Continuous Space Translation Models for Phrase-Based Statistical Machine Translation

Holger Schwenk
University of Le Mans
Avenue Laennec
72085 Le Mans Cedex, France
Holger.Schwenk@lium.univ-lemans.fr

ABSTRACT
This paper presents a new approach to perform the estimation of the translation model probabilities of a phrase-based statistical machine translation system. We use neural networks to directly learn the translation probability of phrase pairs using continuous representations. The system can be easily trained on the same data used to build standard phrase-based systems. We provide experimental evidence that the approach seems to be able to infer meaningful translation probabilities for phrase pairs not seen in the training data, or even predict a list of the most likely translations given a source phrase. The approach can be used to rescore *n*-best lists, but we also discuss an integration into the Moses decoder. A preliminary evaluation on the English/French IWSLT task achieved improvements in the BLEU score and a human analysis showed that the new model often chooses semantically better translations. Several extensions of this work are discussed.

KEYWORDS: Statistical machine translation, phrase probability estimation, continuous space models, neural network.

Proceedings of COLING 2012: Posters, pages 1071–1080,
COLING 2012, Mumbai, December 2012.

1 Introduction

In the statistical approach to machine translation (SMT), all models are automatically estimated from examples. Let us assume that we want to translate a sentence in the source language **s** to a sentence in the target language **t**. Then, the fundamental equation of SMT is:

$$\mathbf{t}^* = \arg\max_{\mathbf{t}} P(\mathbf{t}|\mathbf{s}) = \arg\max_{\mathbf{t}} P(\mathbf{s}|\mathbf{t})P(\mathbf{t})/P(\mathbf{s}) = \arg\max_{\mathbf{t}} P(\mathbf{s}|\mathbf{t})P(\mathbf{t}) \quad (1)$$

The translation model $P(\mathbf{s}|\mathbf{t})$ is estimated from bitexts and the language model $P(\mathbf{t})$ from monolingual data. A popular approach are phrase-based models which translate short sequences of words together (Koehn et al., 2003; Och and Ney, 2003). The translation probabilities of these phrase pairs are usually estimated by simple relative frequency. We are only aware of few works to perform more sophisticated smoothing techniques, for instance (Foster et al., 2006). The log-linear approach is commonly used to consider more *feature functions* (Och, 2003). In the Moses system, four feature functions are usually used for the translation model: the forward and backward phrase translation probabilities and lexical probabilities in both directions. These four feature functions together could be seen as a particular smoothing technique of the translation model. In other works, hundreds or thousands of features are used.

The dominant approach in language modeling are so-called back-off n-gram models. An alternative approach was proposed by (Bengio and Ducharme, 2001; Bengio et al., 2003). The basic idea is to project the words into a continuous space and to perform the probability estimation in that space. The projection as well as the estimation can be jointly performed by a multi-layer neural network. The continuous space language model (CSLM) was very successfully applied to large vocabulary speech recognition, and more recently to SMT, e.g. (Schwenk et al., 2006; Le et al., 2010; Zamora-Martínez et al., 2010; Schwenk et al., 2012).

Given this success for language modeling, there were also two attempts to apply the same ideas to the translation model. Both were developed for tuple-based translation systems, e.g. based on bilingual units. This allows to to see the translations model like a standard n-gram LM task and it is straight forward to apply the CSLM (Schwenk et al., 2007). In the second work, this idea was improved by considering different factorization of the joint probability, in particular word-based ones: the principal idea is to predict the probability of a target word given the context of the previous source and target words (Le et al., 2012). The authors report good improvements in the BLEU scores for several tasks, but the approach seems to be complicated, in particular with respect to the training of the model or direct integration into the decoder.

In this work we propose a generic architecture which can be used in the standard pipeline to build a phrase-based SMT system. The continuous space translation model (CSTM) is trained on exactly the same data, i.e. the so-called `extract` files and no additional word alignments, segmentation, etc is necessary. In machine learning, we are generally not interested in memorizing perfectly the training data, but in learning the underlying structure of the data, and being able to generalize well to unseen events. We give experimental evidence that the architecture proposed in this paper can provide meaningful probability estimations for new *phrase pairs* which were not seen in the training data. The architecture can be also used to provide a list of the most likely translations given (an unseen) source phrase.
Our implementation is based on the open-source CSLM toolkit described in (Schwenk, 2010; Schwenk et al., 2012). This allows us to take advantage of all the possibilities of this software, in particular weighted resampling of large corpora and fast training on GPU cards.[1].

