Atlantic Area Transnational Cooperation Programme, 2015). With the improvement in broadband infrastructure, SMEs such as web design companies, ICT developers and support services, and companies in the translation, creative industries, and culture and tourism sectors, are able to establish

themselves in a wide range of locations, including previously challenging rural and semi-rural areas. These are often regions of high cultural and linguistic richness and diversity, where a bilingual or multilingual workforce already exists and could benefit from readily accessible LT tools and resources.

The remainder of this paper presents case studies demonstrating how complex LT tools and resources may be packaged, containerized, presented and disseminated to empower such language communities, helping them to help themselves and actively participate in the endeavour to form a truly multilingual information society in Europe.

2 Software Containerization Case Studies

Many LRs are sophisticated pieces of open source software with complex and intricate build dependencies. In addition, some require an in-depth knowledge of the language technology field in question in order to operate the software.

Recent developments in software containerization, such as the launch of the Docker platform, mitigates these issues and provides opportunities to package and distribute pre-built LRs with accompanying scripts as accessible and useful offerings to a wider audience of new users.

Docker provides the same resource isolation and allocation benefits for a given application as virtual machines, but a different architectural approach allows Docker containers to be much more efficient and portable. Docker images contain an application's entire filesystem, and images are built from instructions normally issued at a command prompt in a Dockerfile.

2.1 Case Study 1: Moses-SMT

The Moses statistical machine translation system is an ideal candidate for packaging with Docker, due to its very complex build dependencies, requirements for additional LRs and its usefulness to translators who would benefit from being able to possess their own machine translation engines.

An instance of a Moses statistical machine translation system packaged within a Docker container first appeared in March 2015 as a deliverable from a national project conducted by Bangor University's Language Technology Unit and funded by the Welsh Government. This project's goal was the establishment the Welsh National Language Technologies Portal, to promote and make accessible prioritised Welsh language technology resources (Prys & Jones, 2015). Docker containerization of the project's Moses-SMT ensured that the project deliverable was available to translators and machine translation practitioners in the Welsh translation industry in a highly accessible form.

The Dockerfile developed builds a basic Moses-SMT decoder Docker image. Instructions include installing build dependencies such as specific compilers and libraries, downloading and building third-party LRs that Moses utilizes such as GIZA++, IRSTLM and CMPH before setting up environment variables and finally building the Mosesdecoder itself with XML-RPC support included. The following simple Docker pull command can be issued to obtain the pre-built Docker image from the central Docker hub repository:

```
$ docker pull techiaith/moses-smt
```

Other Dockerfile instructions add bespoke Python scripts to the Docker image's file system that make the Moses decoder useful for Welsh<>English translation. Upon running the image the user is able to state a pre-trained domain specific machine translation engines available for download from the Welsh National Technologies Portal website at http://techiaith.cymru/. For example:

```
$ docker run --name moses-smt \

      -p 8080:8080 \
```

```
-p 8008:8008 \

techiaith/moses-smt \

start -e CofnodYCynulliad -s en -t cy
```

Interfaces to both a JSON and XML-RPC server are then available (at ports 8008 and 8080 respectively) as well as a simple HTML form for demonstration or verification purposes at http://localhost:8008. The JSON server in addition is able to use a recaser included with the downloaded machine translation engine.

The Welsh <> English Moses-SMT image, has been pulled 139 times to date - an encouraging figure for a pairing involving a less resourced language. Other examples of Moses-SMT packaged with Docker have been created by individuals with the initial Dockerfile to provide language independent images with additional capabilities and flexibility for more advanced MT practitioners and researchers.

2.2 Case Study 2 : HTK

A recent project by Bangor University's Language Technology Unit to stimulate progress on Welsh language speech recognition involved using the HTK Speech Recognition Toolkit to train new acoustic models (Jones & Cooper, 2016).

The stable version of the HTK available to the project (version 3.4, released in March 2009) had a build dependency on certain 32-bit libraries. It was therefore not immediately possible to build the latest stable HTK on the more recent and prominent 64-bit operating systems. However a workaround can be put into place and utilized repeatedly if included in the instructions of a Dockerfile that builds a Docker image for an HTK acoustic model training environment.

The project subsequently packaged bespoke scripts for:

- downloading and preparing the crowd sourced speech corpus (Cooper et al, 2014),
- building pronunciation lexicons,
- training the actual acoustic models with single, multiple or every individual contributions from the speech corpus,
- testing and reporting on word and sentence accuracies
- packaging for easy integration into the Julius large vocabulary CSR engine.

The portability and simplification provided by the Docker based HTK environment allowed for consistent reproduction of test results amongst developers, as well a cross-discipline collaboration with Welsh language phoneticians able to use the facility to contribute their expertise for language dependent HTK files, such as the decision tree clustering script file (tree.hed).

