Learning Performance of a Machine Translation System: a Statistical and Computational Analysis

Marco Turchi, Tijl De Bie, Nello Cristianini

University of Bristol (UK)

Outlines

- Motivation.
- Experimental Setup.
- Experiments.
- Conclusion and discussion.

Motivation

- Performance of a learning system is result of (at least) two effects:
  - representation power of the hypothesis class:

    how well the system can approximate the target behaviour;

  - statistical effects:

    how well the system can estimate the best element of the hypothesis class.

Motivation

- They interact, with richest classes being better approximators of the target behaviour, but requiring more training data to identify the best hypothesis.

- In SMT, learning task is complicated by the fact that the probability of encountering new words or expressions never vanishes.

Motivation

- These observations lead us to analyze:
  - learning curves;
  - flexibility of the representation class;
  - stability of the model;
  - computational resources needed to train the system.

Experimental Setup

1. Role of training set size on performance on new sentences.
2. Role of training set size on performance on known sentences.
3. Effect on performance of increasing noise levels in parameters.
4. Computational Cost.

Experimental Setup

- Software
  - Moses.
  - Giza++: IBM model 1, 2, 3, and 4 with number of iterations for model 1 equal to 5, model 2 equal to 0, model 3 and 4 equal to 3.
  - SRILM: n-gram order equal to 3 and the Kneser-Ney smoothing algorithm.
  - Mert: 100 the number of nbest target sentence for each develop sentence.
  - Training, development and test set sentences are tokenized and lowercased.
  - Maximum number of tokens for each sentence in the training pair is 50.
  - TMs were limited to a phrase-length of 7 words. LMs were limited to 3.
- Data
  - Europarl Release v3 Spanish-English corpus.
  - Training set: 1,259,914 pairs.
  - Test and Development sets 2,000 pairs each.

Experimental Setup

- Evaluation Scores
  - BLEU, NIST, Meteor, TER, Unigram Recall, Unigram Precision, FMean, F1, Penalty and Fragmentation.
  - NIST is used as evaluation score after we observed its high correlation to the other scores on the corpus.

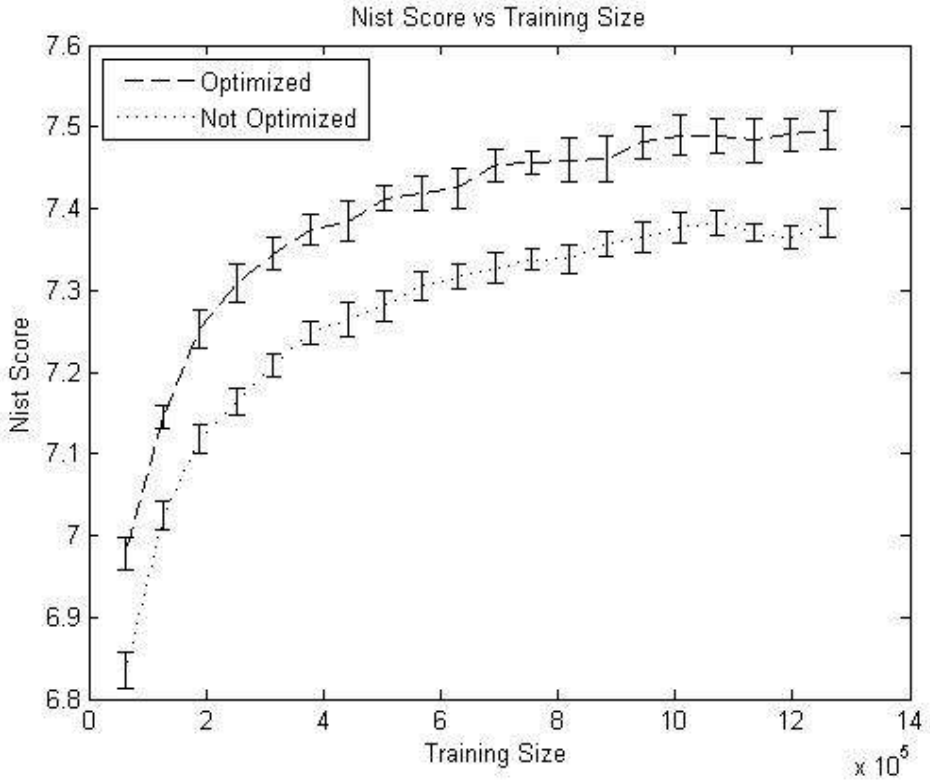
- Hardware
  - University of Bristol cluster machine, http://www.acrc.bris.ac.uk/acrc/hpc.htm.

