IberSPEECH 2021
24-25 March 2021, Valladolid, Spain

Confidence Measures for Interactive Predictive Neural Machine Translation

Ángel Navarro[1], *Francisco Casacuberta*[1]

[1]Universitat Politècnica de València

annamar8@prhlt.upv.es, fcn@prhlt.upv.es

Abstract

Confidence Measures (CMs) can be used to estimate the reliability of the words of a hypothesis generated by a machine translation system. In the Interactive-Predictive Machine Translation (IPMT) paradigm, they are used to determine which words of the generated predictions need to be corrected, reducing the total number of words typed by the user. The CMs used must be fast enough in order to not affect the interaction between the user and the machine negatively. In this paper, we present several fast CMs for Interactive-Predictive Neural Machine Translation: IBM Model 1 and 2, Fast Align and Hidden Markov Model. These estimators let the system to achieve a reduction in the number of words typed by getting less-quality translations. The experiments done proved that these CMs are fast enough to use them in an IPMT system, and obtained a high relative reduction on the number of words corrected while getting good-quality translations.

Index Terms: confidence measure, neural machine translation, interactive machine translation, interactive predictive machine translation

1. Introduction

Although the quality of the translations generated by Machine Translation (MT) systems has highly improved in recent years with the apparition of the Neural Models, the MT systems are not able to generate error-free translations yet. The Interactive-Predictive Machine Translation (IPMT) field uses human experts to translate interactively with the system the sentences provided, where the machine guarantees high-efficiency and the human the quality of the translations. There are a large variety of approaches that reduce the effort done in the process, one of them is the use of Confidence Measures (CMs).

CMs provide a correctness estimation for each word of a hypothesis. The system uses this information to classify the words as correct if their confidence is above a threshold. The words classified as correct do not need to be checked by the user or corrected, reducing the number of words typed as well the time that the user spends on reading and deciding whether to accept a prediction.

Not all the CMs approaches are adequate for the IPMT paradigm, some of them use large numbers of features, or use the N-best translations for its calculation. In this field, we want to obtain the estimation value as accurate and quick as possible in order to not interrupt the translation process.

2. Related Work

Confidence Estimation has been extensively studied in the Speech Recognition field [1] and opened to MT in the last decade. Blatz et al. (2004) [2] introduced various methods to determine the correctness of the translations based on Statistical Machine Translation (SMT) model and target language features, translation tables and word posterior probabilities. A new method to measure confidence measures is presented by Bach et al. (2011) [3] with a representation of how to visualize the confidence estimations in a post-editing framework.

Recently, these measures have been implemented in IPMT systems [4]. The works presented by González et al. (2010) [5, 6] in a Interactive-Predictive Statistical Machine Translation (IPSMT) system use these confidences based on IBM Model 1 to reduce the number of words to correct by the user, only checking the words with a confidence estimation lower than the threshold set. The workbench CasMaCat [7] shows the CMs to the user using different colours and only displays the predictions up to the first word classified as incorrect.

In this work, we have implemented in an Interactive-Predictive Neural Machine Translation (IPNMT) system four different CMs at word level with the aim of reducing the number of words that the user has to check risking the less quality as possible.

3. Confidence Measures

We have studied different CMs which main features can be pretrained and saved in data matrixes. The system only has to access the matrixes to obtain the confidence estimation, getting the values very fast during the search, which is crucial to do not interrupt the user-machine interaction. The features used are based on the translation probability of the target word and its alignment probability.

The project has focused on the use of computationally efficient CMs over getting high-quality confidence estimations of the words.

3.1. IBM Model 1

The first CM is based on the IBM Model 1 [8], similar to the one described in Blatz et al. 2004 [2]. As performed in related works [5, 6], we modified this CM by replacing the average with the maximal lexicon probability for its dominance over it [9]. Having the sequence $e_1^I = e_1, ..., e_I$ from the target language, and the sequence $f_1^J = f_1, ..., f_J$ from the source langugae, the confidence value of the word e_i can be calculated as follows:

$$c(e_i) = \max_{0 \leq j \leq J} \mathrm{p}(e_i \mid f_j) \tag{1}$$

where $\mathrm{p}(e_i \mid f_j)$ is the lexicon probability obtained from the IBM Model 1, and f_0 is the empty source word.

