N-gram-based Machine Translation enhanced with NNs for the French-English BTEC-IWSLT'10 task

Francisco Zamora-Martínez[1], **María José Castro-Bleda**[2], **Holger Schwenk**[3]
[1]Universidad CEU-Cardenal Herrera, Spain
[2]Universidad Politécnica de Valencia, Spain
[3]Université du Le Mans, France
fzamora@dsic.upv.es, mcastro@dsic.upv.es, Holger.Schwenk@lium.univ-lemans.fr

Introduction

▷ Statistical Machine Translation: maximum entropy approach, under a log-linear combination:

$$\hat{\mathbf{e}} = \text{argmax} \sum_{m=1}^{M} \lambda_m h_m(\mathbf{f}, \mathbf{e}),$$

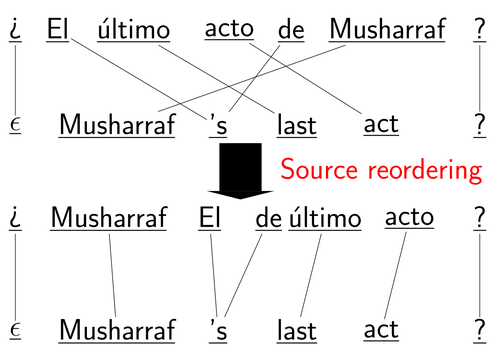
▷ N-gram-based SMT: Stochastic Finite State Transducer (SFST) trained as a N-gram Language Model (LM) of bilingual linguistic units (tuples).

▷ Neural Network Language Models (NNLMs): N-gram LM trained as a Neural Network, with a better smoothing due to a continuous space representation of words.

▷ The presented system enhances a standard state-of-the-art N-gram-based SMT system with NNLMs via log-linear combination. The novelty: NNLMs are fully integrated in the core of the search procedure of the system.

N-gram-based Machine Translation

▷ SFST trained as a N-gram LM of tuples extracted from a bilingual corpus after a segmentation and reordering procedure:



Extracted tuples		
(¿	,	ϵ)
(Musharraf	,	Musharraf)
(El de	,	's)
(último	,	last)
(acto	,	act)
(?	,	?)

▷ Maximum entropy approach adapted for tuples with a source sentence word reordering search.

▷ Log-linear combination of models: N-gram translation model (SFST following GIATI technique), N-gram target LM,

- N-gram translation model: a SFST trained from a bilingual tuple segmented corpus, following GIATI technique.
- N-gram target LM: trained from monolingual corpora.
- Lexicon direct and inverse translation models: based on IBM-1 models at the tuple level.
- Distortion model: penalization of long reorderings between tuples.
- Lexicalized reordering model: six models like in phrase-based approach implementation in Moses, but adapted for tuples.
- Word and tuple bonus models: penalization of number of words, and number of tuples.

▷ The N-gram translation model and N-gram target LM could be composed of a standard N-gram LM and a NNLM.

Neural Network Language Models

▷ Compute the probability of a sequence of words $\mathbf{s} = s_1 s_2 \ldots s_{|\mathbf{s}|}$ with a Neural Network following the same equation as N-grams:

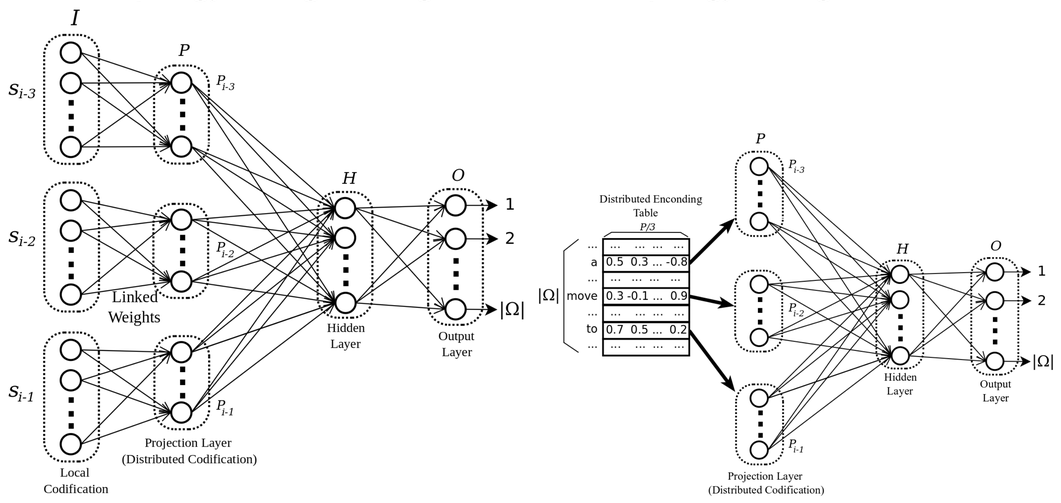
$$p(\mathbf{s}) \approx \prod_{i=1}^{|\mathbf{s}|} p(s_i | s_{i-1} s_{i-2} \ldots s_{i-N+1}) \,.$$

▷ Better smoothing than standard N-grams.