Role of training set size on performance on new sentences.

- Analyse how performance is affected by training set size, by creating learning curves.
- Create subsets of the complete corpus by sub-sampling sentences from a uniform distribution, with replacement.
- Ten random subsets for each of the 20 chosen sizes (each size 5%, 10%, etc of the complete corpus).
- For each subset, a new instance of Moses has been created.
- Development and test sets contain 2,000 pairs each.

Role of training set size on performance on new sentences.

- The experiments have been run for the models with and without the optimization step.

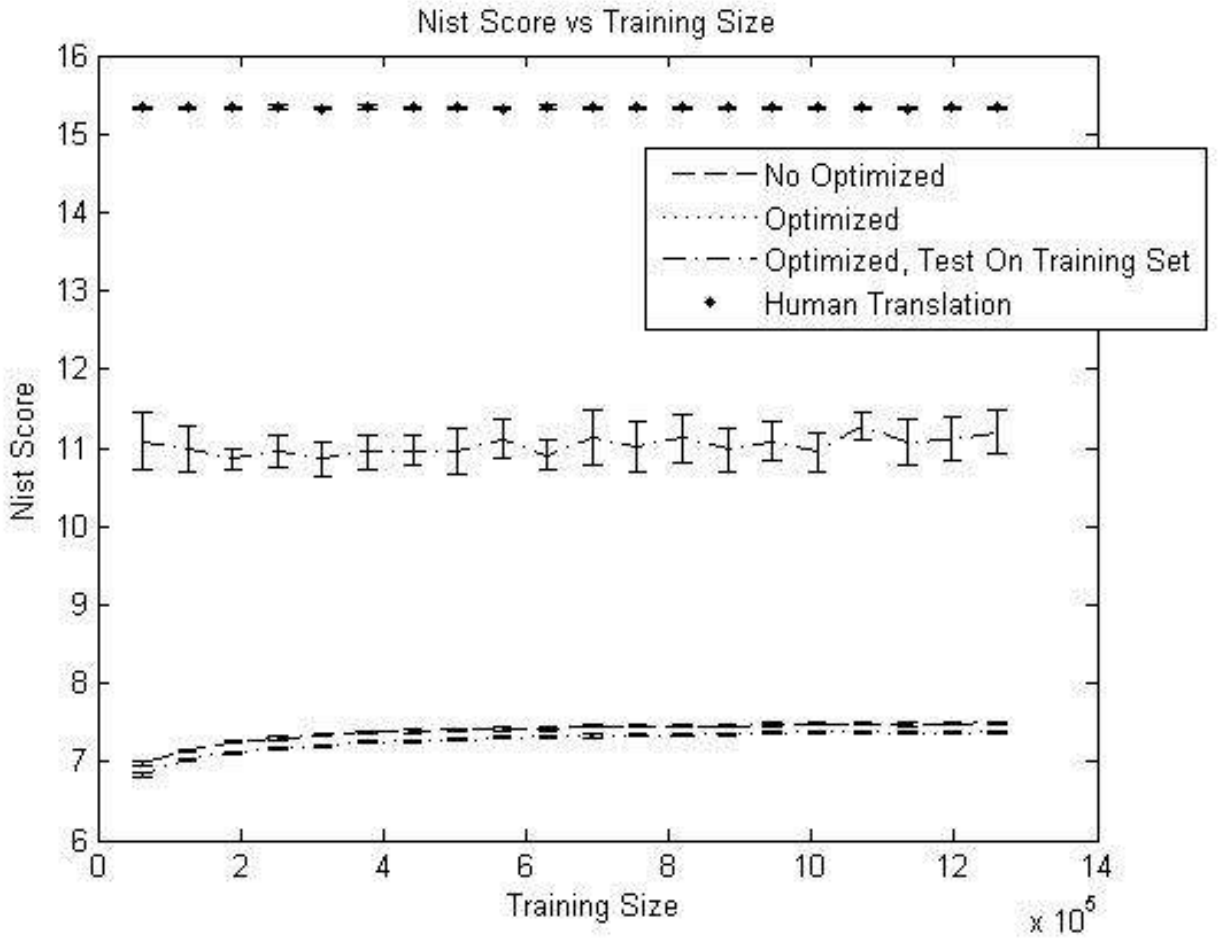


1. Addition of massive amounts of data result into smaller improvements.
2. Small error bars.
3. Benefits of the optimization phase.

Role of training set size on performance on known sentences

- Experiment much like the one described above.
- Key difference: the test set was selected randomly from the training set (2,000 pairs after cleaning phase).
- An upper bound on the performance achievable by this architecture if access to ideal data was not an issue.
- Performance on translating training sentences are not due to simple memorization of the entire sentence.
- ”Human Translation” identifies the curve obtained using the reference sentences as target sentences.

Role of training set size on performance on known sentences



Role of training set size on performance on known sentences

- If the right information has been seen, the system can reconstruct the sentences rather accurately.
- System can represent internally a good model of translation.
- It seems unlikely that good performance will ever be inferred by increasing the size of training datasets in realistic amounts.
- Process with which we learn the necessary tables representing the knowledge of the system is responsible for the performance limitations.

Effect on performance of increasing noise levels in parameters

- The training step results in various forms of knowledge: translation table, language model and parameters from the optimization.
- The internal models learnt by the system are lists of phrases, with probabilities associated to them.
- Which of these components is mostly responsible for performance limitations?

