Electronic Preprint for Journal of Information Processing Vol.27

Regular Paper

On Cross-Lingual Text Similarity Using Neural Translation Models

Kazuhiro Seki[1,a)]

Received: September 4, 2018, Accepted: November 7, 2018

Abstract: Accurately computing the similarity between two texts written in different languages has tremendous value in many applications, such as cross-lingual information retrieval and cross-lingual text mining/analytics. This paper studies the important problem based on neural networks. Specifically, our focus is on the neural machine translation models. While translation models are utilized, we pay special attention *not* to the translation itself *but* to the intermediate states of given texts stored in the translation models. Our assumption is that the intermediate states capture the syntactic and semantic meaning of input texts and are a good representation of the texts, avoiding inevitable translation errors. To study the validity of the assumption, we investigate the utility of the intermediates states and their effectiveness in computing cross-lingual text similarity in comparison with other neural network-based distributed representations of texts, including word and paragraph embedding-based approaches. We demonstrate that an approach using the intermediate states outperforms not only these approaches but also a strong machine translation-based one. Furthermore, it is revealed that intermediate states and translated texts work complementarily each other despite the fact that they are generated from the same NMT models.

Keywords: document similarity, distributed representation, neural network, cross-lingual information retrieval

1. Introduction

Similarity between two documents is a crucial measure commonly used for various applications from text clustering to information retrieval (IR). The most common way of computing document similarity would be first representing two documents by word vectors with/without some term weighting schemes such as TFIDF [25], and then computing their inner product or cosine similarity [16]. An assumption underlying such similarity measures is that two documents are more similar if they share more terms in common. Although the idea is intuitive and valid in many cases, it cannot be applied to cases where the languages used in the two documents are different from each other.

The most straightforward approach to solving the problem is to translate one document to the language in which the other document is written [4], [22] by a machine translation (MT) system. The MT-based approach has been shown effective but has potential drawbacks. For instance, a word in one language may be polysemous (e.g., "crane" and "bank") and may not be correctly translated to the right word in the other language depending on the performance of the MT system utilized. Furthermore, even if a word in one language has only a single sense, it may have multiple corresponding words/expressions (i.e., synonyms) in the other language. For example, suppose that we are to measure similarity between an English document and a Japanese document and that the latter contains a Japanese word "警察" (the people or the department who enforce laws and investigate crimes). The word can be translated to "police", "policemen", "authorities", "constabulary", etc. in English depending on the context. Although all the translations can be considered correct and refer to the same concept (with possibly different nuances), only the one actually used in the English counterpart positively contributes to their similarity.

Another approach is to convert two documents into a common semantic space in which they can be directly or indirectly compared for their similarity [5], [8], [31]. For instance, a seminal work was done by Dumais et al. [8], who applied latent semantic analysis (LSA) [7] to an English and French parallel corpus in order to obtain a reduced dimension semantic space where terms/documents in both languages are mapped. The present study explores this direction but uses sequence-to-sequence neural machine translation (NMT) models [26] *without* translation to derive vector representation of a given document pair so as to measure their similarity.

The contribution of this work is four-fold: First, as far as we know, this is the first attempt to study the utility of NMT models for cross-lingual texts similarity. Second, as a proof of concept, empirical evaluation is carried out on an English-Japanese translation corpus to study its effectiveness as compared with alternatives including other neural network-based and machine translation-based approaches. Third, despite its simplicity, the NMT-based approach is shown to work strikingly well. Fourth, combining the NMT-based approach with resulting machine translations considerably boosts the performance of cross-lingual document similarity.

The rest of the paper is constructed as follows: Section 2 introduces representative work related to cross-lingual document simi-

[1] Konan University, Kobe, Hyogo 658–0062, Japan
[a)] seki@konan-u.ac.jp

larity and neural machine translation models. Section 3 describes the approach based on the intermediate representation stored in NMT models. Section 4 details the evaluative experiments and reports on the results. Section 5 concludes the paper with a brief summary, limitations, and possible future directions.

2. Related Work

With the potential impact on a wide range of applications including IR and data/text mining, there have been a number of studies on cross-lingual text similarity, partly motivated by the SemEval workshop [4]. A straightforward yet effective approach to the task is to use machine translation (MT) systems to make the problem monolingual [22]. However, this approach is dependent on the availability of an MT system for a given language pair and, even if it is available, the approach is likely to be sensitive to the errors made by the system. As mentioned in the previous section, antonyms and synonyms are problematic for this approach, especially for short texts (e.g., microblogs) where a few word mismatches will have a relatively large impact on the overall similarity.