▷ Big computational problems due to the size of output layer and softmax activation function:

$$o_i = \frac{exp(a_i)}{\sum_{j=1,\ldots,|\Omega|} exp(a_j)} \,.$$

Topology during training **Topology during evaluation**



Integrating NNLMs in the decoding

▷ An important speed up of NNLMs evaluation is needed to fully integrate the model inside the decoder.

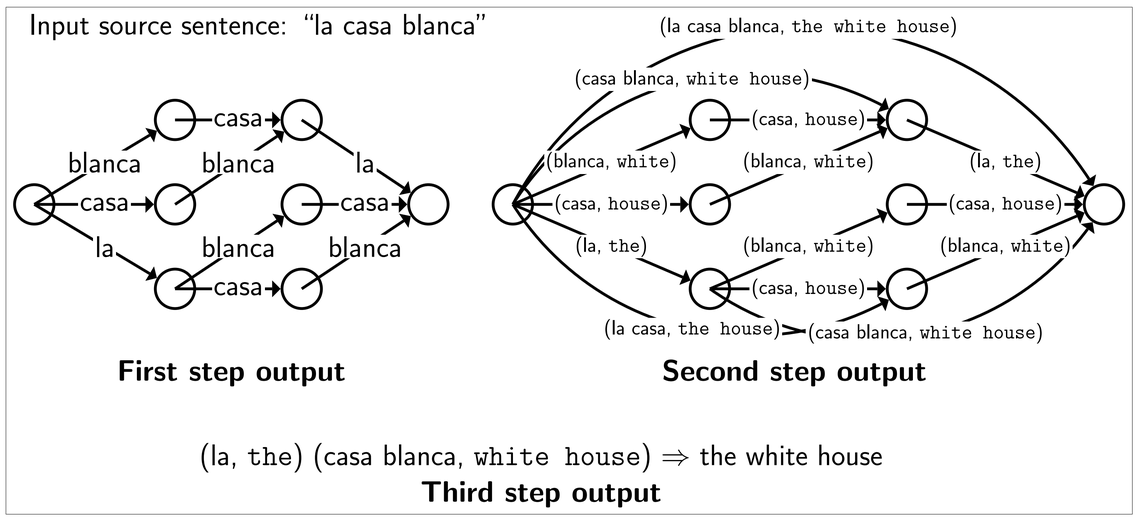
▷ The proposed approach consists on pre-computing and storing the softmax normalization constants most probably needed during the LM evaluation, since the cost of retrieving this value from a table is negligible compared with the cost of computing it.

- A space/time trade-off has to be considered.
- When a softmax normalization constant is not found, another simpler model (for instance, a lower order NNLM or statistical N-gram model) is used.

▷ Implementation of our own decoder inside the April toolkit. Decoding process in three steps:

1. Source word constrained reordering search graph, with histogram pruning.
2. Extension of search graph with tuples.
3. A Viterbi decoding with beam search and histogram pruning over the tuple extended search graph, with integrated NNLMs.

Input source sentence: "la casa blanca"



First step output **Second step output**

(la, the) (casa blanca, white house) ⇒ the white house
Third step output

Results

▷ System trained with BTEC and CSTAR'03 sets. Tuning done by means of MERT over the IWSLT'04 set. IWSTL'05 set used as internal test set for select the best system.

▷ BLEU Tuning results for the Dev2 (IWSLT'04) and Dev3 (IWSLT'05) partitions. The number of processed source words per second was measured for each system. NNTM is the addition of the NN for the N-gram translation model, NNTLM is the addition of the NN for the N-gram target LM, and R is the addition of the lexicalized reordering model.

▷ Official BLEU Test results for the test sets IWSLT'09 (Test1) and IWSLT'10 (Test2) partitions, primary results are bolded. Every result is with the integrated system. System retrained over BTEC + all available development data:

Tuning results

System	Dev2	Dev3	Ws/Sec.
N-gram-based	65.8	65.2	21
+ R	65.8	65.4	21
Integrating NNLMs in the decoder			
+ NNTLM	67.0	65.7	8
+ NNTM	66.6	65.4	10
+ NNTLM + NNTM	67.4	66.3	7
+ NNTLM + NNTM + R	67.7	67.1	7
Rescoring 1000-best list			
+ NNTLM + NNTM	66.9	66.4	–
+ NNTLM + NNTM + R	67.8	66.6	–

Test results

System	Test1	Test2
N-gram-based + R	61.4	52.2
+ NNTLM + NNTM + R	62.5	54.3
Adding all available data		
N-gram-based + R	61.8	51.2
+ NNTLM + NNTM + R	**63.6**	**53.6**

Conclusions

▷ An N-gram-based SMT system enhanced with NNLMs for the French-English BTEC task of the IWSLT'10 evaluation campaign was presented.

▷ Improvement between 1.8 and 2.4 BLEU points was obtained.

▷ The performance in time of the NNLMs integrated system only decreased 3 times over the baseline.

▷ The integration improves in 0.5 points of BLEU the rescoring scheme.

▷ System positioned as second in the preliminary automatic evaluation results.

Partially supported by the Spanish Ministerio de Ciencia e Innovación, HITITA (TIN2010-18958)