Effect on performance of increasing noise levels in parameters

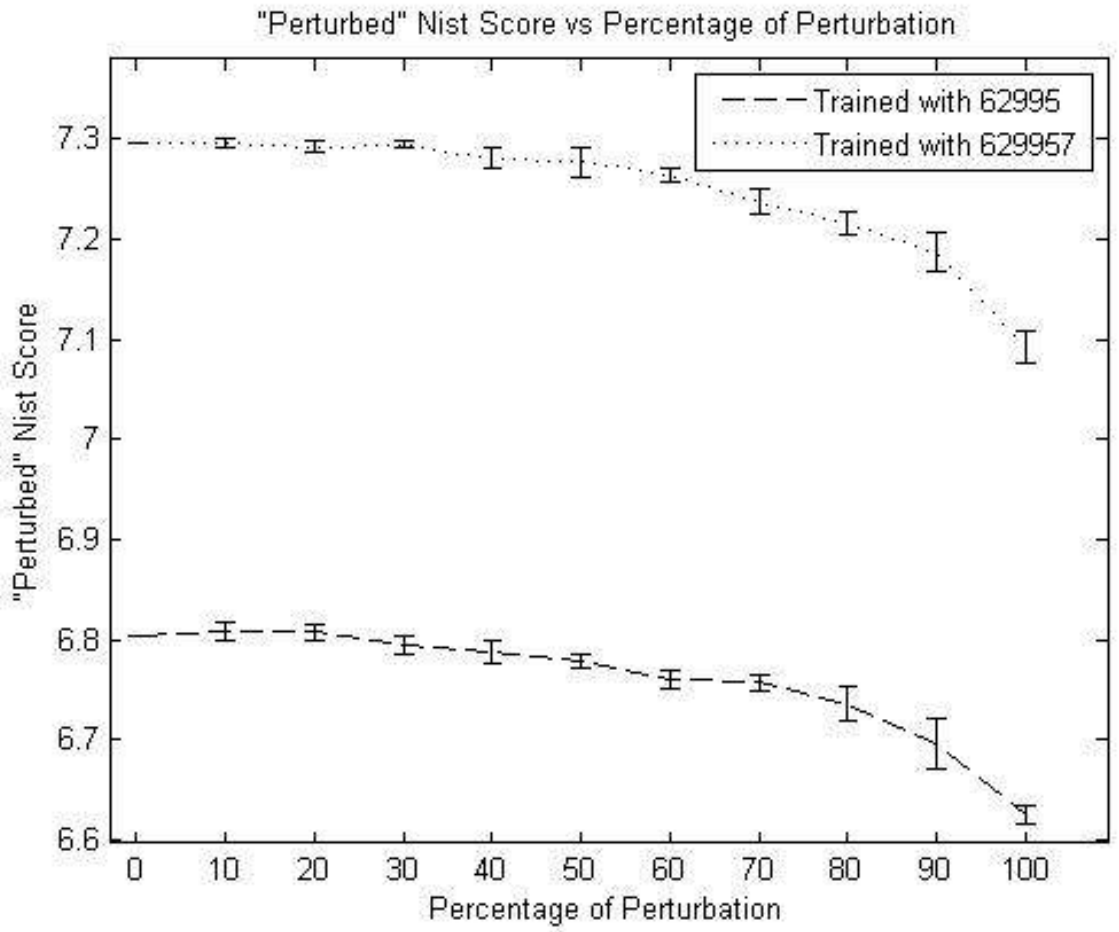
- In order to simulate the effect of inaccurate fine tuning of numeric parameters, a percentage of noise has been added to
  - each probability in the language model, including conditional probability and back off;
  - translation model, bidirectional translation probabilities and lexicalized weighting.

- We have measured the effect of this corruption on performance.

Effect on performance of increasing noise levels in parameters

- Given a probability *p* and a percentage of noise, *pn*, a value has been randomly selected from the interval [-*x*,+*x*], where *x* = *p* * *pn*, and added to *p*.
- This kind of noise does not alter the orders of magnitude of the probabilities. It modifies the less important digits of each probabilities.
- Two models trained with 62,995 and 629,957 pairs of sentences.
- For each model, for each value of *pn*, ten experiments have been run.
- Optimization step has not been run.

Effect on performance of increasing noise levels in parameters



Effect on performance of increasing noise levels in parameters

- Gentle decline in performance seems to suggest that fine tuning of parameters is not what controls the performance.

- Perhaps advanced statistical estimation or more observations of the same n-grams, would not lead to much better performance.