Another approach is based on the family of matrix decomposition [8], [24], [30]. This type of approach requires a parallel translation corpus and constructs a term-document matrix for each language. The matrices are then merged into a single large matrix such that each column of the matrix corresponds to a pair of translations. The resulting matrix is decomposed into a set of orthogonal factors from which the original matrix is approximated. In the approximated vector space, semantically similar words and documents occur near each other independent of their languages. Then, a new document pair can be folded into the common semantic space, where their equivalence can be directly computed by, for example, cosine similarity.

Yet another approach employs some form of a neural network and typically attempts to learn mapping between two languages [9]. For example, bilingual autoencoders [5] extend the idea of autoencoders [11] which learn efficient codings of an input given the input itself as the output. The bilingual autoencoders learn mapping between two languages by reconstructing an input sentence in one language not to itself but to its translation in the other language. Similarly, S2Net [31] trains a Siamese network for each language and learns a transformation matrix such that the similarity of translations in different languages is minimized. Despite these efforts, however, it has been shown that approaches including matrix decomposition do not perform as well as a simple MT-based approach [10].

The present work also explores an approach based on neural networks but specifically focuses on neural machine translation (NMT) models [2], [6], [26]. A representative work on NMT was presented by Sutskever et al. [26]. They proposed a sequence-to-sequence model which maps an input sequence to another sequence using two recurrent neural networks (RNN), each composed of multi-layered Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU). The first RNN encodes an input sequence to a vector of fixed dimensionality, and the second RNN decodes the target sequence from the vector. Sequence to sequence models are general models and are applicable to many tasks other than machine translation, including dialogue generation [28] and question answering [13]. With sequence to sequence models, a machine translation model can be learned in an end-to-end manner by feeding a sequence of words in the source language and another sequence of words in the target language.

3. Cross-Lingual Text Similarity

3.1 Existing Approaches

We first introduce representative, existing approaches to be compared in this study, and then describe the details of the idea and evaluation framework of the NMT-based approach.

3.1.1 Word Embedding Pooling

Word embedding [18] is obtained as a by-product of a neural probabilistic language model [3] and has been successfully used in various NLP tasks. Word embedding pooling is a simple extension of word embedding to a text (a sequence of words) and is defined as the element-wise average of the word embeddings of the words (w) composing the text (d). More formally, it is defined as $\sum_{w \in d} \vec{v}_w / |d|$, where $\vec{v}_w$ is a word embedding vector of w and $|d|$ is the length (in words) of d. This approach disregards word order as can be seen in the definition but generally works moderately well and is often used in part for computing cross-lingual text similarity [4].

3.1.2 Paragraph Embedding

Paragraph embedding (also known as paragraph vectors) was proposed by Le and Mikolov [15] to learn a distributed representation of paragraphs as an extension of the word embedding model. In the paragraph vector model (called the distributed memory model), a paragraph itself is treated as a pseudo word and its embedding vector is learned through learning a neural language model. By including a paragraph as another word in the model, the paragraph acts as a kind of memory or context. It should be noted that while the original word embedding model learns only the words appearing in training data, the paragraph embedding model is able to infer the paragraph embedding vector for a new paragraph not found in the training data through the standard backpropagation by feeding the paragraph to the model.

3.1.3 Siamese Neural Network

This approach is a cross-lingual extension of the Siamese LSTM neural network by Mueller and Thyagarajan [20] for learning mono-lingual sentence similarity. The extended model consists of two stacked RNNs to encode English and Japanese text pairs separately and an output layer to estimate their similarity as shown in **Fig. 1**, where y denotes a true label (defined as 1 for corresponding translation pairs and 0 for the others). It should be mentioned that this model uses word embedding as an input layer of the RNN and is different from the cross-lingual Siamese model by Yih et al [31] in which the input text was represented as a classic bag-of-word TFIDF vector.

3.2 NMT-based Approach

A sequence-to-sequence NMT model first encodes an input text to a dense real-valued vector (sometimes called a "thought vector"), which is passed to a decoder for generating a translation (**Fig. 2**). Thus, the intermediate vector can be naturally seen as a concise, good representation of the input. In addition, since

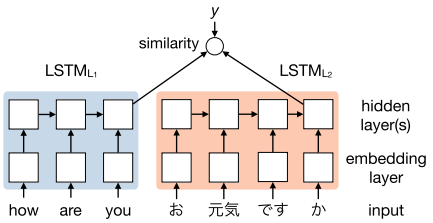


Fig. 1 Siamese neural network for estimating cross-lingual (Japanese-English) similarity.

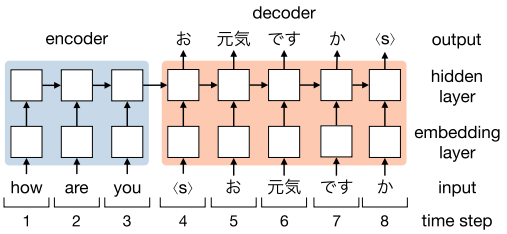


Fig. 2 An example of a sequence-to-sequence NMT model translating an English sentence to Japanese.

