International Journal of Semantic Computing
© World Scientific Publishing Company

Unsupervised Translation Quality Estimation for Digital Entertainment Content Subtitles

Prabhakar Gupta Mayank Sharma
Amazon
{*prabhgup, mysharm*}*@amazon.com*

We demonstrate the potential for using aligned bilingual word embeddings in developing an unsupervised method to evaluate machine translations without a need for parallel translation corpus or reference corpus. We explain different aspects of digital entertainment content subtitles. We share our experimental results for four languages pairs English to French, German, Portuguese, Spanish and present findings on the shortcomings of Neural Machine Translation for subtitles. We propose several improvements over the system designed by Gupta et al. [1] by incorporating custom embedding model curated to subtitles, compound word splits and punctuation inclusion. We show a massive run time improvement of the order of $\sim 600\times$ by considering three types of edits, removing *Proximity Intensity Index (PII)* and changing post-edit score calculation from their system.

Keywords: Quality Estimation; Word Embeddings; Unsupervised Alignment; Machine Translation

1. Introduction

Digital Entertainment industry ecosystem consists of a large catalog of content *viz.* movies, TV series and video clips with their corresponding subtitles across multiple languages. Conversion of subtitles from a source language to a target language is predominantly done by human translators with bilingual/multilingual proficiency. This process is costly, time consuming and does not scale across multiple language pairs. In order to reduce human effort and the translation cost, we use a Neural Machine Translation (NMT) [2–6]) system to convert a subtitle from its source language to a target language. However, there are many known problems with NMT [7]. One particular problem in case of subtitles is that NMT does not capture the nuances present in subtitles. Hence, accessing the quality of machine translation is imperative.

Translation evaluation has two facets and we focus on *adequacy* of translation (meaning retention) without focusing on *fluency* aspects such as grammatical construction of a sentence. Estimating the translation quality of sentences is primarily a manual process. Currently, professionals with multilingual proficiency rate the translation quality [8]. Problems with human evaluation are its scalability to large corpora spanning across multiple language pairs; subjective bias of the evaluator;

large costs of evaluation; consistency across multiple evaluators and sensitivity of the data. Thus, researchers use automated translation quality scores. Industry-popular translation quality estimation systems like BLeU [9], TER [10], NIST [11] or METEOR [12] are used to analyze a machine translation system but require a reference corpora. However, a reference corpora for translated subtitles does not exist. Thus, we need an automated system to estimate translation quality of subtitles without a reference corpus.

In Gupta et al. [1], authors present an unsupervised system to estimate translation quality for source language English with four target languages *viz.*, French, German, Portuguese, Spanish. In this work, we improve the system's $F_{0.5}$ score by adding processing for compound words, retaining punctuation and using custom embedding models trained on a combination Wikipedia[a] and subtitles. The embedding models can create embedding vectors for out-of-vocabulary (OOV) words using subword information [13]. We also present improvements of the order of $\sim 600\times$ for run time of the system during the testing phase by reducing the number of operations required to generate post-edit score. We remove *Proximity Intensity Index (PII)* from the system as it does not contribute to the increasing accuracy as much as it slows down the process.

The outline of this paper is as follows. Section 1 presents the problem statement and the introduction. Section 2 describes the Subtitle Validation Program findings. Section 3 presents the methodology to generate translation scores. Section 4 presents the experiments and analysis of performance of the system on run time, custom embedding, compound word splitting for OOV words, punctuation inclusion and on length of source and target sentences. Finally, with section 5, we present the concluding remarks and future endeavors.

2. Subtitle Validation Program

To understand the problems and their intensity in translating subtitles using manual translators and NMT, we randomly choose over 50 English subtitles containing $20k$ subtitle blocks in total. We asked humans to translate these subtitle files to four languages *viz.*, French, German, Spanish, Portuguese. We also translated the same English files using an NMT system trained using [2] to all four languages. We sent these machine translated files to humans for post-editing. The results showed that the human translation time is greater than the human post-edit time. For French, post-editing takes 25% less time than human-translation. The difference in average number of hours is not too high since humans are still required to go through the complete subtitle file again proving the need for a system to automate the process of quality estimation.

