Language Packs

Language packs are pre-built translation models with an included instance of the Joshua runtime environment. A key feature is that **there are no dependencies** (apart from Java 8). Getting a machine translation system running on your own machine is as easy as downloading the tarball, unpacking it, and running the included shell script.

> **Version 3 Language Packs Coming Soon**
>
> (June 2018) We had intended for Version 3 language packs with Kenlm (via Docker) and more complete Google Translate API support to be prepared however this will not happen any time soon. If you would be interested in taking this project on as a Google Summer of Code project. If you have questions, comments, concerns, or wish to help, please post questions to the Joshua mailing list: dev@joshua.apache.org.

Table of Contents

- Language Packs
- Using Language Packs
- Improved Translation With KenLM
- Decoder Options
- Versions
- Citation

Language Packs

The following language packs are available for Joshua. Click the links on the full language pair names to download the models directly. You might be interested in notes on how most of these models were built, including information about how to make them faster (with a little elbow-grease), better (with a little knowledge), and what you might want to do with them.

ISO 639	Language pair	Release Date	Size	Version	Notes
en-en	English–English	2016-11-18	various	2	English paraphrase packs from the Paraphrase Database
am-en	Amharic–English	2016-11-18	841 MB	2	
ar-en	Arabic–English	2016-11-18	1.4 GB	2	
az-en	Azerbaijani–English	2016-11-18	846 MB	2	
bg-en	Bulgarian–English	2016-11-18	2.2 GB	2	
bn-en	Bengali–English	2016-11-18	893 MB	2	
bs-en	Bosnian–English	2016-11-18	1.4 GB	2	
ca-en	Catalan–English	2016-11-18	936 MB	2	
cs-en	Czech–English	2016-11-18	2.7 GB	2	
da-en	Danish–English	2016-11-18	3.5 GB	2	
de-en	German–English	2016-11-18	4.0 GB	2	
dv-en	Dhivehi–English	2016-11-18	873 MB	2	
el-en	Greek–English	2016-11-18	3.2 GB	2	
en-de	English–German	2017-01-31	4.5 GB	2	Phrase-based model
en-ru	English-Russian	28 Oct 2016	4.6 GB	2	Language model data sources can be found within the artifact README file
es-en	Spanish–English	2016-11-18	4.8 GB	2	
et-en	Estonian–English	2016-11-18	2.2 GB	2	
eu-en	Basque–English	2016-11-18	877 MB	2	
fa-en	Persian–English	2016-11-18	1.3 GB	2	
fi-en	Finnish–English	2016-11-18	2.6 GB	2	
fr-en	French–English	2016-11-18	4.0 GB	2	
ga-en	Irish–English	2016-11-18	866 MB	2	
gl-en	Galician–English	2016-11-18	879 MB	2	
ha-en	Hausa–English	2016-11-18	853 MB	2	
he-en	Hebrew–English	2016-11-18	1.4 GB	2	
hi-en	Hindi–English	2016-11-18	858 MB	2	

hr-en	Croatian–English	2016-11-18	1.4 GB	2	
hu-en	Hungarian–English	2016-11-18	2.0 GB	2	
id-en	Indonesian–English	2016-11-18	1.4 GB	2	
is-en	Icelandic–English	2016-11-18	1.1 GB	2	
it-en	Italian–English	2016-11-18	3.9 GB	2	
ka-en	Georgian–English	2016-11-18	849 MB	2	
ku-en	Kurdish–English	2016-11-18	827 MB	2	
lt-en	Lithuanian–English	2016-11-18	2.0 GB	2	
lv-en	Latvian–English	2016-11-18	2.0 GB	2	
mg-en	Malagasy–English	2016-11-18	907 MB	2	
mk-en	Macedonian–English	2016-11-18	1.4 GB	2	
ml-en	Malayalam–English	2016-11-18	851 MB	2	
ms-en	Malay–English	2016-11-18	1014 MB	2	
mt-en	Maltese–English	2016-11-18	1.4 GB	2	
nl-en	Dutch–English	2016-11-18	3.6 GB	2	
no-en	Norwegian–English	2016-11-18	1.4 GB	2	
pl-en	Polish–English	2016-11-18	2.8 GB	2	
pt-en	Portuguese–English	2016-11-18	4.5 GB	2	
ro-en	Romanian–English	2016-11-18	2.5 GB	2	
ru-en	Russian–English	2016-11-18	1.9 GB	2	
ru-en	Russian-English	04 Nov 2016	4.4 GB	2	Language model data sources can be found within the artifact README file
sd-en	Sindhi–English	2016-11-18	837 MB	2	
si-en	Sinhala–English	2016-11-18	862 MB	2	
sk-en	Slovak–English	2016-11-18	2.4 GB	2	
sl-en	Slovenian–English	2016-11-18	2.3 GB	2	
so-en	Somali–English	2016-11-18	850 MB	2	
sq-en	Albanian–English	2016-11-18	1.3 GB	2	
sr-en	Serbian–English	2016-11-18	1.5 GB	2	
sv-en	Swedish–English	2016-11-18	3.4 GB	2	
sw-en	Swahili–English	2016-11-18	859 MB	2	
ta-en	Tamil–English	2016-11-18	832 MB	2	
te-en	Telugu–English	2016-11-18	823 MB	2	
tg-en	Tajik–English	2016-11-18	851 MB	2	
tt-en	Tatar–English	2016-11-18	840 MB	2	
ug-en	Uighur–English	2016-11-18	838 MB	2	
uk-en	Ukrainian–English	2016-11-18	984 MB	2	
ur-en	Urdu–English	2016-11-18	866 MB	2	
vi-en	Vietnamese–English	2016-11-18	1.2 GB	2	

