Keep It or Not: Word Level Quality Estimation for Post-Editing

Prasenjit Basu[1], Santanu Pal[2,3], Sudip Kumar Naskar[4]
[1]Future Institute of Engineering and Management, India
[2]Saarland University, Germany,
[3]German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany
[4]Jadavpur University, India
basuprasen@gmail.com, santanu.pal@uni-saarland.de,
sudip.naskar@jdvu.ac.in

Abstract

The paper presents our participation in the WMT 2018 shared task on word level quality estimation (QE) of machine translated (MT) text, i.e., to predict whether a word in MT output for a given source context is correctly translated and hence should be retained in the post-edited translation (PE), or not. To perform the QE task, we measure the similarity of the source context of the target MT word with the context for which the word is retained in PE in the training data. This is achieved in two different ways, using *Bag-of-Words* (*BoW*) model and *Document-to-Vector* (*Doc2Vec*) model. In the *BoW* model, we compute the cosine similarity while in the *Doc2Vec* model we consider the Doc2Vec similarity. By applying the Kneedle algorithm on the F1-mult vs. similarity score plot, we derive the threshold based on which OK/BAD decisions are taken for the MT words. Experimental results revealed that the Doc2Vec model performs better than the BoW model on the word level QE task.

1 Introduction

Evaluating and estimating quality of a machine translation (MT) system without referring the actual translation is now one of the key research areas in MT domain (Blatz et al., 2004; Specia et al., 2009). In a machine translated document quality estimation can be performed at various granularities like word level, phrase level or sentence level (Specia et al., 2010, 2013). Scarton et al. (2016) produced their task in WMT16 in document level quality estimation with winning result in two different models (Bojar et al., 2016). One model used discourse features and SVR and another model employed word embedding feature and Gaussian Process for quality estimation. (Biçici, 2017) predicted translation performance with referential translation machines at word level, sentence level and at phrase level. (Blain et al., 2017) submitted task on bi-lexical word embedding in WMT17 QE shared task, which produced promising results in sentence level Quality Estimation. Some studies (Fiederer and OBrien, 2009; Koehn, 2009; DePalma and Kelly, 2011; Zampieri and Vela, 2014) show that the quality of MT output along with PE can produce better result than human editor in certain situations.

In our work we mainly focus on word level quality estimation. The distributional structure of words was first described by (Harris, 1954). (Turian et al., 2010) illustrated representations of words in semi-supervised learning. Bengio et al. (2003) proposed neural probabilistic language model by using a distributed representation of words. Collobert and Weston (2008), described how a convolutional neural network architecture could be used to make different language processing predictions, such as semantically similar words, etc. Mnih and Hinton (2008) proposed a fast hierarchical language model along with a feature based algorithm which automatically builds word trees from data. Mikolov et al. (2013b) proposed vector representation of words with the help of negative sampling (instead of softmax function) that improves both word vector quality and training speed. Their work showed prediction of a word from a context by adding two word vectors from the same context. (Mikolov et al., 2013a) proposed a novel approach to represent words as fixed length vectors, widely known as word2vec model and they reported state-of-the-art performance on word similarity task. (Le and Mikolov, 2014) extend their model to vector representation of a document known as Paragraph Vector model or commonly Document-to-Vector (Doc2Vec) model.

This paper reports our submission in the WMT 2018 Shared Task on Word-Level Quality Estima-

Proceedings of the Third Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 759–764

https://doi.org/10.18653/v1/W18-64084

tion (QE task-2) on English–German (IT domain) SMT data. The proposed model has been developed in two ways - one using the standard Bag-of-Words model and another using the Doc2Vec model. The motivation behind the use of Doc2Vec model is to achieve more accurate semantic similarity compared to the simple cosine similarity on Bag-of-Words model. The Doc2Vec model captures semantic similarity which the Bag-of-Words model can not. Our word level error estimation is mainly based on Translation Error Rate (Snover et al., 2006) between MT and PE.