[1] `http://wwww.lium.univ-lemans.fr/~cslm`

2 Architecture

Let us first the recall the principles of the CSLM, using the same notion as (Schwenk, 2007). The inputs to the neural network are the indices of the $n-1$ previous words in the vocabulary h_j and the outputs are the posterior probabilities of *all* words of the vocabulary: $P(w_j = i|h_j), \forall i \in [1,N]$, where N is the size of the vocabulary. The input uses the so-called 1-of-n coding, i.e., the ith word of the vocabulary is coded by setting the ith element of the vector to 1 and all the other elements to 0. The ith line of the $N \times P$ dimensional projection matrix corresponds to the continuous representation of the ith word. These continuous projections of the words are concatenated. This layer is followed by one or more tanh hidden layers. The output layer uses a softmax normalization. The value of the ith output neuron is used as the probability $P(w_j = i|h_j)$. Training is performed with the standard back-propagation algorithm minimizing the cross-entropy between the output and the target probability distributions and a weight decay regularization term. The CSLM has a much higher complexity than a back-off LM, in particular because of the high dimension of the output layer. We use the option proposed by the CSLM toolkit to limit the size of the output layer to the most frequent words (short list). All the words are still considered at the input layer. Other options are explored in (Le et al., 2011).

2.1 Continuous space translation model

The central question of a phrase-based SMT system is how to estimate the probability of a phrase-pair. In practice, the length of the phrases is limited to a small value, e.g. $p,q \in [1,7]$.

$$P(\tilde{\mathbf{t}}|\tilde{\mathbf{s}}) = P(t_1 \dots t_p|s_1 \dots s_q) \tag{2}$$

This equation can be factorized as follows:

$$\begin{aligned} P(t_1,\dots,t_p|s_1,\dots,s_q) &= P(t_1|t_2,\dots,t_p,s_1,\dots,s_q) \times P(t_2,\dots,t_p|s_1,\dots,s_q) \\ &= P(t_1|t_2,\dots,t_p,s_1,\dots,s_q) \times P(t_2|t_3,\dots,t_p,s_1,\dots,s_q) \times P(t_3,\dots,t_p|s_1,\dots,s_q) \quad (3) \\ &= \prod_{k=1}^{p} P(t_k|t_{k+1},\dots,t_p,s_1,\dots,s_q) \approx \prod_{k=1}^{p} P(t_k|s_1,\dots,s_q) = \prod_{k=1}^{p} P(t_k|\tilde{\mathbf{s}}) \quad (4) \end{aligned}$$

At a first look, our model seems to be based on the approximation in the last line of the above equation, i.e. we drop the dependence between the target words. By these means we actually get p independent "n-gram models" which try to predict the kth word in the target phrase given all the words of the source phrase $\tilde{\mathbf{s}}$. This naturally leads to the neural network architecture depicted in Figure 1 left. Note that there are no constraints to use the same vocabulary at the input and the output of the neural network. In this first architecture, we do not use p completely independent neural networks, but all the target words share the same projection of the words of the source phrase into the continuous space. This idea can be pushed further by adding one common hidden layer (see Figure 1 middle). Both architectures are trained by the same back-propagation algorithm than the CSLM – we just have a target vector for each output layer. The common hidden layer forces the neural network to learn a distributed representation suitable to predict each one of the p words in the target phrase. We argue that this re-introduces a dependence between the target words which we had initially dropped in equation 4. This can even be made more explicit with neural network architectures like the one in Figure 1 right.