We also asked humans to classify each subtitle block into one of the three cases *viz.* adequacy error, fluency error and no error. We computed the frequency of

[a] `https://dumps.wikimedia.org/`

adequacy and fluency errors across aforementioned language pairs. Figure 1 shows the distribution of errors present in the movies. We observe that, adequacy error is the biggest problem present in about 70% of the subtitle blocks followed by fluency errors which occur in about 15% of the blocks. In this work, we develop a solution to assess the quality of translation based on adequacy errors. The next section highlights the key steps in building an unsupervised translation quality estimation system.

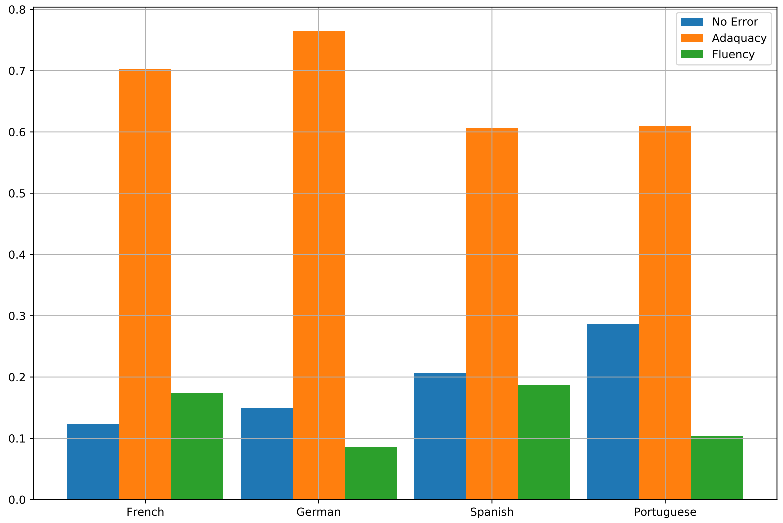


Fig. 1: Distribution of errors in each subtitle blocks marked by post-editors for machine translated subtitles. Any given block could have both adequacy and fluency error.

3. Unsupervised Quality Estimation

The procedure for translation quality estimation is divided into 4 steps namely, bilingual word embedding alignment, creation of cosine similarity matrix, generation of word pair list and calculation of post-edit score. We describe each step in detail below.

3.1. *Bilingual Word Embeddings Alignment*

To compare the distance between the two words in different languages, the spaces of two language embeddings must be aligned. Monolingual spaces individually are not inherently aligned as shown in Figure 2(a). We align the two monolingual embeddings such that they retain their individual monolingual spatial integrity as shown in Figure 2(b). We observe that words such as *and* and *und* which were not close in original space, comes close to each other after alignment. Earlier approaches for alignment used identical words/phrases in source and target languages [14, 15] or considered spatial distribution of vectors [16]. State-of-the-art approaches use adversarial training [17, 18] to identify the best overlap between two language embeddings.

In our experiments, we use Wikipedia articles and the subtitles to generate custom FastText [19] embeddings for various languages using Gensim [20]. Let $X \in \mathbb{R}^{m_1 \times d}$ and $Y \in \mathbb{R}^{m_2 \times d}$ denote the source and target embeddings containing m_1 and m_2 unigrams respectively. Both the embeddings are in $d = 300$ dimension. Table 1 present the vocabulary sizes. We assume that for both the embeddings $X$and Y the following holds true; $X = \{x : \|x\|_2 = 1, \ \forall x \in X, \ x \in \mathbb{R}^d\}$. In this work, we use supervised orthogonal Procrustes method [18] to align source embedding to a target embedding as shown in eq. 1. This problem is equivalent to finding the nearest orthogonal matrix to a given matrix $Z = X^T Y$. To find the matrix W^* we use singular value decomposition of $Z = U\Sigma V^T$ to write $W^* = UV^T$.