2 Proposed Approach

Our system highlights the retention of a word in MT translation and thus it helps human post-editors to increase their productivity with less effort. Our QE system is built over the Translation Error Rate (TER) (Snover et al., 2006) alignment between MT output and the corresponding PE output in the training data. TER alignment shows whether words from MT data (hypothesis in TER) will be continued, deleted or substituted with respect to the PE data (reference in TER). Based on the TER alignment, we build binary classification models that suggests OK for continuation and BAD for deletion or substitution.

Our QE system follows two models: Bag-of-Words Model and Document-to-Vector based model as described in the following subsections.

2.1 Bag-of-Words Model

MT words that are retained in PE are identified through TER alignment. In the Bag-of-words (BoW) model, for each word (w_i) in MT that is retained in PE in the training set, we find the corresponding source texts ($src^*_{w_i}$). A BoW (B_{w_i}) is then formed from the $src^*_{w_i}$ for each such w_i that are present in both MT and the corresponding PE in the training set. Algorithm 1 presents the BoW creation method. B_{w_i} contains more repetition of the source words which actually bear the meaning of w_i.

On the development set, we also establish TER alignment between the MT text (MT_{dev}) and the PE text (PE_{dev}). For each word (say, w_j) appearing in each sentence in MT_{dev}, we consider the corresponding src as the source context (say, src_{w_j}) and keep track of the post-editing operation required on the word (through TER alignment), i.e., whether the word is retained (OK) in PE or not (BAD). Then we compute the cosine similarity between src_{w_j} and B_{w_j}.

Input: src–mt–pe parallel training data and TER alignments between mt and pe
Output: source BoW (B_{dict}) for each target word
begin
  $V_{list} \leftarrow NULL$
  $B_{dict} \leftarrow NULL$
  foreach *sentence* $mt_i \in mt$ **do**
    foreach $T_{i,j} \in mt_i$ **do**
      if $T_{i,j}$ *is retained in* pe_i **then**
        if $T_{i,j} \notin V_{list}$ **then**
          $V_{list}.add(T_{i,j})$
        end
        $B_{list} \leftarrow NULL$
        forall $S_{i,k} \in src_i$ **do**
          $B_{list}.add(S_{i,k})$
        end
        $B_{dict}[T_{i,j}].add(B_{list})$
      end
    end
  end
  return B_{dict}
end

Algorithm 1: Creation of source BoW; $T_{i,j}$ is the j^{th} word of the i^{th} mt sentence and $S_{i,k}$ is the k^{th} word of i^{th} src sentence.

The similarity scores range between 0 and 1 with varying distribution. We aim to arrive at a threshold on the similarity score above which the system takes the OK decision, otherwise the BAD decision. This threshold is trained on the development set. However, the datasets, both training and development, are highly imbalanced; 85.66% and 83% of the mt tokens are retained (i.e., OK) in pe in the training set and the development set respectively, and the rest are discarded or changed (i.e., BAD), which indicates that the mt data was generated by a strong MT system. Such imbalance in the dataset proves to be a major hurdle in automatic QE or post-editing. The imbalance in the dataset leads to the fact that a very simple baseline of setting the threshold to 0 results in 85% F1-score on the development set (we consider only the non-stop words), which is very difficult to defeat.

The similarity scores obtained for the development set MT words are divided into a number of segments (or ranges) for equal distribution such that there are roughly equal number of instances in each range (cf. Table 3). The upper bound of each segment corresponds to a threshold.

We compute F_1-$mult$[1] for each of the segments

[1] F_1-$mult$ is the multiplication of F_1 scores for the OK and BAD classes, and is the official evaluation metric for the WMT QE shared task.

and produce the F_1-$mult$ curve. Figure 1 shows the F_1-$mult$ curve on the development set which does not lead to any peak or intermediate threshold. We use the Kneedle algorithm (Satopaa et al., 2011) to find a knee point on the F_1-$mult$ curve which serves as the threshold for our model and based on this threshold we take the OK/BAD decision.

