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

Marco Turchi, Tijl De Bie, Nello Cristianini

University of Bristol (UK)

Outline

- What is a Phrase Based Statistical Machine Translation (PBSMT) system?
- Motivation.
- Experimental Setup.
- Experiments.
- Conclusion and discussion.

Phrase Based SMT

- Given a foreign language sentence “f”, find the most probable translation “e”.
- “e” and “f” are split in phrases (set of consecutive words).
- Data driven: learns translations of words and phrases from parallel corpora.
- Language independent: the only thing we need is a parallel corpora.
- No need for language experts.
- Probabilities are determined automatically by training a statistical model using the parallel corpus.

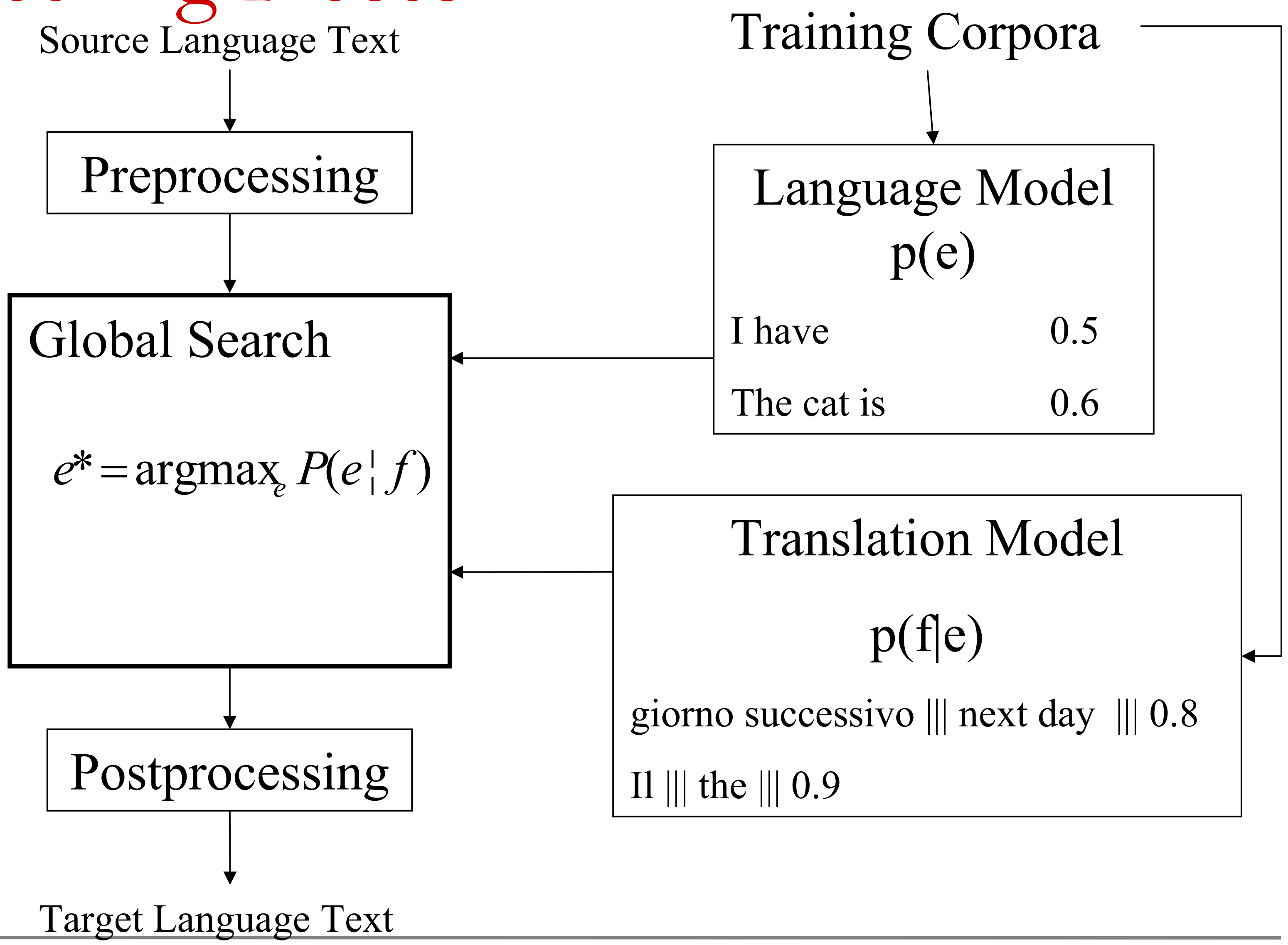
Parallel Corpus

I declare resumed the session of the European parliament adjourned on Friday 17 December 1999, and i would like once again to wish you a happy new year in the hope that you enjoyed a pleasant festive period.	Dichiaro ripresa la sessione del parlamento europeo, interrotta venerdi' 17 Dicembre e rinnovo a tutti i miei migliori auguri nella speranza che abbiate trascorso delle buone vacanze.
(the house rose and observed a minutes silence)	(il parlamento osserva un minuto di silenzio)

Three Big Pieces

- p(e) is the LANGUAGE MODEL (LM)
  - Assigns a higher probability to fluent/grammatical sentence.
  - Estimated using monolingual corpora.
- p(f|e) is the TRANSLATION MODEL (TM)
  - Assigns higher probability to sentences that have corresponding meaning.
  - Estimated using parallel corpora.
- GLOBAL SEARCHER:
  - Uses LM and TM to compute p(e|f).

Three Big Pieces



Source Language Text

Training Corpora

Preprocessing

Language Model

p(e)

I have	0.5
The cat is	0.6

Global Search

$$e^* = \text{argmax}_e\, P(e|f)$$

Translation Model

p(f|e)

giorno successivo ||| next day ||| 0.8

Il ||| the ||| 0.9

Postprocessing

Target Language Text

Phrase Based Model

- All components can be weighted differently

$$\hat{e} = \arg\max_{e} p(e \mid f) = \arg\max_{e} \prod_{i=1}^{I} p_{LM}(e_i \mid e_1 \ldots e_{i-1})^{\lambda_{LM}} \prod_{i=1}^{I} p_{TM}(\overline{f}_i \mid \overline{e}_i)^{\lambda_{TM}}$$

- In a general log-linear fashion:

$$p(x) = \exp \sum_{i=1}^{n} \lambda_i h_i(x)$$

- More features *h* can be added to the log linear model.

Motivation

- We study a PBSMT system as a learning system.
- Performance of a general learning system is result of (at least) two effects:
  - representation power of all the possible mathematical models that can approximate a human behaviour;
  - how well the system can estimate the best element of all the possible mathematical models (statistical effects).

Motivation

- They interact, with richest models being better approximators of the human behaviour, but requiring more training data to identify the best mathematical model.

- In SMT, the 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:
  - how performance change as function of the training data: learning curves;
  - flexibility of the chosen mathematical model;
  - computational resources in term of CPU time and hard drive space 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. Computational Cost.