Table 1: Size of vocabulary used to generate custom FastText embeddings for each language

Language	**Vocabulary size**	**Dimension**
English	941380	300
French	841717	300
German	961344	300
Portuguese	545279	300
Spanish	861593	300

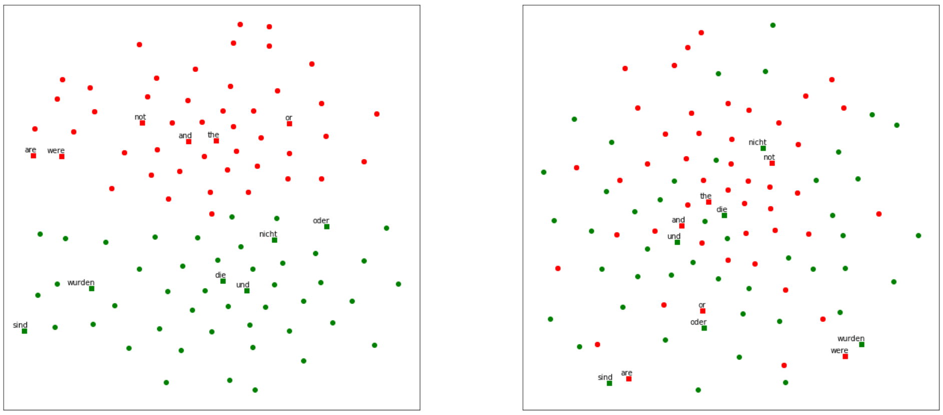
$$W^* = \arg \min_{W \in \mathbb{R}^{d \times d}} \|XW - Y\|_F, \tag{1}$$

such that,

$$W^T W = I. \tag{2}$$

where, $I \in \mathbb{R}^{d \times d}$ is an identity matrix. Here, we use a parallel dictionary of most frequent 5000 words to align the two embeddings. In all our cases the source language is considered to be English and target language belongs to one of the four languages. Once the solution W^* to the orthogonal Procrustes problem has been found, the aligned vector x_a in the target language for a source word x is given by:

$$x_a = xW^* \tag{3}$$



(a) Before alignment

(b) After alignment

Fig. 2: 2-dimensional t-SNE [21] plots of most frequent 50 words in English (Red markers) and German (Green markers) before and after alignment projected in same plane

3.2. *Creating Cosine Similarity Matrix*

For every input request we generate a cosine similarity matrix using aligned word embeddings [19, 22, 23]. The cosine similarity between two vectors x_a and y given by:

$$sim(x_a, y) = \frac{x_a^T y}{\|x_a\|_2 \|y\|_2}. \tag{4}$$

Figure 3 presents a cosine similarity matrix between an English sentence, *“or we have a variety of paddles and straps”* and its German translation *“oder wir haben eine Vielzahl von Paddeln und Gurten”*. We observe that words closer to each other have a high similarity scores.

3.3. *Generating Translation Word Pair List*

Using the cosine similarity matrix, we estimate the number of edits human would have to do to make it perfect (acceptable) translation. We do not remove any punctuation as punctuation also contribute the quality of translation [24]. Let us consider an example, English sentence *“I started very young.”* and its German translation *“Ich fing sehr jung an.”*. Cosine matrix for the same is shown in Figure 4. There are 5 tokens in English sentence and 6 tokens in German sentence. We create a list of word pairs which are directly translated. For each row and each column we check the highest similarity scores to generate the pairs list. If there is an OOV word, we consider cosine similarity to be 0. From Figure 4, we can see there are 5 pairs and no pair for German word *“an”*.

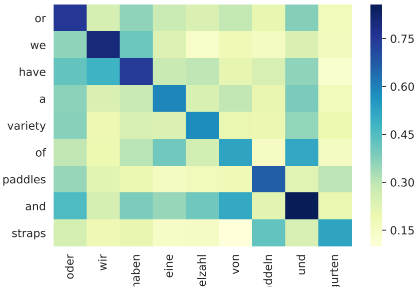


Fig. 3: Cosine similarity scores heatmap for english sentence, *"or we have a variety of paddles and straps"* and German translation *"oder wir haben eine Vielzahl von Paddeln und Gurten"*

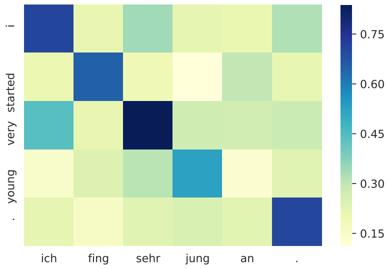


Fig. 4: Cosine similarity scores heatmap for english sentence, *"I started very young."* and German translation *"Ich fing sehr jung an."*