For each test set MT word (say w_k), we generate the similarity score between B_{w_k} and the current source src_{w_k}. If the score is above the threshold, the word is predicted as OK, otherwise BAD.

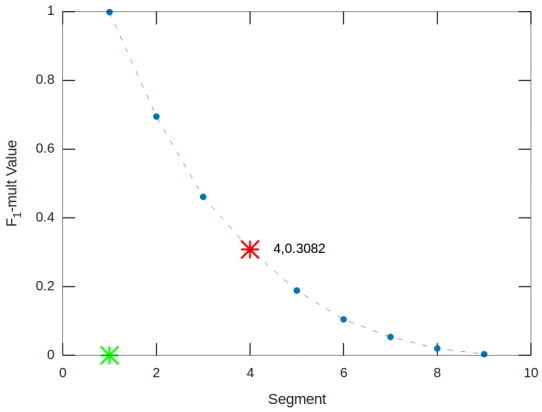

Figure 1: Segment vs. F_1-$mult$ plot on the development set for the BoW model. Red Mark denotes the (segment, F_1-$mult$) co-ordinate value for knee point and green Mark describes segment starting position.

2.2 Document-to-Vector based Model

In the Document-to-Vector (Doc2Vec) model for QE, for an MT word w_i, we also compute similarity between src_{w_i} and B_{w_i}. However, here instead of the considering them as BoW, we treat them as documents and measure their Doc2Vec similarity score (Sim_{D2V}). For this, we prepare document vector for each src_{w_i} and B_{w_i} using gensim (Rehurek and Sojka, 2010). Gensim has its own implementation of document embedding via distributed memory or distributed Bag-of-Words model. In its model each document is represented as a fixed length vector. It is a generalization of and derived from the word2vec model. The QE decision is taken based on whether the Sim_{D2V} for the word is above or below the threshold which is trained on the development set, as in the case of the BoW model. To train our Doc2Vec model we remove all stop words from the training data. For obtaining the threshold, the Doc2Vec similarity scores are divided into a number of segments of equal distribution. Like the BoW model, we generate the F_1-$mult$ curve on those similarity scores and use the Kneedle algorithm to find the threshold.

3 Experiments

We used the WMT-2018 English–German (EN–DE) word level QE dataset for our experiments. Table 3 presents the statistics of the training, development and test sets. Stop words generally occur very frequently and their number of occurrences across BoW could easily mislead word-level QE. Therefore we process the training data by removing stop words for both German[2] and English from all the data sets, i.e., neither we consider them while building our context bags, nor we consider their QE.

	Sentences	Tokens		
		src	*mt*	*pe*
Train	26,299	389,070	393,000	400,058
Dev	1000	14,600	14,773	14,970
Test	1926	28,312	28,785	-

Table 1: Statistics of the the WMT-2018 Word Level QE Shared Task Data Set.

We considered 9 thresholds for the BoW model. Table 3 shows the segments and the corresponding thresholds.

Seg. No	Threshold
1	0.075
2	0.15
3	0.2
4	0.25
5	0.31
6	0.38
7	0.47
8	0.58
9	1

Table 2: Segment versus Threshold values for the BOW model

Table 3 shows word specific assignment of binary scores to each threshold. For a word with QE decision OK, a word–threshold cell is assigned to 1 if the similarity score for the corresponding word is higher than the corresponding threshold, and