Experimental Setup

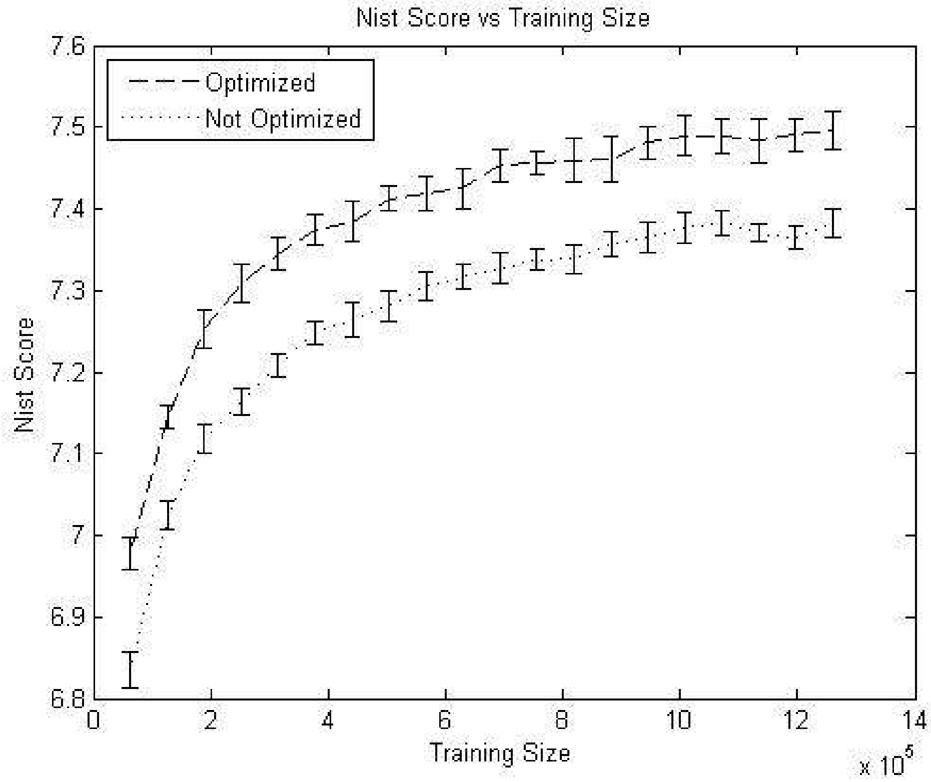
- Data: Europarl Release v3 Spanish-English corpus.
  - **Training set**: 1,259,914 pairs. Used to train the model and to compute the LM and TM probabilities.
  - **Development set**: 2,000 pairs. Used to estimated the best lambdas in log linear equation.
  - **Test set**: 2,000 pairs.
- Software
  - Moses, Giza++, SRILM. Each of these tools can not be parallelized.
- Evaluation Scores
  - BLEU, NIST, Meteor, TER. NIST is used as evaluation score.
- Hardware: ***University of Bristol cluster machine.***

Experimental Setup

- 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.
- Each set of experiments runs 200 instance of Moses.
- Each instance has been run on a single node of Bluecrystal. Each process uses more than 2 Gbs of memory, and 15 Gbs of hard drive space.

Role of training set size on performance on new sentences.

- Analyse how performance is affected by training set size, by creating learning curves.