3.2. IBM Model 2

The second CM is based on the IBM Model 2 [8] and extends the previous CM by adding the alignment probabilities. This extra information lets the system take cognizance of where words appear in either string. The confidence value of the word e_i, which is positioned at i in the target sequence, can be calculated as follows:

10.21437/IberSPEECH.2021-42

$$c(e_i) = \max_{0 \le j \le J} \mathrm{p}(e_i \mid f_j)\ \mathrm{p}(a_i = j \mid i, J, I) \quad (2)$$

where a_i is the alignment position from the source sequence corresponding to position i from target, $\mathrm{p}(a_i|\ i, I, J)$ is the alignment probability obtained from the IBM Model 2, and $a_i = 0$ represents the empty source word.

3.3. Fast Align

The third CM that we have studied is based in Fast Align [10], which appeared as a simpler and faster reparameterization of the IBM Model 2, and presents a different method to calculate the alignment probability of the confidence estimation. In the previous model, we have to compute the alignment probability of each position of the target sentence, alignment and sentence length. In Fast Align this probability is based on favour the alignment points close to the diagonal, and just need to train two parameters, the null alignment probability p_0 and a precision $\lambda \geq 0$ which controls how strongly the model favours the alignment points near the diagonal. The alignment probability can be calculated as follows:

$$\mathrm{p}(a_i{=}j \mid i, J, I) = \begin{cases} p_0 & a_i = 0 \\ (1 - p_0) \times \frac{e^{\lambda h(a_i, i, J, I)}}{Z_\lambda(i, J, I)} & 0 < a_i \leq n \\ 0 & otherwise \end{cases} \quad (3)$$

where $h(a_i, i, J, I)$ can be computed as follows:

$$h(a_i, i, J, I) = -\left| \frac{i}{I} - \frac{a_i}{J} \right| \quad (4)$$

the normalization $Z_\lambda$term is computed as follows:

$$Z_\lambda(i, J, I) = \sum_{j'=1}^{n} \exp \lambda h(j', i, J, I) \quad (5)$$

In their paper [10], Dyer et al. (2013) described in detail how to reduce the time complexity of the method to 1, drastically reducing computing time and obtaining the same time complexity that we had with the previous methods where we only had to get the value from a matrix.

3.4. Hidden Markov Model

The last CM is based in HMM [11], which differs from the previous CMs by taking a different approach to obtain the alignment probabilities. HMM does not take in count the position of the target word, its alignment probability is calculated from the alignment positions on the source sentence of a target word and the previous one, more specifically it depends only on the jump width $(a_i - a_{i-1})$. The confidence value can be calculated as follows:

$$c(e_i) = \max_{0 \le j \le J} \mathrm{p}(e_i \mid f_j)\ \mathrm{p}(a_i{=}j|\ a_{i-1}, J) \quad (6)$$

where $\mathrm{p}(a_i{=}j|\ a_{i-1}, J)$ is the alignment probability obtained from HMM that can also be represented as $p(j|j', J)$. To obtain the confidence value of a target word the method requires to calculate the optimal alignment of the previous word. This requires to use dynamic programming as follows:

$$\hat{a}_{i-1} = \arg\max_{1 \le a_{i-1} \le J} p(a_i|a_{i-1}, J) Q(i-1, a_{i-1}) \quad (7)$$

where $Q(i, j)$ is a sort of partial probability that we can calculate recursively using the following formula:

$$Q(i, j) = \mathrm{p}(e_i|f_j) \max_{1 \le j' \le J} \left[p(j|j', J)\ Q(i-1, j') \right] \quad (8)$$

To calculate it efficiently, we need to save in memory all the partial probabilities of the previous target positions. This information is used recursively for each partial probability, so we are saving computation time by keeping it on memory.

4. Experimental Setup

4.1. System Evaluation

As explained previously, CMs are a classifying task where we want to tag the words of the hypothesis generated by the system as correct or incorrect, depending on its confidence value and the threshold used to decide correctness. The metrics used to evaluate the CMs, Classification Error Rate (CER) and Receiver Operating Characteristic (ROC), capture the discriminability of the classification function across the range of all thresholds used. We are also going to report the mean execution time of the confidence measures in milliseconds.