Currently, we have only performed experiments with architecture depicted in the middle of Figure 1, using seven words at the input and output respectively. In practice, many phrase

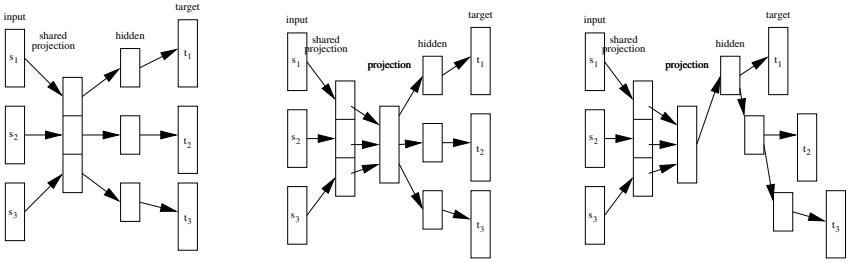

Figure 1: Different neural network architectures for a continuous space translation model. In this example, the input and output phrases are limited to a size of three words each. Left: simple extension of the CSLM. Middle: addition of a common hidden layer in order to introduce a dependence between the target words. Right: hierarchical dependence.

pairs are shorter since long phrases rarely match new test data. This is handled as follows. For an incomplete source phrase, i.e. with less than seven words, we set the projections of the "missing" words to zero. By these means, they have no influence on the calculation of the subsequent neural network layers. We could also use a special NULL word token whose projections are initialized to zero. This may enable the neural network to learn a different, more suitable, projection. Incomplete target phrases at the output layer are handled by simply not back-propagating a gradient for the "missing" words. In future work, we will also investigate the use of a special NULL word token at the output layer. By these means, we can try to learn the length of the target phrase for a given input phrase.

We have performed some initial experiments to analyze whether predicting multiple words is a difficult task for the neural network. For these experiments, we took a small corpus of phrase pairs of length of up to three source and two target words. We first trained a neural network with only one output layer to predict the first target word. This is actually a continuous space *language* model with different input and output vocabularies. Therefore, we can calculate the perplexity to measure its quality. Another neural network of the same architecture is used to predict the second target word. The results are given in Table 1, first two rows. We then trained a neural network on the same data to predict both target words, but evaluating the perplexity of only one target word (first or second). As can be seen in Table 1, 3rd row, this led to lower perplexities. From these experiments we can conclude that predicting multiple words is actually better than predicting separately individual words. The individual output layers of the NN seem to benefit of the gradient back-propagated to the common hidden and projection layer.

To measure the quality of the prediction of a phrase, i.e. a sequence of words, we define the *multi-word perplexity* as $ppl = e^{-H}$, using the approximation of equation 4:

$$\begin{aligned} H &= \frac{1}{n}\sum_e \log P(\tilde{\mathbf{t}}^{(e)}|\tilde{\mathbf{s}}^{(e)}) \approx \frac{1}{n}\sum_e \log \sqrt[p]{\prod_{k=1}^{p} P(t_k^{(e)}|\tilde{\mathbf{s}}^{(e)})} = \frac{1}{n}\sum_e \frac{1}{p}\sum_{k=1}^{p} \log P(t_k^{(e)}|\tilde{\mathbf{s}}^{(e)}) \\ &= \frac{1}{p}\sum_{k=1}^{p} H_k \quad \text{with} \quad H_k = \frac{1}{n}\sum_e \log P(t_k^{(e)}|\tilde{\mathbf{s}}^{(e)}) \end{aligned} \tag{5}$$

Therefore, the multi-word perplexity of a phrase pair is identical to the geometric mean of

Architecture	Perplexity Target 1	Target 2
Separate network which predicts first target word only	64.0	n/a
Separate network which predicts second target word only	n/a	81.9
One network which predicts both target words	62.8	80.8
multi-word perplexity:	71.3	

Table 1: Comparison of single and multi-word prediction (networks with 1 or 2 output layers)

the perplexity when predicting the individual target words separately. This is experimentally verified (see last line of Table 1). Note that this measure is pretty meaningless for classical phrase-tables with probability estimates obtained by relative frequency. Many phrase-pairs are singletons and their translation probability is estimated to be 1.0.

3 Experimental evaluation

First experimental results were performed on the data of the 2011 IWSLT evaluation which addressed translation of public lectures from English into French. The main resource provided for this task is a parallel corpus of about 100k sentences (2M words). A development and test corpus with one reference translation is also available (see Table 2). For LM we used the French side of the bitext and about 1.3G words of the LDC Gigaword corpus and other provided news corpora. According to the system description of the best performing system (Rousseau et al., 2011), adding more (out-of domain) parallel training data yields only small improvements. Our baseline phrase-based system achieves a BLEU score of 23.12 on the development and 24.84 on the test data, respectively. This system is build with the Moses toolkit using default parameters. Fourteen feature functions were used: five scores for the reordering model, four scores for translation model, a LM score, a distortion, phrase and word penalty. The coefficients of these feature functions are tuned with MERT to maximize the BLEU score.