3.4. *Calculating Post-Edit score*

To calculate the post-edit score, we transform the target tokenized sentence by replacing each target word with the source word from the translated word pair list generated in last step. For our running example, tokenized German sentence *"[ich, fing, sehr, jung, an, .]"* is transformed to *"[i, started, very, young, an, .]"*. We calculate 3 types of word edits *viz.*, insertion, deletion, substitution on the original tokenized German sentence and the transformed target sentence. There would be one edit in this case which is deletion of word *"an"*. As opposed to [1], which uses a greedy search to get shifts similar to what explained in [10]; we calculate edit distance. We then normalize number of edits with the maximum number of tokens in source and target sentences. Finally, we threshold the scores to generate the labels for a given pair. In the next section, we describe the experiments conducted to validate the efficacy of our system.

4. Experiments

We created a dataset of random 1500 subtitle blocks in English, used an NMT system trained using [2] to translate each sentence to the four aforementioned languages. Human translators were asked to rate each translation on scale of one to six with six being a perfect translation. We considered translation with score > 3 as *GOOD* and others as *BAD*. For French 96.3%, German 68.7%, Portuguese 79.9% and Spanish 81.9% of 1500 subtitle blocks were *GOOD*.

4.1. *Time Performance*

We improve the three techniques use to generate the score as discussed in [1] to improve the run time of the method. First, we use a word level post-edit score instead of a character-level post-edit score after creating transformed target sentence. Second, we use three edits (insertion, deletion, substitution) instead of using four (insertion, deletion, substitution, shifts) and third, we remove PII which eliminate pairs from the list to ensure that there are no random pairing. These improvisations enables quick computation of the translation score. Table 2, presents the comparison for our baseline system using a word-level post-edit score without PII with the results. For this experiment we use the pre-aligned embedding provided by MUSE [18, 25]. While the performance of the system decreases by $\sim 1\%$ in terms of $F_{0.5}$ Score as compared to [1], the improvements in run time are of the order of $\sim 600\times$, making it a *near-real-time* system. This massive gain in run time of the system allows us to use more complicated techniques such as compound word split for score calculation. Henceforth, in all our other experiments, we use these results (with word-level post-edit calculation without PII) as baseline since the run time improvement is too large to be ignored.

Table 2: Performance comparison between base system and system with word-level post-edit calculation without PII

Language	**Approach**	$F_{0.5}$	P	R	**Avg. Time (ms)**
French	(Gupta et al.) [1]	97.1	96.4	99.9	2830
	Baseline	96.4	96.3	96.7	4.5
German	(Gupta et al.) [1]	73.7	70.1	92.9	2930
	Baseline	71.7	67.9	92.6	4.4
Portuguese	(Gupta et al.) [1]	83.2	79.9	99.9	2480
	Baseline	82.2	79.4	95.5	4.3
Spanish	(Gupta et al.) [1]	85.4	82.8	97.6	2690
	Baseline	84.1	81.8	95.2	4.4

4.2. *Domain specific aligned Word Embeddings*

The baseline results presented in Table 2 are generated using pre-aligned word embeddings made available by [18, 25]. However, there are two problems with these embeddings. First, these embeddings clips the number of words to top $200k$ words for each language rendering them impractical for OOV words and second, the authors in [18] provide the final word vectors making it impossible to fine-tune these embeddings [26] for subtitles. Therefore, we trained custom Word2Vec [22] word embedding models using FastText [19] implementation provided in Gensim [20] for five languages (English, French, German, Portuguese, Spanish) using Wikipedia and subtitles. These custom models can handle OOV words using sub-word information.

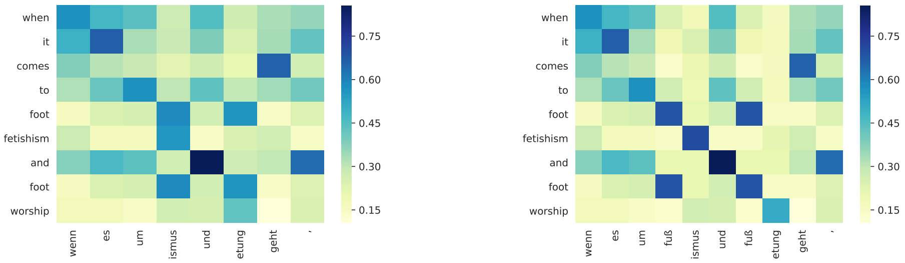
To generate a custom alignment matrix for each language, we followed the process explained in subsection 3.1 and [18] (Custom alignment). Each alignment matrix was a 300×300 matrix. We compared our scores with alignment matrices provided by Babylon [15] (Babylon Alignment). Table 3 presents the comparison for different alignment matrices. The $F_{0.5}$ scores for both the custom and Babylon alignment are comparable and better than the baseline. We used same process as described in subection 4.1 for all three alignments.