The ground truth of the words classified by the models is obtained from the reference translations of the parallel corpus. We consider a word as correct if it occurs in the same position as the reference translation. As we are highly restricting the number of words that could be classified as correct the metrics obtained are pessimistic.

The CER [5] is computed as the proportion of words that our CM has classified incorrectly given a threshold value. The area under the ROC curves [12], called IROC, gives us a global indication of the CM discriminability. These curves show the plot correct-reject ratio (true correct / n_0) vs correct-accept ratio (true incorrect / n_1) for different thresholds, where n_0 and n_1 are the total number of correct and incorrect words of the ground truth. The ROC curve lies in the unit square, the diagonal corresponding to the random choice and the edges to a perfect classification.

To evaluate the improvement of the IPNMT system with the CM implementation, we compare the improvement on BiLingual Evaluation Understudy (BLEU) with the reduction of Word Stroke Ratio (WSR) [13]. BLEU computes a geometric mean of the precision of n-grams multiplied by a factor to penalise short sentences. WSR is computed as the proportion of words that the user needs to correct to generate the reference translation.

4.2. Corpora

All experiments have been carried out on the Spanish-English language pair of the EU corpus. The corpus was cleaned, lowercased and tokenized using the scripts included in the toolkit Moses [14]. We applied the subword subdivision BPE, described in Sennrich et al. [15], with a maximum of 32000 merges.

The EU corpus [16] is formed from the Bulletin of the European Union, which exists in all official languages and is publicly available on the internet.

4.3. Experimental Setup

First of all, we built our Neural Machine Translation (NMT) models using NMT-Keras [17]. We used an encoder-decoder

Table 1: *Statistics of the Spanish-English EU corpus. K and M stands for thousands and millions respectively.*

		Es-En	
Training	Sentences	214K	
	Average Length	27	24
	Running Words	6M	5M
	Vocabulary	84K	69K
Dev.	Sentences	400	
	Average Length	29	25
	Running Words	12K	10K
Test	Sentences	800	
	Average Length	28	25
	Running Words	23K	20K

architecture with attention model [18] and LSTM cells [19]. The dimensions of encoder, decoder, attention model and word embedding were set to 512. We used a single hidden layer of encoder and decoder. The learning algorithm used for the NMT system was Adam [20], with a learning rate of 0.0002. We clipped the L_2 norm of the gradient to 5. The batch size was set to 50 and the beam size to 6.

Secondly, we built our Confidence Measures Models. We use the toolkit GIZA++ [21] to train the IBM Model 1 and 2; and the HMM Model. To built the Fast Align Model we used the scripts developed in Dyer et al. (2013) [10].

4.4. Confidence Measures Evaluation Results

We carried out experimentation intended to study the performance of the CM on an IPNMT system. First of all, we carried out an IPMT session that we used to produce a corpus of words tagged as correct or incorrect. These words are compared with the references to classify them correctly and use them as the ground truth to calculate the CER and ROC.

Figure 1 displays the CER evolution through different threshold values for each one of the CMs used. The three models that used an alignment probability for their confidence calculation have similar behaviour. The IBM Model 2 obtained the best CER score, 0.24 for a threshold value of 0.125.

Figure 2 compares the ROC curves of the CMs used, the diagonal shows the random choice curve. This time the behaviour of the IBM Model 2 and Fast Align are very similar related, though Fast Align is some points lower. At the same time, the IBM Model 1 behaves like HMM.

Table 2 shows the performance of the confidence measures in terms of CER and IROC. The baseline is a classifier which tags all the words as the most frequent class, $\text{CER}_b = min(n_0, n_1)/n$. The values of CER displayed are based on threshold optimized on the validation set. All the CMs obtain a relative improvement over the baseline CER of more than 7%. The best CM is the model based on the IBM Model 2 that gets a relative improvement over the baseline CER of 20%.

All the CMs obtained an execution time lower than 100 milliseconds, a threshold set by Nielsen (1994) [22] that marks the limit for having the user feel that the system is reacting instantaneously. This makes the confidence measures optimal for an IPMT system and do not break the human-machine interaction.

4.5. User Simulated IPNMT Results

In the previous section, we have studied the discriminability of the different confidence measures across the range of all thresholds used. We have integrated these confidence measures in an IPNMT system to study the trade-off between the effort that the translator needs to do and the final quality of the translations.