In our initial experiments, we trained a neural network to estimate the forward phrase translation probability $P(\tilde{\mathbf{t}}|\tilde{\mathbf{s}})$. The maximal phrase length was set to seven words, as it is also used during the standard phrase extraction process. The extraction process of the Moses toolkit produced 7M phrase pairs after word alignment of the 100k parallel sentences. The resulting phrase table has five 5M phrase-pairs. Our neural network has the following architecture: a 320 dimensional projection layer for each input word, one common hidden layer of dimension 768, one 512 dimensional additional hidden layer for each output, and seven output layers of dimension 16384 (we use the mechanism of a short list provided by the CSLM toolkit). In our case, a phrase is processed by the neural network when all the target words fall into the short list. This was the case for about 93% of the observed phrases in the training data. Note that the source and target vocabulary of the translation model are much smaller than the one of the LM (about 50k). Phrase pairs which are not processed by the CSTM are obtained from a classical phrase table. We used binary phrase-tables which are kept on disk. Training of this configuration takes about 2 days on a standard multi-core server and less than 10 hours on a GPU card.

We experimented with three different possibilities to use the translation model probabilities provided by the neural network: 1) evaluation of the CSTM to provide meaningful probabilities for unseen phrase pairs; 2) rescoring of *n*-best lists provided by Moses; and 3) direct integration into the decoding algorithm of Moses. These options are discussed in the following sections.

Corpus	#lines	#words English	#words French
Train (TED)	107k	1.8M	2.0M
Dev	2026	36.6k	39.0k
Test	572	9.0k	9.1k

Table 2: Statistics of the parallel corpora provided for the 2011 IWSLT evaluation.

Source phrase	Translation provided by the CSTM
a nice car	une jolie voiture
a nice bike	un vélo sympa
a nice woman	une jolie femme
a nice garden	un joli jardin
a nice man	un homme sympa

Table 3: Example translations proposed by the CSTM. All these phrase-pairs were not in the training data. No target LM was used.

3.1 Generalization to new phrase-pairs

Table 3 shows some examples of phrase pairs and the most likely translation provided by the CSTM. These translations are obtained by selecting the output target words with the highest probability. It is important to note that none of these phrase pairs are included in the training data. The standard Moses phrase table only contains many single word-to-word translations of the words *nice*, *car*, *bike*, *woman*, *garden* and *man*. Therefore, the full translation process must completely rely on the LM to select the best individual translations of the three words so that a correct French sentence will be created. It can be clearly seen that the CSTM does not seem to perform a simple word-by-word translation. In our setting, we translate from English into French, a morphologically rich language. French has two genders and the adjectives must be adapted to the noun. In our example, the source phrases only differ by the third word, but this induces changes of all the three words in the target phrase because of the morphology of the French language. The CSTM was able to produce in all cases the correct translation and to adapt the article and adjective to the noun. Note that this is obtained without an additional LM on the target words. We interpret this is as experimental evidence that our architecture is able to capture relations between the target words. The CSTM can also propose *word reorderings*: in some translations the adjective is correctly placed in front of the noun (e.g. *une jolie voiture*), and in others behind the noun (e.g. *un vélo sympa*).

3.2 Rescoring n-best lists

The CSTM can be used to rescore n-best lists produced by the baseline system. This is in analogy to the use of the continuous space *language* model which is usually not integrated into beam search. Rescoring the translation model probabilities requires the phrase alignments in the n-best list. Table 4 gives some statistics on this process. Although there are almost 17M requests for phrase translation probabilities, only 53k were actually different. We take advantage of this redundancy to speed-up the translation model rescoring. In our case, this takes less than five minutes, but half of the time is actually needed to load and parse the n-best lists. The CSTM processes almost 95% of the probability requests, this means using a short list is not a limitation. The forward phrase probability estimated by the neural network is added as 15th score and the coefficients are retuned with MERT. By these means we were able to achieve an improvement of about 0.3 BLEU on the development and 0.2 BLEU on the test set (see Table 5). This is not a huge improvement, but we have changed only one score of the phrase-table. Exactly the same approach can be used to estimate the inverse phrase translation probability $P(\mathbf{s}|\mathbf{t})$.