Table 3: Performance comparison between different alignments for source language English

Language	**Approach**	$F_{0.5}$	P	R	**Avg. Time (ms)**
French	Baseline	96.4	96.3	96.7	4.5
	Custom Alignment	**97.0**	96.3	99.7	9.3
	Babylon Alignment	96.9	96.2	99.4	10.5
German	Baseline	71.7	67.9	92.6	4.4
	Custom Alignment	73.0	68.5	99.1	13.3
	Babylon Alignment	**73.1**	68.6	99.5	9.7
Portuguese	Baseline	82.2	79.4	95.5	4.3
	Custom Alignment	**83.2**	79.9	99.7	8.3
	Babylon Alignment	**83.2**	79.9	99.9	9.9
Spanish	Baseline	84.1	81.8	95.2	4.4
	Custom Alignment	**85.0**	81.9	99.8	8.0
	Babylon Alignment	**85.0**	81.9	99.8	9.7

4.3. *Compound Word Split*

In languages (like German), multiple words can be combined to make new words posing a problem for words not appearing or appearing infrequently in training data making it a OOV word [27]. For example, German word *Breitflügelfledermaus* meaning *wide-wing bat* can be split to *Breit flügel fledermaus* while retaining exact



(a) Without word break

(b) With word break

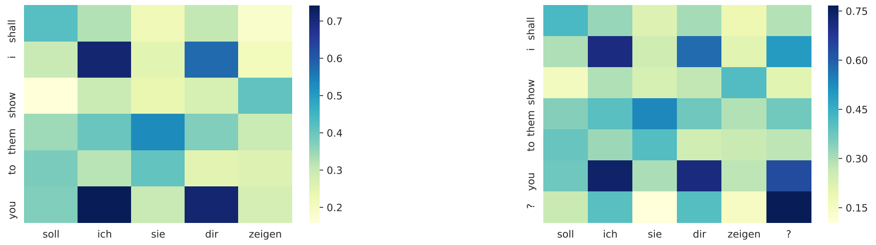
Fig. 5: Cosine similarity scores heatmaps showing the difference between calculating score with and without compound word splitting for English subtitle block, *"When it comes to foot fetishism and foot worship,"* and German translation *"Wenn es um Fußfetischismus und Fußanbetung geht,"*

same meaning. Each of the split words is present in vocabulary while the compound word is OOV. With fixed embeddings, if a word is unknown, we can only assign cosine similarity as 0 making it impossible to pair with any other word. For every unknown word, we try to split the word into smaller chunks in order to capture valid embeddings for compound words. We used Python port of SymSpell library[b] which uses a Triangular Matrix approach instead of Dynamic Programming to split a compound word into smaller known words. For our 1500 dataset the scores did not improve significantly because there were not many OOV words. However, we observe an improvement in the performance on the set with OOV words using compound word split algorithm. For example, consider a English subtitle block, *"When it comes to foot fetishism and foot worship,"* and its correct German translation *"Wenn es um Fußfetischismus und Fußanbetung geht,"*. Without compound word splits we get an $F_{0.5}$ score of 0.80 whereas with compound word split, $F_{0.5}$ score improves to 0.91. German words *"Fußfetischismus"* and *"Fußanbetung"*, meaning foot fetish and foot worship respectively, are OOV but when splited to *"[Fuß, fetischismus]"* and *"[Fuß, anbetung]"*, all splits are present in the dictionary and our score improves by a considerable margin. Figure 5 presents the alignment matrices with and without compound word splitting. We observe that the words align better with word splitting resulting in a better $F_{0.5}$ score.