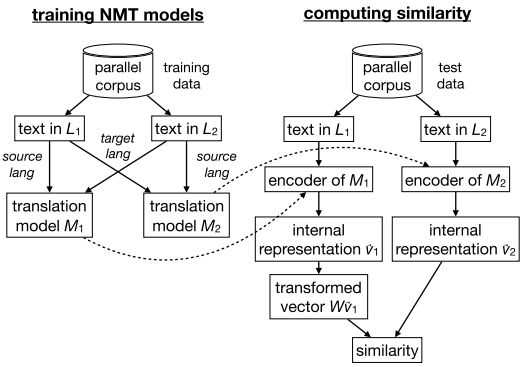


Fig. 3 An illustration of the data and processing flows for training translation models (left) and computing similarity between two texts written in different languages (right).

the intermediate vector is not yet translated to a particular word sequence, it could potentially avoid the issues of polysemy and synonymy (see Section 1) associated with MT-based approaches. Given the observation, we hypothesize that these intermediate states capture syntactic and semantic properties of input text and that they can be effectively used for measuring the similarity between two texts written in different languages.

To validate the hypothesis, we design a modeling and evaluation framework as illustrated in **Fig. 3**. The left-hand side of the figure depicts the data flow for training two NMT models, and the right-hand side of the figure depicts the data flow for computing cross-lingual similarity for a given text pair, where L_1 and L_2 denote two languages used in two groups of texts, respectively. The following subsections describe the key components in the figure.

3.3 Learning Neural Machine Translation Models

Suppose that there are two texts conveying the same contents but written in different languages L_1 and L_2. Then, two NMT models M_1 and M_2 are independently trained on the a set of such text pairs, where M_1 translates L_1 to L_2 and M_2 translates L_2 to L_1. As an NMT model, the present work adopts the sequence-to-sequence model with GRU layers and an attention mechanism by Vinyals et al. [27], but it can be any other NMT model with the encoder-decoder architecture.

When computing the similarity between two texts in evaluation, one in L_1 and the other in L_2, they are fed to the encoders of M_1 and M_2, respectively. The outputs of the encoders (vectors $\vec{v}_1$ and $\vec{v}_2$) are used as a representation of the input texts. In precise, $\vec{v}_1$ and $\vec{v}_2$ are vectors of the outputs of the last GRU layer of the encoders of M_1 and M_2, respectively, at the last input time step so as to capture all the information contained in the input. It should be pointed out that it is also possible to use the outputs of encoders at every time step as done by the attention mechanism [2]. However, since our aim is not to generate a good translation, their effects would be limited and so are not examined further in the present work.

3.4 Estimating Transformation Matrix

Since the translation models M_1 and M_2 are learned independently and also for the opposite directions of translation, the outputs of their encoders are not directly comparable to each other. In other words, each pair of corresponding elements in the resulting vectors $\vec{v}_1$ and $\vec{v}_2$ represent different properties of the respective input texts. Thus, simply computing the similarity between the two vectors will not work. This is similar to the case for word embeddings for different languages [17].

Following Mikolov et al. [17], we assume a linear relationship between the two language spaces. A transformation matrix W which maps the space for $\vec{v}_1$ to the one for $\vec{v}_2$ can then be estimated using known pairs of corresponding texts (translations) in L_1 and L_2 as:

$$\arg\min_{W} \|V_1^T W^T - V_2^T\|^2 \quad (1)$$

where V_1 (or V_2) is a matrix formed by merging multiple $\vec{v}_1$ (or $\vec{v}_2$) of the known translations. Such W can be computed as $(V_1^+ V_2^T)^T$, where V_1^+ is the pseudo-inverse of matrix V_1.

After converting the vectors using the transformation matrix W, a standard similarity function such as cosine similarity can be used for computing the similarity of the input texts.

4. Evaluation

To investigate the utility of the intermediate states of NMT models in computing cross-lingual similarity as described in Section 3, several experiments were carried out by using English and Japanese translation pairs. The two languages do not belong to the same family of languages and differ greatly in various aspects from their characters to syntax to semantics and would serve as a good test bed for the purpose of this study.

4.1 Experimental Setup

As training and test data, an English-Japanese translation corpus was needed. The experiments in this paper were based on the Eijiro English-Japanese dictionary Ver. 139. The dictionary includes not only an English-Japanese equivalent word list but also 492,007 pairs of equivalent sentences (translations), of which 10,000 randomly chosen pairs were used for testing and the remaining 482,007 pairs were used for training two NMT models. It should be mentioned that Japanese sentences were split

Table 1 Descriptive statistics of the translation data from the Eijiro dictionary. "Average sentence length" is the average number of terms composing a sentence.