3.3 Integration into the decoder

As far as we know, there is only one attempt to integrate the CSLM directly into the translation process (Zamora-Martínez et al., 2010). This is in fact tricky since many LM probabilities are requested and it is not straight forward to delay a bunch of requests so that we can use the CSLM more efficiently. A possible implementation could be based on the work on distributed LMs, for instance (Brants et al., 2007). Previous works on continuous space translation models in an bilingual tuple system only used rescoring (Schwenk et al., 2007; Le et al., 2012). On the other hand, it seems to be easier to integrate the approach proposed in this paper directly into the decoder. When translating a sentence, Moses first enumerates all the possible segmentations of the source sentence, given the known phrases in the phrase-table. Once all those probabilities are obtained, the translation model is not queried any more. This process largely simplifies the use of continuous space methods. Two options come to mind: 1) keep the segmentations proposed by the classical phrase table, but get the translation probabilities from the CSTM instead of the phrase-table; or 2) don't use a classical phrase-table any more, but request all possible phrase-pairs directly from the CSTM. The integration of the phrase-table into the decoder according to the first option can be in fact simulated by creating a "fake" phrase-table that has the same form than the standard Moses phrase-table, but replacing the probabilities with the ones calculated by the CSTM. Alternatively, we could add the CSTM probability as an additional feature function. The results of these experiments are summarized in Table 5. When replacing the forward translation model probability in the phrase-table, we achieve a slight improvement of the BLEU score on the test data, in comparison to rescoring n-best lists (25.03→25.13). This seems to indicate that the CSTM probabilities trigger the exploration of new paths during the beam search which were pruned in the n-best list.

Some example translations are provided in Figure 2. In the first example, the English word *"right"* can have several translations which have different meanings: *"à droite"* if we are referring to a direction, *"correct"*, etc if we approve something, and *"pas vrai ?"* or *"non ?"* if we ask for confirmation. The CSTM made the right choice, but this did not improve the BLEU score. The same observations hold for the second example where the CSTM selects the correct translation of *"little"*. In the third example, the phrase-based baseline system fails and performs a very bad word by word translation. The CSTM provides a much better translation.

When using the CSTM to completely replace a phrase-table, we are able to apply phrases of any length at each position in the source phrase. Limiting the source phrase length to the usual seven words, there are at most $7 \times q$ possible segmentations of the source sentence into phrases, where q is the length of the source sentence. For each of these segmentations, the CSTM could provide a large number of translations. The goal would be to obtain an ordered list of the most likely translations given an input source. Initial work has shown that the CSTM can provide a meaningful list of possible translations, but this list also contains wrong translations. We

Nb of sentences	2026
Nb of phrase pairs	16.8M
Nb of different phrase pairs	53k
Aver. nb of phrases per sentence	10.3
Phrases processed by CSTM	94.9%

Table 4: Statistics of rescoring 1000-best lists with the CSTM.

System	Dev	Test
Baseline	23.12	24.84
CSTM rescoring	23.45	25.03
CSTM decode	23.13	25.13

Table 5: n-best rescoring versus integrating the CSTM into beam-search (BLEU scores).

SRC:	now this sounds crazy **right**
BASE:	cela peut paraître fou **à droite**
CSTM:	c'est fou **pas vrai**
REF:	cela paraît incroyable **non**
SRC:	and sometimes a **little** prototype of this experience is all that it takes to ...
BASE:	et parfois un **peu** prototype de cette expérience est qu'il faut pour ...
CSTM:	et parfois un **petit** prototype de cette expérience est qu'il faut pour ...
REF:	et parfois un **petit** prototype de cette expérience sera la seule chose qui ...
SRC:	but the things i constantly hear far too many chemicals pesticides ...
BASE:	mais **les choses je constamment entendre** bien trop de produits chimiques ...
CSTM:	mais **ce que j'entends constamment** beaucoup trop de produits chimiques ...
REF:	**ce que j'entends souvent** c'est trop de produits chimiques ...

