Proposal for changes to Machine Translation

Adam Lopez

March 1, 2016

1 Overview of proposed changes

I propose two changes, which could be applied independently.

1. Changes to assessment so that it is based 100% on coursework.

2. Changes to the course summary and syllabus.

1.1 Rationale for changes to assessment.

Currently, the course is assessed by coursework (30%) and exam (70%). This assessment structure makes sense for courses that focus on theoretical foundations, but machine translation is an application of natural language processing, machine learning, and software engineering to a very specific domain. Within the broader field of natural language processing, it much more akin systems than theory, and the most natural way to demonstrate mastery of systems is to build them. Since most of the students have already taken a course in natural language processing, and many of them have taken one or more courses in machine learning, they will have already been assessed on necessary theoretical background in the exams for those courses. (82% of students fully enrolled in this year's course have taken FNLP or ANLP; 73% have taken MLPR, PMR, and/or IAML).

Because past instructors have recognized the necessity of practical assessment, the course has always included a substantial coursework component: in six out of the past seven offerings, including this year's, the course has had three or four courseworks (in 2013 there were two). Although the course descriptor says that students should expect to spend 40 hours on coursework, many students anecdotally report spending much more time than this. They also spend substantial time on exam revision. In light of the emphasis on reducing assessment for 10-point courses—with which I agree—the choice is between reducing the amount of coursework or eliminating the exam. With the above arguments in mind, it is preferable to eliminate the exam.

1.2 Rationale for changes to course summary and syllabus.

These changes are meant to reflect the way the course is currently taught, which in turn reflects recent changes in the field. They designed to give more flexibility in the choice of which specific statistical models and machine learning methods are taught, since state-of-the-art in model design evolves rapidly. For example, factored models and class-based language models, which are mentioned on the syllabus, are not widely used. On the other hand, neural network encoder-decoder models have gained substantial traction in the last few years and have been the subject of intense interest in the industry, but they had not even been invented at the time the current syllabus was written.

2 Proposed changes to assessment

The new assessment will consolidate and extend changes that were introduced last year, with a set of courseworks in which students design and implement the major elements of a machine translation system, test them on real data, and attempt to improve them.[1] The courseworks were developed by the current lecturer for a course at Johns Hopkins University (taught in 2012 and 2014) and have since been adopted by all major universities that I am aware of with a course in machine translation, including Carnegie Mellon, the University of Pennsylvania, the University of Illinois, Bar Ilan University, and several others. They have also been used in several summer schools and a massively open online course. As a consequence of widespread use they have been substantially tested and improved.

[1] http://alopez.github.io/dreamt/

2.1 Coursework Design

Although the 2016 course only includes three courseworks, the full set includes four, each corresponding to a major subproblem in machine translation: **alignment**, **decoding**, **evaluation**, and **reranking**. From the more general perspective of AI, they emphasize the key problems of unsupervised learning, search, evaluation design, and supervised learning, respectively. For each problem the students are provided data, a naïve solution, and an evaluation program that compares program output to correct answers on a development subset of the input data. The programs are implemented in python with no external dependencies.[2] By default, each baseline system reads the test data and generates output in the evaluation format, so setup requires zero configuration, and students can begin experimenting with the code and adapting it immediately. For example, on receipt of the alignment code, aligning data and evaluating results requires only typing:

```
> python align.py | python score-alignments.py
```

This design ensures that students can always measure how their new implementations perform against a default algorithm on known data. Each coursework also includes unlabeled test data (except the decoding assignment, a combinatorial problem for which it is always possible to confirm the result). Test results are evaluated against a hidden key when assignments are submitted, and students are informed about how well their implementations compare against a benchmark algorithm produced by the instructor, as well as state-of-the-art solutions (though students are not required to reproduce the state of the art, which can require months or years of engineering). This encourages students to design robust solutions.

2.2 Assessed Skills

The courseworks are supplied with a set of detailed questions, which guide the students through a set of tasks.

1. Evaluate the behaviour of the default algorithm or model on real data and understand where it succeeds and fails.
2. Understand and implement a standard algorithm or model from a specification.
3. Motivate and design a new mathematical model or algorithm based on observations about the behaviour of the default and standard models or algorithms.
4. Implement the new model and evaluate its behaviour.

Each aspect of the assignment is assessed, so that the assessment is not overly biased towards one particular skill (e.g. data analysis, maths, or implementation). Furthermore, though the result of each coursework is a system component, they are independent, so that a failed implementation in one coursework does not adversely affect subsequent coursework. The markers provide feedback on all aspects of the assignment, as they are skills that will serve the students when they leave the university.

2.3 Hours

Based on this model, the projected hours breakdown is as follows.

- 20 hours lecture.
- 20 hours reading.
- 60 hours coursework, distributed across 4 assignments (15 hours each).