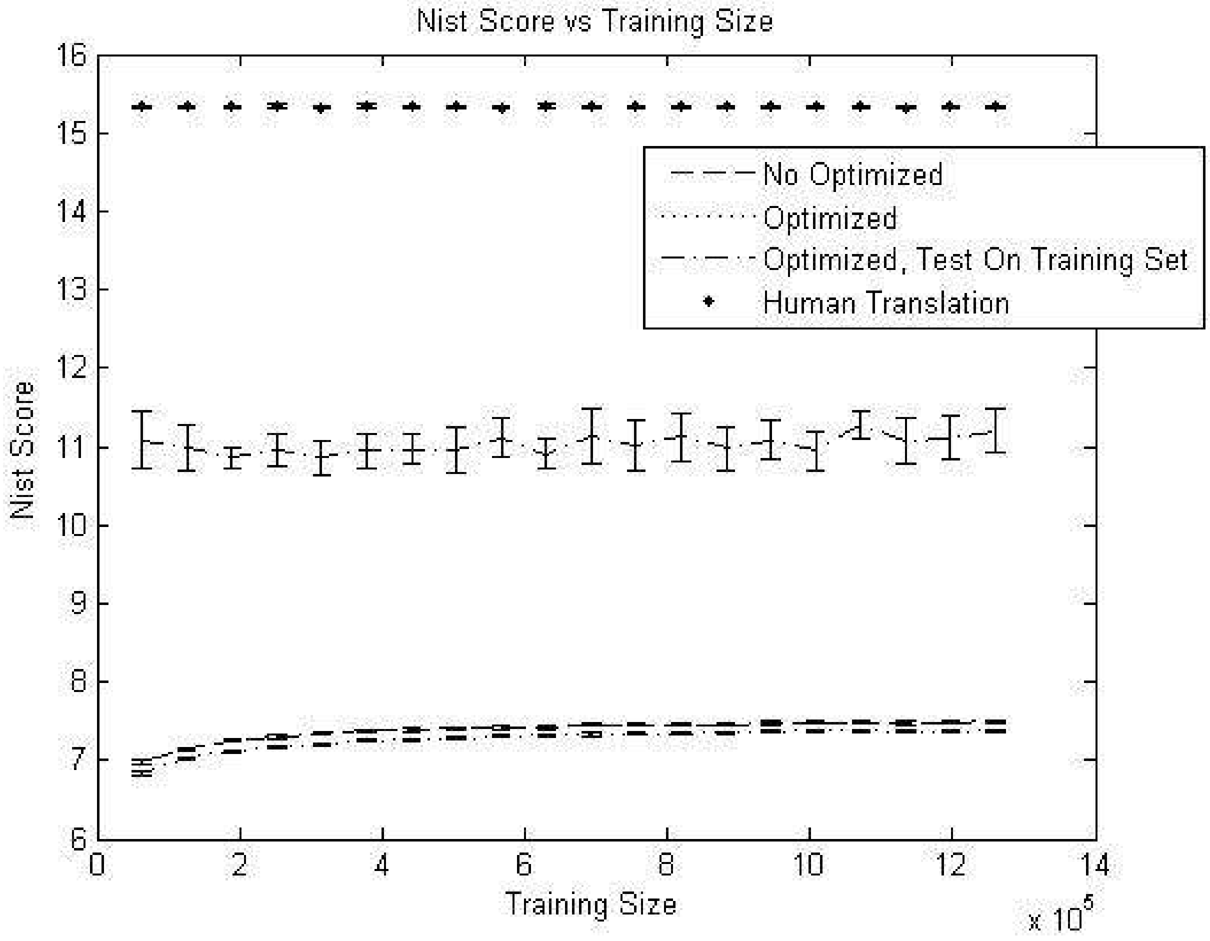


1. Addition of massive amounts of data result into smaller improvements.
2. Small error bars.
3. Benefits of the optimization phase.

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.
- ”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

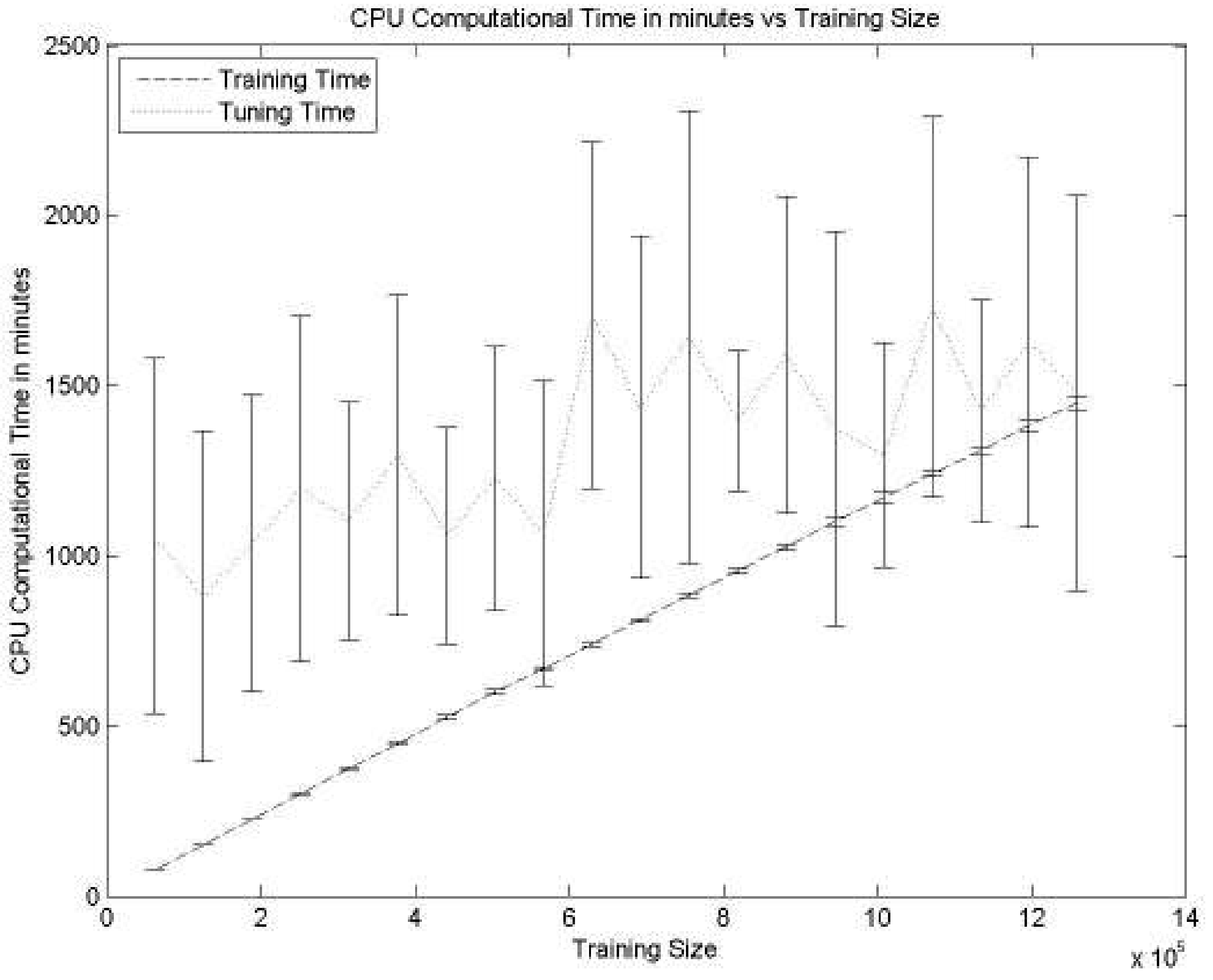
- If the right information has been seen, the system can reconstruct the sentences rather accurately.
- System can represent internally a good model of translation.
- It seems unlikely that good performance will ever be inferred by increasing the size of training datasets in realistic amounts.
- Process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