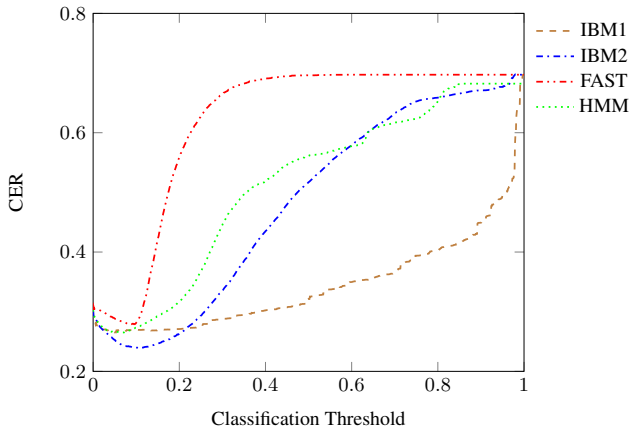

Figure 1: *CER for IBM Model 1, IBM Model 2, Fast Align and HMM across the range of all thresholds used.*

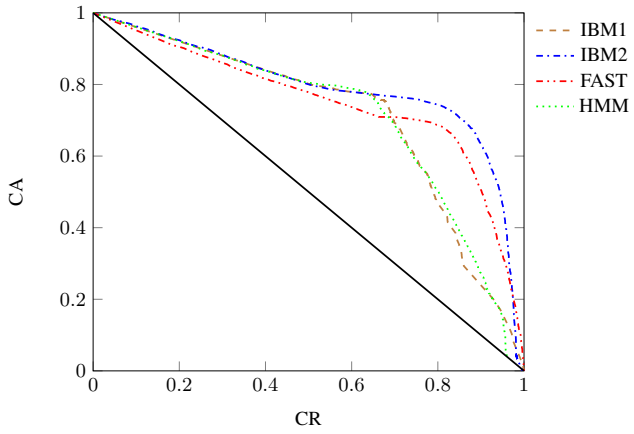

Figure 2: *ROC curves for IBM Model 1, IBM Model 2, Fast Align and HMM.*

For the user simulation, we only check and correct those words that have a confidence estimation under the threshold value. The words are compared with the ones that have the same position on the reference sentence and are corrected if they are different. We correct the words typing the ones that appear on the reference without taking in count the context.

In this section, we present a range of experiments using different thresholds values from 0.0 where the system behaves as an unsupervised NMT system, to 1.0 where the user has to check and correct all the words as an IPNMT system. For each threshold used, we compare the user effort using the metric WSR, and the quality of the sentences with BLEU.

Figure 3 shows the WSR and BLEU scores for all the CM used across the transition between an unsupervised NMT system with a 0.0 threshold and the conventional IPNMT system with a 1.0 threshold. As we raise the threshold more words are tagged as incorrect increasing the number words that the user has to check and correct, which improves the quality of the translations. Although IBM Model 1 and HMM obtained the lowers IROC values, they present more gradual transitions that let us have a larger range of useful thresholds values to use.