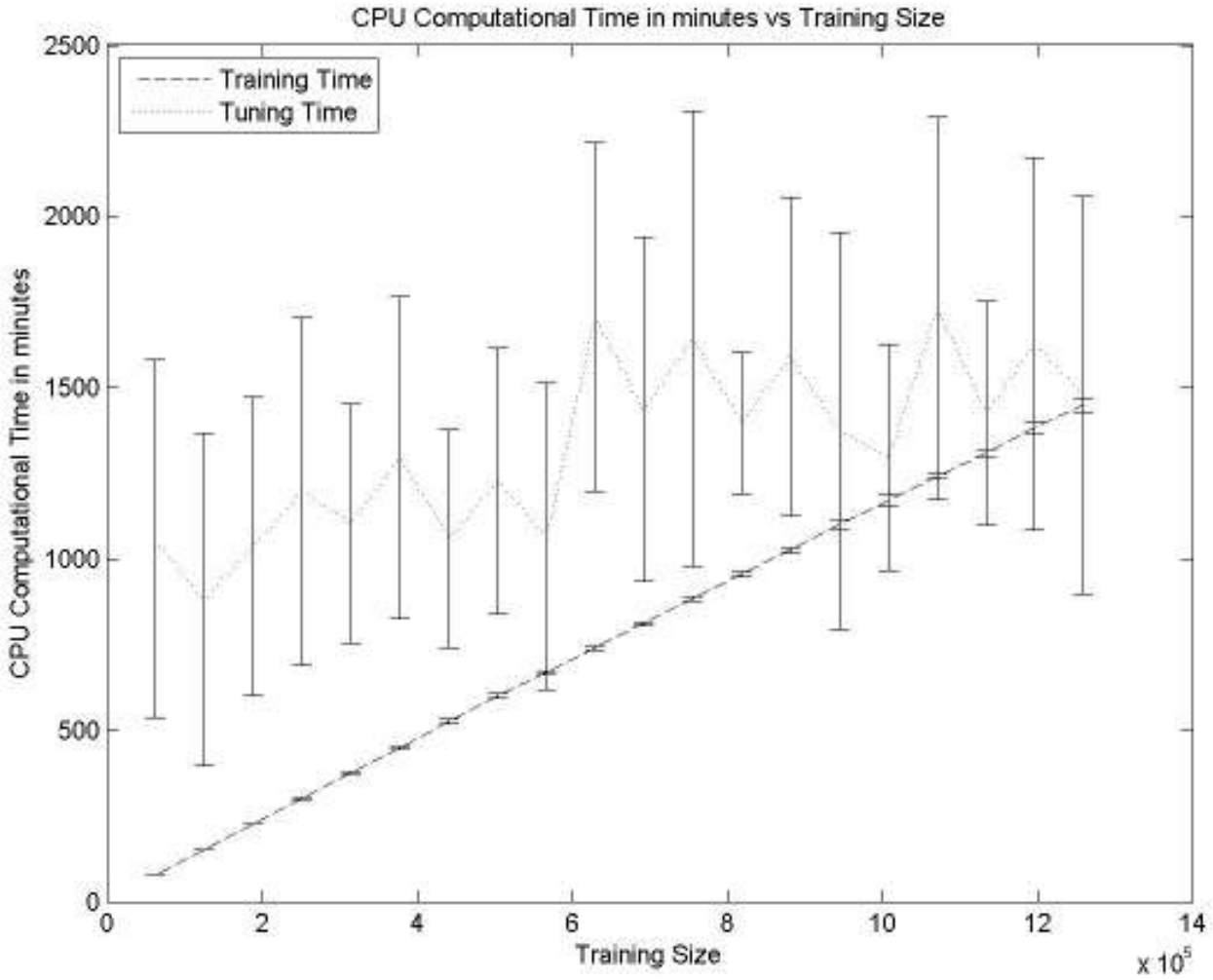
- More investigations are required in this direction.

Computational Cost

- The computational cost of models creation, development-phase and testing phase has been measured during the creation of the learning curves.

- The user CPU time is computed as the sum of the user time of the main process and the user time of the children.

Computational Cost



Computational Cost

- Despite its efficiency in terms of data usage, the development phase has a high cost in computational terms.
- Increasing the training size causes an increase in training time in a roughly linear fashion.
- Small development set size can require a large amount of tuning time.

Conclusion

- Our experiments suggest that:
  - Expressive power does not seem to be a limitation at the moment.
  - Fine tuning of parameters is not what controls the performance.
  - Adding more i.i.d. (independent and identically-distributed) data does not seem to produce particular increases in performance.
    - Way forward: involve changing data acquisition and incorporating linguistic constraints?
- Open Question:
  - Which of these components, phrases or probabilities, is mostly responsible for performance limitations?

THANKS

ANY QUESTIONS?