Computational Cost

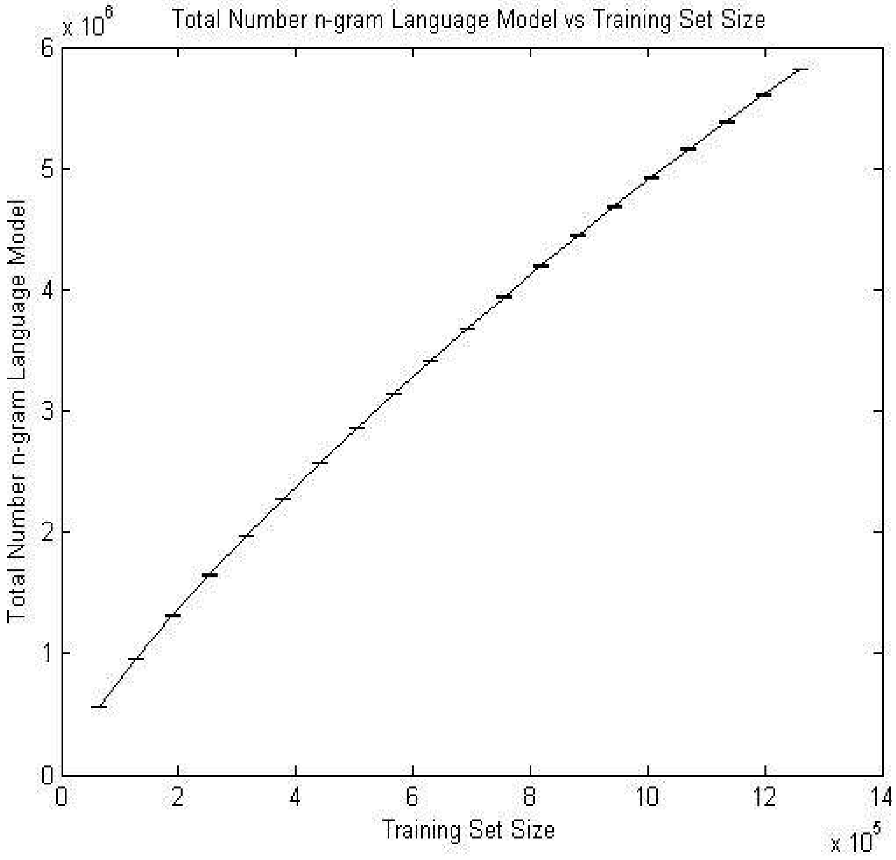
- The computational cost of models creation, development-phase and testing phase has been measured during the creation of the learning curves.