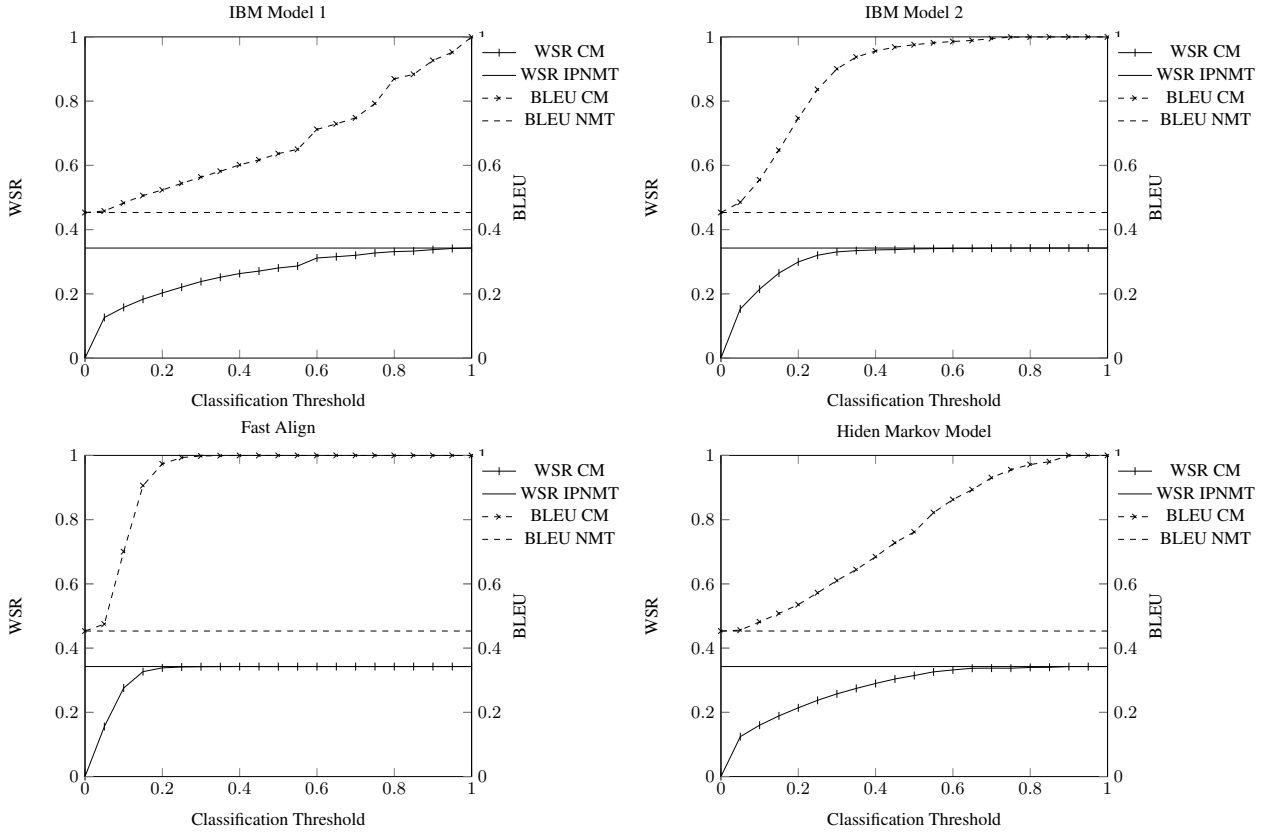

Figure 3: *BLEU translation scores versus WSR across the range of all thresholds used.*

Table 2: *CER[%], IROC and execution time (ms) of the Confidences Measures on the test set.*

Confidence Measure	CER	IROC	ms
baseline	30.0	-	-
IBM Model 1	26.5	0.713	**3.41**
IBM Model 2	**23.9**	**0.791**	4.24
Fast Align	27.8	0.752	4.32
HMM	26.5	0.714	8.93

We can compare the relative WSR reduction obtained for each CM while getting similar quality translations. For this purpose, we compare the relative WSR reductions of the experiment samples with BLEU scores closer to 0.70. The higher relative reduction is obtained by Fast Align with a relative reduction of 19.5%. IBM Model 1, IBM Model 2 and HMM obtained 6.6%, 12.6% and 11.3% respectively.

5. Conclusions and Future Work

5.1. Conclusions

In this paper, we have proposed four different CMs that can be computed very fast while obtaining a good discriminability evaluation, which makes them a perfect option to implement into IPMT systems. We compared the CM using the CER and IROC metrics. The CM based on IBM Model 2 obtained the best results in both metrics.

We have tested the confidence measures in an IPNMT system, comparing the effort that the user has to do for each threshold value used with the quality of the translations obtained. Around 0.70 of BLEU score Fast Align obtained the best WSR reduction, almost 20%.

5.2. Future Work

The word confidence measures obtained can be combined to compute a sentence correctness value. As future work, we plan to investigate different methods to combine them and compare the effort reduction.

Also, we will try in future work to use more complex CMs with higher computational time, like neural models, and try to use them in IPMT systems.

In the experiments that we have performed, we simulated the user interaction and used for the evaluation of very pessimistic ground truth. We need to compare our results with those obtained with real translators that will take into account different possible translations in the correction process.