	English	Japanese
Num of sentences	482,007	482,007
Total num of term occurrences	7,027,160	10,154,559
Average sentence length	14.6	21.1
Num of unique terms	351,676	152,288

- Success cannot be measured solely on the basis of income. ↔ 成功は、収入だけで測れるものではない。
- Once you get to know her, she's kinda cool. ↔ 一度彼女のことを知ると、彼女は格好いい。
- Sap is collected during a three-to-six-week period in early spring before the maple trees begin putting out leaves. ↔ 樹液は、カエデの木が芽を出し始める前の、早春の3〜6週間の間に集められます。

Fig. 4 Examples of English-Japanese translation pairs in the Eijiro dictionary.

into words by the MeCab morphological analyzer [*1] in advance as they do not have explicit word boundaries (e.g., spaces in English). **Table 1** shows the descriptive statistics of the data and **Fig. 4** presents a few examples of translation pairs from the corpus.

The parameters of both translation models (English to Japanese and Japanese to English) were empirically set as follows: learning rate = 0.5, learning rate decay factor = 0.99, batch size = 64, number of GRU layers = 3, number of units per layer = 1,024 (including an embedding layer), and vocabulary size = 40,000 for each language. Stochastic gradient descent was used for optimization. In addition, parameters regarding buckets were set in a way that 1,000 longest sentences in the training data (1 k/482 k ≈ 0.2%) were excluded from training for efficiency. The NMT models were trained until the perplexity of the model on the training data reached around 5.0 or the global learning step reached around 100,000.

The test data (10,000 English and Japanese sentence pairs) were further split into 1,000 and 9,000 pairs. The former was used for evaluating the performance of the intermediate state-based approach and alternative approaches to be described in Section 4.2, and the latter was used for obtaining the vector transformation matrix W (see Section 3.4).

After applying the transformation to the 1,000 English sentences by matrix W, the similarity of every combination of an English sentence and a Japanese sentence from the 1,000 pairs was computed using cosine similarity. When a pair of equivalent English and Japanese sentences had the highest similarity among the 1,000 possible combinations, it was regarded as correct. Differently put in IR terminology, an English sentence can be thought of as a query to retrieve the corresponding Japanese sentence as its sole relevant document in a collection of 1,000 sentences.

As an evaluation metric, precision at 1 (Prec@1) and precision at 5 (Prec@5) were used. Prec@1 was defined as the number of corrects divided by the total number of sentence pairs used in the evaluation (i.e., 1,000), whereas Prec@5 is more relaxed and looked at five sentences from the top in the order of descending similarity scores.

[*1] http://taku910.github.io/mecab/

4.2 Baseline Approaches

For comparison, the following four alternative approaches were implemented and tested in the experiments.

- Paragraph embedding (Doc2Vec): A Doc2Vec model was learned for each language by feeding the training data (482 k translation pairs) to the Gensim Python package [23] with the default parameters. Using this model, each sentence in the test data was represented as an embedding vector. As is the case with the approach described in Section 3.4, a transformation matrix was obtained by the 9,000 English-Japanese translation pairs and cosine similarity was computed for the remaining 1,000 pairs after transformation.
- Word embedding pooling (Word2Vec): This approach simply takes the average of the word embedding vectors of words composing a sentence. For this experiment, pretrained English [*2] (GoogleNews-vectors-negative300.bin.gz) and Japanese [*3] models were utilized. Note that more recent pretrained models [*4] were also tested but did not improve the performance, hence the results are not reported here. Again, W was obtained by the 9,000 English-Japanese translation pairs and cosine similarity was computed for the 1,000 pairs after transformation.
- Siamese neural network (S2Net): This approach used the extended model described in Section 3.1.3. The configuration of the model was empirically set on the same training data (482 k translation pairs). Specifically, the number of hidden layers and the number of units per layer in RNN were set to 3 and 256, respectively. For the input word embedding layer, the number of dimensions was set to 300. After training, the same 1,000 pairs as the other approaches were used as the test data for fair comparison. Note that because S2Net transforms English and Japanese texts into the common semantic space, computing W is not needed.
- Machine translation (MT): This approach translated English sentences to Japanese by an NMT model and computed their similarity in a standard way, namely, representing input texts by word vectors weighted by TFIDF and computing their cosine similarity. The same 1,000 translation pairs were used as test data, the same as in the other approaches. This is a strong baseline as mentioned in Section 2. For translation, the same NMT model as in Section 4.1 was used to avoid the effect of the performance difference of MT systems used in evaluation.