Using Language Packs

Once you download the model, unpack it. The simplest use-case is then to run Joshua as a standard UNIX tool, accepting a single line of input and writing a single line of output. Assuming your language pack is downloaded to "apache-joshua-language-pack.tgz":

```
# SRC and TRG are the two-character ISO 639-1 language codes
tar xzf apache-joshua-SRC-TRG-YYYY-MM-DD.tgz
cd apache-joshua-SRC-TRG-YYYY-MM-DD
cat example.SRC | ./prepare.sh | ./joshua
```

Here, "example.SRC" is a file containing sentences in your input language (e.g., "es" for Spanish), one per line. Joshua expects to be given one sentence at a time; it will not do this for documents by itself.

There is some startup cost associated with the models, however. You may find it more beneficial, therefore, to run it as a server. Joshua can run in two server modes: raw TCP, and HTTP.

```
# start in server mode, taking direct TCP/IP connections
./joshua -server-port 5674 -server-type tcp
cat example.SRC | nc localhost 5674

# start in server mode, answering web queries.
./joshua -server-port 5674 -server-type http
# Then open "web/index.html?port=5674" in your browser
```

Improved Translation With KenLM

The goal in releasing the language packs above was to make it easy for people to run translation systems. Part of this meant having no external dependencies (apart from Java). This means that we were not able to include the excellent KenLM language modeling code. If you are able to compile this, you can use it instead of the provided BerkeleyLM. This will result in significantly better translation quality, load time, and memory usage.

Docker Support

Shortly (February 2017) we will release a docker module for compiling KenLM and loading and running any of the Joshua language packs with KenLM, providing an easy way to get these improvements that hides some of the complexity below.

1. Download KenLM. You need to clone the Joshua repo, set some variables, and compile KenLM:

   ```
   mkdir joshua
   cd joshua
   export JOSHUA=$(pwd)
   curl -L https://api.github.com/repos/apache/incubator-joshua/tarball | tar --strip-components=1 -xzvf -
   RUN echo y | bash download-deps.sh kenlm
   ```

   If everything compiles correctly, this will produce a file in "lib/libken.so" (under Linux).
2. Make a "lib" directory in your language pack, and copy the file "lib/libken.so" to it.
3. Within the language pack, there should be a file named "joshua.config.kenlm". Rename that file to "joshua.config".

You can now start the language pack per normal, and it will use KenLM instead of BerkeleyLM. Depending on your environment, you may have some trouble compiling KenLM and the Joshua JNI library. In general, it requires GCC 4.8+ and the Boost libraries.

Decoder Options

Joshua supports many command-line options controlling its output. By default, it outputs only a single hypothesis per input line. Here are some options that may be useful to you:

- "-m XXg" — increase the amount of memory provided to Joshua. The default is 8g, but for the larger language packs, you will want 16 or 24. In general, 50% more memory than the raw model size should be more than sufficient.
- "-top-n N" — output up to N translation candidates, instead of just one.
- "-output-format STRING" — change the output format. By default, Joshua outputs just the single, tokenized translation with the highest model probability.
  Here are some other options:
  - %s: the raw translated string
  - %S: the detokenized translated string
  - %e: the source string
  - %i: the sequence number (0-indexed)
  - %c: the model score
  - %f: the feature string

  These can all be combined in a single string, e.g., -output-format "%i ||| %s ||| %f ||| %c"

Versions

The language pack version history:

Version	Description	Release Date

3	Includes KenLM language model files (recommended) in addition to BerkeleyLM. The latter is the default, with the former recommended and facilitated with a Docker container. Google API now multithreaded.	March 2017
2	Contains a "joshua" top-level script and "prepare.sh" for preparing data. Operates in server mode or from the command line. Entirely BerkeleyLM-based. Includes a Joshua 6.1 release candidate jar file.	November 2016

Citation

Please cite the following paper if you use Joshua in your research.

```
@article{post2015joshua,
  Author = {Post, Matt and Cao, Yuan and Kumar, Gaurav},
  Journal = {The Prague Bulletin of Mathematical Linguistics},
  Title = {Joshua 6: A phrase-based and hierarchical statistical machine translation system},
  Year = {2015}
}
```