6. Acknowledgements

This work received funds from the Comunitat Valenciana under project EU-FEDER (*IDIFEDER*/2018/025), Generalitat Valenciana under project ALMAMATER (*PrometeoII*/2014/030), and Ministerio de Ciencia under project MIRANDA-DocTIUM (RTI2018-095645-B-C22).

7. References

[1] F. Wessel, R. Schluter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Transactions on speech and audio processing*, vol. 9, no. 3, pp. 288–298, 2001.

[2] J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing, "Confidence estimation for machine translation," in *Coling 2004: Proceedings of the 20th international conference on computational linguistics*, 2004, pp. 315–321.

[3] N. Bach, F. Huang, and Y. Al-Onaizan, "Goodness: A method for measuring machine translation confidence," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 211–219.

[4] M. Domingo, A. Peris, and F. Casacuberta, "Segment-based interactive-predictive machine translation," *Machine Translation*, vol. 31, no. 4, pp. 163–185, 2017.

[5] J. González-Rubio, D. Ortíz-Martínez, and F. Casacuberta, "On the use of confidence measures within an interactive-predictive machine translation system," in *Proceedings of the 14th Annual conference of the European Association for Machine Translation*. Saint Raphaël, France: European Association for Machine Translation, May 27–28 2010. [Online]. Available: https://www.aclweb.org/anthology/2010.eamt-1.18

[6] J. González-Rubio, D. Ortiz-Martínez, and F. Casacuberta, "Balancing user effort and translation error in interactive machine translation via confidence measures," in *Proceedings of the ACL 2010 Conference Short Papers*, 2010, pp. 173–177.

[7] V. Alabau, R. Bonk, C. Buck, M. Carl, F. Casacuberta, M. García-Martínez, J. González, P. Koehn, L. Leiva, B. Mesa-Lao *et al.*, "Casmacat: An open source workbench for advanced computer aided translation," *The Prague Bulletin of Mathematical Linguistics*, vol. 100, no. 1, pp. 101–112, 2013.

[8] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1993. [Online]. Available: https://www.aclweb.org/anthology/J93-2003

[9] N. Ueffing and H. Ney, "Application of word-level confidence measures in interactive statistical machine translation," in *Proceedings of the 10th EAMT Conference: Practical applications of machine translation*. Budapest, Hungary: European Association for Machine Translation, May 30–31 2005. [Online]. Available: https://www.aclweb.org/anthology/2005.eamt-1.35

[10] C. Dyer, V. Chahuneau, and N. A. Smith, "A simple, fast, and effective reparameterization of IBM model 2," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, Jun. 2013, pp. 644–648. [Online]. Available: https://www.aclweb.org/anthology/N13-1073

[11] S. Vogel, H. Ney, and C. Tillmann, "HMM-based word alignment in statistical translation," in *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*, 1996. [Online]. Available: https://www.aclweb.org/anthology/C96-2141

[12] R. O. Duda, P. E. Hart *et al.*, *Pattern classification*. John Wiley & Sons, 2006.

[13] J. Tomás and F. Casacuberta, "Statistical phrase-based models for interactive computer-assisted translation," in *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, 2006, pp. 835–841.

[14] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*. Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 177–180. [Online]. Available: https://www.aclweb.org/anthology/P07-2045

[15] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. [Online]. Available: https://www.aclweb.org/anthology/P16-1162

[16] S. Barrachina, O. Bender, F. Casacuberta, J. Civera, E. Cubel, S. Khadivi, A. Lagarda, H. Ney, J. Tomás, E. Vidal, and J.-M. Vilar, "Statistical approaches to computer-assisted translation," *Computational Linguistics*, vol. 35, no. 1, pp. 3–28, 2009. [Online]. Available: https://www.aclweb.org/anthology/J09-1002

[17] Álvaro Peris and F. Casacuberta, "NMT-Keras: a Very Flexible Toolkit with a Focus on Interactive NMT and Online Learning," *The Prague Bulletin of Mathematical Linguistics*, vol. 111, pp. 113–124, 2018. [Online]. Available: https://ufal.mff.cuni.cz/pbml/111/art-peris-casacuberta.pdf

[18] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in neural information processing systems*, 2015, pp. 577–585.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2017.

[21] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[22] J. Nielsen, *Usability engineering*. Morgan Kaufmann, 1994.