[2] https://www.ranks.nl/stopwords/german

Token	PE Decision	Score	Threshold								
			0.075	0.15	0.2	0.25	0.31	0.38	0.47	0.58	1
hinzugefgt	OK	0.32	1	1	1	1	1	0	0	0	0
verhalten	OK	0.26	1	1	1	1	0	0	0	0	0
zustzliche	BAD	0.23	0	0	0	1	1	1	1	1	1
verknpfen	OK	0.23	1	1	1	0	0	0	0	0	0
wird	OK	0.32	1	1	1	1	1	0	0	0	0
verborgene	OK	0.37	1	1	1	1	1	0	0	0	0
enthlt	BAD	0.03	0	1	1	1	1	1	1	1	1
balken	OK	0.21	1	1	1	0	0	0	0	0	0
fenster	OK	0.17	1	1	0	0	0	0	0	0	0
sol	BAD	0.5	0	0	0	0	0	0	0	0	1

Table 3: A snapshot of the intermediate table showing word–threshold pair assignment

Seg No	Th. Value
1	0.001
2	0.12
3	0.21
4	0.4
5	0.1

Table 4: Segment vs. threshold values for the Doc2Vec model

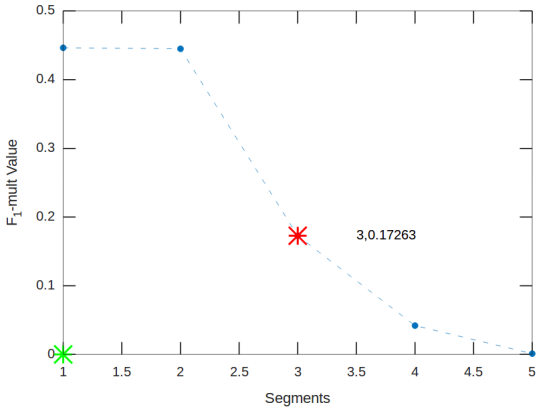

Figure 2: Segments versus F_1-$mult$ plot on training set of Doc2Vec model. Red Mark denotes the (segment, F_1-$mult$) co-ordinate value for knee point and green Mark describes segment starting position.

0 otherwise. For words with PE decision BAD, scores are assigned the other way round. It is to be noted that our model can only predict the QE decision for words that are already seen in the training set. Words that are not present in the training set (including stop words) are simply retained.

Kneedle algorithm on the Segments vs. F_1-$mult$ plot on the development set (cf. Figure 1) leads to the segment 4 as the knee point and the corresponding similarity score of 0.25 (cf. Table 3) serves as the threshold, which produces the optimal F_1-$mult$ for the BoW model.

For the Doc2Vec based experiment, gensim creates models using distributed Bag-of-Words. Doc2Vec similarity is measured between the vector representation of the Bag-of-Words and the source context for each target word from training data. The scores were distributed among 5 segments (cf. Table 3). Figure 2 shows the Segments vs. F_1-$mult$ plot for the Doc2Vec model. From the plot we take the knee value of the graph, i.e. segment 3 and the corresponding similarity score 0.21 (cf. Table 3) is considered as the threshold for the Doc2Vec model.

According to the WMT18 published results for the word level quality estimation task (Task 2), the results of our two models along with baseline are shown in Table 3. The evaluation results suggest that the Doc2Vec based word level QE model performs better than the Bag-of-Words based model for both the OK class and the BAD class on the WMT18 testset.

The expected results could have been better if we could use larger dataset as Doc2Vec model performs better for bigger data sources (Azunre et al., 2018). For Bag-of-Words based model we have removed stop words from those Bag-of-Words for the target German word of MT which itself is not a stop word. We also removed all stop words from test data. Removal of stop words from training data and test data leads to not-up-to-the-mark performance.

Participant	*Model*	F_1-BAD	F_1-OK	F_1-$mult$
fblain	BASELINE	0.4115	0.8821	0.3630
basuprasen	Doc2Vec	0.2889	0.7547	0.2180
basuprasen	BagOfWords	0.2784	0.7335	0.2042

Table 5: Evaluation Results on the WMT18 Word level Quality Estimation (Task 2)

4 Conclusions and Future Work

The paper reports our participation in the WMT 2018 shared task on word level quality estimation (QE task2) on English–German SMT data. The task of word level QE is treated as a binary classification problem — i.e., decision is taken about whether a word under consideration is to be retained or not. The prediction is performed by measuring the similarity of the source context of the target word with the context for which the word is retained. This is achieved in two ways, using BoW model and Doc2Vec. Experimental results suggest that the Doc2Vec model can model this much more effectively than the Bag-of-Words model. An obvious extension of this work would be to extend our model to phrase-level QE and determining missing words and source words that lead to errors.

