MT Marathon 2015 (Champaign)

Proposed Projects

Feel free to start a new entry or add your comments anywhere, in the text or on side.
Projects can be proposed until the first day of MT Marathon, but announcing them earlier might attract more participants, come better prepared etc.

List of proposed projects (in order of proposal time):

1. **Log-linear LM Interpolation** - Lane Schwartz
2. **Moses improvement grab-bag** - Jeremy Gwinnup
3. **Neural Network bootcamp** - Gaurav Kumar / Philipp Koehn
4. **Improving Bilingual Corpus Extraction from CommonCrawl** - Christian Buck
5. **Something related to discriminative training (MIRA)?** - David Chiang
6. **Stream decoding for Speech Translation -** Stephan Vogel
7. **Hybrid Neural Translation Modeling with PRO** - Chris Dyer

---

1. Log-linear LM Interpolation

Proposed by: Lane Schwartz

Description: Add description here

Examples:

Desirable skills: C++ and KenLM

Comments/Questions:

- why not linear interpolation? *(philipp koehn)*
  - *maybe because of simplicity? log linear -> addition , linear -> multiplication that might lead to underflow issues ?*

Interested? Add your name below:

- Lane Schwartz
- Joao Sedoc
- Jeremy Gwinnup

---

2. Moses improvement grab-bag

Proposed by: Jeremy Gwinnup

Description: Collection of (hopefully) minor improvements to moses

Examples:

- Print word alignments when using Incremental-Search algorithm
- **defer** - Fallback rule tables for chart decode (similar to --decoding-graph-backoff)
- **already implemented** - Expand placeholder functionality to work for more than one type of placeholder
- Fix chart decode fuzzy match
- Lattice MBR in chart decode
- **already implemented** - Generation-before-decode step

Desirable skills: C++ and Moses

Comments/Questions:

- Feel free to add similarly scoped items to the list above!

Interested? Add your name below:

- Jeremy Gwinnup
- Jatin Ganhotra
- Hieu Hoang (remotely)

3. Neural Network Bootcamp

Proposed by: Gaurav Kumar and Philipp Koehn (Slides)

Description: Implementation of the training of various neural network models for specific components (word alignment, reordering model, morphological prediction, …) and integration into Moses. We may use the Python toolkit Theano to train these models.

Desirable skills: Python and/or C++ and Moses

Comments/Questions:

Interested? Add your name below:

- David Chiang - I've tried reordering and it gave only a small improvement.
- Hieu Hoang (remotely) - how about a bilingual version of
  - http://jmlr.org/proceedings/papers/v32/botha14.pdf
  - The code for the mono LM hasn't even been released it
- David Chiang: how about a focus on recurrent networks -- maybe a new recurrent translation model and/or facilities for decoding with continuous states.
  - decoding with continuous states: status quo would be to turn off dynamic programming, but maybe something better is possible, like quantizing the state space?

4. Improving Bilingual Corpus Extraction from CommonCrawl

Proposed by: Philipp Koehn and Christian Buck

Description: In a previous MT Marathon, a pipeline was developed to extract parallel corpora from the large CommonCrawl dump of web pages. There are several components in the pipeline that require improvement (document alignment, parallel sentence extraction, etc.).

Desirable skills: Programming (any language, even Java), willingness to get hands dirty

Comments/Questions:

- It would be great to have someone with Amazon EC2 experience
- Does anyone have free EC2 credits?

Interested? Add your name below:

- Christian Buck
- Anthony Kimball

5. Something related to discriminative training (MIRA)?

Proposed by: David Chiang
Description: Discriminative training methods that scale to large numbers of features have been shown to be useful, but there have been a lot of challenges converging on a method that is as easy to use/implement as MERT is. There is an implementation of MIRA in Moses, but are there any training methods or bells and whistles that need to be added? E.g., maximum expected BLEU? Searching for hope/fear translations in the whole lattice? Or, since neural networks are the thing, replace the standard linear model with a feedforward neural network, whose inputs are typical MT features.

Comments/Questions:

Interested? Add your name below:

6. Stream decoding for Speech Translation

Proposed by: Stephan Vogel and Francisco Guzmán

Description: Implement a low-latency, stream-based decoding algorithm in Moses to be used in conjunction of stream-based speech recognition systems. Such an architecture has been described in (Kolss et al, 2008), and some of the challenges of incremental decoding (i.e. word by word) have been addressed (at a sentence level) by (Sankaran et al, 2010).

Often decoding is performed on segmented input, e.g. on entire sentences. This allows to first apply the translation model to build a graph containing all phrasal translations, and then perform a first-best or n-best path search through this graph, at which time the language model is applied and also reordering is considered. This means that the decoder outputs translations in chunks. While this works just fine in many applications it is not ideal in speech translation applications, where low latency is desired, i.e. in automatic simultaneous interpretation. And it is also not the preferred modus operandi for interactive translation.

The goal of the project is to add an additional decoding algorithm to Moses, that works similarly to stack/beam decoding, only that it accepts a continuous stream of words and generates a continuous, low-latency stream of target words.

This requires a different operation of the decoder: input is taken one word at a time, phrasal translations including this new word are harvested from the phrase table, search is expanded over the newly inserted translations, optionally target words from the currently best hypothesis are committed to the output, which then triggers the deletion of competing hypothesis. In addition, preparation for a different type of rich input format (e.g. CTM) should be available.

There are many challenges in this project: how to make stream-decoding efficient memory-wise? how to make it fast? can we use multiple threads? can we accept multiple streams?

Desirable skills: C++ and Moses

References:

Muntsin Kolss, Stephan Vogel, Alex Waibel. Steam Decoding for Simultaneous Spoken Language Translation. *Proc of Interspeech 2008*. Brisbane, Australia, 22-26 Sept 2008.

Baskaran Sankaran, Ajeet Grewal, Anoop Sarkar. Incremental Decoding for Phrase-Based Statistical Machine Translation. Proc of WMT 2010.

Comments/Questions:

Interested? Add your name below:

Francisco Guzmán
Ahmed Abdelali
Nadir Durrani

7. Hybrid Neural Translation Modeling with Pairwise Ranking Optimization

Neural translation models, even when they are used to provide rescoring features to phrase-based translation systems (e.g., Devlin et al. 2014, Auli et al., 2014, etc), are trained independently of the phrase-based systems where they will be used to provide features. This project will add nonlinear scores computed by neural networks to phrases used, but in contrast to previous work, these will be trained alongside the conventional translation features (heuristic translation probability estimates, language models, etc) to optimize a pairwise ranking objective (similar to the one proposed by Hopkins and May, 2011). Rather than modeling translation from scratch, the intention is that the neural net component will only need to explain *divergences* from the phrase-based system rather than "relearning" what the translation model or language model already knows. This will simplify the learning problem and hopefully lead to better results.

I went to the **bank** to get some **money**.
ich ging in die **Bank** , um **Geld** abzuheben.

There will be two components:

1) decoder integration (extraction of training instances, computation of feature values in context)

2) model training

My preference is to use cdec since I understand the architecture better.
For learning, we will use CNN (https://github.com/clab/cnn), a new library my group has been developing that is designed with NLP problems in mind.

Desirable skills: C++, Moses or cdec, CUDA

Interested? Add your name below:
Gaurav Kumar