4.4. *Inclusion of Punctuation*

Punctuation define the sentence type. For example, a sentence ending with a '.' implies an assertive sentence, however, a sentence ending with a '?' implies an interrogative sentence. If punctuation are missing or added incorrectly in target sentence, it

[b] `https://github.com/wolfgarbe/SymSpell`



(a) Excluding punctuation

(b) Including punctuation

Fig. 6: Cosine similarity scores heatmaps showing the difference between calculating score with including and excluding punctuation for English subtitle block, *"Shall I show them to you?"* and German translation *"Soll ich sie dir zeigen"*

changes their sentence type. In above experiments, we included punctuation for each language as opposed to suggested in [1]. Table 4 shows the comparison of baseline and improved system with and without punctuation. We observe that system with punctuation gives higher $F_{0.5}$ scores as opposed to system without punctuation. It is to be noted that the improved system consists of compound word split and uses custom embeddings. Consider English sentence *"Shall I show them to you?"* and its German translation *"Soll ich sie dir zeigen"*. If we exclude punctuation, final score is 0.83 whereas if we include punctuation as well, score increases to 0.86. Figure 6 presents the alignment matrices with including and excluding punctuation. We observe that the words align better by including punctuation, resulting in a higher $F_{0.5}$ score.

4.5. *Performance of system is inversely proportional to length of sentences*

We assess the performance of the system with increasing sentence length as mentioned in Gupta et al. [1]. During our experiments, we observed that performance of our system is inversely proportional to the length of input sentences. We used our improved system with custom alignment, word splits and punctuation inclusion. In figure 7, we show the variation in $F_{0.5}$ scores as we change the maximum allowed number of words in target and source sentence for English to German translation. We observe a sharp decline in $F_{0.5}$ scores for our system as sentence length increases and the performance remains constant after a sentence length of 20 words. This empirically demonstrates the efficacy of our system for small sentences. The subtitle block usually consists of sentences with length less than 10 words. Figure 8, shows the distributions of number of tokens per subtitle block for each of the 5 languages. This justifies the use of our improved system for subtitle quality estimation.

Table 4: Comparison between system with and without using punctuation as separate tokens

Language	**Approach**	$F_{0.5}$	P	R	**Avg. Time (ms)**
French	Baseline	96.4	96.3	96.7	4.5
	Baseline without Punctuation	96.3	96.3	96.6	3.4
	Improved System	97.0	96.3	99.7	9.3
	Improved System without Punctuation	96.9	96.4	99.1	8.0
German	Baseline	71.7	67.9	92.6	4.4
	Baseline without Punctuation	71.6	67.9	92.0	3.9
	Improved System	73.0	68.5	99.1	13.3
	Improved System without Punctuation	71.6	67.9	91.0	9.2
Portuguese	Baseline	82.2	79.4	95.5	4.3
	Baseline without Punctuation	82.1	79.4	94.8	3.2
	Improved System	83.2	79.9	99.7	8.3
	Improved System without Punctuation	83.1	79.8	99.2	7.6
Spanish	Baseline	84.1	81.8	95.2	4.4
	Baseline without Punctuation	84.0	81.7	94.9	3.3
	Improved System	84.9	81.9	99.8	8.0
	Improved System without Punctuation	84.1	81.7	95.4	7.3

