Project Ideas

This page hosts ideas for Joshua projects. Many of these will also be present as tickets in Joshua's JIRA issue tracker. If you are interested in these projects, please comment on this page, the associated JIRA page, or email the Joshua development mailing list (dev@joshua.incubator.apache.org) for assistance and direction (and to ensure someone else is not already working on it!).

The projects here are designed to draw on a wide range of talents and interests. If you have a particular skill and would like to contribute to Joshua, please email that same address and we will do our best to find something for you!

Smaller Models

Estimated time: 2 weeks
Skills required: Moderate programming abilities
Difficulty: low–moderate

Phrase tables and grammars can get very big when trained on lots of parallel data, which makes it hard to distribute them in Language Packs. A quick way to reduce model size is to reduce the amount of parallel data used to build models, but sampling a subset of it. This is the very naive approach used in the construction of the original language packs (November 2016), but there are much better ways. One relatively simple one is the Vocabulary Saturation Filter (VSF), proposed by Will Lewis and Sauleh Eetemadi and described in this paper. It would be wonderful to implement this and use it to do a better job selecting which sentences to include for our general-purpose language packs.

It would be ideal to implement this in Java, but Python or Scala would also fit well inside Joshua.

Revamp Tokenization and Normalization

Estimated time: 1 week
Skills required: Reading Perl, writing regular expressions in Java
Difficulty: low

As part of the preprocessing, Joshua tokenizes input sentences, for example splitting punctuation off from words. This is currently done with a set of Perl preprocessing scripts, but it would be nice to move this to the decoder itself.

Integrated Transliteration

Estimated time: 2 weeks
Skills required: Java, Perl (for integration into the training pipeline), familiarity with expectation maximization
Difficulty: moderate

Many of the language packs released translated from languages with non-Latin scripts. Words that cannot be translated are therefore pushed through to the translation and cannot even be read by someone who doesn't know that script. At the same time, many untranslatable words are simply transliterated words. For example, an Arabic word might be an English word (like a name or technical term) that has simply been written in Arabic. These words can be transliterated. It would be good to add built-in transliteration models that can be applied to all out-of-vocabulary words and enabled for certain languages. Transliteration models can be built over the same bitext using techniques like Sajjad, Fraser, and Schmid (2012).

Demo Interface Improvements

Estimated time: 1 week
Skills required: Web front-end (JQuery, or whatever it is folks are using these days)
Difficulty: low–high, depending on what you do (design is hard!)

Joshua and Joshua language packs include a web-based demonstration that runs in a browser and contacts Joshua running as a server. The demo itself could be a lot prettier. There is also some additional functionality that would be nice to have:

- Add phrase highlighting. For each translation, it would highlight phrasal translation features as a unit, and link them to the point in the source sentence.
- Add better widgets for setting model weights (e.g., slider bars)
- Add better tokenization of the input (a proper sentence-splitter, for example)
- Prompt the user for a server connection at page-load time, dimming the screen until a proper connection is established
- Add cookies that would store the last server connected to, and perhaps other information.

. There are also changes to Joshua that could be made to support fancier front-end features, such as:

- Extend the JSON API to include alignments.