Unsupervised quality estimation without reference corpus for subtitle machine translation using word embeddings

Prabhakar Gupta
Amazon
prabhgup@amazon.com

Shaktisingh Shekhawat
Amazon
shaktis@amazon.com

Keshav Kumar
Amazon
kumakesh@amazon.com

***Abstract*—We demonstrate the potential for using aligned bilingual word embeddings to create an unsupervised method to evaluate machine translations without a need for parallel translation corpus or reference corpus. We explain why movie subtitles differ from other text and share our experimental results conducted on them for four target languages (French, German, Portuguese and Spanish) with English source subtitles. We propose a novel automated evaluation method of calculating edits (insertion, deletion, substitution and shifts) to indicate translation quality and human aided post edit requirements to perfect machine translation.**

I. Introduction

Translation quality evaluation primarily remains a manual process that requires professional human evaluators with multilingual proficiency. Evaluators use their professional judgment relying upon experience and knowledge to rate translations [1]. However there are problems with this approach. Firstly, it does not scale for high volume evaluation. Secondly, it relies on subjective criteria (an evaluator may rate same translations differently on reevaluation[1]). Thirdly, it incurs heavy cost in terms of money and lastly, for organizations dealing with sensitive data, it also limits usage. Existing automated scoring metrics like BLeU [2], TER [3], NIST [4] or METEOR [5] require reference corpora to establish coherence with the human translated sentence. Generating a dependable corpus for such use is costly and requires time. There has not been a lot of success in completely automated quality estimation at runtime (where a reference corpus is missing) which is at-par with human scores. In this paper, we present a strategy to evaluate movie subtitle translation without using a reference corpus for four language pairs, viz., English-French, English-German, English-Portuguese and English-Spanish. We use aligned bilingual word embeddings to estimate the amount of human post edit [6] that would be required on the given source-translated sentence pair to be at par with the human translated translation. A translation evaluation has many facets and our system focuses upon meaning retention during translation process without focusing on aspects such as punctuation, grammar-based capitalizations, etc.

[1]For our experiment, we took 1500 sentences out which 1494 were unique and for French 2 times, for Portuguese 6 times and for Spanish 1 time, evaluators gave different scores to same sentence pairs.

A. Problem motivation

Amazon Prime Video[2] has a large catalog of titles (movies and TV show episodes) that support subtitles and captions. Subtitles are transcriptions of spoken dialogue or narrative in the title playback. When these transcriptions capture non-spoken text like *"laughs softly"* or *"explosion in the background"*, they are called captions. Subtitles and captions (collectively called subtitles in the remaining paper) drive meaningful engagement with the title. Poor quality subtitles disengage viewers and they prefer to move on to other content / other platforms than reporting the problem. This leads to under reporting of subtitle quality issues and impacts creation of automated processes to generate labelled corpus of *GOOD* / *BAD* translations. We did not discover any public research available for translation quality estimation dedicated to subtitles. Moreover, aforementioned systems rate the quality of translation output on one scale whereas we want a system that can evaluate each translation and hence provide us a score on runtime which lacks in the current state-of-the art systems.

Structurally, a subtitle file (*.srt*, *.dfxp*, etc.) has a list of subtitle blocks each having a start time, an end time and a text string. Every block may also contain formatting rules (like bold, position or color). Blocks may or may not hold a complete dialogue, they can also contain more than one dialogue. Typically, good subtitles conform to specifications[3] that are designed to make reading easier without intruding upon viewing experience and mandate that content displayed on screen to match the subtitles. These factors additionally constrain subtitle making the linguistic property of the sentence slightly different from written text. As an example, a random sample of $8,883$ subtitles in Amazon Prime Video catalog contained $6,112,642$ subtitle blocks with an average word density of 6.41 words per block.