4.3 Results and Discussion

4.3.1 Sentence Retrieval

The results of finding corresponding English and Japanese sentence pairs are summarized in **Table 2**, where "IntRe" refers to the approach using the intermediate representation of NMT models and "Doc2Vec", "Word2Vec", "S2Net", and "MT" refer to the approaches using paragraph vectors, word embedding pooling,

[*2] https://code.google.com/archive/p/word2vec
[*3] https://github.com/Kyubyong/wordvectors
[*4] https://research.fb.com/fasttext

Table 2 Results in precision for corresponding sentence retrieval.

Approach	Prec@1	Prec@5
Doc2Vec	0.002	0.006
Word2Vec	0.125	0.229
S2Net	0.094	0.269
MT	0.264	0.427
IntRe	0.512	0.739

Siamese neural network, and machine translation, respectively.

It is found that "IntRe" achieved strikingly better results than the others including the strong MT-based approach; Prec@1 almost doubled as compared to "MT". On the other hand, word/paragraph embedding approaches did not perform well. Word embedding pooling "Word2Vec" was not very effective presumably due to the fact that it ignores the contextual information (word order) by simply averaging word vectors. Another reason may be that the word embedding models used here are mono-lingual. Recently, Lample et al. [14] proposed a cross-lingual word embedding model, which was reported to work better than mono-lingual word embedding for cross-lingual tasks.

On the other hand, paragraph vectors "Doc2Vec" performed the worst despite that they are designed to represent a short text and deemed more suitable for this task than word embedding. These results imply that the two language spaces learned by paragraph vector models do not have a simple linear relationship. Lastly, Siamese neural network "S2Net" yielded slightly lower Prec@1 and slightly better Prec@5 than Word2Vec.

It should be mentioned that training Doc2Vec models does not require translation pairs and mono-lingual corpora larger than the Eijiro data could be used for training. The relatively small Eijiro data used for training may explain the poor performance of Doc2Vec. To examine the possibility, much larger data, specifically, English and Japanese Wikipedia dumps, were used for training English and Japanese Doc2Vec models, respectively. The performance did improve but was still below the other approaches and thus omitted here. Also, there are other models more recently proposed for sentence embedding [1], [19], [21] and it would be interesting to examine them for a cross-lingual setting in future work.

4.3.2 Relation to the Performance of NMT models

As discussed above, the approach, IntRe, using intermediate representation of NMT models outperformed the others including the MT-based approach directly using translated texts. However, the MT models used in the present work may be considered rather weak given the current size of the models (i.e., three layers of 1,024 units) and the limited size of the training data. Therefore, it is possible that, if the translation models' performance was better, the MT-based approach may have performed better than IntRe.

Therefore, we examined the effect of MT model's performance on this task by conducting two additional experiments, where the size of the training data and the capacity of the translation model were changed. For the former, the size of the training data (i.e., the number of translation pairs) was gradually changed from 100,000 to 482,007 with the unit size being fixed to 1,024. If the precision of MT-based approach increases more rapidly than IntRe as the training data grow, it implies that MT-based approach would eventually outperform IntRe.

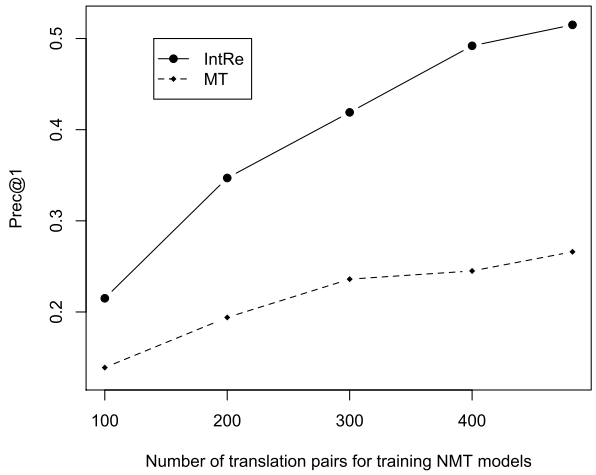


Fig. 5 Relation between precision and data size for training translation models.

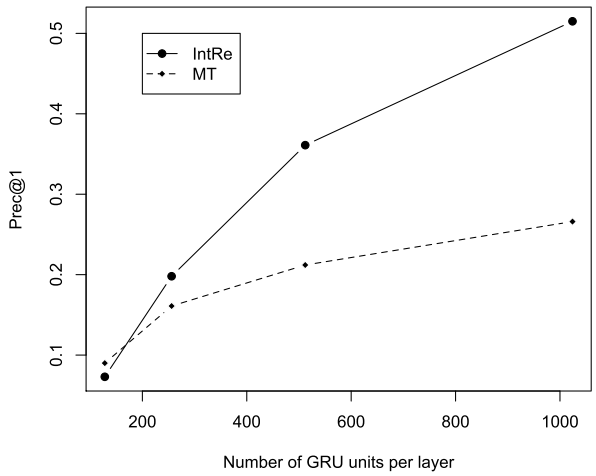


Fig. 6 Precision comparison between IntRe and MT-based approaches with increasing unit size.