Figure 2: Example translations of the test set: English source, translation provided by the baseline systems, decoding with a CSTM, and reference translation.

are currently experimenting with various thresholds to discard unreliable translation options. Finally, it is also possible to combine both options to integrate the CSTM into the beam-search: keep the original segmentations of the source sentence into phrases, only add the most reliable new phrase pairs proposed by the CSTM and calculate all the translation model probabilities with the neural network. This research is ongoing.

Conclusion and perspectives

This paper has presented a new technique to estimate the translation probabilities in a phrase-based SMT system. This can be seen as an extension of the continuous space language model: all the words of the source phrase are projected onto a continuous space and the neural network predicts the joint probability of all the words in the target phrase. To the best of our knowledge, previous research to apply continuous space methods to the translation model, were limited to tuple-based translation models (Schwenk et al., 2007; Le et al., 2012). The system proposed in this paper is trained on the same data than a standard phrase-based systems.

An interesting feature of the approach is the ability to provide translation model probabilities for any possible phrase-pair. We have provided experimental evidence that the system actually seems to be able to provide meaningful translations for source phrase which were not seen in the training data. An interesting extension of this idea is to use large amounts of *monolingual* data to pre-train the projections of the source words onto the continuous space. The neural network could learn from the monolingual data that words are synonyms since they often appear in similar contexts. This could be used in the CSTM to provide translations for source words not seen in the bitexts. This could be also interesting when translating from a morphologically rich language into English since many verb forms actually translate into the same English word.

The implementation of the continuous space translation model is based on an extension of the CSLM toolkit and it will be freely available. By these means, we can benefit of all the infrastructure of the toolkit, in particular training on large amounts of data using resampling techniques and a fast implementation on GPU cards, or weighting of the training data.

Acknowledgments

This work was partially financed by the French government (COSMAT, ANR-09-CORD-004), the European Commission (MATECAT, ICT-2011.4.2 – 287688) and the DARPA BOLT project.

References

Bengio, Y. and Ducharme, R. (2001). A neural probabilistic language model. In *NIPS*, volume 13, pages 932–938.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *JMLR*, 3(2):1137–1155.

Brants, T., Popat, A. C., Xu, P., Och, F. J., and Dean, J. (2007). Large language models in machine translation. In *EMNLP*, pages 858–867.

Foster, G., Kuhn, R., and Johnson, H. (2006). Phrasetable smoothing for statistical machine translation. In *EMNLP*, pages 53–61.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based machine translation. In *HLT/NACL*, pages 127–133.

Le, H.-S., Allauzen, A., Wisniewski, G., and Yvon, F. (2010). Training continuous space language models: Some practical issues. In *EMNLP*, pages 778–788.

Le, H.-S., Allauzen, A., and Yvon, F. (2012). Continuous space translation models with neural networks. In *NAACL*.

Le, H.-S., Oparin, I., Allauzen, A., Gauvain, J.-L., and Yvon, F. (2011). Structured output layer neural network language model. In *ICASSP*, pages 5524–5527.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *ACL*, pages 160–167.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignement models. *Computational Linguistics*, 29(1):19–51.

Rousseau, A., Bougares, F., Deléglise, P., Schwenk, H., and Estève, Y. (2011). Lium's systems for the IWSLT 2011 speech translation tasks. In *International Workshop on Spoken Language Translation*.

Schwenk, H. (2007). Continuous space language models. *Computer Speech and Language*, 21:492–518.

Schwenk, H. (2010). Continuous space language models for statistical machine translation. *The Prague Bulletin of Mathematical Linguistics*, (93):137–146.

Schwenk, H., Costa-jussà, M. R., and Fonollosa, J. A. R. (2007). Smooth bilingual n-gram translation. In *EMNLP*, pages 430–438.

Schwenk, H., Déchelotte, D., and Gauvain, J.-L. (2006). Continuous space language models for statistical machine translation. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 723–730.

Schwenk, H., Rousseau, A., and Attik, M. (2012). Large, pruned or continuous space language models on a GPU for statistical machine translation. In *NAACL Workshop on the Future of Language Modeling for HLT*.

Zamora-Martínez, F., Castro-Bleda, M. J., and Schwenk, H. (2010). N-gram-based machine translation enhanced with neural networks for the French-English BTEC-IWSLT'10 task. In *IWSLT*, pages 45–52.