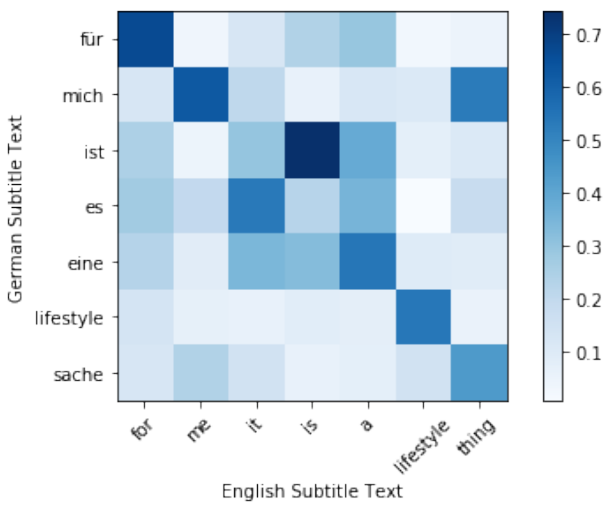
II. Our Approach

Word embeddings[4] (like Word2Vec [7], Glove [8] or FastText [9]) excel at capturing synonymous information [10]. This can be leveraged to get further insight about language

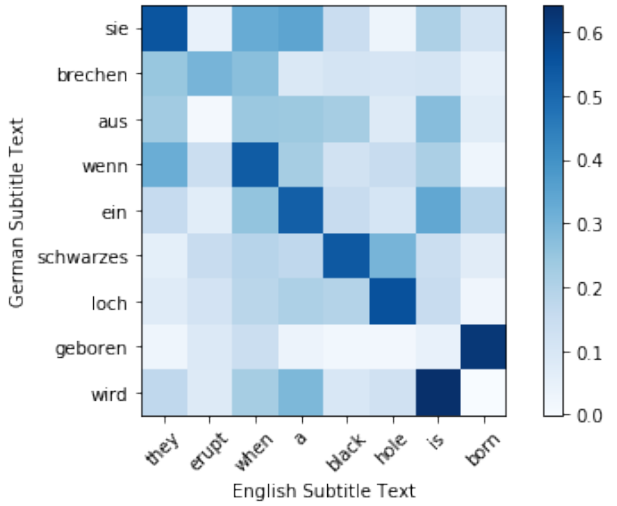
[2]https://www.primevideo.com

[3]BBC Subtitle Guidelines http://bbc.github.io/subtitle-guidelines

[4]Word embeddings are distributed word representations of text allowing words with similar meaning to have a closer representation.



(a) Heatmap of similarity scores for an English-German sentences pair where identifying pairs is simple



(b) Heatmap of similarity scores for an English-German sentences pair where clearly identifying pairs is not simple

Fig. 1.

constructs. When referring to the cosine similarity[5] score, we observed two types of cases, one where source and target (translated) sentences have similar (if not equal) number of words and their order is retained and the other, where the number of words is very different or the order is not maintained. As shown in Figure 1, for a sentence pair: *"For me, it is a lifestyle thing."*, and *"Für mich ist es eine Lifestyle Sache"*, words *"For"* and *"Für"* are in same order in their respective languages. Similarly, for all other words we see a strong one-to-one similarity mapping. On the other hand, consider the example, *"They erupt when a black hole is born."* and *"Sie brechen aus, wenn ein schwarzes Loch geboren wird."*, the one-to-one mapping is relatively absent for all the words. In the example, two correctly translated one-to-one mapped word pairs (*"black"* and *"schwarzes"*, *"hole"* and *"Loch"*) exist.

Recently, there have been attempts to use bilingual word embeddings for unsupervised word translation by aligning two monolingual embeddings [11] but use of these embeddings for translation quality estimation is unexplored. In this work, we use the word translation property of aligned bilingual embeddings to quantify *"how much of the meaning has been correctly translated"*. This insight is core to our approach.

III. Method

We trained a monolingual embeddings for each of the five languages (English, German, French, Spanish and Portuguese) to generate their normalized word embeddings [12]. For any two given languages, their embeddings did not have any natural alignment. As shown in Figure 2, we aligned these independent monolingual embeddings while retaining their individual monolingual spatial integrity. Mathematically, it is equivalent to finding an alignment W of dimensions $d \times d$ (where d is size of one word's embedding) for source embedding X and target embedding Y which maximize the overlap between source and target embeddings such that:

