Group Projects: MT

Alexander Fraser

CIS, LMU München

2017-05-23 Machine Translation

Group Projects

- We only have a limited time for presenting projects at the end of the semester
- As a result, we will do group projects in this class
- You will send me preferences about projects in about a week and a half
  - I wanted to present possible topics sooner so there is time to read up on possible topics
- Further details will be discussed later

- Project: Moses X-EN (or X-DE)
  - Download and install the open-source Moses SMT system (you may want to use the virtual machine distribution)
  - Choose a language X for which X-EN is not well-studied but make sure that parallel corpora are available
  - Download an English/X parallel corpus, e.g., from Opus or statmt.org
  - Build a Moses SMT system for X to EN
  - Test your system on held out training data and also on Wikipedia or similar (be sure to check that the English Wikipedia does not contain this content!)
  - Perform an overall error analysis of translation quality
  - Pick some polysemous X words and show whether Moses can correctly select all of the senses

- Project: Moses EN-DE
  - Download and install the open-source Moses SMT system (you may want to use the virtual machine distribution)
  - Download the Europarl English/German parallel corpus, e.g., from Opus or statmt.org
  - Build a Moses SMT system for EN to DE
  - Test your system on English-German data from the old release of the UN multilingual corpus (Chen, Eisele)
  - Perform an overall error analysis of translation quality
  - Document domain adaptation problems in the output
    - Time allowing, add a small amount of UN data to the training data and show the difference

- Project: Google Translate X-DE (Pivoting through EN)
  - Select a Language X text for which there is unlikely to be parallel English or German parallel data available (i.e., don't take a classic novel or news!). Suggestion: Wikipedia articles (on topics with no English or German pages)
  - Run this text through Google Translate X-DE
    - Split sentences to be separated by blank lines
    - Carefully save the results and record dates for all translations
  - Explicit pivot
    - Run this text through Google Translate X-EN
    - Post-edit the EN output to fix some obvious major errors
    - Run the original EN output and the post-edited EN through Google EN-DE
  - Perform a careful analysis of Google Translate's performance in translating these texts
    - Is Google Translate "pivoting" when translating from X-DE directly?
    - What are common problems in each translation?
    - Is there useful information which is easier to get from the original X input than from the intermediate EN?
    - Is the correct EN (after post-editing) somehow better than the original X input in some way?
    - Does post-editing the EN help DE translation quality? By how much?

- Project: Translate a German lexical substitution data set
  - There was a shared task on lexical substitution at GSCL 2015 called LexSub
  - They created a set of sentences where a small set of selected German words are annotated with possible synonyms
  - The task is to translate this data set to English using Google Translate
    - Evaluate and correct a number of the translations
  - This task is not trivial, because the annotated synonyms are lemmas, and also the sentences may no longer be entirely fluent when substitution occurs
  - Describe how to do a basic cross-lingual lexical substitution task using this data

- Project: Predicting case given a sequence of German lemmas (or lemmas from another case-marking language, e.g., a Slavic language)
  - Given a German text, run the CIS tools Marmot and Lemming to obtain rich part-of-speech tags and lemmas (be sure that such models are available, they are for German)
  - Suppose you have an MT system (from, e.g., English) which produces German lemmas
  - Then build a classifier to predict the correct case, given the sequence of German lemmas as context
  - (see also my EACL 2012 paper)

- Project: Wikification of ambiguous entities
  - Find disambiguation pages on Wikipedia which disambiguate common nouns, e.g. http://en.wikipedia.org/wiki/Cabinet
  - Download the web pages for the disambiguated pages, e.g.

    http://en.wikipedia.org/wiki/Cabinet_(furniture) or (government)
  - Automate this data set construction to the extent possible
  - Build a classifier to predict the correct disambiguation
    - Use only the sentences in the unambiguous articles in which the word itself occurs as your training data (this is like so-called "distant supervision")

Topics from Fabienne Braune and Matthias Huck