Since the HTK license restricts source code redistribution, there is no image for HTK on the Docker Hub Registry that would facilitate a simple 'docker pull' command. Users are able however to build their own images by obtaining separately the HTK source code and cloning the Dockerfile and associated files from GitHub:

```
~/src$ git clone https://github.com/techiaith/seilwaith.git
~/src$ cd seilwaith
~/src/seilwaith$ wget --user <your registered HTK username> --ask-
password http://htk.eng.cam.ac.uk/ftp/software/HTK-3.4.1.tar.gz
~/src/seilwaith$ wget --user <your registered HTK username> --ask-
password http://htk.eng.cam.ac.uk/ftp/software/HTK-samples-
3.4.1.tar.gz
~/src/seilwaith$ make
```

This make command automatically runs the Docker images in order to perform HTK's installation verification tests. The user should exit the Docker environment with an 'exit' command.

This HTK Docker image serves as a base which further Dockerfiles are able to inherit and extend for packaging into a derived Docker image for containing bespoke scripts that facilitate easy training of acoustic models with Welsh language specific speech data. An example sequence of commands for producing acoustic models from a subset of Paldaruo contributors' recordings would be:

```
~/src/seilwaith/srdk$ make
root@4fb715613e2:/usr/local/srdk# cd cy/paldaruo
root@4fb715613e2:/usr/local/srdk/cy/paldaruo# ./fetch.sh
root@4fb715613e2:/usr/local/srdk/cy/paldaruo# ./init.sh
root@4fb715613e2:/usr/local/srdk/cy/paldaruo# cd
/srdk_projects/cy/paldaruo
root@4fb715613e2:/srdk_projects/cy/paldaruo# SRDK_0_PrepareAudio.py -f
userids.csv -n paldaruo
root@4fb715613e2:/srdk_projects/cy/paldaruo# SRDK_Train.py
```

Further information can be found at the GitHub repository https://github.com/techiaith/seilwaith.git

2.3 Case Study 3 - META-SHARE

The Bangor University Language Technology Unit recently deployed its own META-SHARE node using the facilities provided by Docker. Its resources, along with accompanying high quality metadata, can be seen at http://metashare.techiaith.cymru.

META-SHARE's code and accompanying comprehensive documentation for facilitating deploying your own META-SHARE nodes can be found on GitHub. Installation proved challenging due to dependencies for dated versions of certain web stack components.

It became obvious that such difficulties could be mitigated by developing a Dockerfile that would be able to include the required versions of web stack components in a META-SHARE Docker image. The META-SHARE Docker image requires a local settings file, copied into the image by Dockerfile instructions, to be present at initial startup. Consequently it is not possible to provide a META-SHARE image from the Docker Hub Registry and a single simple 'docker pull' command. A clone from GitHub is necessary:

```
~/src$ git clone https://github.com/techiaith/docker-metashare.git
```

The local_settings.py file can be edited according to the official documentation and a METASHARE node deployed by executing:

```
~/src/docker-metashare$ make

~/src/docker-metashare$ make run
```

Access to the META-SHARE node's file system for administration purposes can be obtained from issuing:

```
~/src/docker-metashare$docker exec -it metashare bash

root@6a1a810049fb:/META-SHARE-3.0.2/metashare#
```

It is hoped that other language technology resource providers may be able to benefit from this approach that simplifies the deployment of META-SHARE nodes and thus aid in expanding the network of repositories of language data, tools and web services.

3 Conclusion

This paper has presented three brief case studies demonstrating how software containerization can simplify complicated technology so that they become accessible to local digital communities and serve as a means of empowering those communities to develop and support their own language technology infrastructure. In our case studies, Docker was utilized to tame three complex language technologies, namely Moses-SMT, HTK and META-SHARE in order to eliminate difficulties in building and installation, as well as to package accompanying bespoke scripts that make these resources usable to non-LT experts.

As usage of Docker grows, with individuals and other third-party organisations also producing their Docker images containing various complex language technology resources, it may become incumbent on the original developers of systems such as Moses-SMT, HTK and META-SHARE to provide official and supported Docker images. Digital communities, in particular those of lesser resourced languages, stand to gain from the common infrastructure and support that such an approach provides.

References

1. Adams, R. Empowerment, participation and social work, p. xvi. New York, Palgrave Macmillan (2008)
2. Atlantic Area Transnational Cooperation Programme 2014-2020 Version 1.1, p. 23. European Union (2015)
3. Cooper, S., Prys, D., Jones, D.B. Developing further speech recognition resources for Welsh. In: Judge, J., Lynn, T., Ward, M. and Ó Raghallaigh, B. eds. Proceedings of the First Celtic Language Technology Workshop at the 25th International Conference on Computational Linguistics (COLING 2014), pp. 55-59. Dublin, Ireland (2014)
4. Jones, D.B. & Cooper, S. Building Intelligent Digital Assistants for Speakers of a Lesser-Resourced Language. In LREC 2016 - CCURL Workshop Proceedings. Portorož, Slovenia (2016)
5. META_NET Strategic Research Agenda for Multilingual Europe 2020. `http://www.meta-net.eu/sra/` [Accessed 18/05/2016].
6. Prys, D. & Jones, D.B. National Language Portals for LRLs: a Case Study in Language Technologies. In LRL 2015, pp. 355-360. Poznan, Poland. (2015).