$$W^* = \arg \min_{W \in R^{d \times d}} \|WX - Y\|_F$$

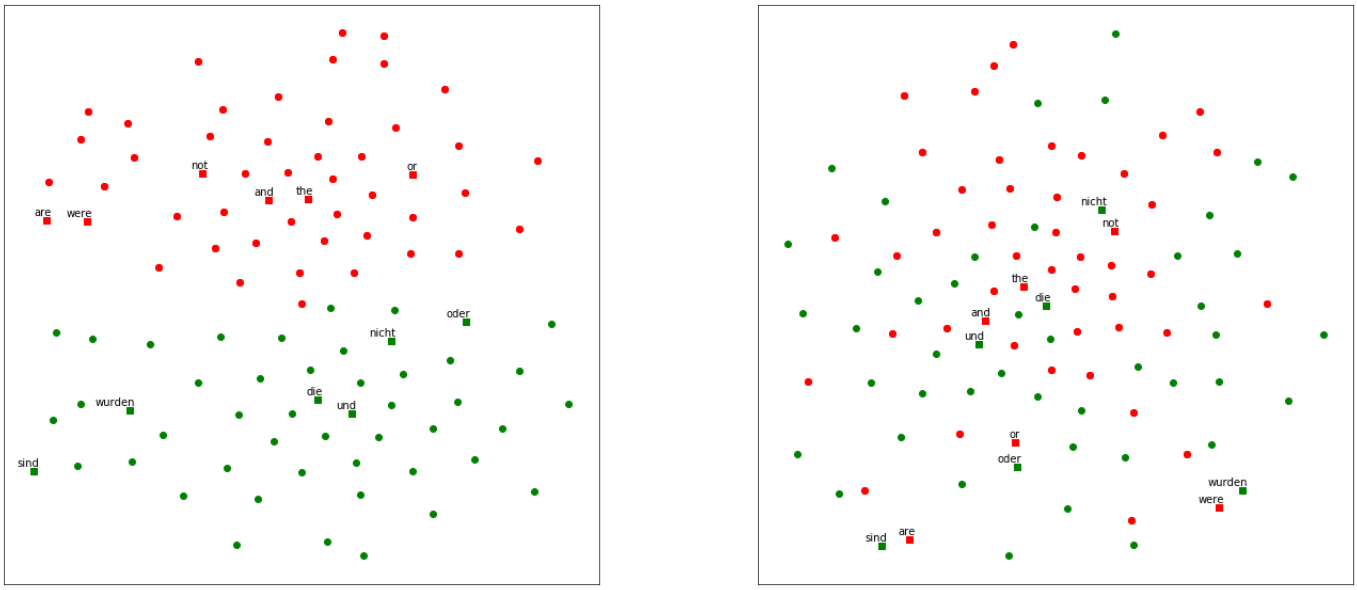
The documented approaches to reduce human supervision in alignment process have used either identical words/phrases in source and target languages [13], considered spatial distribution of vectors [14], or adversarial training [15] to identify the best overlap between two language embeddings. The most successful attempt utilized a hybrid approach to adversarial training [16] and it outperformed all supervised state-of-the-art methods. They learn an initial mapping W using adversarial training and then use the known test query pairs for Procrustes[6]. They also perform training over rare words altering the monolingual space which spreads rare words out of the dense patches. These aligned embeddings were aligned in such a way that these were reusable in the sense that we could very use two target languages to do the same experiment between them. For example, we could use French and German embeddings and have one of them as source language and one as target lannguage.

Using the aligned bilingual embeddings, we followed a 3-step process to estimate a normalized score for a given source-translated sentence pair:

1) Step 1: Generated a list of candidate pairs.: We created a list of word pairs that were correctly translated and calculated the cosine similarity (connection) between each source word's embedding and each target word's embeddings. We selected words which had the highest similarity scores to generate a

[5]Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them.

[6]The orthogonal Procrustes problem [17] is a matrix approximation problem in linear algebra. In its classical form, one is given two matrices A and B and asked to find an orthogonal matrix R which most closely maps A to B.