Figure 5 plots Prec@1 of the two approaches with varying training data size. The result indicates that, overall, the precision between the IntRe and the MT-based approach becomes more widely separated as more training data are used.

Then, the number of GRU units per layer was changed from 128 to 256, 512, and 1,024 where the training data size was fixed to the maximum (i.e., 482,007). The smaller the unit size is, the poorer the translation model would become. Similarly to the previous experiment, if the precision of MT-based approach increases more rapidly than IntRe, it implies the advantage of the MT-based approach with a larger model.

Figure 6 plots Prec@1 of the two approaches with increasing unit size. It turned out that the precision of IntRe increased more rapidly than the MT-based approach. Taken together with the previous experiment, the result strongly suggests the advantage of IntRe and that a wider increase in performance would be expected with higher performing NMT models.

In addition, in order to see the current upper bound of the MT-based approach, we carried out an experiment using one of the state-of-the-art NMT systems, Google Translate [29], as an MT system to translate English sentences to Japanese. Based on the translation, we repeated the sentence retrieval experiment and the

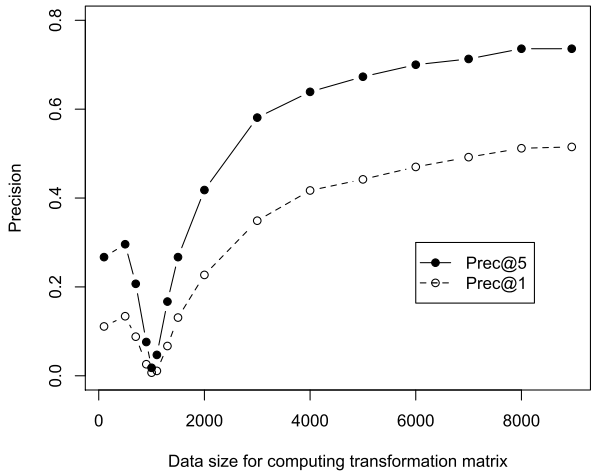


Fig. 7 Relation between precision and data size for computing a transformation matrix.

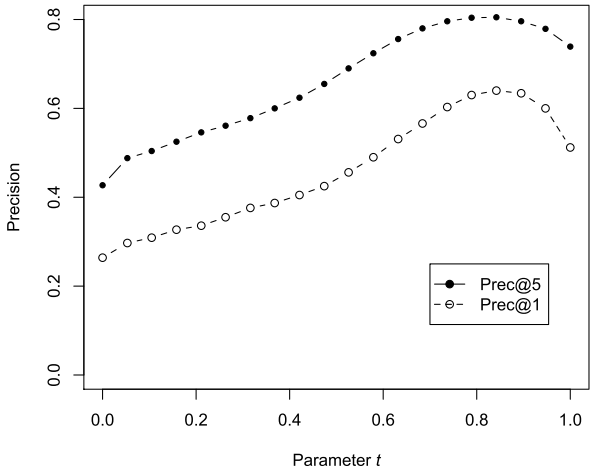


Fig. 8 Precision curves when MT and IntRe are linearly interpolated.

resulting P@1 and P@5 were found to be 0.693 and 0.801, respectively, which are significantly better than the MT-based approach using our weaker model (Table 2). Due to the lack of the access to the internal states of Google's NMT models, we are not able to evaluate the performance of "IntRe" approach based on the models. However, the observations made in the above experiments suggest that even greater performance could be achieved by exploiting the intermediate states of the models.

4.3.3 Transformation Matrix

In this section, the relation between the data size for computing a transformation matrix W and precision was studied, which would tell us if more data for estimating W would be beneficial for further improving the performance of IntRe. **Figure 7** shows the plots of Prec@1 and Prec@5 with different data sizes.

After a sudden drop at the data size of around 1,000, which may be due to overfitting to the small amount of data, the accuracy steadily increased and appears to come close to a plateau at the data size of 9,000. This result suggests that larger data may help increasing the performance to some degree but the effect would be limited.

4.3.4 Model Combination

A common approach to further improving the absolute performance of a given task is to take advantage of multiple models/approaches and combine them. To investigate if any improvement could be accomplished, we tested a simple ensemble scheme to interpolate two similarity values independently obtained by MT and IntRe from the experiment in Section 4.3.1. To be precise, we computed a linearly weighted sum of their similarity matrices, $\mathbf{M_{IntRe}}$ and $\mathbf{M_{MT}}$,

$$\mathbf{M'} = t \cdot \mathbf{M_{IntRe}} + (1 - t) \cdot \mathbf{M_{MT}} \quad (2)$$

where an element m_{ik} of the matrices is a similarity between an English sentence e_i and a Japanese sentence j_k and t is an interpolation parameter to control the relative effect of $\mathbf{M_{IntRe}}$ and $\mathbf{M_{MT}}$.