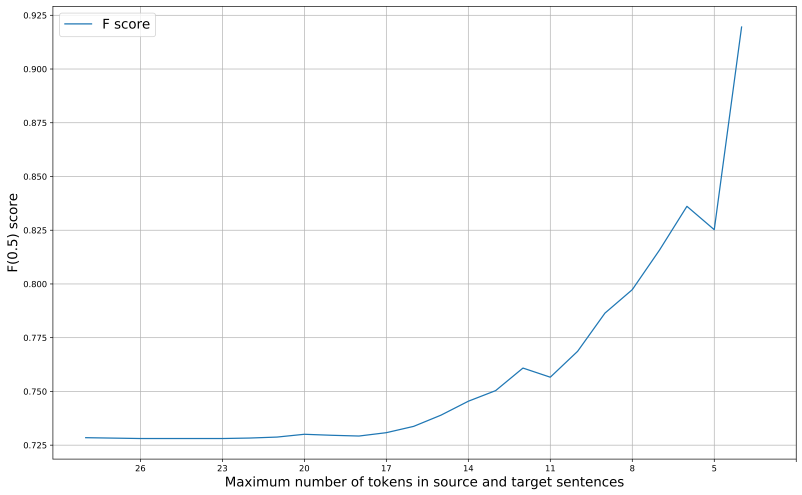
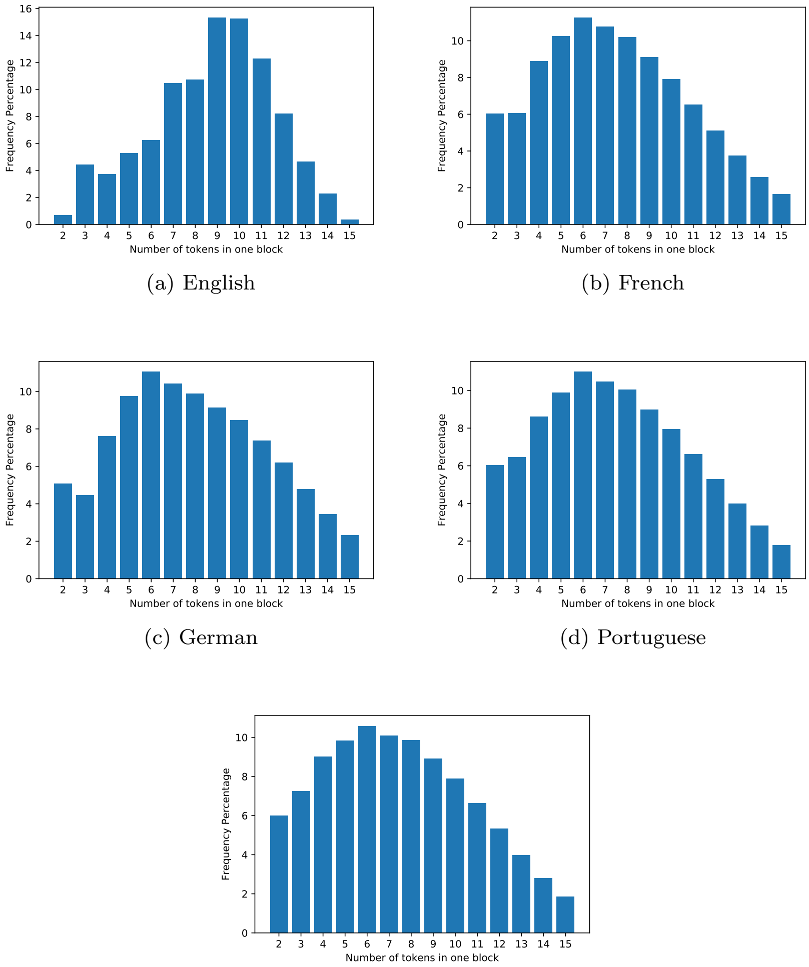


Fig. 7: Increase in $F_{0.5}$ score as length of sentences decrease for English-German



(a) English

(b) French

(c) German

(d) Portuguese

(e) Spanish

Fig. 8: Frequency distributions of number of tokens per subtitle block for each language

5. Conclusion

In this paper, we present various run time and $F_{0.5}$ score improvements for unsupervised translation quality evaluation over the system presented in Gupta et al. [1].

We suggest that training custom monolingual embeddings curated to subtitles and aligning them improves the system. The custom model can reduce the problem of OOV words due to sub-word enrichment since it generate embeddings for the words which are not present in the dictionary of Word2Vec. Pre-aligned word embeddings such as [18, 25] are limited to most frequent $200k$ words and fine-tuning these embeddings to a separate dataset such as subtitles is not possible. Such embeddings cannot tackle the problem of OOV words. We show a massive improvement of the order of $\sim 600\times$ without a significant loss in $F_{0.5}$ score in the run time performance over the system presented in [1]. We remove PII from their system and propose using word-level post-edit scoring instead of a character-level post-edit scores. We further enhanced our system using compound word split as the embeddings generated from sub-word information using our model may not align well with the source words. However, the embeddings generated after compound word splits align well with the words in the source sentence.

We also present other improvements to make system more resilient to different types of language errors. In future, we shall focus on detecting fluency errors and generating a translation score for both adequacy and fluency errors.