(a) Before alignment

(b) After alignment

Fig. 2. 2-dimensional t-SNE plots of most frequent 50 words in English and German before and after alignment projected in same plane

list of candidate pairs. Consider the example subtitle block *"we crush him and become the guys"* from TV series 'Silver and Gold'[7] and its German translation *"wir knutschen ihn und werden die kerle"*. For this selection, we faced two issues:

1) Word missing from its respective monolingual dictionary. For example, German word *"knutschen"* was not present in the German monolingual dictionary built from Wikipedia data. We rectified it by assuming the word's distances with every other word in source sentence to be infinite, hence similarity score with each word was 0.
2) Two words having maximum similarity to one target word. For example, For English words *"crush"* and *"him"*, the highest similarity was with German word *"ihn"* (As shown in Table I, 0.2798 and 0.7317 respectively) so we consider the one with highest similarity scores.

	wir	***knutschen***	***ihn***	***und***	***werden***	***die***	***kerle***
we	**0.8063**	0	0.2847	0.3361	0.2424	0.3148	0.3913
crush	0.1187	0	0.2798	0.2282	0.1476	0.1516	0.2579
him	0.2807	0	**0.7317**	0.3342	0.2126	0.2929	0.2685
and	0.2546	0	0.3702	**0.8566**	0.4254	0.5962	0.2065
become	0.2554	0	0.3118	0.3703	0.3001	0.2691	0.0985
the	0.2534	0	0.3234	0.5312	0.3574	**0.6811**	0.1460
guys	0.5607	0	0.2234	0.2190	0.1955	0.2423	0.4735

TABLE I
EXAMPLE TO DEMONSTRATE TWO TYPES OF CONFLICTS

2) Step 2: Eliminated incorrectly paired words.: We wanted to ensure that no two unrelated words are considered as a pair on our candidate list. For each pair, words in the pair should

[7]Silver and Gold https://www.imdb.com/title/tt6254644

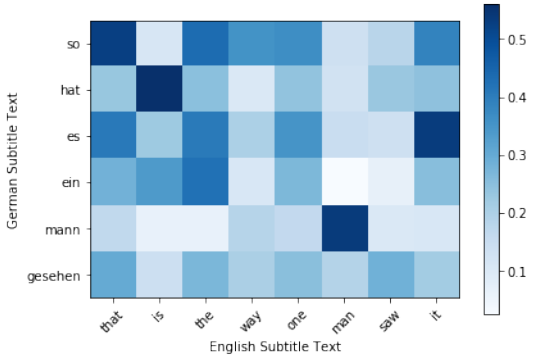


Fig. 3. Cosine similarity scores heatmap for explaining importance of PII

be present in each other's proximity. We introduced a new hyperparameter called *Proximity Intensity Index (PII)* which allows us to check the authenticity of a candidate pair. If the PII value is x, we select x nearest neighbors of source word's vector in target language embeddings and if the target word is not in these neighbors, we discard the pair. This ensured that random selections were avoided. For example, for a sentence pair as shown in Figure 3, *"That is the way one man saw it."* and *"So hat es ein Mann gesehen."*, if we do not set PII, we get 4 pairs whereas, if we set PII to be equal to 10, then we get only 3 pairs. We discard the pair (*"that"*, *"So"*) since they do not belong to each other's neighbour set. Once elimination of incorrect pairs was completed, all remaining pairs were assigned same weightage. As reported later in Table III, setting an optimal PII is very crucial. Results also match the intuition that PII should not be a very large or a very small value.

3) Step 3: Calculated number of post edits.: Once we have all the correctly translated pairs of words, we calculated number of minimum edits required for the target sentence to exactly match the source sentence. We modified the target sentence by replacing target words with the corresponding source word for each pair. We calculated 4 types of edits: insertion, deletion, substitution and shift giving equal weightage to each type of edit. Calculating shifts is computationally expensive as explained in original Translation Error Rate (TER) work [3]. A greedy search on various shifts was used that minimized the number of insertions, deletions and substitutions to select optimum shifts. To find an optimal sequence of edits, the following constraints were added:

- Shifts must exactly match the target words in the destination position.
- Original word sequence of source and corresponding target must not exactly match.
- Destination position of the target word sequence must be misaligned before the shift.
- If a shift reduces the number of insertions, deletions, substitutions by just one, that shift would be considered since alignment is more correct subjectively.

Insertions, deletions and substitutions are calculated using dynamic programming. Original TER calculates the number of edits and normalizes it by the number of unigram tokens in reference sentence (target sentence in our case). We modified this normalization with maximum of number of unigram tokens in source and target sentence to avoid penalizing shorter translations (general lengths of subtitles are small < 6 unigram tokens). Since a sentence pair can have edits ranging from 0 to the length of source target sentence, our modified normalized error rate ranges from 0 to 1 where a score 0 represents no edit was required on the sentence marking it as a perfect translation and score 1 represents none of the words in target sentence made any sense making it a word salad (unacceptable translation).