Based on the combined similarity matrix $\mathbf{M'}$, we performed the same sentence retrieval experiment again. **Figure 8** plots the resulting Prec@1 and Prec@5 for different values of t. Notice that the leftmost dots at $t = 0$ correspond to using only MT and that the rightmost dots at $t = 1.0$ correspond to using only IntRe.

Both Prec@1 and Prec@5 increased as t increased up to around $t = 0.85$ where Prec@1 and Prec@5 reached 0.640 and 0.805, respectively. Comparing with IntRe alone, the percent increases are 24% for Prec@1 and 9.5% for Prec@5. It is interesting to observe that the intermediate states and translated texts work complementarily despite the fact that both of them were generated by the same NMT models.

Lastly, it should be emphasized that Prec@1 came close to that of Google's NMT model and Prec@5 marginally exceeded Google's. Considering the limited amount of our training data used for building the rather small NMT models on which IntRe and MT were based, these results are encouraging and could lay the foundation for cross-lingual text similarity.

5. Conclusions

This paper dealt with the important problem of cross-lingual text similarity with a focus on neural networks. Specifically, we explored the utility of NMT models for the problem but did not rely on translated text. Instead, we looked at the internal states of the models as the semantic vector representation of text to avoid translation errors introduced by an MT system. The vectors were then transformed to the same language space as the other language and then used for computing similarity. The validity and effectiveness of the approach were evaluated on an English-Japanese translation corpus in comparison with word/paragraph embedding-based approaches and a strong MT-based approach. The results demonstrated that the approach using the intermediate states performed better than the other approaches by a wide margin. In addition, when the intermediate state-based approach was combined with an MT-based approach, they worked complementarily and the performance was further improved despite that they originated from the same NMT models.

The present study focused on an English and Japanese pair but the observations made in the present study should be language-independent and will be beneficial for other language pairs. However, there still remain some limitations. First, it requires a translation corpus to train the models, which may or may not exist for a given pair of languages. Zero shot translation [12] may be beneficial in this regard. Second, the models were built for short texts, such as sentences, and not suitable for representing documents. Third, cross-lingual extensions of word/sentence embedding are

a hot area of research and there are other interesting approaches to compare, which were only briefly discussed in this paper. As well as tackling these issues, future work would include testing the usefulness of the approach in actual applications, such as IR and data mining for cross-lingual texts.

Acknowledgments This work is partially supported by JSPS KAKENHI Grant Numbers 18K11558 and MEXT, Japan.

References

[1] Arora, S., Liang, Y. and Ma, T.: A Simple but Tough-to-Beat Baseline for Sentence Embeddings, *Proc. 5th International Conference on Learning Representations* (2017).

[2] Bahdanau, D., Cho, K. and Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate, *Proc. 3rd International Conference on Learning Representations* (2015).

[3] Bengio, Y., Ducharme, R., Vincent, P. and Janvin, C.: A Neural Probabilistic Language Model, *The Journal of Machine Learning Research*, Vol.3, pp.1137–1155 (2003).

[4] Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I. and Specia, L.: SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation, *Proc. 11th International Workshop on Semantic Evaluation* (*SemEval-2017*), pp.1–14 (2017).

[5] Chandar A.P.S., Lauly, S., Larochelle, H., Khapra, M., Ravindran, B., Raykar, V.C. and Saha, A.: An Autoencoder Approach to Learning Bilingual Word Representations, *Proc. 27th International Conference on Neural Information Processing Systems*, pp.1853–1861 (2014).

[6] Cho, K., van Merrienboer, B., Bahdanau, D. and Bengio, Y.: On the Properties of Neural Machine Translation: Encoder–Decoder Approaches, *Proc. 8th Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp.103–111 (2014).

[7] Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K. and Harshman, R.: Indexing by latent semantic analysis, *Journal of the American Society for Information Science*, Vol.41, No.6, pp.391–407 (1990).

[8] Dumais, S., Letsche, T., Littman, M. and Landauer, T.: Automatic cross-language retrieval using latent semantic indexing, *Proc. AAAI Spring Symposium on Cross-language Text and Speech Retrieval*, pp.18–24 (1997).

[9] Glavas, G., Franco-Salvador, M., Ponzetto, S.P. and Rosso, P.: A resource-light method for cross-lingual semantic textual similarity, *Knowledge-Based Systems*, Vol.143, pp.1–9 (2018).

[10] Gupta, P., Banchs, R.E. and Rosso, P.: Continuous Space Models for CLIR, *Information Processing & Management*, Vol.53, No.2, pp.359–370 (2017).