2.4 Learning Outcomes

The learning outcomes in the course descriptor will be slightly revised to make them more concise. The current learning outcomes are:

- Provide a written description of the main algorithms used in the system.
- Design and justify an approach to the evaluation of the system using state of the art tools and metrics.
- Analyse the data collected by such an evaluation.
- Where the system is designed to deal with large volumes of data the student should also be able to describe how the system handles large data volumes and critically compare the system's solution with other common solutions to the problem.

[2]Several significant machine translation research systems have been implemented in python, and with the trend towards systems built atop python neural network libraries like Theano and Tensorflow this is likely to continue.

- Identify where linguistics knowledge is relevant in the design of the system and what influence of linguistic knowledge has on the translation quality and performance of the system.

I propose revising these to:

- Understand the main linguistic challenges involved in machine translation.
- Understand state-of-the-art models and algorithms used to address challenges in machine translation.
- Design, implement, and apply modifications to state-of-the-art machine translation models algorithms.
- Understand the computational and engineering challenges that arise in the use of different models for machine translation.
- Understand, design and justify approaches to evaluation of machine translation systems.

The coursework addresses each of the learning outcomes.

Understand the main linguistic challenges involved in machine translation. In the course of the assignments, the students will be faced with many problems caused by the ambiguity of language, the many linguistic properties that cause data sparsity, and the many tradeoffs in expressivity and computational tractability of statistical models of language; so the coursework fulfills this learning outcome.

Understand state-of-the-art techniques used to address challenges in machine translation. A central component of each coursework is to implement a key algorithm in machine translation. To do so, the student must understand the algorithm; so the coursework fulfills this learning outcome.

Design, implement, and apply modifications to state-of-the-art machine translation algorithms. Each coursework includes a component in which students are required to implement something beyond a standard algorithm; so the coursework fulfills this learning outcome.

Understand the computational and engineering challenges that arise in the application of machine translation models. The questions supplied with each coursework specifically ask students to vary parameters of algorithms which give a stark picture of the tradeoffs in the speed, memory use, and accuracy of different statistical models. In designing their models, students must confront these problems directly; so the coursework fulfills this learning outcome.

Understand, design and justify approaches to evaluation of machine translation systems. Each coursework includes an objective evaluation measure that the students must understand in order to improve. Data analysis questions also ask the students to evaluate performance at a detailed level. Finally, a coursework on evaluation of systems requires students to confront this problem directly; so the coursework fulfills this learning outcome.

3 Proposed changes to course descriptor

I propose the following changes.

Current summary. Machine Translation deals with computers translating human languages (for example, from Arabic to English). The field is now sufficiently mature that Google use it to allow millions of people to translate Web Documents each day. This course deals with all aspects of designing, building and evaluating a range of state-of-the-art translation systems. The systems covered are largely statistical and include: word-based, phrase-based, syntax-based and discriminative models. As well as exploring these systems, the course will cover practical aspects such as using very large training sets, evaluation and the open problem of whether linguistics can be useful for translation.

Proposed summary. Google translate can instantly translates between any pair of over one hundred human languages like French and English. It and other translation systems translate by reading millions of words of already translated text and learning statistical models to translation new text. This course will show you how they work, and what work is still to be done. It focuses on the use of fundamental ideas from algorithms, machine learning, and linguistics, showing how they apply to a real and difficult problem in artificial intelligence.

Current syllabus

- History of MT
  - rule-based systems, ALPAC report, IBM models, phrase-based systems
- Models
  - word-based
  - phrase-based
  - syntax-based

  - discriminative
  - Factored Models

- Reordering
  - Lexicalised reordering
  - Distortion
  - Changing the source

- Language Modelling
  - Ngram models
  - Scaling LMs (cluster-based LMs, Bloom Filter LMs)

- Decoding
  - Knight on complexity, problem statement
  - Stack decoding

- Evaluation
  - Human evaluation
  - Automatic methods
  - NIST competitions

- Adding linguistics
  - Reranking
  - As factors

- Parallel corpora etc (data)
  - What they are, where they come from
  - Comparable corpora
  - Multi-parallel corpora

Proposed syllabus

- Statistical models of translation
  - Probabilistic models
  - Latent variable alignment models
  - n-gram language models
  - linear models
  - neural models

- Learning and inference for translation models
  - Maximum likelihood
  - Expectation maximization
  - Discriminative learning
  - Stochastic methods
  - Dynamic programming
  - Approximate search

- Linguistic phenomena and their associated modeling problems
  - Morphology
  - Syntax
  - Semantics

- Evaluation of machine translation systems
  - By humans
  - By machines
  - Use in design of loss functions for learning algorithms
- Engineering concerns
  - Scaling
  - Approximations
  - Efficient data structures
- History of machine translation