IV. Experiments

We selected English as the source language and French, German, Portuguese, and Spanish as 4 target languages for the experiment. AWS SageMaker Blazing Text algorithm [12] was used to generate monolingual normalized word embeddings for all 5 languages referring to the data dump[8] of Wikipedia articles. As a part of preprocessing data, we converted every word to lower case, replaced all numeric unigram tokens with zero, removed all the punctuation and expanded all the contractions (*you're* - *you are* or *I'm* - *i am*). Once the preprocessing was completed, we used each Wikipedia article as a single sentence to retain contextual information within one article for training. This information was then dumped all the articles in separate files per language (11.0 GB for English, 3.4 GB for French, 4.6 GB for German, 1.3 GB for Portuguese and 2.8 GB for Spanish). The vocabulary size for each language post embeddings generation was 2278269 words for English, 1023892 words for French, 2065380 words for German, 881525 words for Spanish and 488418 words for Portuguese.

We used Pytorch [18] implementation of hybrid adversarial training approach to align each source-target language pair. We generated ~ 2800 test queries to calculate precision in $'k'$ (1, 5, 10) nearest neighbors for all target language monolingual embedding transformations (in all 5 languages combined). As a standard practice we reported the number of times a correct translation of a source word was retrieved in the target, and tabulated $P@k$ for $k = 1, 5, 10$ in Table II.

Target Language	*# test queries*	*P@1*	*P@5*	*P@10*
French	2903	77.8891	89.5124	91.7168
German	3628	75.8851	89.6460	92.3848
Portuguese	2815	79.8662	89.2308	91.0368
Spanish	2966	83.3222	90.7272	92.9286

TABLE II
PRECISION (@1, @5, @10) FOR ALIGNED WORD EMBEDDINGS FOR ENGLISH TO FRENCH, GERMAN, PORTUGUESE, SPANISH

Our test corpus contained 1500 subtitle blocks (with average of 7.76 words per block for English) taken from subtitles across all genres in catalog (like drama, action, comedy) without any preprocessing. We used Amazon Translate[9] NMT model [19] to translate English sentences to all four target languages. The translated output sentences were later shuffled to lose any contextual information to avoid bias during human evaluation. We engaged professional human translation evaluators (2 evaluators in every language) to score these sentence pairs on the scale of one to six where one was poorest quality (unusable) and six for a perfect translation. Since our system's output score is a measure of edits required for the acceptability of the translated sentence, we defined an acceptability threshold where a translation pair with error score greater than threshold was classified as a bad translation and any error score below threshold classified as good translation. We transformed the human evaluation to align to the acceptability threshold where human scores greater than or equal to 4 were mapped to good translation and scores below 4 were mapped to poor translation.

A. Setting optimum threshold (Choosing β and PII)

We consider the quality of translation critical for customer trust that meant precision had higher preference than recall. Thus we set $\beta = 0.5$ for F_β measure allowing us to give twice as much preference to precision over recall.

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

For choosing PII, we repeated the experiment for different values of PII and calculated the maximum $F_{0.5}$ score for each value of PII. For values of PII where $F_{0.5}$ score was same, we choose the smaller PII value to avoid random pairings.

[8]Wikimedia Downloads https://dumps.wikimedia.org

[9]Amazon Translate https://aws.amazon.com/translate

PII	*French*	*German*	*Portuguese*	*Spanish*
1	0.9704	0.7357	0.8322	0.8522
2	0.9704	0.7336	0.8322	0.8530
5	0.9705	0.7339	0.8322	**0.8539**
10	**0.9707**	0.7345	0.8322	0.8538
20	0.9707	0.7355	0.8322	0.8538
50	0.9707	**0.7368**	**0.8324**	0.8524
100	0.9707	0.7350	0.8324	0.8527
200	0.9707	0.7361	0.8324	0.8531
500	0.9707	0.7360	0.8324	0.8537
1000	0.9707	0.7354	0.8324	0.8536

TABLE III
DIFFERENT VALUES OF MAXIMUM $F_{0.5}$ SCORE FOR DIFFERENT VALUES OF PII (PROXIMITY INTENSITY INDEX) FOR ENGLISH TO FRENCH, GERMAN, PORTUGUESE, SPANISH

V. RESULTS

We experimented with all thresholds starting from 0 to 1 in intervals of 0.001 and calculated $F_{0.5}$ score for each value of threshold. For setting PII, we reported the maximum $F_{0.5}$ score for each PII in Table III.