[11] Hinton, G.E. and Salakhutdinov, R.R.: Reducing the Dimensionality of Data with Neural Networks, *Science*, Vol.313, No.5786, pp.504–507 (2006).

[12] Johnson, M., Schuster, M., Le, Q.V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., Hughes, M. and Dean, J.: Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation, *Trans. Association for Computational Linguistics*, Vol.5, No.1, pp.339–351 (2017).

[13] Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., Zhong, V., Paulus, R. and Socher, R.: Ask Me Anything: Dynamic Memory Networks for Natural Language Processing, *Proc. 33rd International Conference on Machine Learning*, pp.1378–1387 (2016).

[14] Lample, G., Conneau, A., Ranzato, M., Denoyer, L. and Jgou, H.: Word translation without parallel data, *Proc. 6th International Conference on Learning Representations* (2018).

[15] Le, Q. and Mikolov, T.: Distributed Representations of Sentences and Documents, *Proc. 31st International Conference on Machine Learning*, pp.1188–1196 (2014).

[16] Manning, C.D., Raghavan, P. and Schütze, H.: *Introduction to Information Retrieval*, Cambridge University Press (2008).

[17] Mikolov, T., Le, Q.V. and Sutskever, I.: Exploiting Similarities among Languages for Machine Translation, *CoRR*, Vol.abs/1309.4168 (2013).

[18] Mikolov, T., Yih, W. and Zweig, G.: Linguistic Regularities in Continuous Space Word Representations, *Proc. 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (*NAACL-HLT-2013*), pp.746–751 (2013).

[19] Mu, J., Bhat, S. and Viswanath, P.: Representing Sentences as Low-Rank Subspaces, *Proc. 55th Annual Meeting of the Association for Computational Linguistics*, pp.629–634 (2017).

[20] Mueller, J. and Thyagarajan, A.: Siamese Recurrent Architectures for Learning Sentence Similarity, *Proc. 30th AAAI Conference on Artificial Intelligence*, pp.2786–2792 (2016).

[21] Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., Song, X. and Ward, R.: Deep Sentence Embedding Using Long Short-term Memory Networks: Analysis and Application to Information Retrieval, *IEEE/ACM Trans. Audio, Speech and Language Processing*, Vol.24, No.4, pp.694–707 (2016).

[22] Peters, C., Braschler, M. and Clough, P.: *Multilingual Information Retrieval*, Springer-Verlag Berlin Heidelberg (2012).

[23] Řehůřek, R. and Sojka, P.: Software Framework for Topic Modelling with Large Corpora, *Proc. LREC 2010 Workshop on New Challenges for NLP Frameworks*, pp.45–50 (2010).

[24] Rupnik, J., Muhič, A., Leban, G., Škraba, P., Fortuna, B. and Grobelnik, M.: News Across Languages - Cross-lingual Document Similarity and Event Tracking, *Journal of Artificial Intelligence Research*, Vol.55, No.1, pp.283–316 (2016).

[25] Sparck Jones, K.: Statistical interpretation of term specificity and its application in retrieval, *Journal of Documentation*, Vol.28, No.1, pp.11–20 (1972).

[26] Sutskever, I., Vinyals, O. and Le, Q.V.: Sequence to Sequence Learning with Neural Networks, *Proc. 27th International Conference on Neural Information Processing Systems*, pp.3104–3112 (2014).

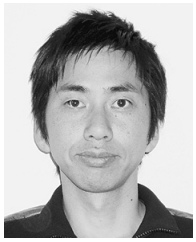
[27] Vinyals, O., Kaiser, L., Koo, T., Petrov, S., Sutskever, I. and Hinton, G.E.: Grammar as a Foreign Language, *CoRR*, Vol.abs/1412.7449 (2014).

[28] Vinyals, O. and Le, Q.V.: A Neural Conversational Model, *CoRR*, Vol.abs/1506.05869 (2015).

[29] Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M. and Dean, J.: Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, *CoRR*, Vol.abs/1609.08144 (2016).

[30] Xiao, M. and Guo, Y.: A Novel Two-step Method for Cross Language Representation Learning, *Proc. 26th International Conference on Neural Information Processing Systems*, pp.1259–1267 (2013).

[31] Yih, W.-t., Toutanova, K., Platt, J.C. and Meek, C.: Learning Discriminative Projections for Text Similarity Measures, *Proc. 15th Conference on Computational Natural Language Learning*, pp.247–256 (2011).



Kazuhiro Seki received his Ph.D. in information science from Indiana University, Bloomington. His research interests are in the areas of natural language processing, information retrieval, and data mining. He is currently an associate professor in Faculty of Intelligence and Informatics at Konan University. He is a member of JSAI and ACM SIGIR.