References

[1] P. Gupta, S. Shekhawat and K. Kumar, Unsupervised quality estimation without reference corpus for subtitle machine translation using word embeddings, *IEEE 13th International Conference on Semantic Computing (ICSC)* 32–38 (jan 2019).

[2] F. Hieber, T. Domhan, M. Denkowski, D. Vilar, A. Sokolov, A. Clifton and M. Post, Sockeye: A toolkit for neural machine translation, *CoRR* **abs/1712.05690** (2017).

[3] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Łukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes and J. Dean, Google's neural machine translation system: Bridging the gap between human and machine translation, *CoRR* **abs/1609.08144** (2016).

[4] T. Luong, H. Pham and C. D. Manning, Effective approaches to attention-based neural machine translation, in *EMNLP* 2015.

[5] D. Britz, A. Goldie, M.-T. Luong and Q. V. Le, Massive exploration of neural machine translation architectures, *CoRR* **abs/1703.03906** (2017).

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, Attention is all you need, in *NIPS* 2017.

[7] P. Koehn and R. Knowles, Six challenges for neural machine translation, in *NMT@ACL* 2017.

[8] F. Guzmán, A. Abdelali, I. P. Temnikova, H. Sajjad and S. Vogel, How do humans evaluate machine translation, in *WMT@EMNLP* 2015.

[9] K. Papineni, S. Roukos, T. Ward and W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in *ACL* 2002.

[10] M. Snover, B. J. Dorr, R. F. Schwartz and L. Micciulla, A study of translation edit rate with targeted human annotation2006.

[11] G. R. Doddington, Automatic evaluation of machine translation quality using n-gram co-occurrence statistics2004.

[12] S. Banerjee and A. Lavie, Meteor: An automatic metric for mt evaluation with improved correlation with human judgments, in *IEEvaluation@ACL* 2005.

[13] P. Bojanowski, E. Grave, A. Joulin and T. Mikolov, Enriching word vectors with subword information, *Transactions of the Association for Computational Linguistics*

5 135–146 (2017).

[14] M. Faruqui and C. Dyer, Improving vector space word representations using multilingual correlation, in *EACL* 2014.

[15] S. L. Smith, D. H. P. Turban, S. Hamblin and N. Y. Hammerla, Offline bilingual word vectors, orthogonal transformations and the inverted softmax, *CoRR* **abs/1702.03859** (2017).

[16] H. Cao, T. Zhao, S. Zhang and Y. Meng, A distribution-based model to learn bilingual word embeddings, in *COLING* 2016.

[17] M. Zhang, Y. Liu, H. Luan and M. Sun, Adversarial training for unsupervised bilingual lexicon induction, in *ACL* 2017.

[18] A. Conneau, G. Lample, M. Ranzato, L. Denoyer and H. Jégou, Word translation without parallel data, *CoRR* **abs/1710.04087** (2017).

[19] P. Bojanowski, E. Grave, A. Joulin and T. Mikolov, Enriching word vectors with subword information, *Transactions of the Association for Computational Linguistics* **5** 135–146 (2017).

[20] R. Řehůřek and P. Sojka, Software Framework for Topic Modelling with Large Corpora, in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* (ELRA, Valletta, Malta, May 2010), pp. 45–50. `http://is.muni.cz/publication/884893/en`.

[21] L. van der Maaten and G. E. Hinton, Visualizing data using t-sne2008.

[22] T. Mikolov, K. Chen, G. S. Corrado and J. Dean, Efficient estimation of word representations in vector space, *CoRR* **abs/1301.3781** (2013).

[23] J. Pennington, R. Socher and C. D. Manning, Glove: Global vectors for word representation, in *EMNLP* 2014.

[24] M. M. Mogahed, Punctuation marks make a difference in translation: Practical examples.2012.

[25] G. Lample, L. Denoyer and M. Ranzato, Unsupervised machine translation using monolingual corpora only, *CoRR* **abs/1711.00043** (2017).

[26] B. J. Lengerich, A. L. Maas and C. Potts, Retrofitting distributional embeddings to knowledge graphs with functional relations, in *COLING* 2018.

[27] M. W.-D. Marco, Simple compound splitting for german, in *MWE@EACL* 2017.