Figure 4 captures the various values of Precision, Recall and F_β score for thresholds ranging from 0 to 1 in interval of 0.001. Table IV captures the optimum thresholds (threshold where $F_{0.5}$ score was maximum) for all 4 target languages with the precision and recall for that threshold.

	Chosen PII	***Maximum*** $F_{0.5}$ ***score***	***Threshold***	***Precision***	***Recall***
French	10	0.9707	0.800	0.9639	0.9986
German	50	0.7368	0.586	0.7006	0.9291
Portuguese	50	0.8324	0.791	0.7991	0.9992
Spanish	5	0.8539	0.695	0.8280	0.9756

TABLE IV
THRESHOLD, PRECISION AND RECALL AT THE POINT WHERE $F_{0.5}$ SCORE IS MAXIMUM FOR NORMALIZED ERROR SCORES

VI. OBSERVATIONS

1) Our system handles polysemy of words well since we employ word embeddings that provides an insight into language constructs. As an example, for the sentence *"but it keeps me entertained"* and its German translation *"aber es hält mich unterhalten"*, our system gave a score of 0 (indicating perfect translation) even though the word *"unterhalten"* means *"to chat"* when used out-of-context (as in this example). It correctly interpreted *"to entertain"* in the context of the given sentence.
2) The performance of the system depends on the lengths of the query sentences. It works better for smaller sentences making it a suitable solution for subtitles (and colloquial usage) and doesn't perform that well for longer sentences. As shown in Table V, if we restrict the number of words in source sentence, we can significantly improve the performance of the system.

# words	***# samples***	***Maximum*** $F_{0.5}$ ***Score***	***Threshold***	***Precision***	***Recall***
all	1500	0.7368	0.586	0.7006	0.9291
10	1206	0.7544	0.594	0.7176	0.9495
9	1055	0.7626	0.594	0.7273	0.9468
8	900	0.7711	0.563	0.7425	0.9120
7	710	0.7893	0.563	0.7636	0.9122
6	508	0.8013	0.566	0.7761	0.9211
5	341	0.8147	0.572	0.7928	0.9163
4	183	0.8356	0.500	0.8311	0.8542
3	80	0.8832	0.584	0.8732	0.9254
2	22	0.9659	0.579	1.0000	0.8500

TABLE V
FOR ENGLISH-GERMAN, PRECISION AND RECALL FOR THE SYSTEM IMPROVES DRASTICALLY AS WE RESTRICT THE LENGTHS OF SOURCE SENTENCES (# OF WORDS IN SOURCE SENTENCE)

3) For sentence pairs that translate into different length sentences, the system tends to perform inefficiently. For example, *"So I did that."* and its German translation *"Also habe ich das getan."*, humans scored these to be a good translation (i.e. a score of 6) whereas our system scored it at 0.625 which is *BAD* translation classification according to our threshold (From Table IV Threshold = 0.586). This was primarily because of difference in lengths of source and translated sentences (Cosine similarity scores in Table VI).

	Also	***habe***	***ich***	***das***	***getan***
So	**0.4790**	0.3436	0.4054	0.2572	0.3893
I	0.3005	0.1523	0.3827	0.1139	0.2939
did	0.1908	0.3796	0.2390	0.1914	0.3622
that	0.3876	**0.4573**	0.2492	0.3277	0.3110