Acknowledgments

Santanu Pal is partly funded by the German research foundation (DFG) under grant number GE 2819/2-1 (project MMPE). Sudip Kumar Naskar is supported by Digital India Corporation (formerly Media Lab Asia), MeitY, Government of India, under the Young Faculty Research Fellowship of the Visvesvaraya PhD Scheme for Electronics & IT. We also want to thank the reviewers for their valuable input, and the organizers of the shared task.

References

Paul Azunre, Craig Corcoran, David Sullivan, Garrett Honke, Rebecca Ruppel, Sandeep Verma, and Jonathon Morgan. 2018. Abstractive tabular dataset summarization via knowledge basesemantic embeddings. *arXiv preprint arXiv:1804.01503*.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155.

Ergun Biçici. 2017. Predicting translation performance with referential translation machines. In *Proceedings of the Second Conference on Machine Translation*, pages 540–544.

Frédéric Blain, Carolina Scarton, and Lucia Specia. 2017. Bilexical embeddings for quality estimation. In *Proceedings of the Second Conference on Machine Translation*, pages 545–550.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *Proceedings of the 20th international conference on Computational Linguistics*, page 315. Association for Computational Linguistics.

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. 2016. Findings of the 2016 conference on machine translation. In *ACL 2016 FIRST CONFERENCE ON MACHINE TRANSLATION (WMT16)*, pages 131–198. The Association for Computational Linguistics.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA. ACM.

Donald A. DePalma and Nataly Kelly. 2011. Project management for crowdsourced translation: How user-translated content projects work in real life. *Translation and Localization Project Management: The art of the possible*, XVI:379–408.

Rebecca Fiederer and Sharon OBrien. 2009. Quality and machine translation: A realistic objective. *The Journal of Specialised Translation*, 11:52–74.

Zellig S. Harris. 1954. Distributional structure. *WORD*, 10(2-3):146–162.

Philipp Koehn. 2009. A process study of computer-aided translation. *Machine Translation*, 23:241–263.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Andriy Mnih and Geoffrey Hinton. 2008. A scalable hierarchical distributed language model. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, NIPS'08, pages 1081–1088, USA. Curran Associates Inc.

Radim Rehurek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Citeseer.

Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. 2011. Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In *Proceedings of the 2011 31st International Conference on Distributed Computing Systems Workshops*, ICDCSW '11, pages 166–171, Washington, DC, USA. IEEE Computer Society.

Carolina Scarton, Daniel Beck, Kashif Shah, Karin Sim Smith, and Lucia Specia. 2016. Word embeddings and discourse information for quality estimation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, volume 2, pages 831–837.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200.

Lucia Specia, Dhwaj Raj, and Marco Turchi. 2010. Machine translation evaluation versus quality estimation. *Machine translation*, 24(1):39–50.

Lucia Specia, Kashif Shah, Jose GC Souza, and Trevor Cohn. 2013. Quest-a translation quality estimation framework. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 79–84.

Lucia Specia, Marco Turchi, Nicola Cancedda, Marc Dymetman, and Nello Cristianini. 2009. Estimating the sentence-level quality of machine translation systems. In *13th Conference of the European Association for Machine Translation*, pages 28–37.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 384–394, Stroudsburg, PA, USA. Association for Computational Linguistics.

Marcos Zampieri and Mihaela Vela. 2014. Quantifying the influence of mt output in the translators' performance: A case study in technical translation. In *EACL Workshop on Humans and Computer-assisted Translation (HaCat)*, pages 93–98.