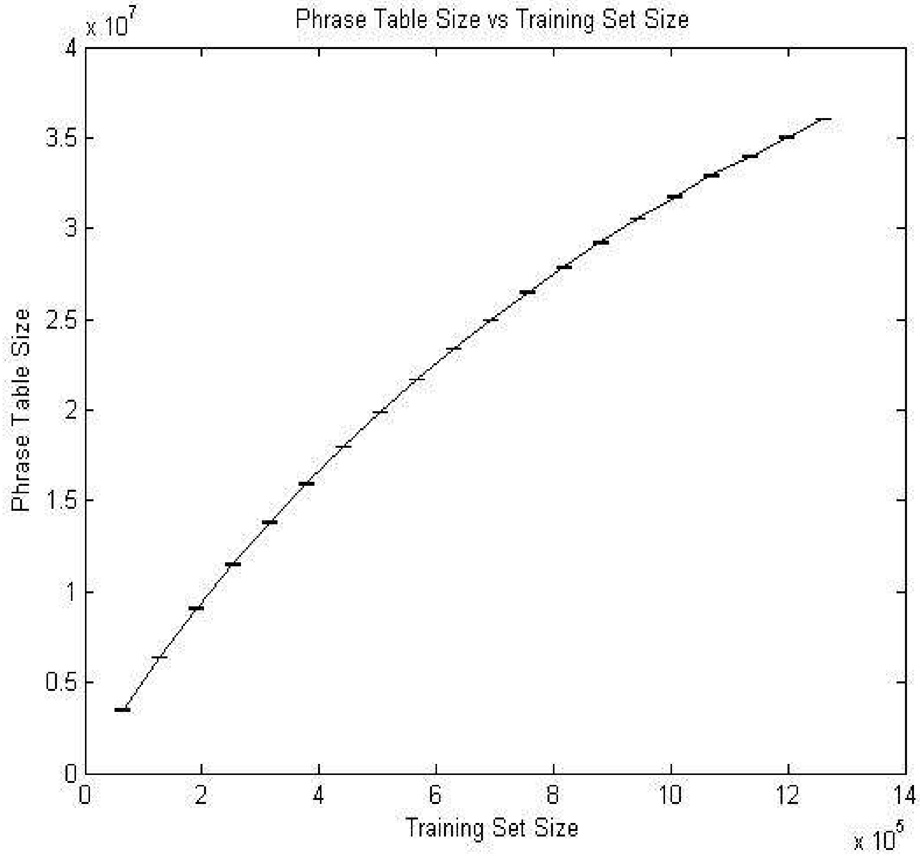
- Hard drive space used by the system has been analysed studying the dimension of Language and Translation models.

Computational Cost



Computational Cost





- Gigabytes and Gigabytes of hard drive space.

Computational Cost

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

Conclusion

- Our experiments suggest that:
  - The chosen model is not a limitation at the moment.
  - 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?
  - More data produces a substantial increasing in computational time and disk usage.

Conclusion

- We have created the highest accuracy learning curves that have never been proposed in SMT.
- We have run software for at least 400,000 CPU minutes (training, optimization, testing, others).
- We have submitted more than 1,000 processes.
- We have used more than 1.5Tb of hard disk space.
- It is only a small part of a more complete set of experiments that we are analysing.

This work would never be possible without BLUECRYSTAL !!!!!

Conclusion

- THANKS TO:
  - High Performance Computing Centre.
  - In particular to:
    - *Callum Wright*
    - Dr Paul Godwin

THANKS

ANY QUESTIONS?