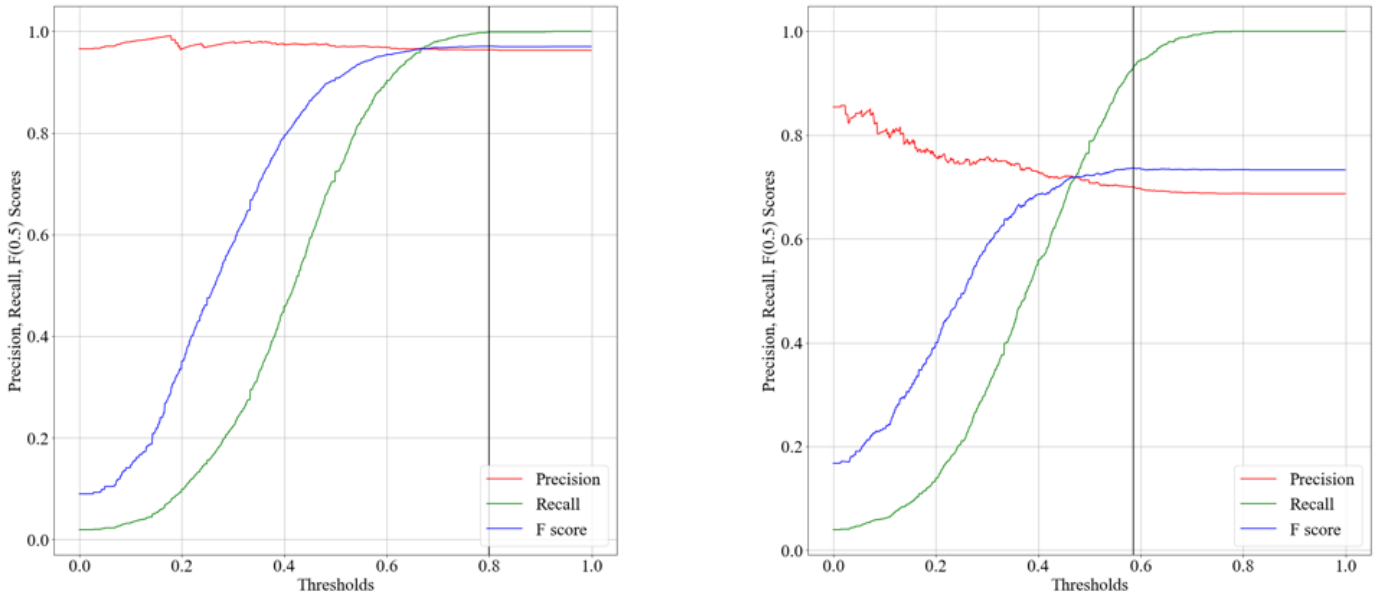
TABLE VI
EXAMPLE TO SHOW THAT FOR THE SENTENCES THAT ARE NOT CLOSE TO WORD-TO-WORD TRANSLATION SYSTEM TEND TO PERFORM POOR

4) The system works better for the languages that are structurally closer[10] to the source language (English). We were able to achieve a precision of 0.96 and recall of 0.99 for French which structurally closest (as compared to other three) to English whereas for languages Portuguese and Spanish which are relatively distant from English do not perform as good.
5) If we ignore the calculation of *shifts* during number of edits calculation, we observe a speed up of at least $2\times$ with statistically insignificant (5% decrease in $F_{0.5}$ scores as shown in Table VII) impact to acceptable scores.

VII. CONCLUSION

We successfully built a system that can evaluate a subtitle source and target sentences pair for one of four language arcs (English-French, English-German, English-Portuguese, English-Spanish) with precision of at least 70% and recall of 90% without a costly translation parallel or a reference corpus

[10]Most Similar Languages http://www.ezglot.com/most-similar-languages.php



(a) English-French

(b) English-German

(c) English-Portuguese

(d) English-Spanish

Fig. 4. Precision Recall $F_{0.5}$ score for different thresholds at the optimum PII. Vertical line represents the threshold chosen where $F_{0.5}$ score is maximum.

	$F_{0.5}$ *score with shifts*	$F_{0.5}$ *score without shifts*	*Time taken (in sec) with shifts*	*Time taken (in sec) without shifts*
German	0.9707	0.9463	2.83	0.80
French	0.7368	0.7228	2.93	1.19
Portuguese	0.8324	0.8143	2.48	0.85
Spanish	0.8539	0.8459	2.69	1.19

TABLE VII
TIME TAKEN AND $F_{0.5}$ SCORES COMPARISON, WITH AND WITHOUT CALCULATING *shifts*

(as needed in state-of-the art automated systems like BLeU, TER or METEOR). We introduced an optimum threshold for each language pair that can classify a translation as *GOOD* or *BAD*. We presented a novel approach to leverage word embeddings and their bilingual alignment which were used to perform word translations but not for translation evaluations. Using word embeddings, made system resistant to polysemy and synonyms. We extended the existing industry accepted scoring system to work without a reference corpus to make them usable at runtime.

VIII. FUTURE WORK / POSSIBLE IMPROVEMENTS

1) In extension to our current study, we shall consider many-to-many translations for entities and other words

that mean different independently but together they mean completely different, e.g.- *"united"* and *"airlines"* independently mean something else but *"united airlines"* completely means something else. We can try to change vanilla edit distance calculations by incorporating word distances [20].

2) For longer sentences proposed solution doesn't perform that good we can improve the system by taking in account the length of sentences and penalizing the types of edits differently in order to let minor edits pass through.

IX. Appendix - Experimental Setup

A. Hyperparameters and Amazon EC2 instances hardware configurations for generating monolingual models

1) Mode: *Continuous Bag of Words (CBoW)*
2) Context Window size: 5
3) Vector Dimension: 300
4) Minimum Frequency Count: 5
5) Learning Rate: 0.05
6) Amazon EC2 instance type: *ml.m5.24xlarge*
7) Instance volume size: 64 *GB*
8) Instance max runtime: 86400 *seconds*

B. Hyperparameters and Amazon EC2 instances hardware configurations for alignment of one language pair

1) Number of iterations: 5
2) maximum vocabulary size: 200000
3) Generator learning rate: 0.1
4) Validation distance: *CSLS (Cross-domain Similarity Local Scaling)*
5) Batch Size: 64
6) Pytorch version: 0.4.0
7) CUDA version: 9.0.176
8) Amazon EC2 instance type: *g3.4xlarge*
9) Instance AMI: *Deep Learning AMI (Ubuntu) Version 9.0*
10) vCPUs: 16
11) Instance memory: 122 *GiB*[11]
12) Instance storage size: 1024 *GiB*
13) Number of GPUs: 1
14) GPU memory: 8 *GiB*

X. Appendix - Human scores analysis

The distribution of scores is subjective to the proficiency and experience of the translators. All the translations were created from state-of-the-art translation engine, Amazon Translate. Therefore, instead of simply asking humans to give a binary label of good or bad translation, we asked them to rate it on scale of 6.

[11]GiB (Gibibyte) are expressed as a base of 2. 1 GiB equals 2^{30} = 1073741824 bytes or 1024 MiB

	Score - 1	*Score - 2*	*Score - 3*	*Score - 4*	*Score - 5*	*Score - 6*
French	8	20	28	206	189	1049
German	20	95	354	326	289	415
Portuguese	1	13	288	566	225	407
Spanish	14	60	197	474	458	297

TABLE VIII
DISTRIBUTION OF SCORES DONE BY HUMAN EVALUATORS (6 BEING A SCORE FOR PERFECT TRANSLATION AND 1 BEING A SCORE FOR POOR TRANSLATION) ON 1500 SUBTITLE BLOCKS WITH AVERAGE OF 7.76 WORDS PER SUBTITLE BLOCK

References

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

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

[3] M. Snover, B. J. Dorr, R. F. Schwartz, and L. Micciulla, "A study of translation edit rate with targeted human annotation," 2006.

[4] G. R. Doddington, "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics," 2004.

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

[6] L. Specia, "Exploiting objective annotations for measuring translation post-editing effort," 2011.

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

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

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

[10] A. Leeuwenberg, M. Vela, J. Dehdari, and J. van Genabith, "A minimally supervised approach for synonym extraction with word embeddings," 2016.

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

[12] S. Gupta and V. Khare, "Blazingtext: Scaling and accelerating word2vec using multiple gpus," in *MLHPC@SC*, 2017.

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

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

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

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

[17] P. H. Schönemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, Mar 1966. [Online]. Available: https://doi.org/10.1007/BF02289451

[18] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

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

[20] H. Wang and P. Merlo, "Modifications of machine translation evaluation metrics by using word embeddings," in *HyTra@COLING*, 2016.